Appli Documentation

Contents

1 Welcome to Appli Documentation

1.1 Disclaimer

This document is a draft version of the Appli documentation. The content is not complete. It is a beta version of the material we are building. We see value in sharing this in this early form because it is already useful for our early users, and also because it helps us gather feedback as we're building it.

Please, send your comments, feedback, and issues to our editorial team.

1.2 Starting points

Reading this manual cover to cover will get a comprehensive understanding of Appli. However, the links below provide quick access to some of the most popular pages in our documentation.

1.2.1 Overview and Core Concepts

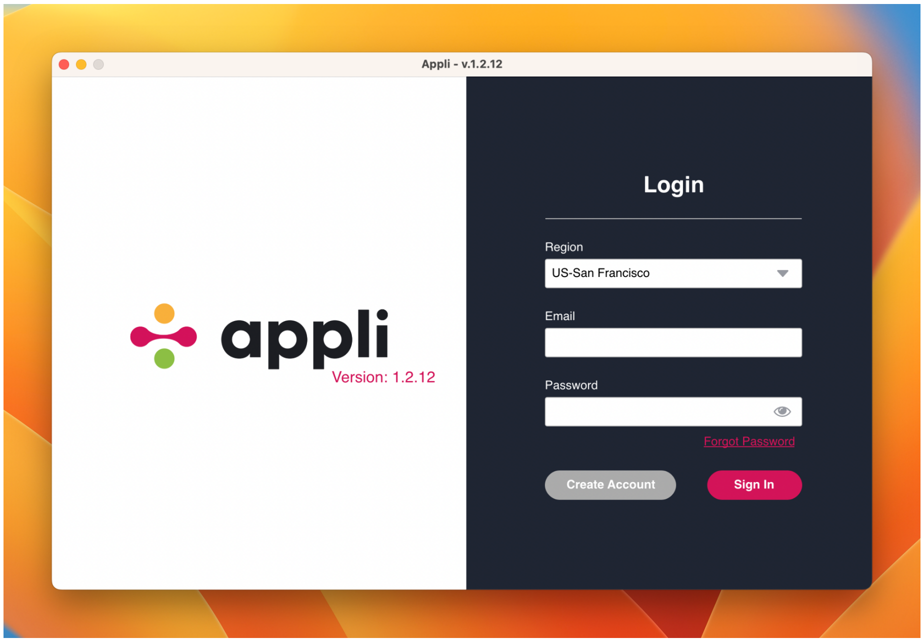
- Appli Overview
- Login
- Low-code
- Elements
- Data Management

1.2.2 Tutorials

- Introductory Tutorial
- Database Tutorial
- Forms Tutorial
- Layout Tutorial

To check which chapters changed recently, read the changelog. To see the patch notes for Appli, read Appli Patch Notes.

2 The Login Screen



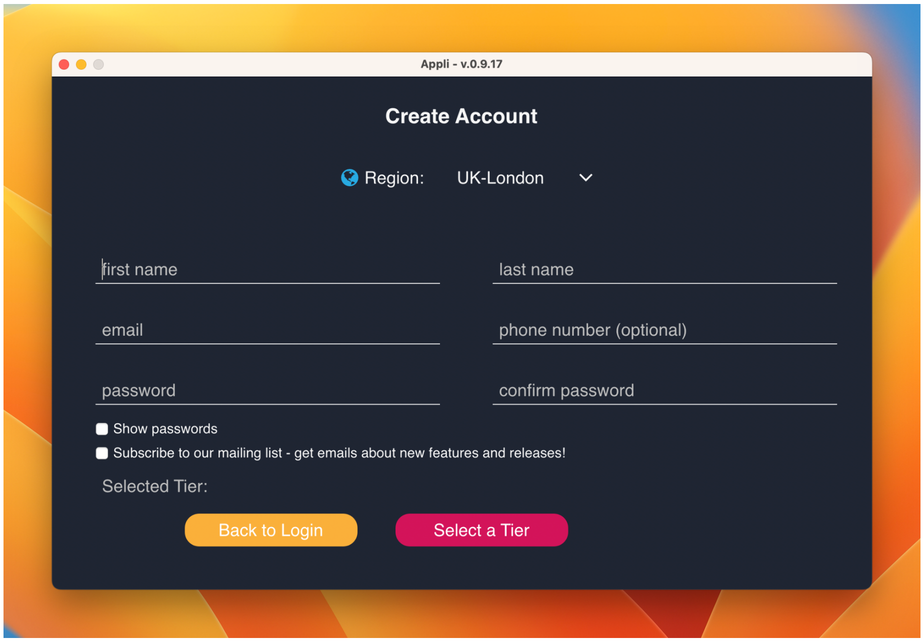
Region
US-San Francisco

Figure 1: Login Screen

The first screen you see when you launch Appli IDE is the login screen. It can create new accounts or to sign into an existing account. To use Appli IDE, fill in your login information, making sure you selected the correct region for your account and click *Sign In*. Once you sign in, you'll be shown the Projects Screen.

2.1 Account Creation

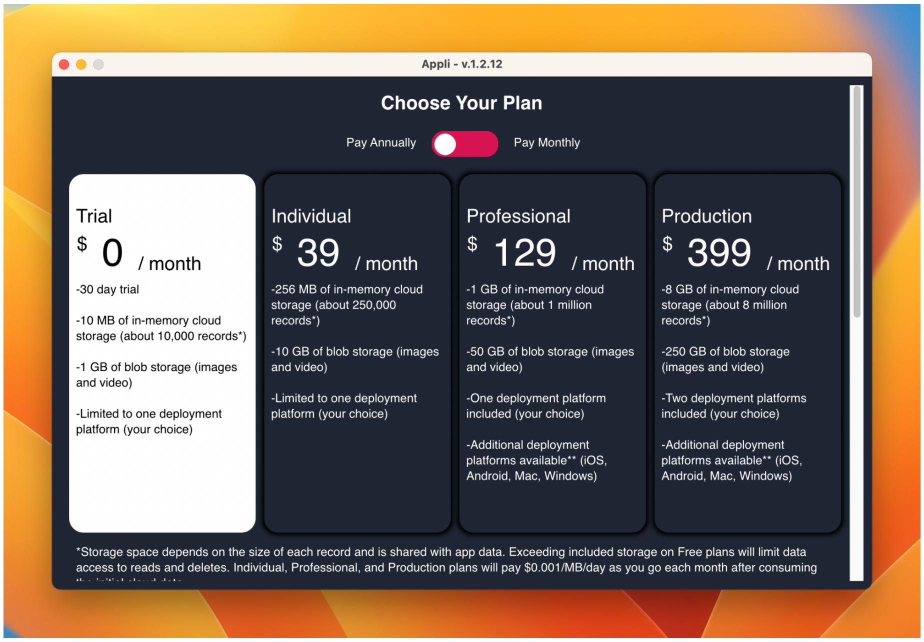
Accounts are unique to each region, so make sure you have selected your correct region first before filling in the login information.



Region: UK-London

Figure 2: Create Account Screen

You must select a *tier* before you're able to continue the account creation workflow.

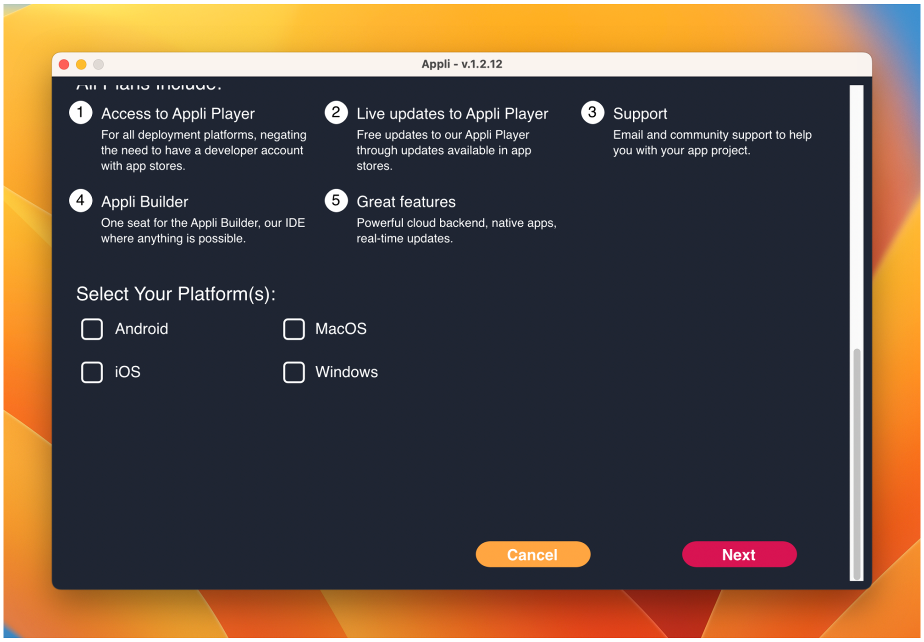


Trial	Individual	Professional	Production
$ 0 / month	$ 39 / month	$ 129 / month	$ 399 / month
-30 day trial	-256 MB of in-memory cloud storage (about 250,000 records*)	-1 GB of in-memory cloud storage (about 1 million records*)	-8 GB of in-memory cloud storage (about 8 million records*)
-10 MB of in-memory cloud storage (about 10,000 records*)	-10 GB of blob storage (images and video)	-50 GB of blob storage (images and video)	-250 GB of blob storage (images and video)
-1 GB of blob storage (images and video)	-Limited to one deployment platform (your choice)	-One deployment platform included (your choice)	-Two deployment platforms included (your choice)
-Limited to one deployment platform (your choice)		-Additional deployment platforms available** (iOS, Android, Mac, Windows)	-Additional deployment platforms available** (iOS, Android, Mac, Windows)

*Storage space depends on the size of each record and is shared with app data. Exceeding included storage on Free plans will limit data access to reads and deletes. Individual, Professional, and Production plans will pay $0.001/MB/day as you go each month after consuming the initial cloud data

Figure 3: Select Tier

Tiers can be configured with additional deployment platforms.



All Plans Include:

1. Access to Appli Player
   For all deployment platforms, negating the need to have a developer account with app stores.
2. Live updates to Appli Player
   Free updates to our Appli Player through updates available in app stores.
3. Support
   Email and community support to help you with your app project.
4. Appli Builder
   One seat for the Appli Builder, our IDE where anything is possible.
5. Great features
   Powerful cloud backend, native apps, real-time updates.

Select Your Platform(s):

□ Android □ MacOS

□ iOS □ Windows

Figure 4: Customizing Tiers

Once you configure your tier to your needs, the green *Create Account* button will become visible.

3 The Project Selection Screen

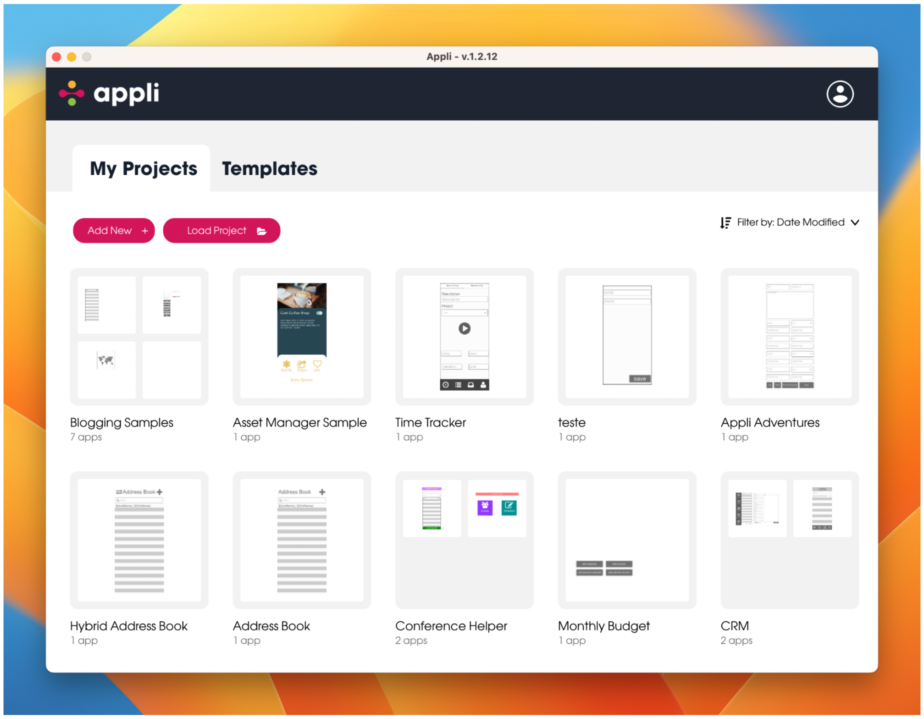
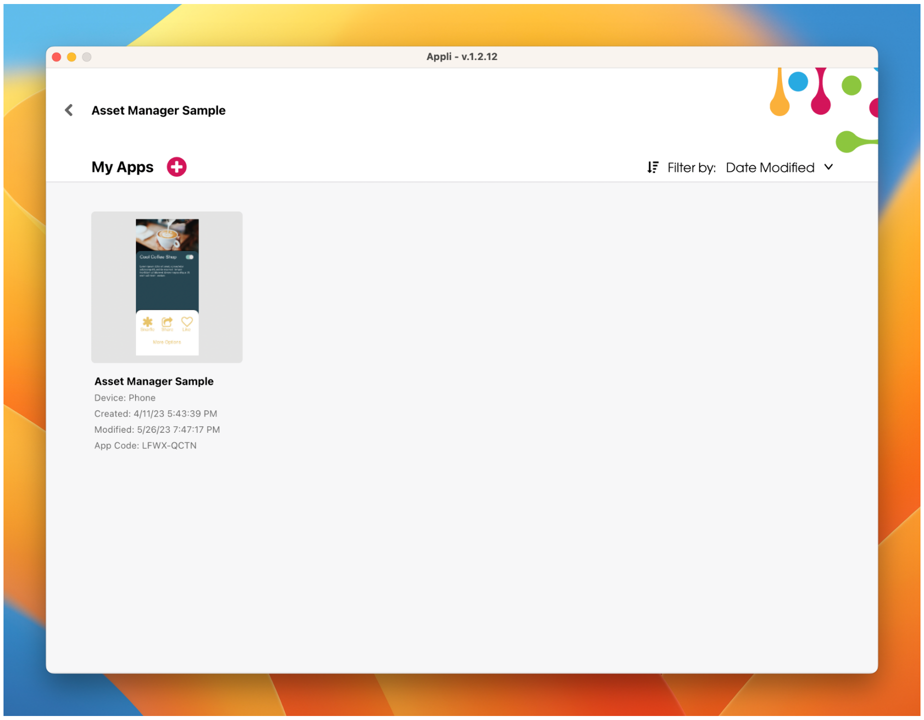


Figure 5: Project Selection Screen

This screen is used to create or open projects. Projects can contain multiple applications. The bottom half of the screen shows your current projects.

Clicking on an existing project lets you select which application from that project you want to open in the Playground. To change how your projects are listed, use the *Sort* dropdown on the right side of the project selection area.

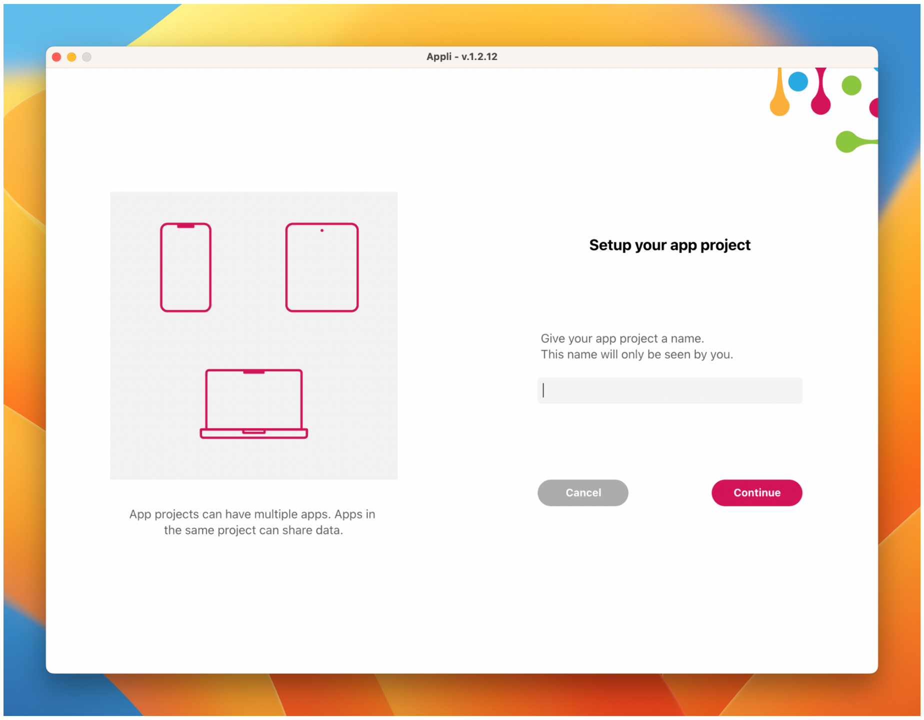


Asset Manager Sample
Device: Phone
Created: 4/11/23 5:43:39 PM
Modified: 5/26/23 7:47:17 PM
App Code: LFWX-QCTN

Figure 6: Application Selection Screen

3.1 Creating a project

Use the *plus button* next to *My App Projects* label to create a new project.



Setup your app project

Give your app project a name.
This name will only be seen by you.

App projects can have multiple apps. Apps in the same project can share data.

Figure 7: Project creation dialog

Once you give the new project a name, choose a platform for your first application on that project. You can have many applications per project. This allows you to support many platforms. You can add more platforms later.

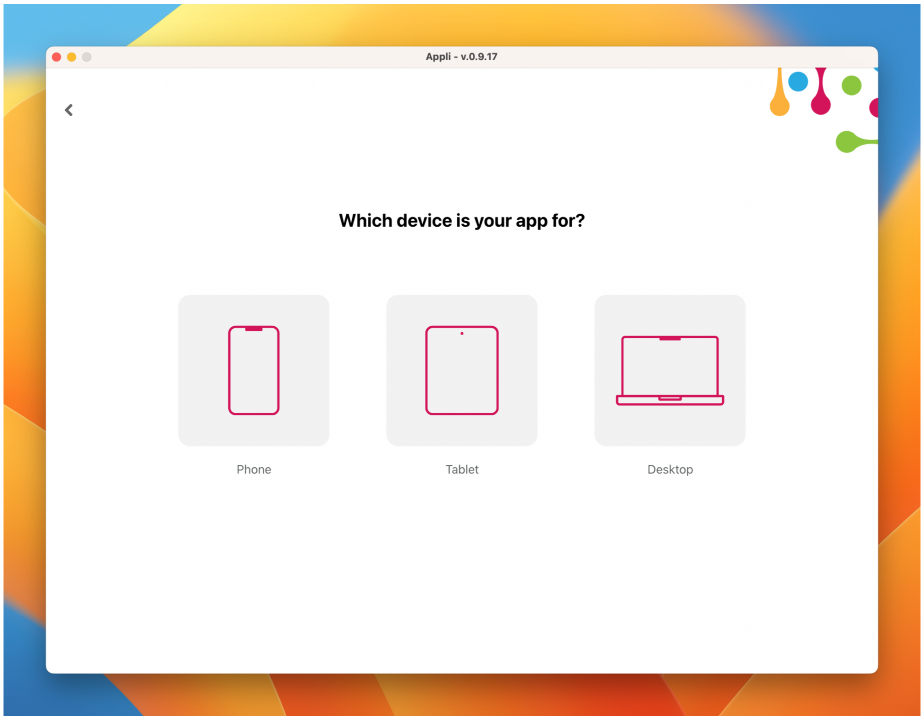
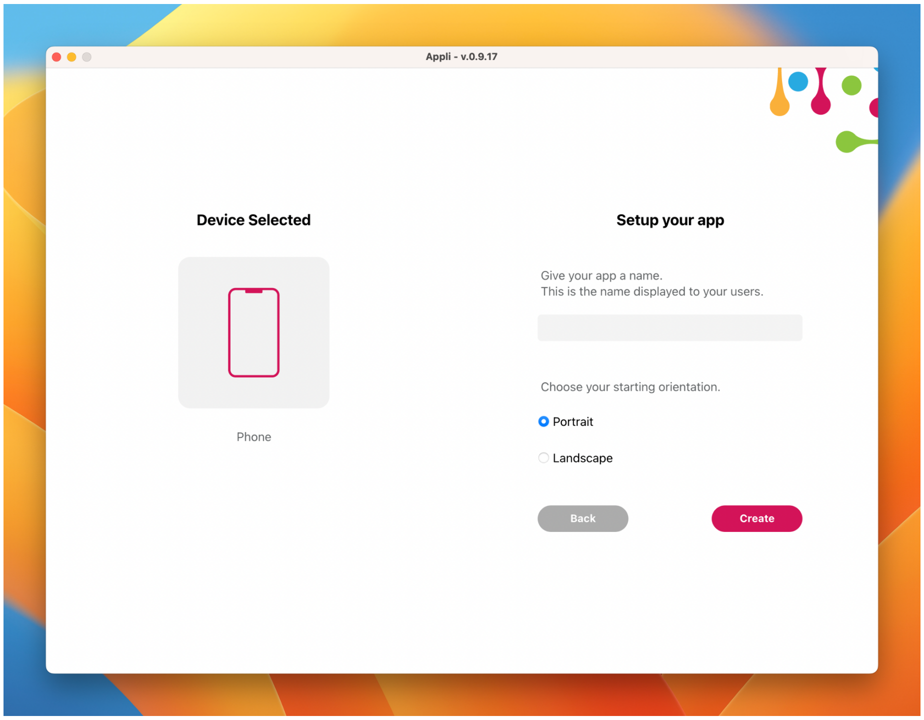


Figure 8: Plataforma selection

After clicking on a platform, Appli Builder displays a dialog allowing you to set the new app's name, and the initial orientation.



Device Selected

Phone

Setup your app

Give your app a name.
This is the name displayed to your users.

Choose your starting orientation.

● Portrait

○ Landscape

Figure 9: App creation dialog

Once the project is created, Appli Builder will load the new project into the Playground.

3.2 Saving a project to disk

Saving a project to disk is easy, just click the *three dots button* next to the project name and select *save to disk* in the pop-up menu.

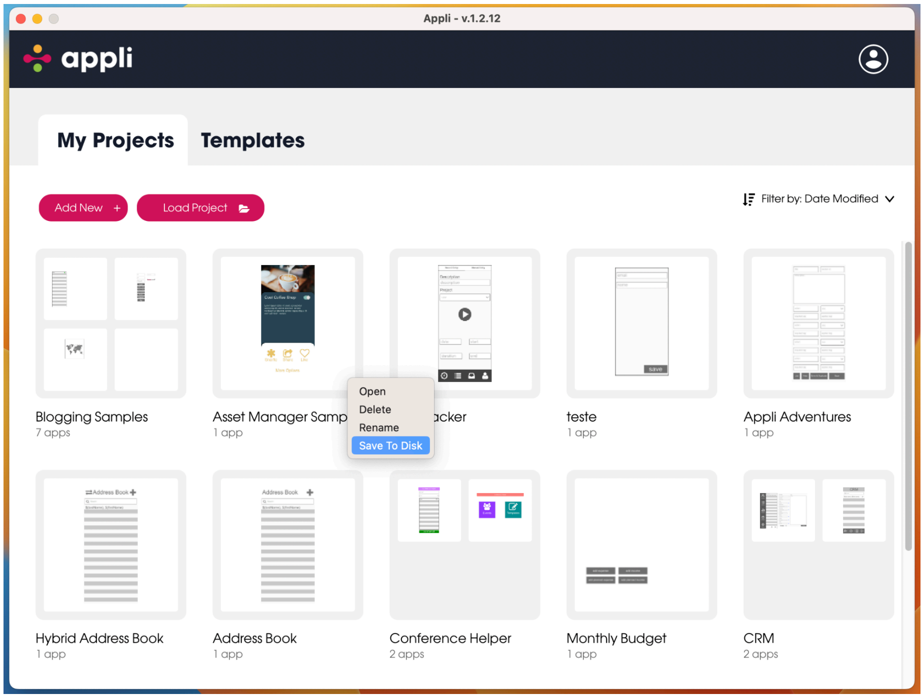


Figure 10: Saving a project to disk

Appli will ask you if you want to save the project using a password. If you do that, only people with the password will be able to open it.

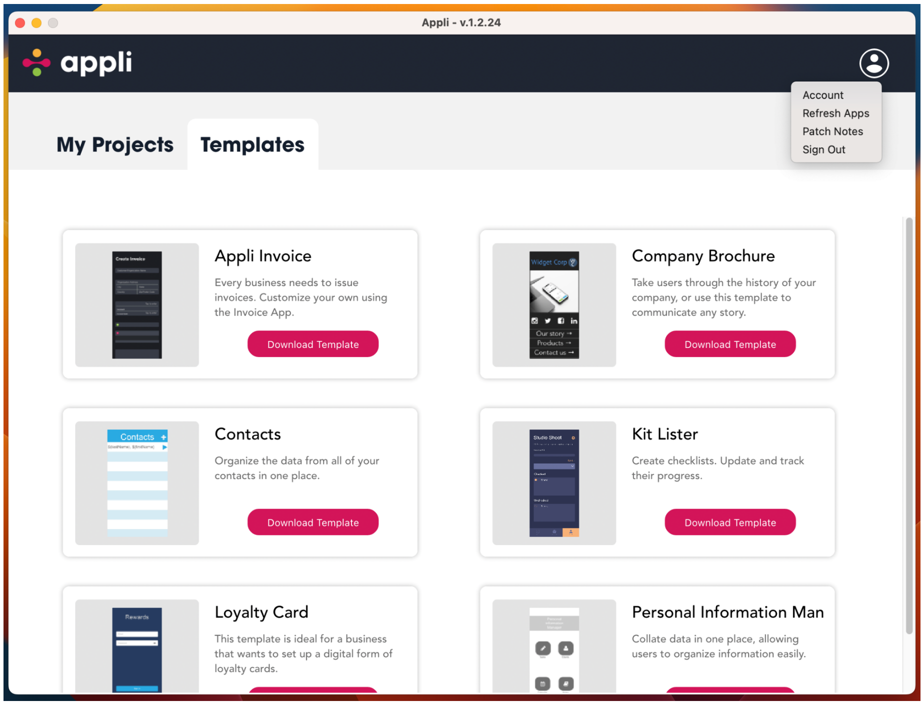
3.3 Opening a project

Use the *open folder button* (just next to the *plus button*) to open a project file. If the project was saved using a password, you'll need to enter it before being able to import the project.

3.4 Templates

The templates tab gives you access to ready-to-use applications from our sample code collection. They are a great way to learn how to use Appli.

The account menu on the top right corner can be used to go back to login, check the patch notes, and refresh the projects and templates listing.



Appli Invoice

Every business needs to issue invoices. Customize your own using the Invoice App.

Company Brochure

Take users through the history of your company, or use this template to communicate any story.

Contacts

Organize the data from all of your contacts in one place.

Kit Lister

Create checklists. Update and track their progress.

Loyalty Card

This template is ideal for a business that wants to set up a digital form of loyalty cards.

Personal Information Man

Collate data in one place, allowing users to organize information easily.

Figure 11: Templates and account menu

4 Overview of Appli Builder Interface

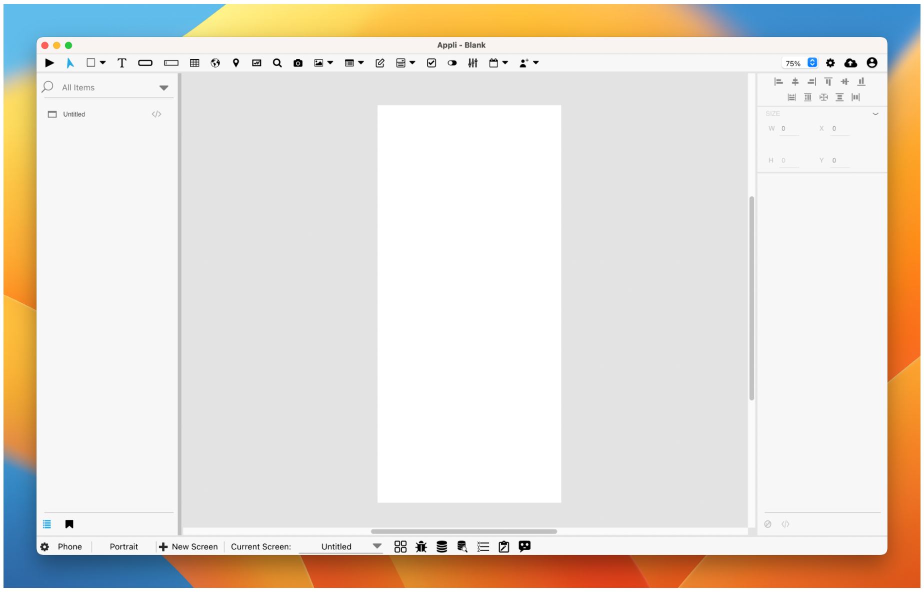


Figure 12: Playground Screen

Welcome to the primary interface of Appli Builder. This is where you'll spend most of your time as a developer working on your next awesome project. This section will help you become familiar with the various features and workflows inside the playground which is the screen you use to design your application.

The playground screen is divided into five regions: tools palette, left pane, playground, property inspector, and footer. In the next sections, we'll provide a high-level overview of each of these regions with links to dive deeper into each of them.

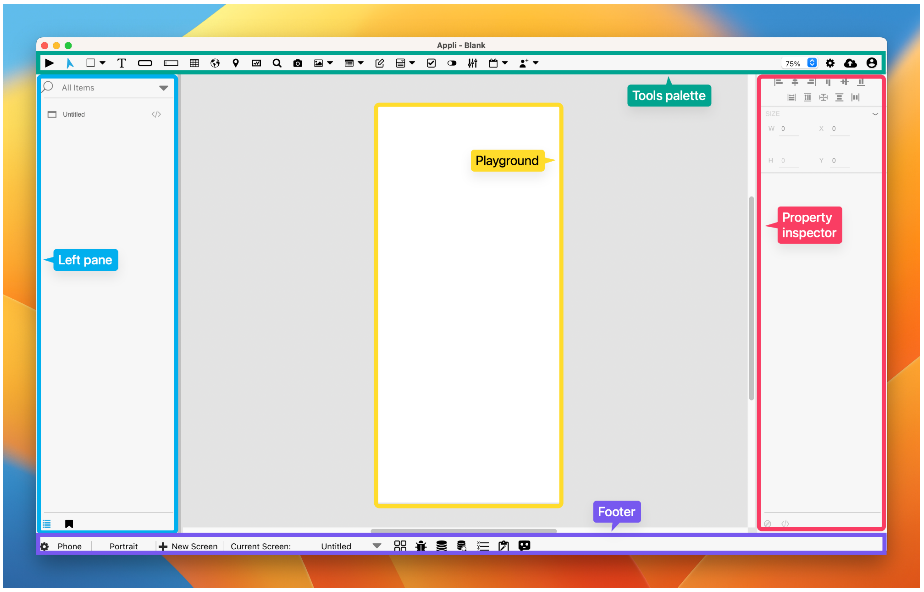


Figure 13: Playground Regions

4.1 Tools Palette

The tools palette on the top of the interface holds the *elements* that the developer uses to build their own application. To add an element to the playground, first select the element, and then use the mouse to create it by clicking and dragging a rectangle that represents its dimensions on the playground.

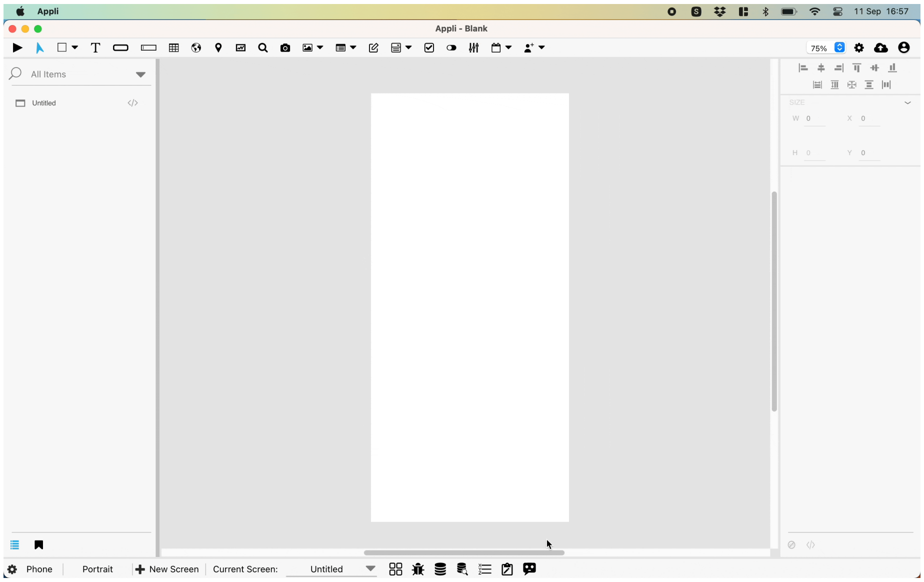


Figure 14: Adding an element to the playground

On the right side of the tools palette is a collection of buttons to enable to you to:

- Save your project on the cloud.
- Change application-wide settings.
- Handle workflow related tasks such as signing out, going back to the home screen,.

To learn more about which elements are available for you, check out the elements gallery.

Check out the Tools Palette documentation to learn more about the tools palette itself.

4.2 Playground

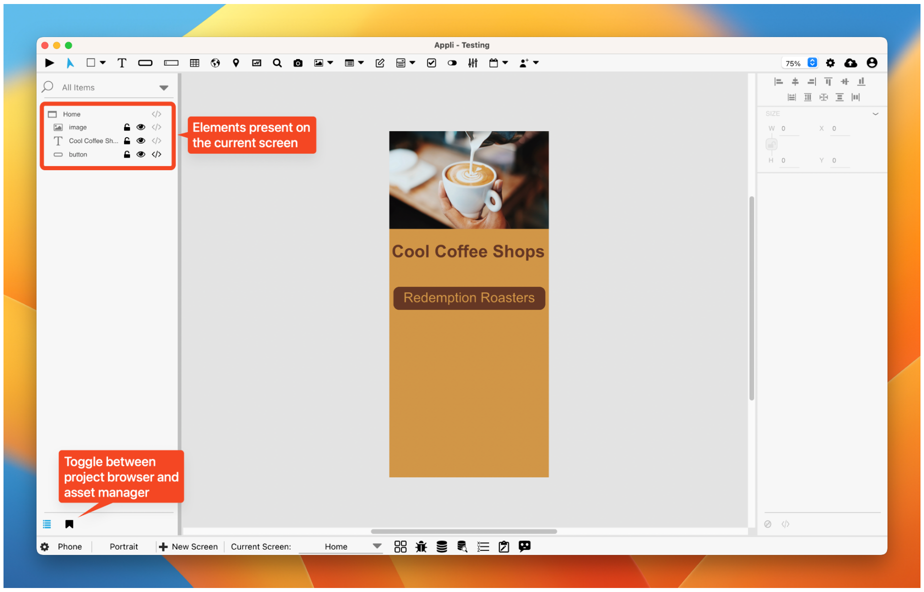
The playground is the most important region in the Appli IDE interface, which is why we call this screen *the playground screen.* This is the canvas in which the developer will build their application.

4.3 The Left Pane

The left side of the IDE hosts two important tools: the project browser and the asset manager. To switch between them, use the buttons at the bottom of the pane. The first button switches to the project browser, the second selects the asset manager.

4.3.1 Project Browser

The project browser lists all the elements on the current screen as a hierarchical list. Some elements are containers. In such cases, the elements they contain will appear under them on the list.



Elements present on the current screen

Toggle between project browser and asset manager

Figure 15: Project browser

A search box is provided in case you need to filter the list of elements. For each element on the list, you can:

- Use the *padlock button* to lock the element and prevent accidental changes to its properties.
- Use the *eye button* to toggle the visibility of the element.
- Use the *brackets button* to change the low-code/no-code flow for the element.

4.3.2 Asset Manager

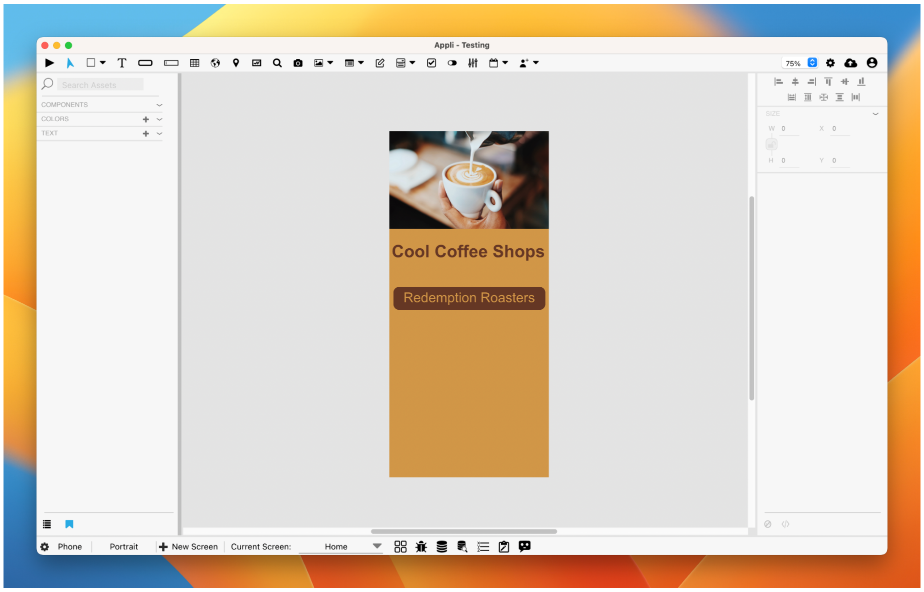


Figure 16: Asset manager

Use the asset manager to define a theme for your application. You can select default colors and text styles that will apply to all elements in the app.

To learn more about this feature, check out the asset manager documentation.

4.4 Property Inspector

On the right side of the IDE, you'll find the property inspector. Elements are configurable via properties. You can change various characteristics and features by changing values in the property inspector. It is through the property inspector that the developer can change the behavior of an element, something that you can learn more by diving deeper into the property inspector.

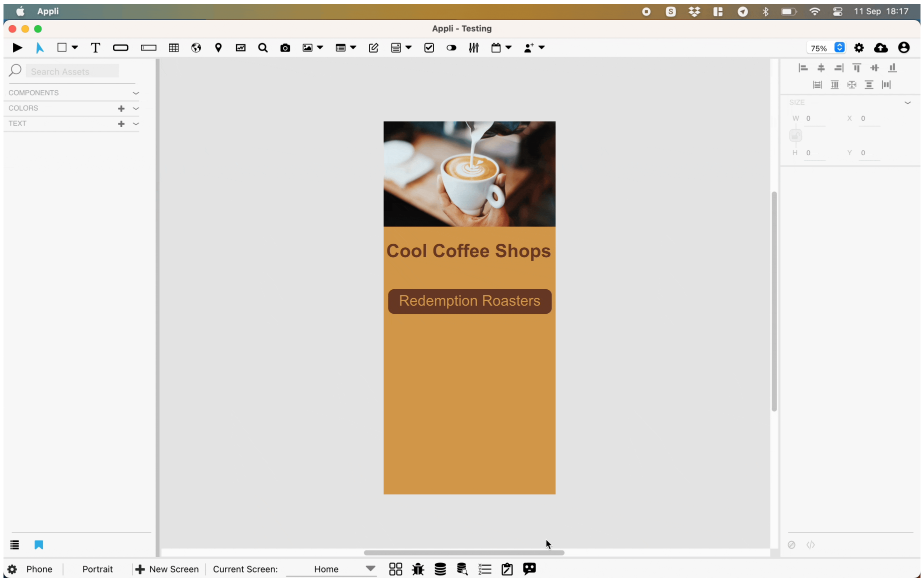


Figure 17: Changing properties on a button

4.5 Footer

The footer is at the bottom of the interface. In that region are many useful tools to manage plataforms, screens, data, variables, and even get help from our AI.

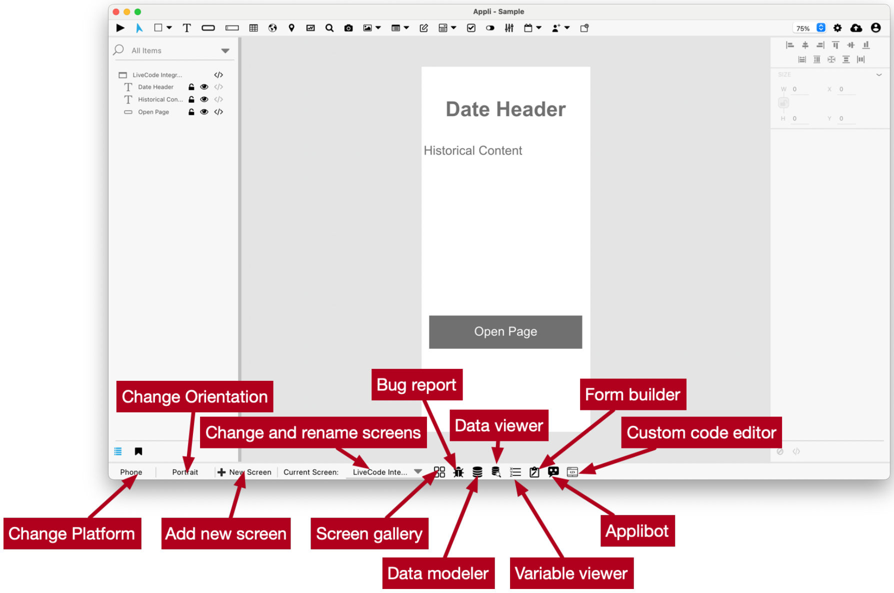


Figure 18: Footer features at a glance

Read the Footer documentation to learn more about the various tools.

5 Tools palette

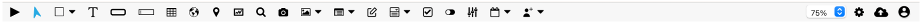


Figure 19: Tools palette

The tools palette can be understood as serving three different broad roles:

- At the right side, we have project management tools.
- Most of the buttons in the center of the palette are *elements* used to construct the application.
- The first button in the palette is a play tool used toggle between interacting or designing the app.

5.1 Project management

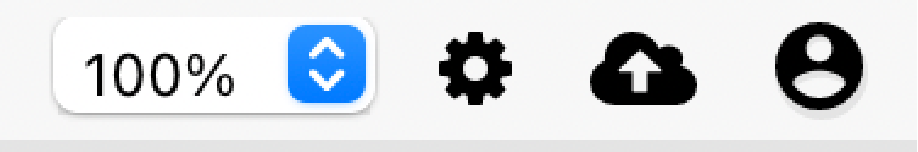


Figure 20: Project Management

These buttons deal with workflows related to your project and your Appli account.

The *gear button* opens the application-wide settings. Use it to change the app name, icon, and some other general settings. Read the application settings documentation for more details.

The *cloud button* saves your application to the cloud.

The last *user button* is actually a dropdown navigation menu with multiple features. It can be used to *sign out* of Appli; go back to the project selection screen; and view the most recent *patch notes*.

5.2 Testing the app

The first button on the toolbar, the one that looks like a *play button* places Appli IDE in *interaction mode*. It disables both the left and right panes and lets the developer interact with the application as if it was running inside the player. To switch back to *design mode*, click it again.

5.3 Elements

Most of the rest of the palette is dedicated to hosting the *elements* used for app designing.

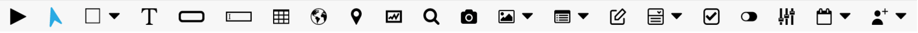


Figure 21: Elements

To draw an element in the playground, first select it on the tools palette and then draw it.

The *pointer tool*, which is the button next to the play button, can select and alter elements already on the playground.

Check out the elements gallery to learn more about each element.

6 Footer

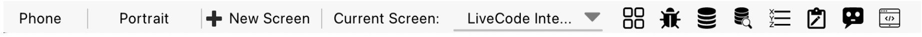


Figure 22: Footer

The footer of Appli builder contains many tools and quick information about the project you're working on.

6.1 Platform indicator

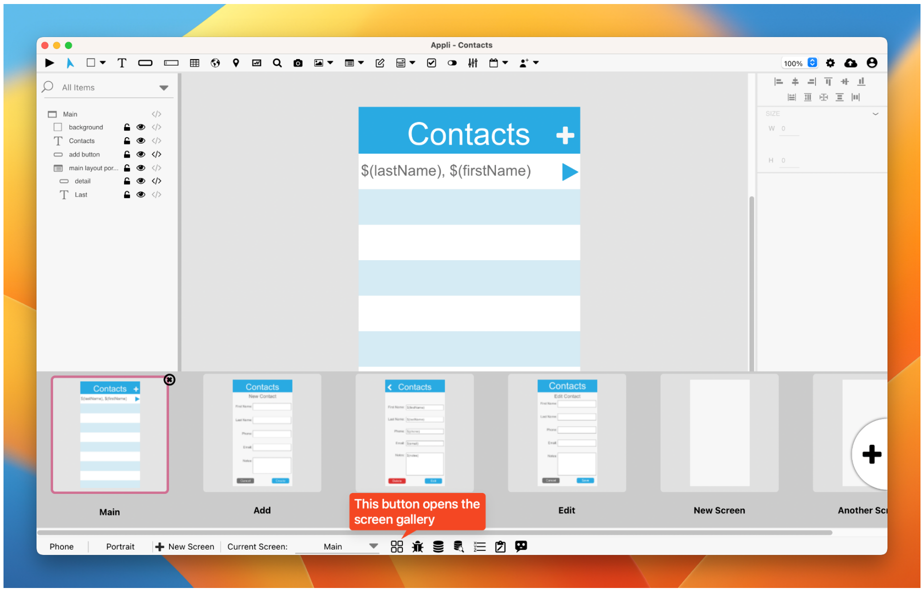
The first item in the footer is an indication of the current platform. In the screenshot above, it indicates it is a phone.

6.2 Screen management section

Following the platform indicator, there multiple items related to screen management. Appli keeps separate versions of the screen for portrait and landscape orientation, so the first item in the screen management section is a quick way to glance which orientation you're working on and also switch between them with a click.

A handy *+ New Screen* button is provided to quickly create a new screen.

The indicator for the current screen shows its name which can be modified by double-clicking it. That label is also a dropdown menu to switch between screens. Clicking it or the four squares next to it will open the screen gallery to switch and reorder screens.



This button opens the screen gallery

Figure 23: Screen Gallery

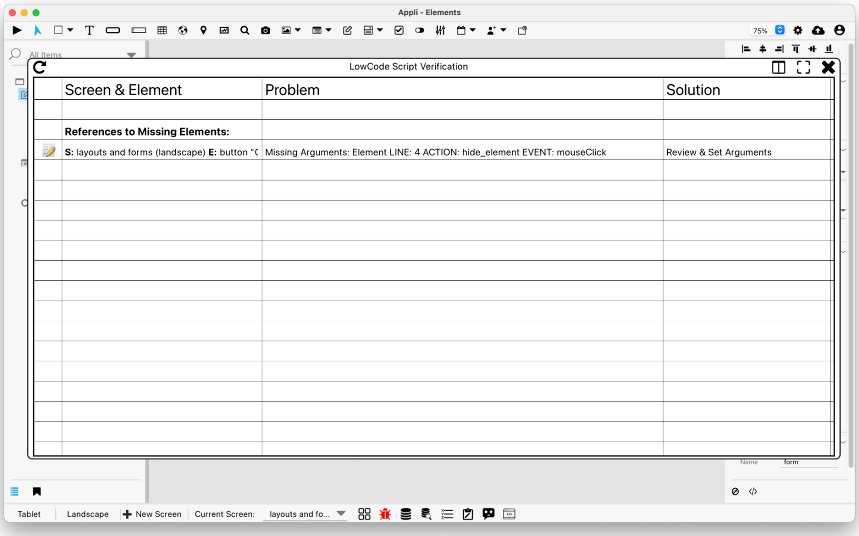
To learn more read the screen management chapter.

6.3 Tools section

All the icons after the screen gallery one are different tools to be used while developing and debugging your app. Each of them opens in their own window.

6.3.1 Low Code Issues tool

The icon is a little bug. You click on it to see any potential low-code issue. If Appli detects a low-code issue on its own, that icon will turn to red to draw your attention to it.



	Screen & Element	Problem	Solution
	References to Missing Elements:		
📝	**S:** layouts and forms (landscape) **E:** button "(	Missing Arguments: Element LINE: 4 ACTION: hide_element EVENT: mouseClick	Review & Set Arguments

Figure 24: Low Code Issues Tool

The tool has three columns:

1. Lists the element that has a problem. Next to it is a handy button shaped like a paper and pencil to go edit the buggy low-code action.
2. A description of the problem.
3. The proposed solution to the problem.

6.3.2 Data Modeler

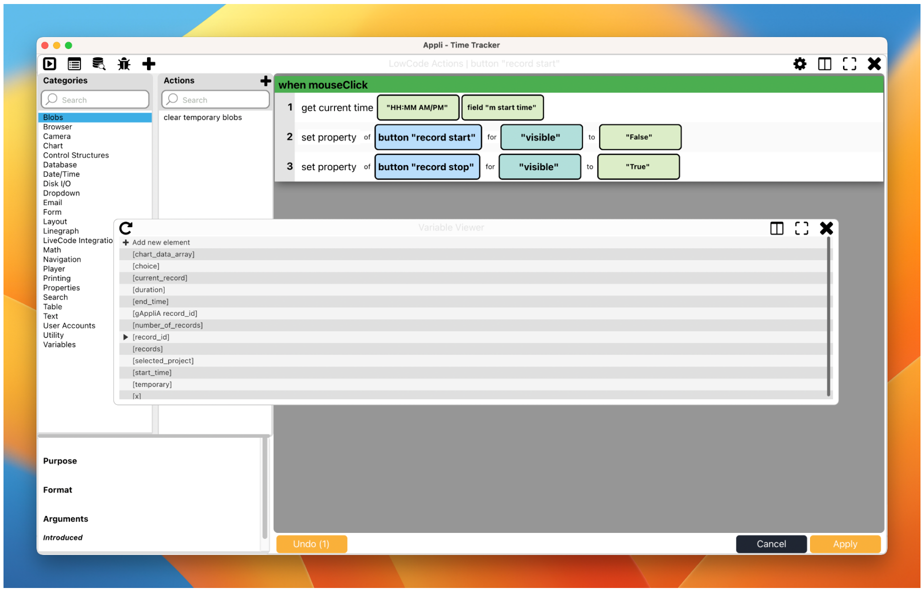
Its icon is a stack of disks. It is used to manage your databases. To learn more about it, double check the data management chapter and the database tutorial.

6.3.3 Data Viewer

A companion to the Data Modeler, its icon is a similar stack of discs but with a magnifying glass next to it. It is used to visualise the data you have on your tables.

6.3.4 Variable Viewer

Its icon is a list of items. Use it to view and alter the contents of variables in your app.



when mouseClick

1 get current time "HH:MM AM/PM" field "m start time"

2 set property of button "record start" for "visible" to "False"

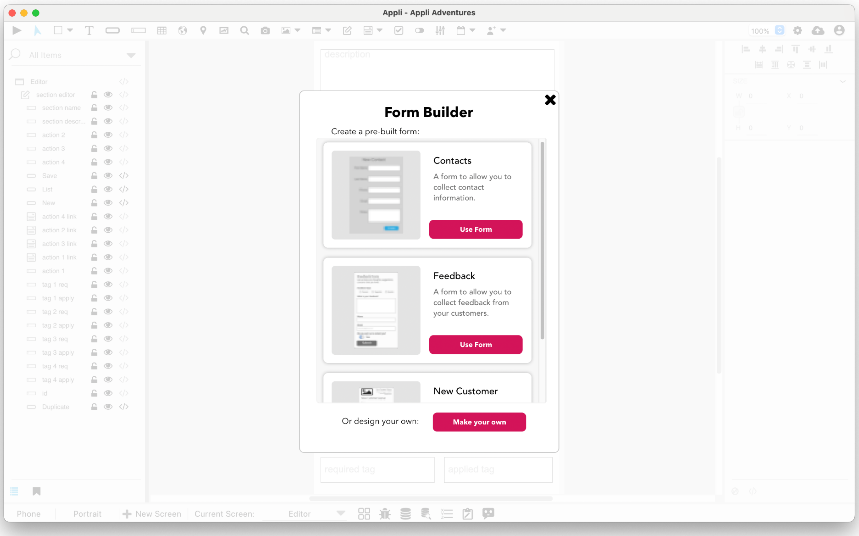
3 set property of button "record stop" for "visible" to "True"

[chart_data_array]
[choice]
[current_record]
[duration]
[end_time]
[gAppliA record_id]
[number_of_records]
[record_id]
[records]
[selected_project]
[start_time]
[temporary]
[x]

Figure 25: Variable Viewer

6.3.5 Form Viewer

With an icon shaped like a clipboard, the form builder is the easiest way to create a form using Appli. Learn more about it in the database tutorial.



Form Builder

Create a pre-built form:

Contacts

A form to allow you to collect contact information.

Feedback

A form to allow you to collect feedback from your customers.

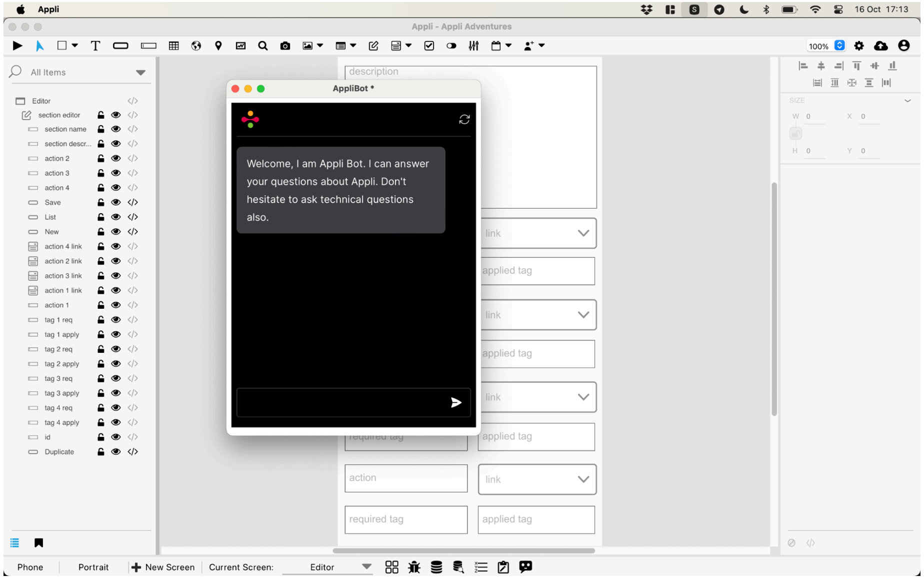
New Customer

Or design your own:

Figure 26: Form Builder

6.3.6 Applibot

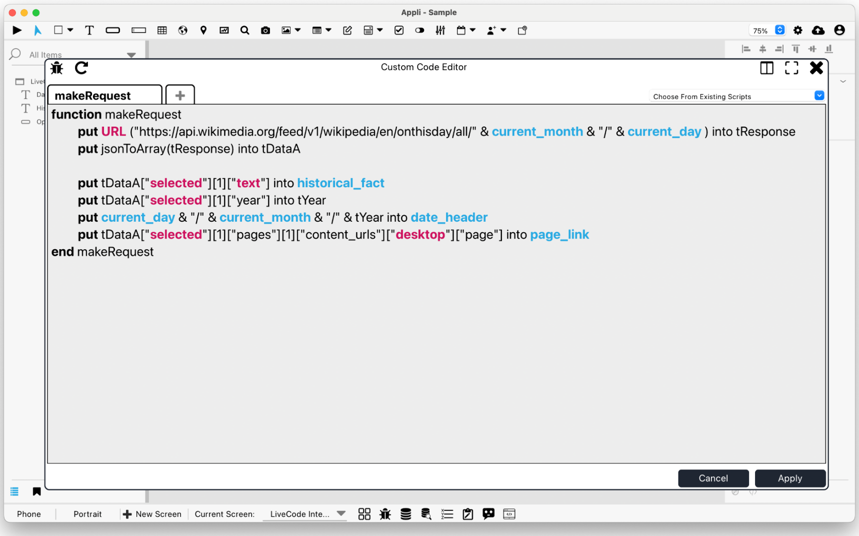
Our handy AI assistant is the final tool in the footer. Shaped like a friendly speech ballon, you can click it to open the chatbot and ask questions about how to solve tasks with Appli.



Welcome, I am Appli Bot. I can answer your questions about Appli. Don't hesitate to ask technical questions also.

Figure 27: Applibot interface

6.3.7 Custom Code Editor



```
function makeRequest
      put URL ("https://api.wikimedia.org/feed/v1/wikipedia/en/onthisday/all/" & current_month & "/" & current_day ) into tResponse
      put jsonToArray(tResponse) into tDataA

      put tDataA["selected"][1]["text"] into historical_fact
      put tDataA["selected"][1]["year"] into tYear
      put current_day & "/" & current_month & "/" & tYear into date_header
      put tDataA["selected"][1]["pages"][1]["content_urls"]["desktop"]["page"] into page_link
end makeRequest
```

Figure 28: Custom code editor showing a LiveCode-based function

A custom code editor to create your own LiveCode-based functions. Check out the Custom Code Editor documentation for more information and the LiveCode Integration tutorial to build a sample app that uses it.

7 Keyboard Shortcuts

Appli has extensive keyboard shortcuts support. Learning them can speed up your development process.

7.1 General shortcuts

Key Combination	Function
V	hide/show thumbnail view.
Shift-O	Alternate orientations.
o	hide/show off screen elements.
ESC	toggle Pointer/Player mode.
Shift-P	Engages pointer mode when you had previously selected an element.

7.2 Copy/paste/cut operations

Key Combination	Operation
CMD-C	Copy.
CMD-V	Paste.
CMD-X	Cut.

7.3 Selection manipulation

Key Combination	Function
CMD-A	Select all.
Shift-CMD-A	Deselects all.

7.4 Undo/Redo management

Key Combination	Function
CMD-Z	Undo.
Shift-CMD-Z	Redo.

7.5 Left Pane

Switching between the various tabs on the left pane can be done with simple keyboard shortcuts.

Key Combination	Function
CMD-D	show data manager.
Shift-CMD-Y	show asset manager.
CMD-Y	show project browser.

7.6 Guidelines

Creating a good design might require one to use multiple combinations of snapping, guidelines, and measuring. These shortcuts will help you handle guideline management with ease.

Key Combination	Function
Tab	Show distance to screen edge.
Space	Do not use guidelines and snapping.
G	Toggle Fixed Guidelines.
~	Prevents elements from snapping to the normal guidelines.
Option/Alt	Displays measurements when an element is selected.

7.7 Elements selection

Learning the element keyboard shortcuts will speed up your development workflow. Quickly change what element type is selected without the need to move the mouse to the toolbar.

Key Combination	Element
a	create account.
b	button.
c	camera.
d	dropdown menu.
e	ellipse.
f	label/text field.
h	header.

Key Combination	Element
i	image.
k	log in.
l	line.
m	map.
r	rectangle.
s	search.
t	input Field.
u	tab menu.
x	switch button.
Shift-B	browser.
Shift-F	form.
Shift-H	footer.
Shift-L	layout.
Shift-M	media.
Shift-R	radio button.
Shift-T	table.

8 Screen Management

A project can contain multiple apps, one per platform. Each app can contain multiple screens in different orientations. That is a mouthful to say that an Appli project is very flexible. You can build apps for various platforms — desktop, tablets, and smartphones — in a single project, and each app will have their own screens.

8.1 Creating a new screen

When you first create a project and an app, you select an initial platform and orientation for the app. The playground screen will open with that platform and screen selected. As your project grows, you'll may need to add more screens. The controls to do that are at the *footer*.

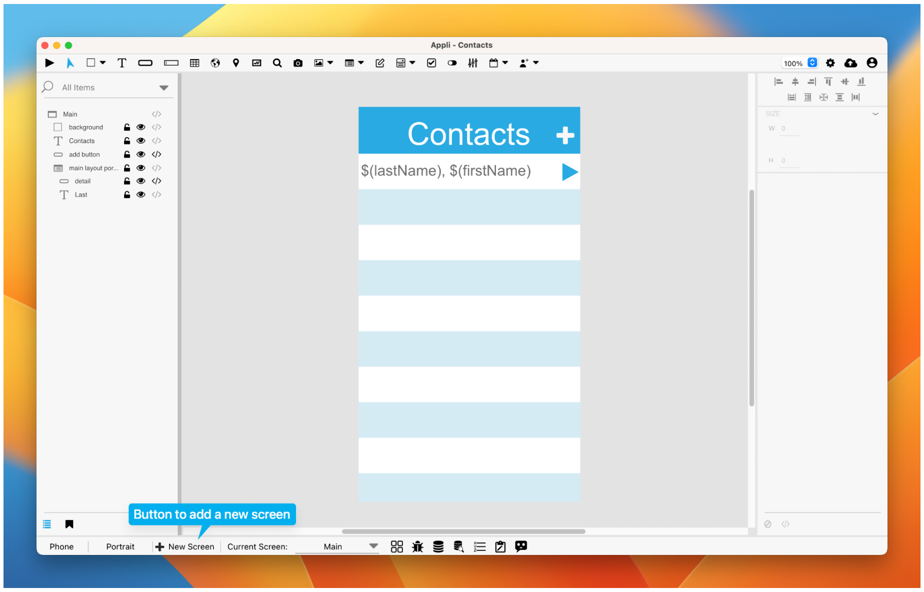


Figure 29: Button to add a new screen

After clicking that button, Appli will ask for the name for this new screen. The playground will automatically switch to the newly created screen as shown in the *current screen display in the footer*. That display is a pop-up menu that allows the developer to switch between the screens in the current platform.

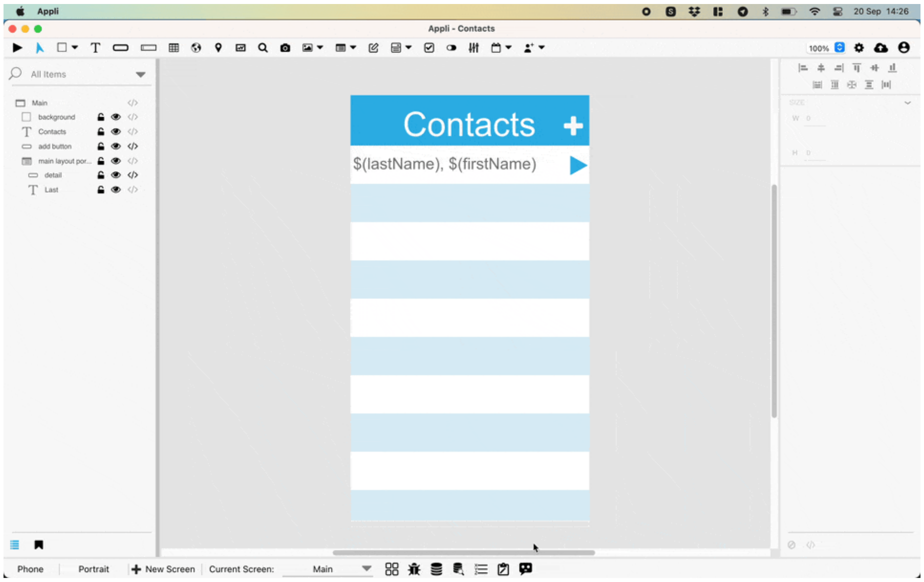
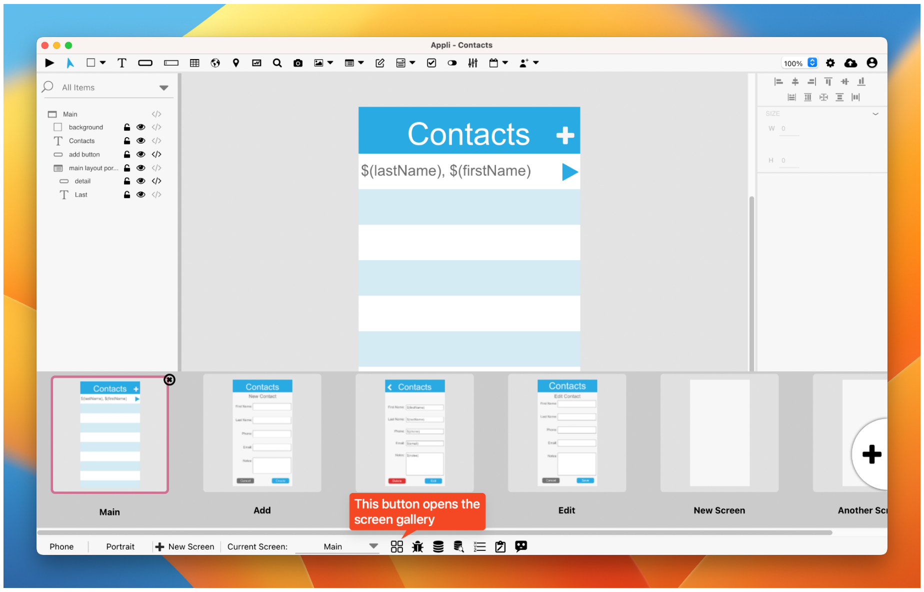


Figure 30: Adding a new screen to a project

8.2 Switching between screens

Besides the pop-up menu shown above, there is another control that opens a gallery of screens thumbnails that makes it easier to find the screen you want.



This button opens the screen gallery

Figure 31: Screen gallery

> **Important:** It is in this gallery view you can delete a screen. Use the little *crossed circle button* in the corner of the current screen thumbnail to delete it. You can also use the large *plus* button at the right side of the gallery to create a new screen.

8.3 Changing orientation

Each screen can have independent versions for each orientation. You can switch between them by using the orientation button on the footer. Both orientations are versions of the same screen and the player will select the correct when it comes time to run the app later.

8.4 Final remarks & What to read next

As we have shown, all the controls to manage both screens and platforms are at the *footer*. By now, you should be able to:

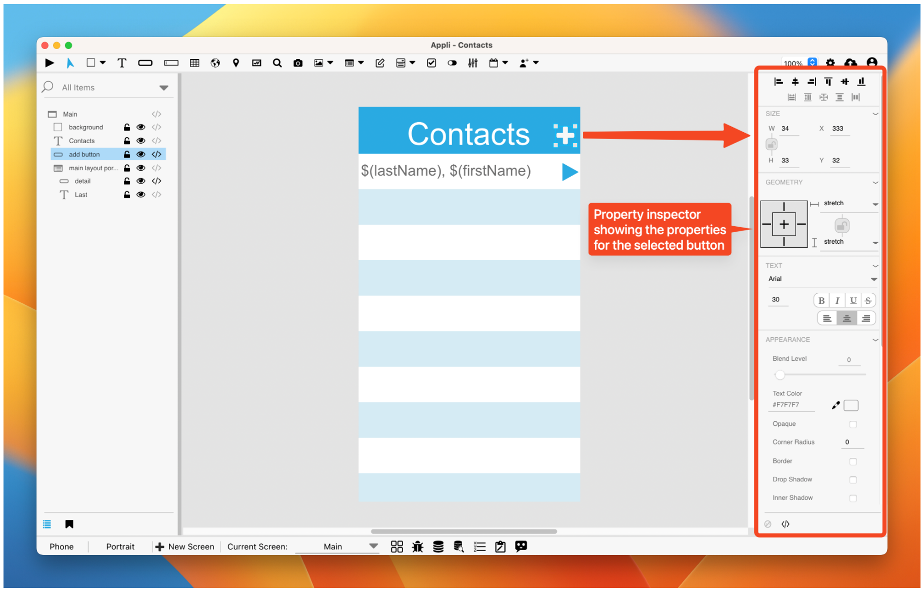
- Create new screens.
- Create versions of a screen for different orientations
- Switch between screens.

Even though you can create as many screens as you want, you still need to learn about responsive design to make sure that your screens can adapt to different resolutions. A screen that was designed for an iPhone 15 Pro Max might not be suitable for a smaller Android device. Using responsive design, you can configure elements to adapt to different resolutions and make sure your app always looks perfect.

9 Property Inspector

In Appli, one builds their app by creating and customizing elements on a screen. Each element has their own characteristics and behaviors. We refer to these characteristics as *properties*. To change them, the developer uses the *property inspector* in the right-side of the screen.

To view the properties of an element, select it using the *pointer tool*. Each type of element has properties that are suitable for them. For example, a *map element* will have a property to specify geolocation markers. The other elements have no need for such property. You won't see it on them. Some properties such as labels and colors are quite common and you'll find them over and over as you work in your app. All this will become second nature as the developer gets familiar with Appli.



Property inspector showing the properties for the selected button

Figure 32: Property Inspector

The interface of the property inspector is divided into sections that always appear in the same order regardless of the selected element's type.

9.1 Alignment tools

Placed at the very top of the property inspector are the alignment tools.

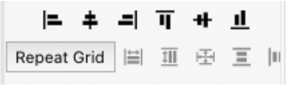
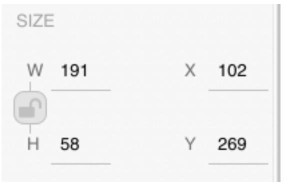


Figure 33: Alignment tools

These tools help the developer align their elements on the screen. Use them when you want to align an element based on the screen borders or other elements.

9.2 Size properties

These are quick ways to change the position and dimensions of an element.


SIZE

W 191 X 102

H 58 Y 269

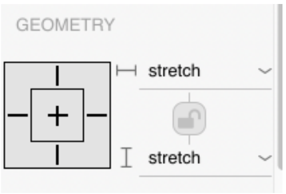
Figure 34: Size properties

There is a handy padlock that enables you to lock the dimensions of an element so that you don't resize them by accident.

9.3 Geometry properties

There are many variations of screen size and resolution for each platform when we factor in the thousands of devices that exist out in the world. A good example is smartphones. They come in various sizes and aspect ratios, even though most of them still qualify for a *portrait smartphone* category.

The geometry properties let the developer configure how an element should behave when the screen size differs from the one being used to design the application. Elements can grow, shrink, etc. To learn more about how to craft resolution independent screens, head on to the responsive design documentation.


GEOMETRY

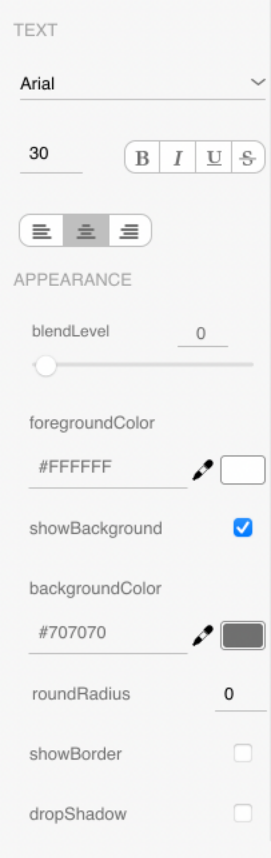
stretch

stretch

Figure 35: Geometry properties

9.4 Text and appearance

These represent all the properties that govern the appearance of an element. Colors, graphic effects, shadows, labels, all are a part of this text and appearances section. Be aware that these can vary depending on the element selected. For example, elements without text will not contain a text section.



Arial

30

blendLevel 0

foregroundColor

#FFFFFF

showBackground ☑

backgroundColor

#707070

roundRadius 0

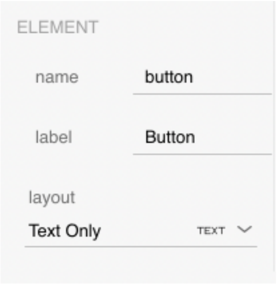
showBorder □

dropShadow □

Figure 36: Text and appearance properties

9.5 Element specific

The properties that are specific for the type of element selected are displayed in a *element specific* section. To learn more about the properties of each kind of element, check out the element gallery.



ELEMENT

name: button

label: Button

layout: Text Only

Figure 37: Element specific properties

9.6 No-code and low-code

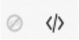


Figure 38: no-code and low-code

These are small buttons at the end of the panel. Some properties can't fit well in a sidebar like this, so you can access them using these buttons.

Some elements are very complex, such as the *camera element*, to access things such as the camera output, use the first button, which is the *no-code* button.

The *low-code* button is used to configure flows. These are the behaviors attached to an element. You can learn more about them by reading the documentation on flows and behaviors.

10 Responsive Design

Computing devices come in various form-factors — from desktop computers to small smartphones — being to adapt and serve your customer is a crucial feature for any application.

No one wants to use an application that was designed for a smartphone and presents itself with the same smartphone-focused user interface on the desktop or tablet. The reverse is also true: an application designed for a desktop is usually a poor citizen on a small smartphone.

Responsive design is the technique of creating designs that adapt to run well in device.

To achieve that, Appli uses three broad platform categories: desktops, tablets, and mobiles. Your design for each of those categories is independent from the others.

Even with per-platform designs, one might worry that many devices in a single category are still too different from one another. While that is definitely true, the properties in the *geometry section* of an element allow the developer to set how the element adapts to different screen sizes. Mobile phones might come in different resolutions and proportions, but they're all just phones: one screen that you interact using touches. Desktops and laptops have different screen sizes, and you mostly interact with them using keyboard and mouse. Tablets sit in between them both and deserve a special design that leverages the best of both worlds.

In this chapter, we're going to learn how to create independent designs for each platform and how to make effective usage of the geometry-related properties to make sure our application shines in all devices.

10.1 Creating designs for each platform

When you are creating a new application, you need to choose the device category you want to work with. Do that by clicking on the desired device on the platform selection screen_:

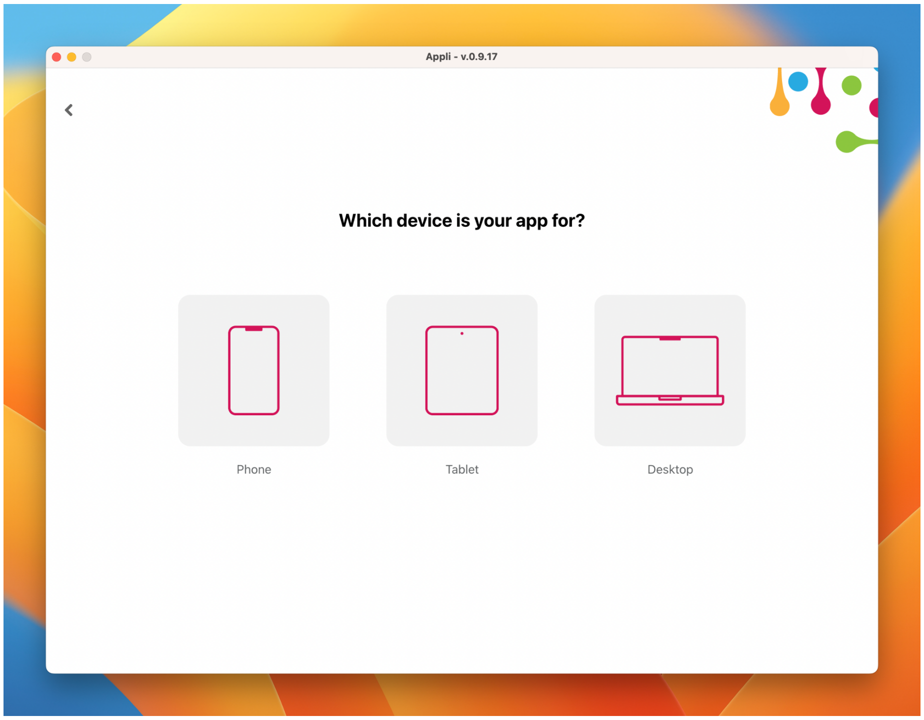


Figure 39: Platform selection screen showing mobile, tablet, and desktop device categories.

Once you complete your application for that category, you can add additional applications with different platforms to your project.

> Be aware that tablets and mobile applications can adapt to orientation changes. Use the orientation toggle at the footer next to the platform indicator to switch between *portrait* and *landscape*. Each orientation design is independent from the other.

10.2 Understanding the geometry properties

Besides creating new designs for each combination of platform and orientation, one should use the properties inside the *geometry section* to configure how an element should behave when the screen does not match the resolution that was used for the original design of the application. A typical case is when you design a mobile app for an iPhone and open it on a Samsung Galaxy Note. They are both phones, but they have different proportions and screen resolutions.

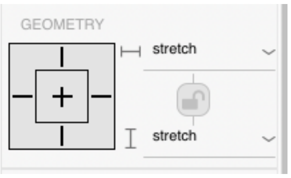


Figure 40: Geometry section as seen in the property inspector.

There are two settings: `responsive-x` and `responsive-y`. They are used to configure how the element behaves when there are changes to the width and the height of the screen.

The values for `responsive-x` are:

- left: move element to maintain its distance to the left side of the screen.
- right: move element to maintain its distance to the right side of the screen.
- left and right: same as having both left and right set.
- stretch: grow or shrink the element as needed.
- center: don't resize the element but center it considering the changes to the screen width.

The values for `responsive-y` are:

- top: move element to maintain its original distance from the top of the screen.
- bottom: move element and maintain its original distance from the bottom of the screen.
- top and bottom: the same as setting both values above.
- stretch: grow or shrink the element as needed.

- center: don't resize the element but center it vertically considering the changes to the screen height.

The values can be changed using the dropdown menus in the section. The visual guide next to them shows the current selection and can also change the value of those properties. Just click on the corresponding line.

Stretch is the default value used for all elements. It means that the object will grow or shrink to cope with the changes in width and height while preserving its aspect ratio.

To make it easier to visualise how these values change how an element behaves, consider the screen below. It was designed for the iPhone XR inside Appli.

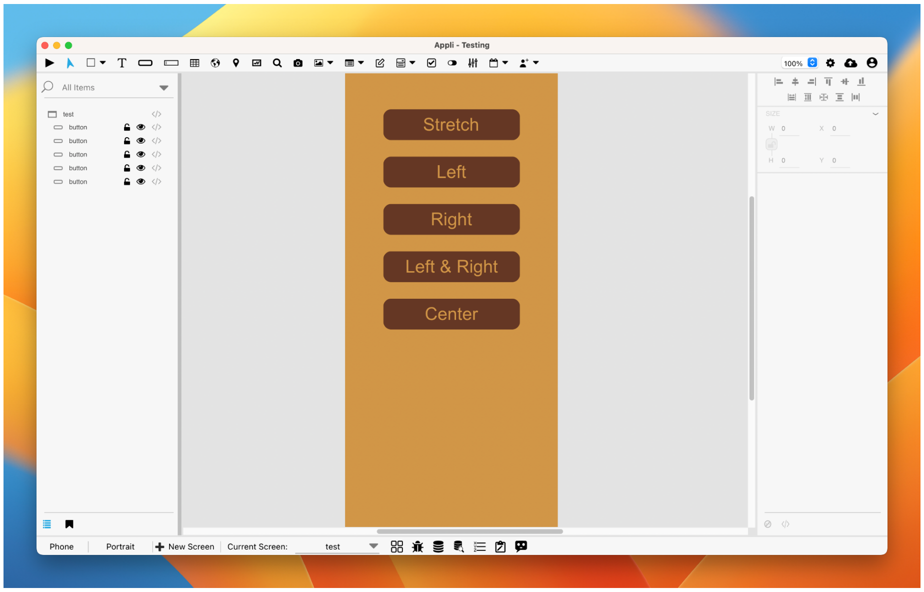


Figure 41: A screen designed for the iPhone XR.

Notice how all buttons have the same size. The label of each button represents the value of the `responsive-x` property. Once I open it in Appli player using a different phone, in this case an iPhone 12 Mini, look at how each button changes according to their `responsive-x` property.

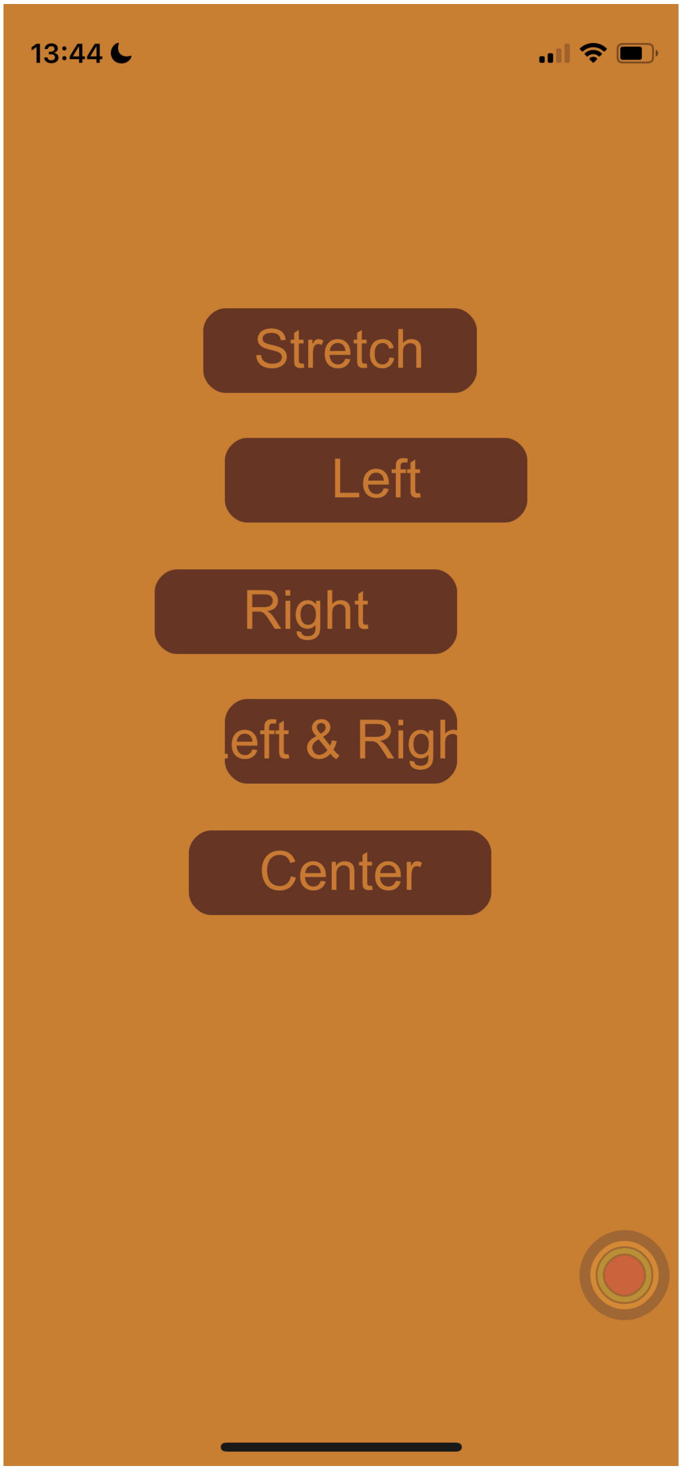


Figure 42: That screen as seen in an iPhone 12 Mini.

As you can see, the default behaviour which is using `stretch` is the best option. With Appli, you get responsive design by default and you only need to change something when you have niche needs.

Oh, and in case you're wondering how their vertical position worked out of the box, that is because they're all have the `responsive-y` property set to `stretch`.

10.3 Tips and Tricks

- When designing a user interface, consider what kind of input method your user will be using. A finger is less precise than a mouse, so larger buttons make for easier targets in mobile platforms while smaller buttons allow you to make the best use of screen real estate on a desktop platform.
- When in doubt, set it to `stretch`.
- All tablet users have access can use touch input. Some of them might have keyboards or even a mouse connected to their tablet. Make sure your tablet design can work with just touch, but make sure that users leveraging keyboards and mouse will benefit from them.

11 Low-Code & Action Scripts

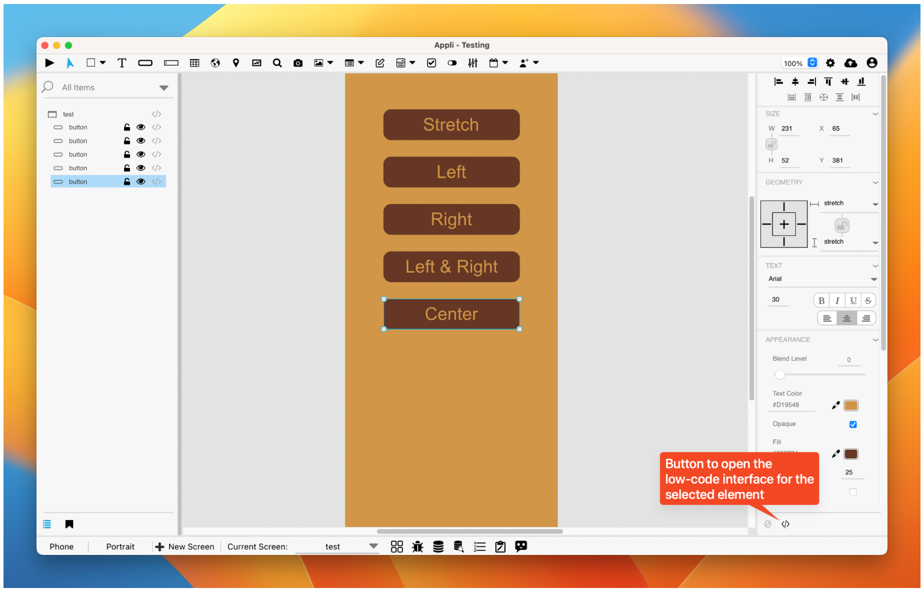
So far, you have learned how to design your application by configuring platforms, creating screens, and laying out and customizing elements. That can get you very far, but to fully leverage the power of Appli, you need to understand how low-code works.

Most development workflows involve the developer typing our commands to craft their source-code in a way that is understood by that programming language. It is an error-prone process in which a typo can render a whole project impossible to compile or worse, it might render the final product buggy in a way that the consequences only surfaces later in its lifetime. Learning a new programming language requires a lot of effort and focus, not that different from learning any new natural language. It is a rewarding process, but given how it works, it has a huge gap between first being exposed to the programming language and actually being an effective developer using that language.

Appli makes this gap smaller by being a next generation development environment. It doesn't force the developer to write their own source-code. Appli features an interactive development paradigm using a graphical interface; Appli writes the source-code for you. This means that an Appli developer doesn't need to memorize all commands using an exotic language before they can be effective. Our interactive low-code system helps you learn how to be a developer and enables you to craft complex programs in a fraction of the time that it would take to do the same task in a previous generation programming language.

11.1 The Low-Code interface

Low-Code is the method used to add behaviours to an element. The property inspector allows one to configure an element. At the bottom of the property inspector, you'll see the Low-Code button, which opens the Low-Code interface for the selected element.



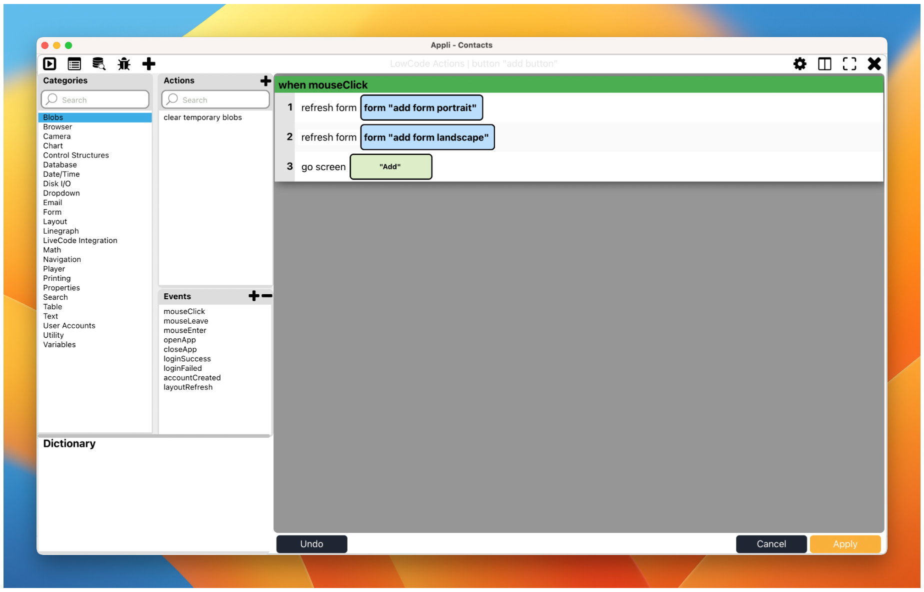
W 231
X 65
H 52
Y 381
stretch
stretch
Arial
30
Blend Level 0
Text Color #D19548
Opaque ☑
Fill
25
Current Screen: test

Button to open the low-code interface for the selected element

Figure 43: Button to open the Low-Code interface

> Let's quickly define what we mean by *behaviour*. We're using it as a loose term to mean whatever happens when the user interacts with the element.

The Low-Code interface is divided into distinct sections that are easy to identify: categories, actions, events, action script, and help.



```
when mouseClick
refresh form form "add form portrait"
refresh form form "add form landscape"
go screen "Add"
```

Figure 44: The Low-Code Interface

By selecting and configuring *actions* and *events*, you create *action scripts* which dictate what happens when a user interacts with the element.

11.2 Events

Events are the trigger for *actions*. When an *event* occurs, it activates the *action* you selected. The most common event is *mouseClick* which is when the user clicks on an element using the mouse. If you don't select an *event* from the list, Appli will default to asking if you want to use *mouseClick*.

To learn more about events, check out the events chapter.

11.3 Actions

Actions are grouped using *categories*. Select a *category* to view all actions available in that group. Select an *event* and click on an *action* to add it to the *action script* on the right side of the interface. The selected *action* might contain some underlined text. That means that part of the *action* is an *argument*. To complete the action, click that *argument* and configure it.

While the action is not properly configured, the *action script* will display with a red tint in the background. Once you set all the arguments, that tint will change to green. This is a quick way to detect mistakes in your script.

On the bottom-left you'll see the help section for the selected *action* explaining all about it and what arguments it uses. Checking that section is a wonderful way to learn all the actions you can use.

11.4 Example: navigating between screens when a button is clicked.

When you open the Low-Code interface for a button, you'll see an empty *action script*.

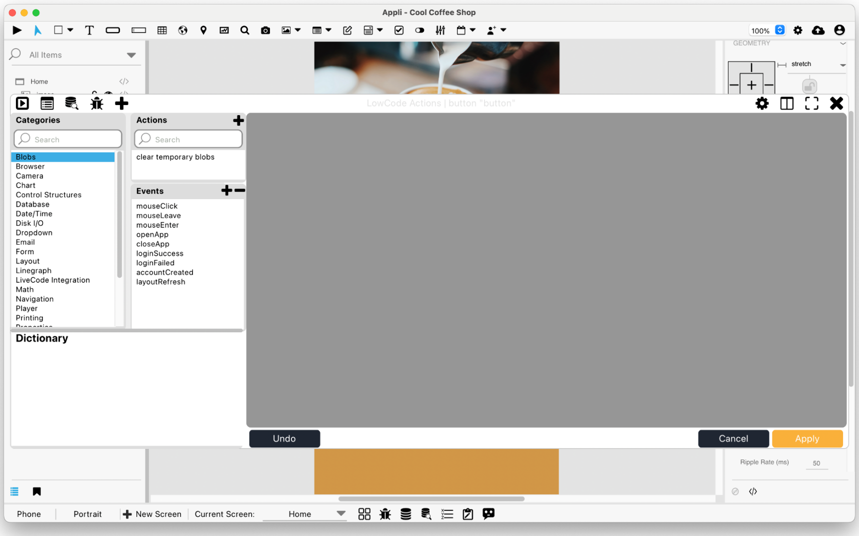
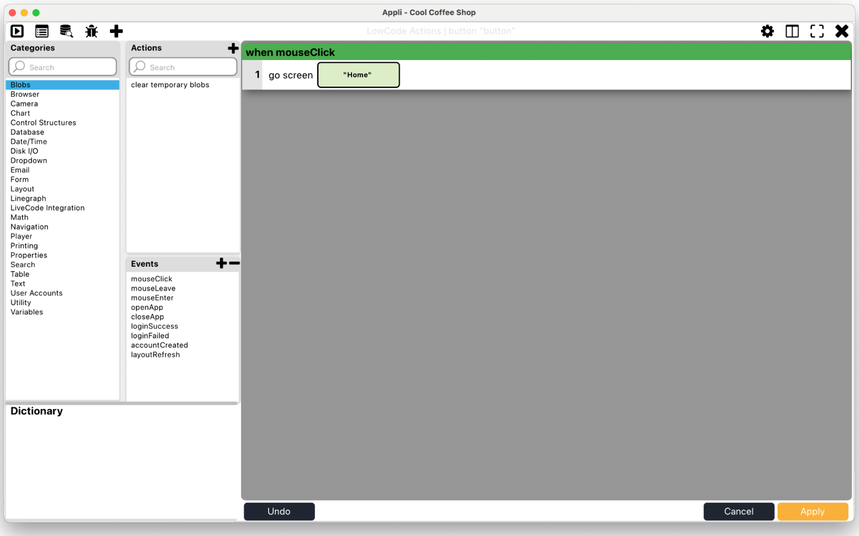


Figure 45: Low-Code interface showing empty action script.

Our aim is to create an *action script* that causes triggers the navigation to another screen when the button is clicked. To do that, first we add a *MouseClick* event from the events section, then we can go to the *Navigation* category and select the *Go to Screen* action.



```
when mouseClick
1  go screen "Home"
```

Figure 46: Low-Code interface after selecting the *Go to Screen* action.

Appli wrote the *action script* for you:

```
when mouseClick
  go screen Screen
```

The last thing you need to do is click on the *Screen* argument to select the destination screen for the *Go to Screen* action. Notice that before selecting the value for the *Screen* argument, the whole script background was tinted red. That is how Appli flags that there are mandatory steps needed before that script is ready. After selecting a screen, the background turns to green.

The *action script* starts with `mouseClick`, that tells you that this behaviour will happen when the user clicks the element, more precisely when they release the mouse button or lift their fingers from the touchpad or screen after clicking. That script can be understood as *go to this specific screen when the user clicks the element*, that's not very different from the actual text of the script. Appli *action scripts* are easy to read and understand.

Action scripts are not restricted to a single action per script. You can select multiple actions to create complex behaviours. They'll happen in the order displayed on the *action script*. If you got something wrong, you can simply click the small *x* button next to the *action* in the script to remove it.

11.5 Low-Code editor toolbar

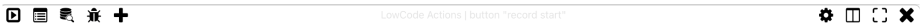


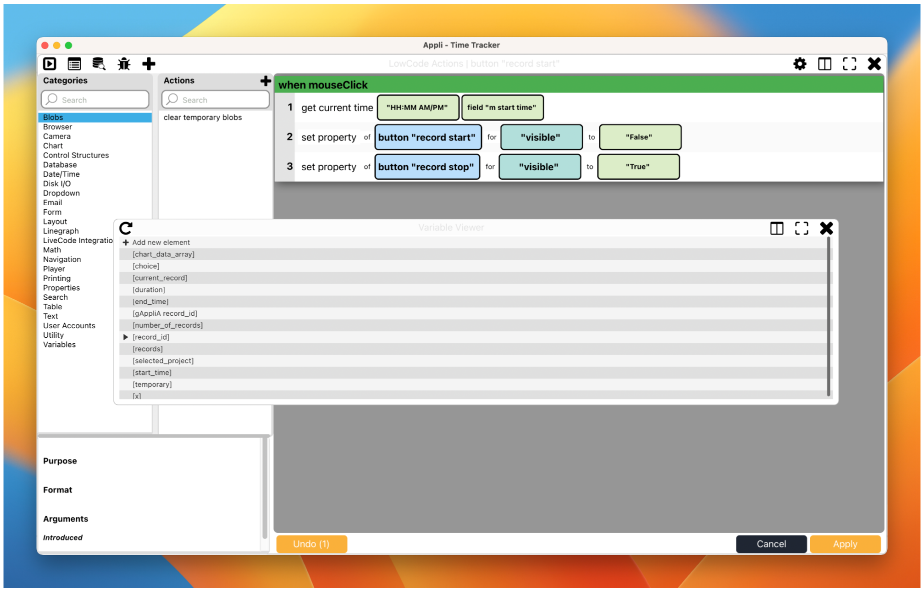
Figure 47: Low-Code editor toolbar

These buttons provide easy access to useful tools and help you manage the Low-Code editor interface. Some of these tools are also present in other panes such as the Data Viewer. They are:

- **Collapse Left Panel:** Handy for smaller screens, allows you to hide the categories, events, and actions panels.
- **Variable Viewer:** Opens the Variable Viewer tool that allows you to view and manage your variables.
- **Data Viewer:** Opens the Data Viewer which provides a way to manage your tables and the records contained in them.
- **Bug Reports:** Checks if your code is bug free and helps you debug it if it is not.
- **Add Blank Action:** Another way of adding actions to a Low-Code script by adding an empty action and then searching for it.
- **Settings:** Settings for the Low-Code editor.
- **Panel Arrangement:** For advanced users, help you arrange multiple panes on the screen.
- **Full Screen:** Makes the current pane full screen.
- **Close:** Closes the current pane.

11.5.1 Variable viewer

The button to open the variable viewer looks like a small bullet list. It displays all the current variables, allowing the developer to check their values and change them if needed.



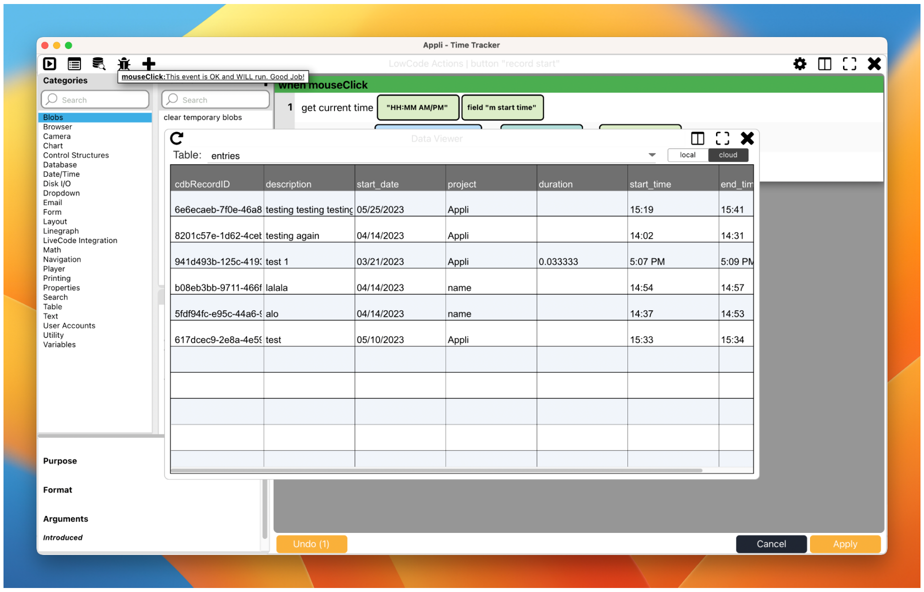
```
when mouseClick
1 get current time "HH:MM AM/PM" field "m start time"
2 set property of button "record start" for "visible" to "False"
3 set property of button "record stop" for "visible" to "True"
```

```
[chart_data_array]
[choice]
[current_record]
[duration]
[end_time]
[gAppliA record_id]
[number_of_records]
▶ [record_id]
[records]
[selected_project]
[start_time]
[temporary]
[x]
```

Figure 48: Variable viewer

11.5.2 Data viewer

The button to open the data viewer looks like a pile of disks with a magnifying glass. It opens a tool to manage your tables and records.



mouseClick:This event is OK and WILL run. Good Job!

clear temporary blobs

```
when mouseClick
1 get current time "HH:MM AM/PM" field "m start time"
```

Table: entries

cdbRecordID	description	start_date	project	duration	start_time	end_tim
6e6ecaeb-7f0e-46a8	testing testing testing	05/25/2023	Appli		15:19	15:41
8201c57e-1d62-4cet	testing again	04/14/2023	Appli		14:02	14:31
941d493b-125c-419	test 1	03/21/2023	Appli	0.033333	5:07 PM	5:09 PM
b08eb3bb-9711-466f	lalala	04/14/2023	name		14:54	14:57
5fdf94fc-e95c-44a6-!	alo	04/14/2023	name		14:37	14:53
617dcec9-2e8a-4e5!	test	05/10/2023	Appli		15:33	15:34

Figure 49: Variable viewer

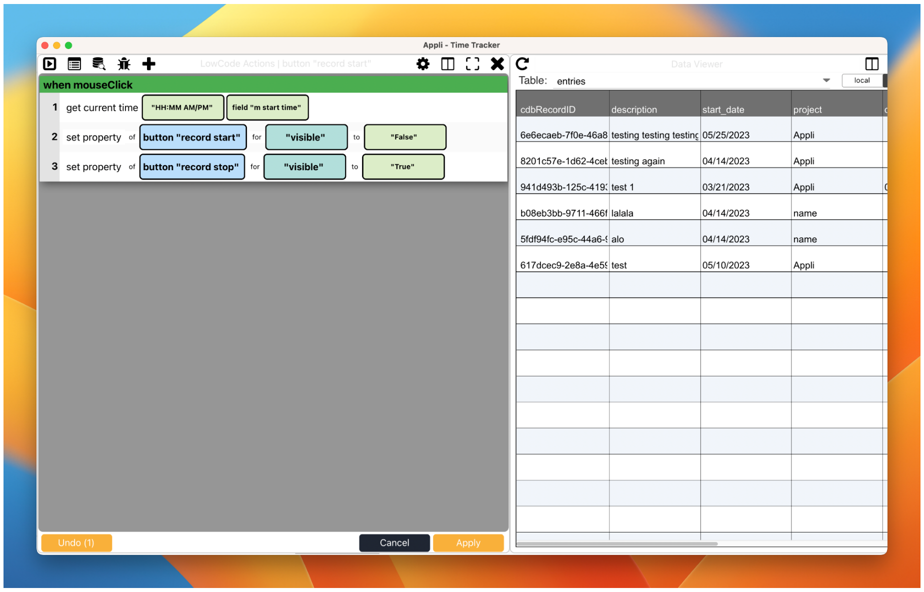
11.5.3 Bug reports

On the toolbar, there is a bug button. When you hover your mouse on top of it, it will display a quick summary report. This applies only to the low-code script currently being worked on.

If you want to get a project-wide bug report, use the bug button at the footer.

11.5.4 Arrange panes

This dropdown is handy if you have more than one pane open such as the Low-Code editor and the data viewer at the same time. You can automatically arrange their sizes so that they fit either vertically or horizontally.



when mouseClick

1 get current time "HH:MM AM/PM" field "m start time"
2 set property of button "record start" for "visible" to "False"
3 set property of button "record stop" for "visible" to "True"

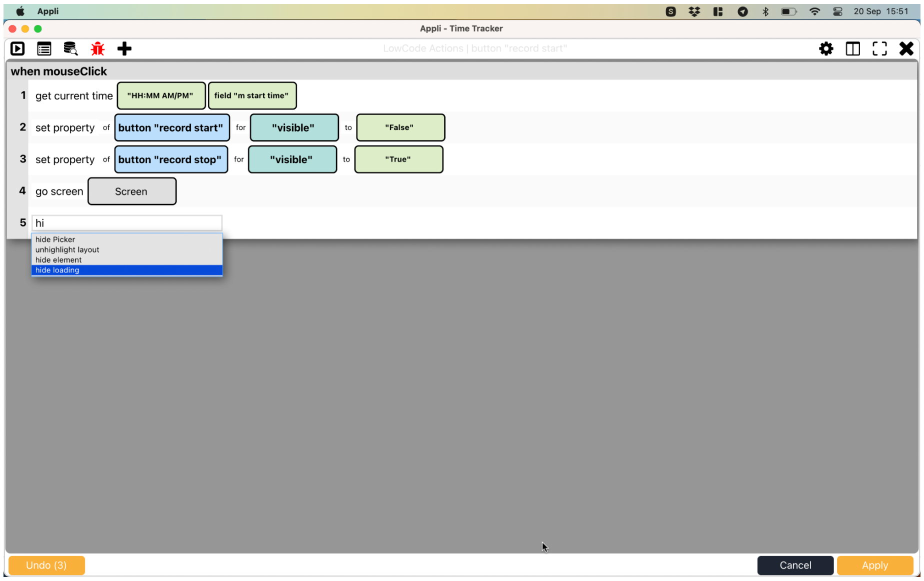
Table: entries local

cdbRecordID	description	start_date	project
6e6ecaeb-7f0e-46a8	testing testing testinç	05/25/2023	Appli
8201c57e-1d62-4cet	testing again	04/14/2023	Appli
941d493b-125c-419:	test 1	03/21/2023	Appli
b08eb3bb-9711-466f	lalala	04/14/2023	name
5fdf94fc-e95c-44a6-!	alo	04/14/2023	name
617dcec9-2e8a-4e5!	test	05/10/2023	Appli

Figure 50: Two panes arranged horizontally with the left pane collapsed in the Low-Code editor

11.5.5 Adding a blank action

This adds a blank action a the bottom of the script. This is an input field. Start typing the action you want and select the matching action from the pop-up menu. Very handy once you get familiar with actions, no more need to go hunting for actions inside categories.



when mouseClick

1 get current time "HH:MM AM/PM" field "m start time"

2 set property of button "record start" for "visible" to "False"

3 set property of button "record stop" for "visible" to "True"

4 go screen Screen

5 hi

Figure 51: Adding an action

11.6 Next Steps

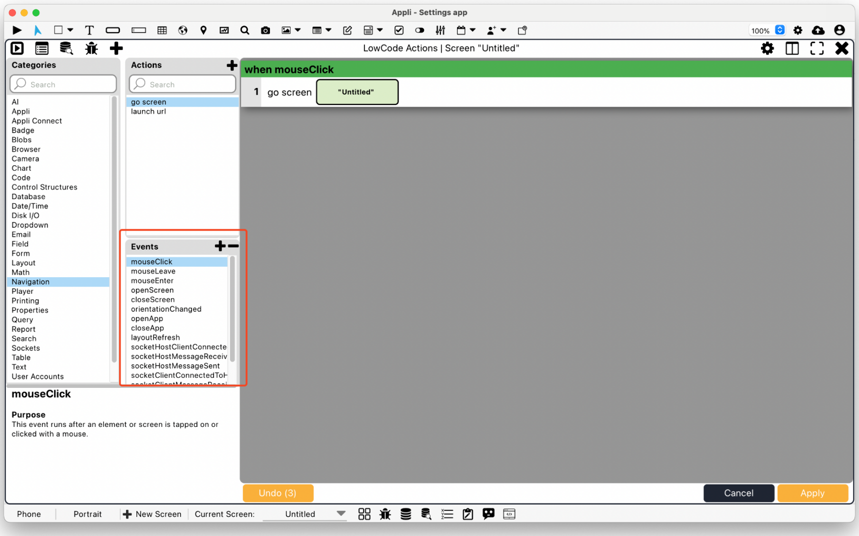
Creating *action scripts* will become easier after you've read the next chapter, which is a tutorial where we create an application from scratch. You'll create multiple *action scripts* for the various elements and that workflow will become a second nature to you in no time.

12 Events

Events are used to activate low-code scripts. When an event happens in your application and that event has an associated low-code script attached to it, that script will execute. Appli has many event types from user interaction events such as clicking a button or typing on a field element to advanced network events such as receiving a message from a web server.

In our chapter about low-code we showed how to use events to implement simple scripts, our tutorials expanded on that. On this chapter we're going to list all the events supported by Appli alongside a brief explanation of them. If you need further help with an event, reach out to us on our forums.

Events appear on their own pane in the low-code editor as can be seen in the screenshot below.



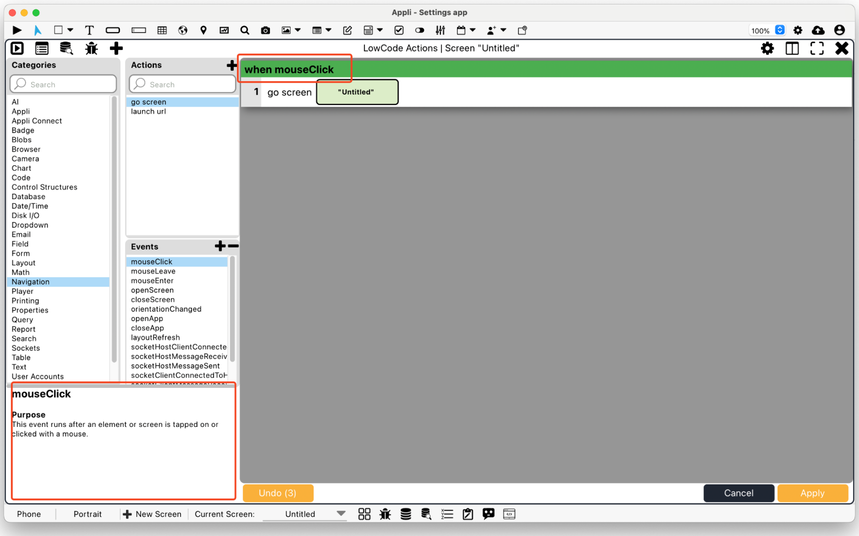
```
when mouseClick
1 go screen "Untitled"
```

mouseClick

Purpose
This event runs after an element or screen is tapped on or clicked with a mouse.

Figure 52: Events panel

When you select an event, a brief explanation is shown in the dictionary pane. A low-code script always begin with an event on top.



```
when mouseClick
1 go screen "Untitled"
```

mouseClick

Purpose
This event runs after an element or screen is tapped on or clicked with a mouse.

Figure 53: Event description and header

To remove an event, right-click the event to display the context menu and select *delete event*.

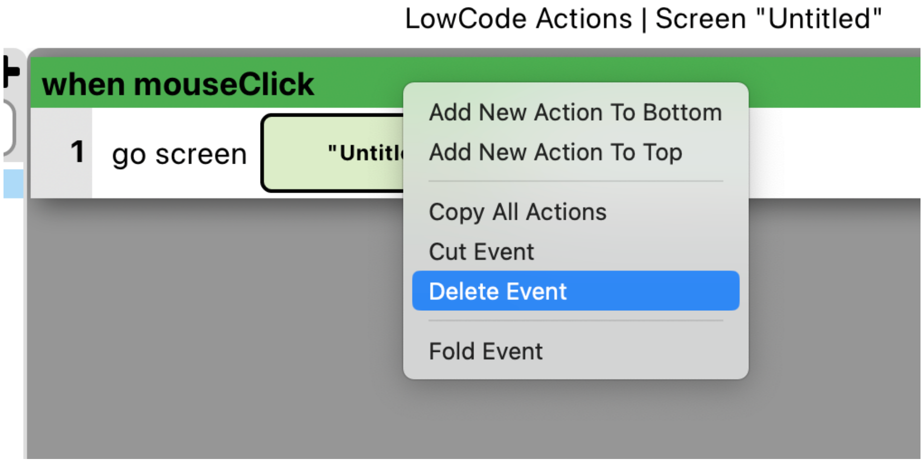


Figure 54: Deleting an event

You can also use the other options to add low-code actions to the event, cut it so you can paste it into another element, or fold it so you can better view other events in the same script.

12.1 Custom events

Custom events allows your low-code scripts to be reused and be more expressive. You can define a custom event and then use the `dispatch` action to execute it. To learn more about it, check out the Code Reuse Tutorial.

12.2 List of events

12.2.1 accountCreated

This event runs after an user account has been successfully created.

12.2.2 cameraCaptured

This event runs after a photo is taken with a camera element, a photo is selected from a mobile device's photo gallery though the camera element, or a barcode or QR code is scanned with the camera element.

12.2.3 cellEdited

This event runs after a cell of a table element has been edited.

12.2.4 cellSelected

This event runs after a cell of a table element has been selected.

12.2.5 checkboxSelected

This event runs after a checkbox element's value has been changed.

12.2.6 closeApp

This event runs after exiting an app.

12.2.7 closeField

This event runs after the focus leaves a field element and the field's contents has changed.

12.2.8 closeScreen

This event runs after leaving a screen.

12.2.9 custom

This special event will prompt you for the name of the custom event you wish to add to this current script.

12.2.10 dateClosed

This event runs after a date element's visibility has been turned off.

12.2.11 dateSelected

This event runs after the date has been changed in the date element.

12.2.12 dropdownSelection

This event runs after a selection has been made in a dropdown element.

12.2.13 exitField

This event runs after the focus leaves a field element and the field's contents has not changed.

12.2.14 layoutRefresh

This event runs after a layout element has been refreshed.

12.2.15 loginFailed

This event runs after a failed user login has occurred.

12.2.16 loginSuccess

This event runs after a successful user login has occurred.

12.2.17 mouseClick

This event runs after an element or screen is tapped on or clicked with a mouse.

12.2.18 mouseEnter

This event runs after the mouse has moved into an element or screen.

12.2.19 mouseLeave

This event runs after the mouse has moved outside of an element or screen.

12.2.20 openApp

This event runs after an app has been loaded.

12.2.21 openScreen

This event runs after a screen is displayed.

12.2.22 orientationChanged

This event runs after the screen's orientation has been changed.

12.2.23 rowSelected

This event runs after a row of a table element has been selected.

12.2.24 searchPerformed

This event runs after a search has been performed in a search element.

12.2.25 selectedTabChanged

This event runs after a tab of the tab element has been selected.

12.2.26 sliderValueChanged

This event runs after the slider element's value has changed.

12.2.27 socketClientConnectedToHost

This event runs after the socket client has connected to the host.

12.2.28 socketClientMessageReceived

This event runs after the socket client message has been received.

12.2.29 socketClientMessageSent

This event runs after the socket client message has been sent.

12.2.30 socketError

This event runs after the socket host or client has experienced an error.

12.2.31 socketHostClientConnected

This event runs after the socket host or client has connected.

12.2.32 socketHostMessageReceived

This event runs after the socket host message has been received.

12.2.33 socketHostMessageSent

This event runs after the socket host message has beent sent.

12.2.34 timeClosed

This event runs after a time element's visibility has been turned off.

12.2.35 timeSelected

This event runs after the time has been changed in the time element.

13 Data Management

Appli offers robust data management features thanks to the many years of experience that the team spent developing LiveCloud. Appli itself is backed by the LiveCloud database. When you save your application to the Cloud, what happens behind the scene is that your app is stored in LiveCloud, where it can be retrieved by the Appli Player.

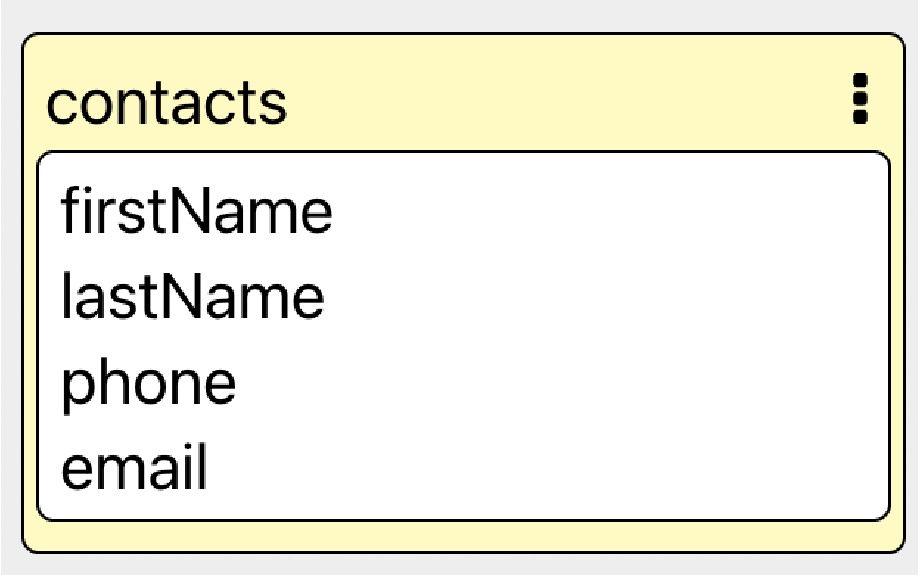
The same powerful database system that powers Appli is available to you as a developer. It is easy to use and matches Appli development workflow in a way that other database systems simply can't.

13.1 Tables

Tables represent a collection of data in Appli. You can think of them in the same mental model you use for spreadsheets or a stack of forms. A collection of structured data you can manipulate and query to find the information you need.

Your app can have as many tables as you need, and tables can store their data locally and on the cloud. Appli provides a robust set of synchronization methods to make sure your local and cloud data match up.

Each table has a set of keys. Much like a form would have fields. A good example is a contacts table. It could look like this:



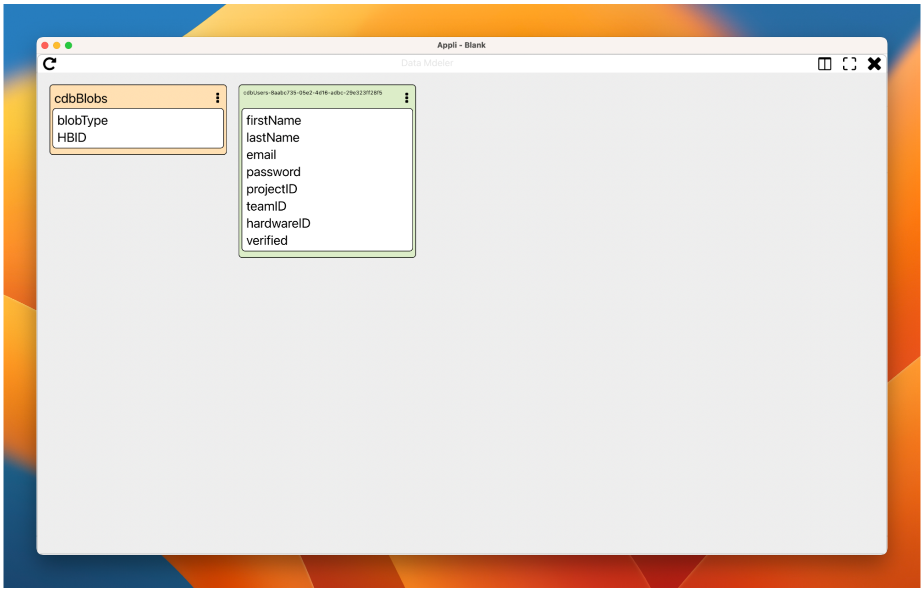
contacts

firstName
lastName
phone
email

Figure 55: Sample contacts table

13.2 Creating a new table from scratch

The *Data Modeler* button is a series of stacked discs on the footer. Clicking it opens the *Data Modeler*.



cdbBlobs

blobType
HBID

cdbUsers-8aabc735-05e2-4d16-adbc-29e323ff28f5

firstName
lastName
email
password
projectID
teamID
hardwareID
verified

Figure 56: Data Modeler

> **Attention:** All Appli apps have two tables from the start, one to hold users and another to hold binary data. They are called *cdbBlobs* and *cdbUsers*. You will see them on the *Data Modeler*.

Create a new table by right-clicking the background and selecting the *Add New Table* from the contextual menu.

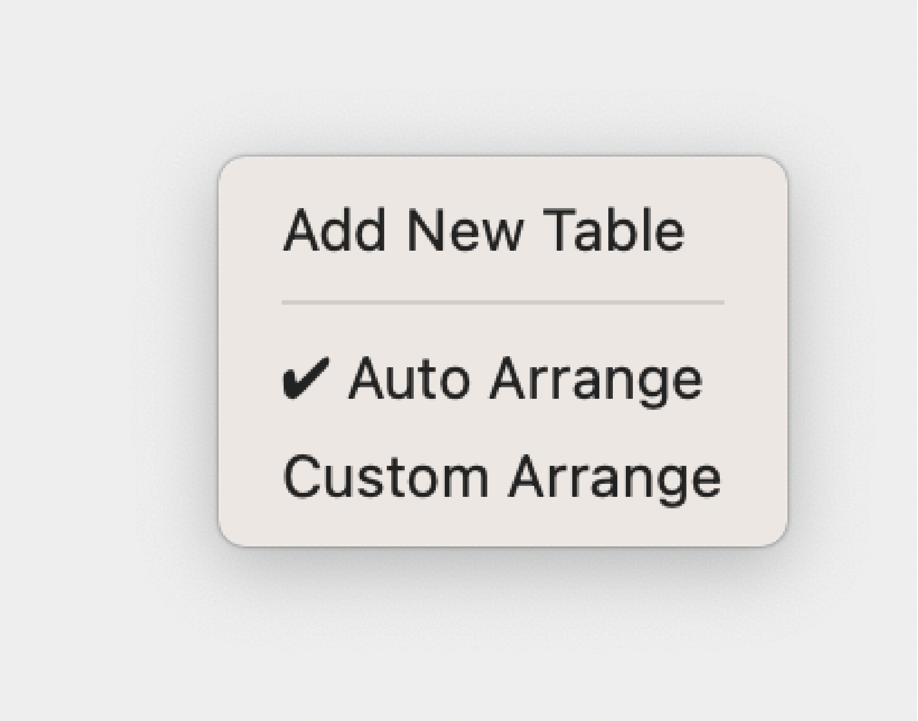


Figure 57: Adding a new table

Once you have the *Table Setup* dialog open. You can fill in your table name and add keys.

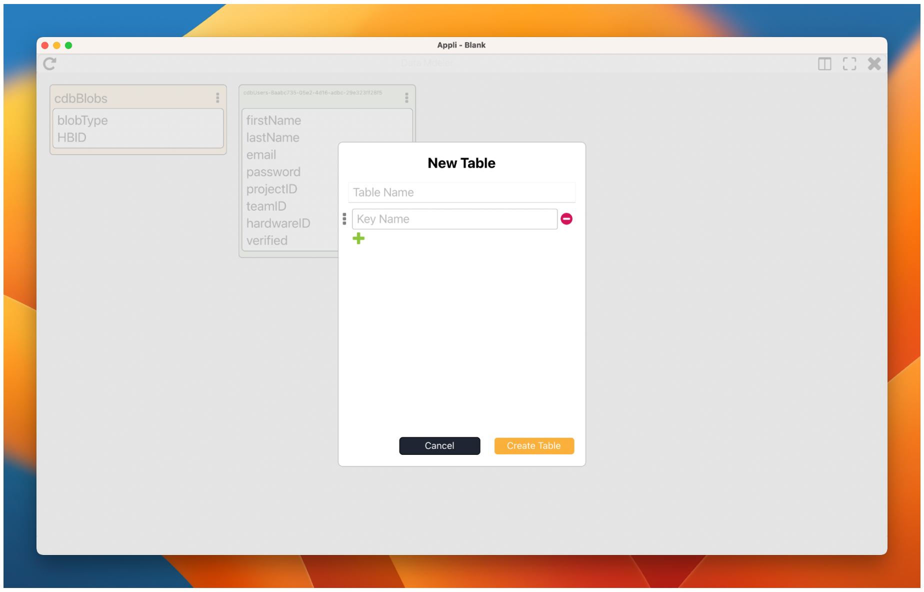
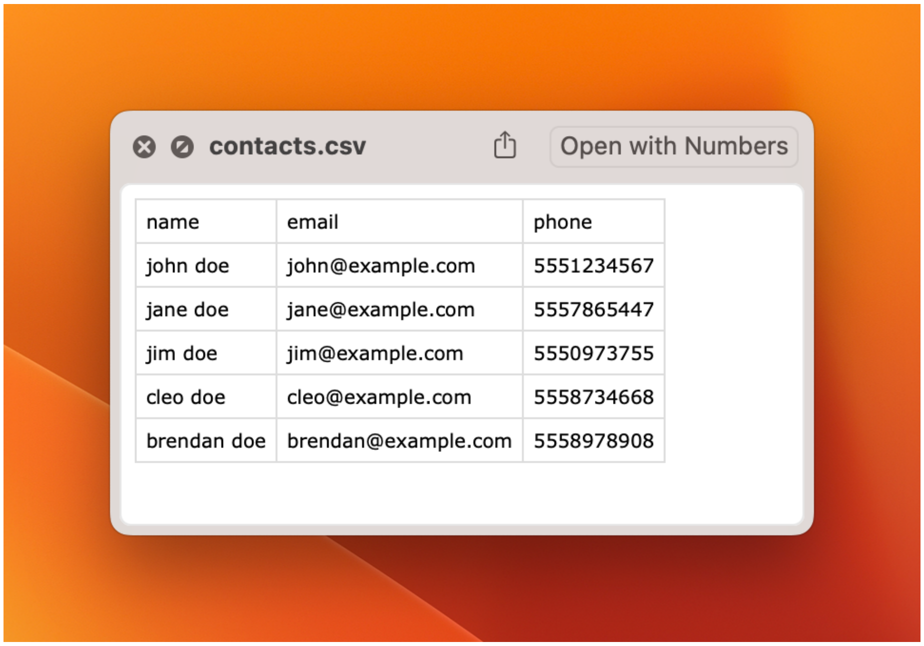


Figure 58: Table Setup

13.3 Using CSV files to bootstrap a database

A good way to bootstrap a table from existing data is to export that data using the CSV format and dragging and dropping it into the playground (be aware, you can't drag and drop a CSV file into the Data Modeler, you need to do it in the playground).



name	email	phone
john doe	john@example.com	5551234567
jane doe	jane@example.com	5557865447
jim doe	jim@example.com	5550973755
cleo doe	cleo@example.com	5558734668
brendan doe	brendan@example.com	5558978908

Figure 59: Sample contacts CSV file

Appli will process the CSV file and open the *Table Setup* dialog pre-filled with the data from that file.

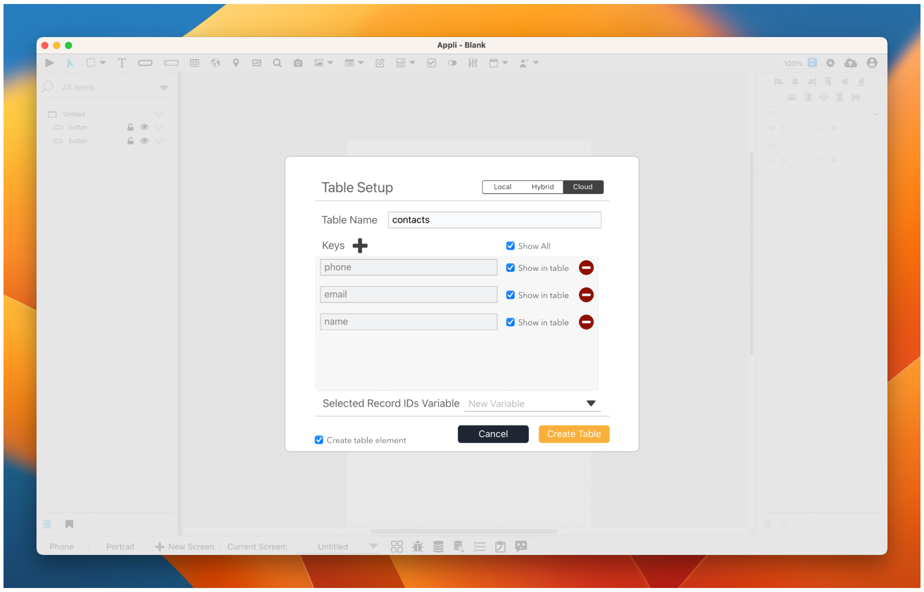


Table Setup

Local | Hybrid | ● Cloud

Table Name: contacts

Keys

☑ Show All

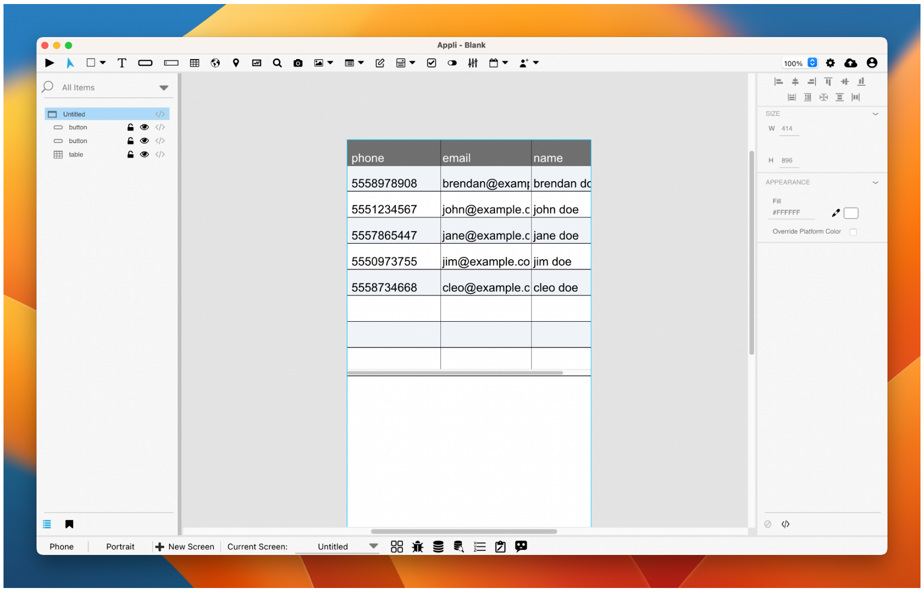
phone	☑ Show in table	⊖
email	☑ Show in table	⊖
name	☑ Show in table	⊖

Selected Record IDs Variable: New Variable

☑ Create table element

Figure 60: Importing a CSV file as a new table

Notice that this also gives you the option of creating a *table element* linked to that table. Select the *source of truth* for the table element — local, cloud, hybrid — and there is a checkbox to flag if you want the table element or not. That is a fast and easy way to create tables, import data into them, and display that data in an app with a single action.



phone	email	name
5558978908	brendan@examp	brendan dc
5551234567	john@example.c	john doe
5557865447	jane@example.c	jane doe
5550973755	jim@example.co	jim doe
5558734668	cleo@example.c	cleo doe

Figure 61: Table element created from importing a CSV

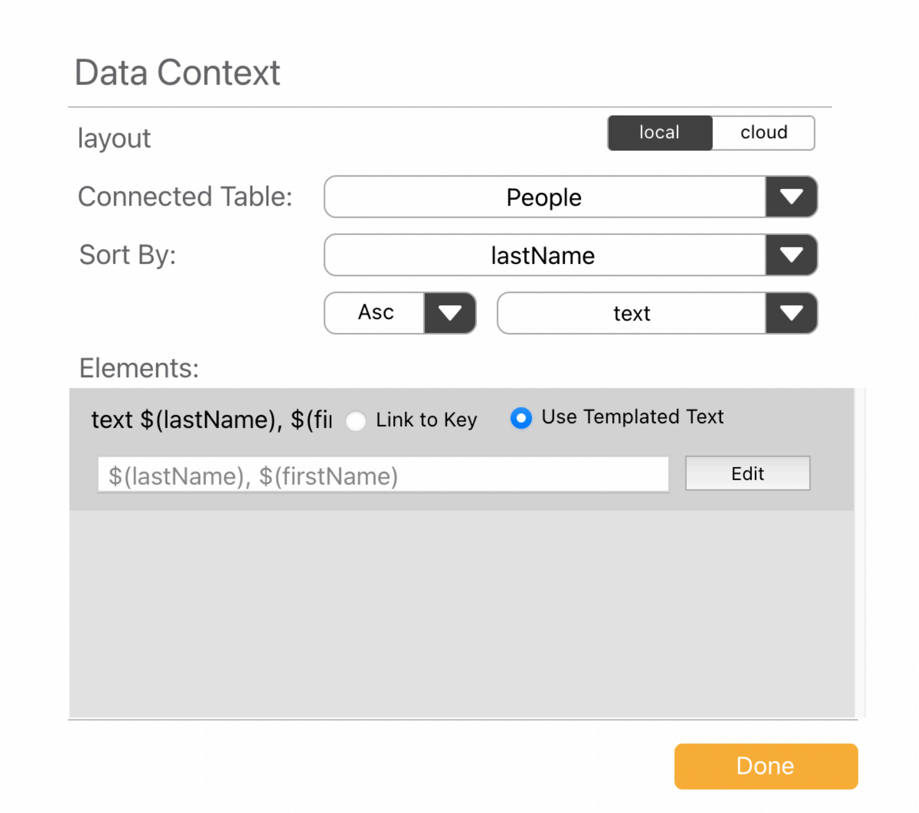
13.4 Binding data to elements

Once you have a table, you can use no-code to bind table data to elements. An obvious element to use is the table element. Using no-code, you can select which table to display and which columns to use.

13.5 Displaying records using the layout element

A *table* can be connected to a *layout element* Using *no-code*. Once connected, the elements placed inside the layout can be bound to data from the table.

The layout can be configured in a way that it is tied to a specific record in the table or display multiple records as a list. Use the *multiple rows* property to change between these two modes. Once that property is set to the desired value, the interface for *no-code* setup will change to reflect that option.



Data Context

layout

Connected Table: People

Sort By: lastName

Asc

text

Elements:

text $(lastName), $(fi ○ Link to Key ● Use Templated Text

$(lastName), $(firstName)

Figure 62: Example of Layout *no-code* table setup with multiple rows.

Elements inside the layout can be linked direct to keys from the table or use *template text* to interpolate data from the table into a formatted text.

> For more information, check the Data Management tutorial for a hands-on guide about building database-aware apps.

13.6 Manipulating data

Low-code has a category just for database manipulation. Everything is fully documented with dictionary entries for each action.

13.7 Using forms to edit and create records

Similarly to *layout elements*, *form elements* can contain other elements inside it. The elements inside a form share a database context. They can be connected to a table via *no-code* configuration and can be further linked to a *Record ID* set using a combination of *low-code* actions:

- `set variable from context`: to store the record id into a variable.
- `set the property from variable`: to set the `dataRecordID` property of the form to the value in the variable set with `set variable from context`.

Elements inside the form will have access to the data from that record. Any change to them using bound fields or *low-code* can be saved back to the same record.

A form that is connected to a table but doesn't have a `dataRecordID` set can create new records in the connected table via:

- `submit form to database` which picks data from the form and save to the database.

  For more information, check the Data Management tutorial for a hands-on guide to building database-aware apps.

14 Asset Manager

If you want to have your app to look professional, you must have a consistent look and feel throughout your user interface. The easiest way to achieve that is to learn how to use the Asset Manager.

You can open the Asset Manager by either clicking the bookmark button at the bottom of the left-side pane in the playground or by pressing CMD+SHIFT+Y on a Mac or CTRL+SHIFT+Y on Windows.

The Asset Manager allows you to keep track and reuse font configurations, colors, and even appli groups.

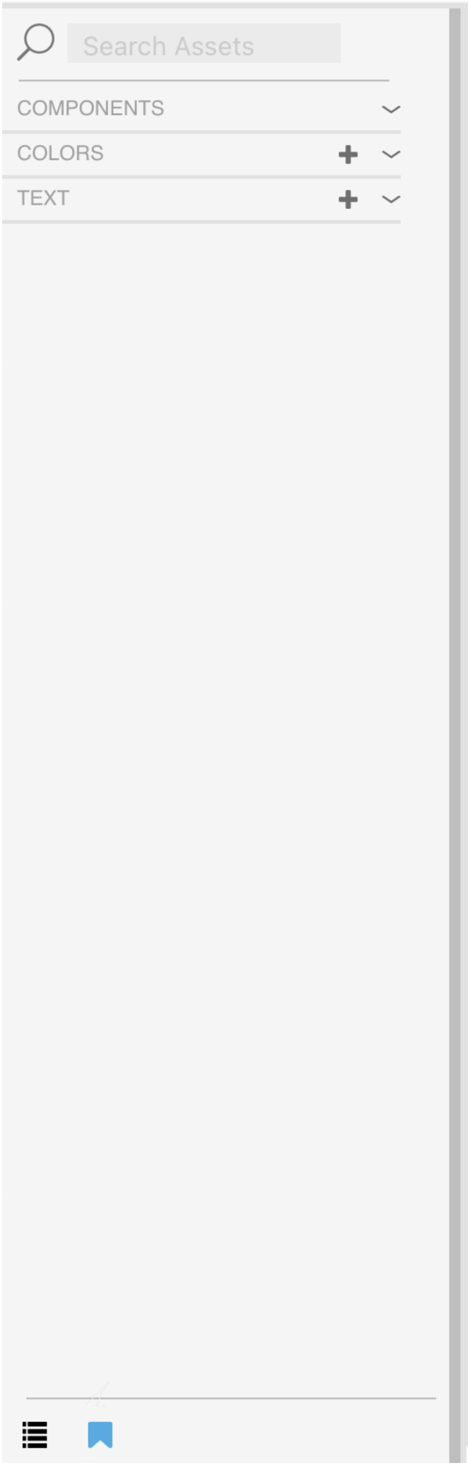
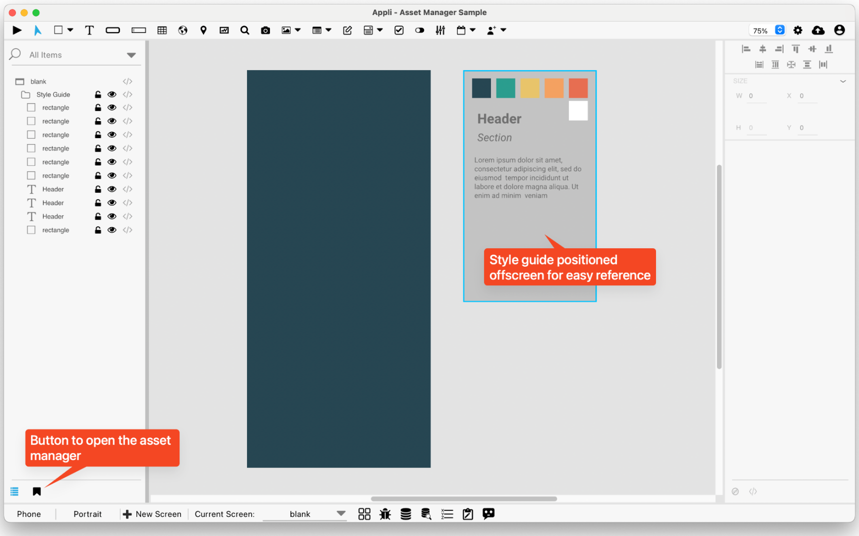


Figure 63: Asset Manager

14.1 Creating a style guide

Before we're able to add items to the Asset Manager, we need to have them on the screen. A good practice is to create a small style guide with all the colors and typography you're going to use on your first screen. By placing it outside the screen boundaries, you can make sure it is visible to you but not your app users.

> Remember to enable Show Offscreen Elements (keyboard shortcut: o) in the app settings to be able to view the style guide.



Style guide positioned offscreen for easy reference

Button to open the asset manager

Figure 64: Style Guide

14.2 Adding colors

With the style guide in place, click on each colored square and click on the plus button on the color section of the Asset Manager. This will add that color to the Asset Manager and you'll be able to reapply it to any element by selecting the element and clicking the color on the left pane.

14.3 Adding fonts

Just like what we did with the colors, we can click each of the text samples and click the plus sign in the fonts section to record that font configuration. Selecting an element in the playground and clicking the font on the Asset Manager will apply those settings.

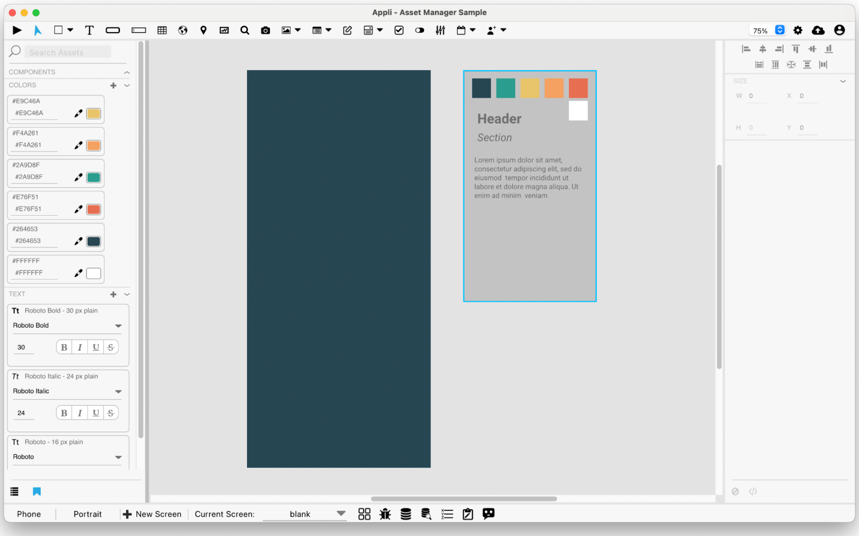
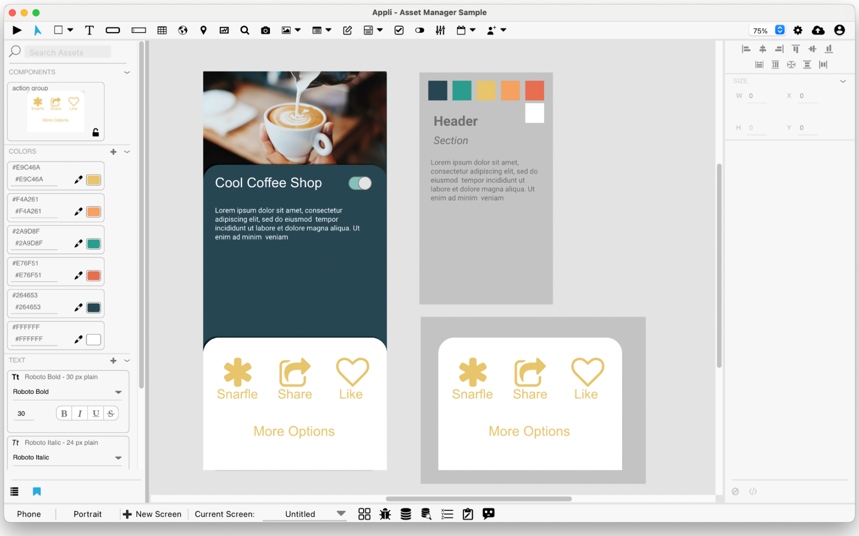


Figure 65: Fonts and colors added to the asset manager

14.4 Adding groups

If you see yourself reusing the same collection of elements over and over again, you might want to consider making them into a group and adding them to the Asset Manager as well. This way, you can easily add them to future screens instead of recreating them from scratch or copying and pasting.



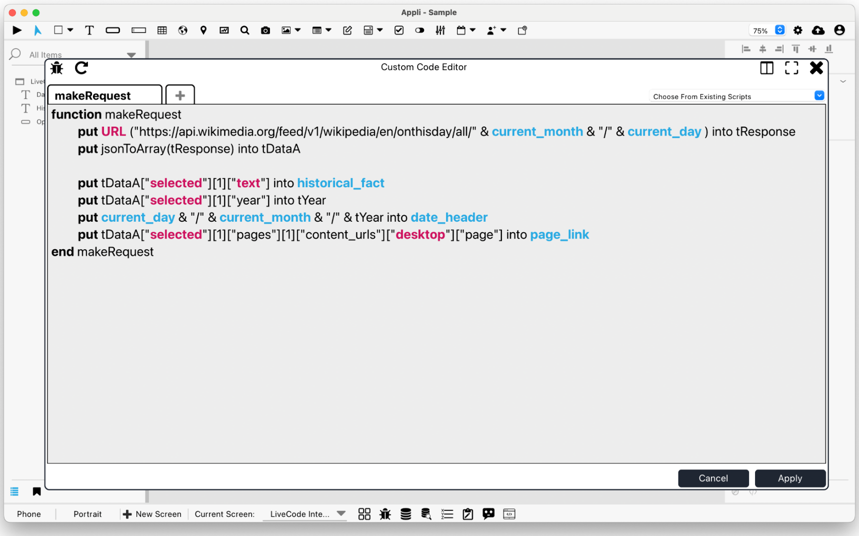
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam

Figure 66: Group added to the Asset Manager

And one of the coolest features is that the group you add to the components section stays in sync with across all your screens. You can update the original group and they'll all reflect the changes.

15 Custom Code Editor



```
function makeRequest
    put URL ("https://api.wikimedia.org/feed/v1/wikipedia/en/onthisday/all/" & current_month & "/" & current_day ) into tResponse
    put jsonToArray(tResponse) into tDataA

    put tDataA["selected"][1]["text"] into historical_fact
    put tDataA["selected"][1]["year"] into tYear
    put current_day & "/" & current_month & "/" & tYear into date_header
    put tDataA["selected"][1]["pages"][1]["content_urls"]["desktop"]["page"] into page_link
end makeRequest
```

Figure 67: Custom Code Editor

The Custom Code Editor allows you to integrate functions built with LiveCode into your app. You might want to do that in the following scenarios:

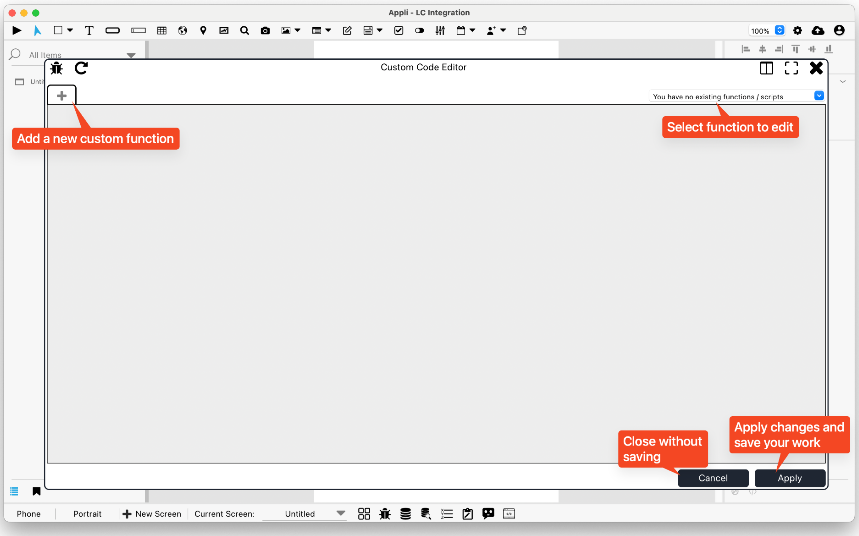
- You're porting a LiveCode-based application to Appli and want to reuse some of the code you already have.
- You need features that are available in LiveCode script but are not yet available through Appli's low-code actions.

Whatever your reason is, our LiveCode integration is easy to use and provides a straight forward path towards building more complex applications. If you want to jump in right away and use it to build a sample app, check out the LiveCode Integration Tutorial or read on to learn how the editor works.

> This chapter assumes familiarity with LiveCode. If you don't know LiveCode, you can learn more about it through their guides.

15.1 Editor interface

The editor uses a tabbed interface and will display one function per tab. If the app has no custom code, the interface opens with a *plus button* in the tab bar to add a new function. On the bottom right of the editor, buttons to cancel changes or apply changes allows the developer to close the editor and save or discard their work.



Add a new custom function

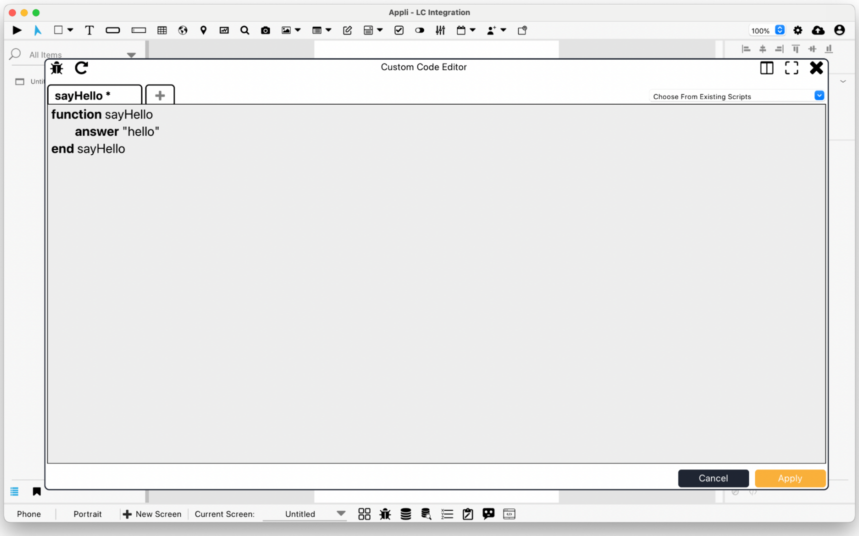
Select function to edit

Close without saving

Apply changes and save your work

Figure 68: Custom Code Editor interface

When you click the plus button, Appli asks for a name for the new function and creates a tab for it. The tab starts empty and you can add a new function written in LiveCode with whatever name you want, but we recommend using the same name as the one you set for the tab. Clicking *Apply* will save your work. When you open the Custom Code Editor, it remembers which tabs were open. If you want to access a function you created that has no open tab, you can select it from the dropbox on the right side of the tab bar.

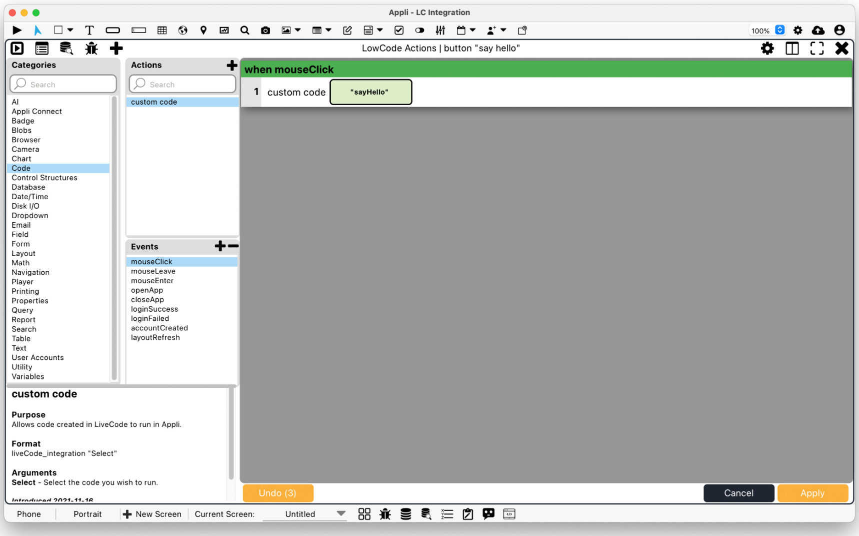


```
function sayHello
    answer "hello"
end sayHello
```

Figure 69: Example of custom code

15.2 Custom Code Action

To execute a custom code function, use the custom code action. The parameter for that action is the name you given when you created a new tab in the custom code editor.



custom code

Purpose
Allows code created in LiveCode to run in Appli.

Format
liveCode_integration "Select"

Arguments
Select - Select the code you wish to run.

Introduced 2021-11-16

Figure 70: Executing the sayHello custom code

15.3 Best practices for custom code

You can add as many functions and commands you want per custom code tab, just remember that the first function is the one that is going to be executed when the custom code is triggered by the *custom code action*.

You can access variables and elements from Appli from the custom code, just start typing their names and use the autocomplete popup to select the variable or element.

Organize your function so that it is divided into three sections:

- **Section 1:** getting data from Appli. Use the start of your function to pick variables and properties from elements in Appli and place them into LiveCode variables. This way, in the middle of your LiveCode script you're only dealing with LiveCode variables (it is easier to debug things this way).
- **Section 2:** LiveCode script. After getting all the data you need, proceed with your function implementation in LiveCode.
- **Section 3:** Return data to Appli. Set values for Appli variables and element properties at the end of your function.

If you follow these guidelines, only the beginning and the end of your function will contain elements or variables from Appli and all the middle part will be pure LiveCode script.

This doesn't affect the execution of your function, this is not something enforced by Appli, these are just best practices that makes it easier for you to implement and debug your custom code.

15.4 Next steps

If you see yourself needing this kind of integration for your apps, it is a good time to go through the LiveCode Integration tutorial.

16 Tutorial: Cool Coffee Shops

In this tutorial, we're going to create a simple application to highlight my favorite coffee shops in London. You are going to go from zero to have a running app in your mobile device in less than one hour. So, fasten your seat belt because working with Appli is fast.

16.1 Creating the project

After logging in to our Appli account, we're presented with the project selection screen:

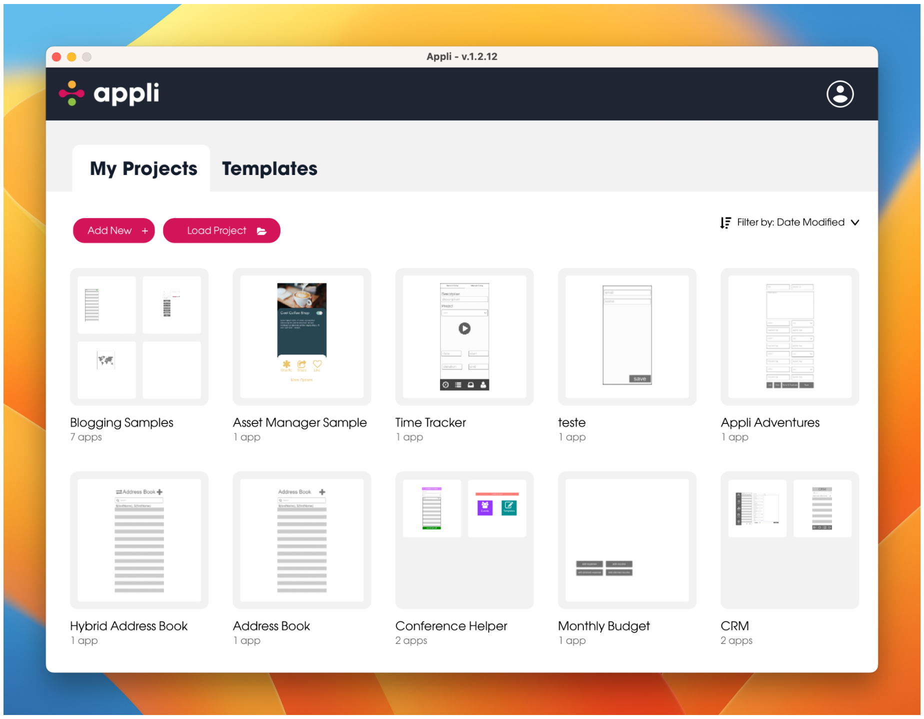
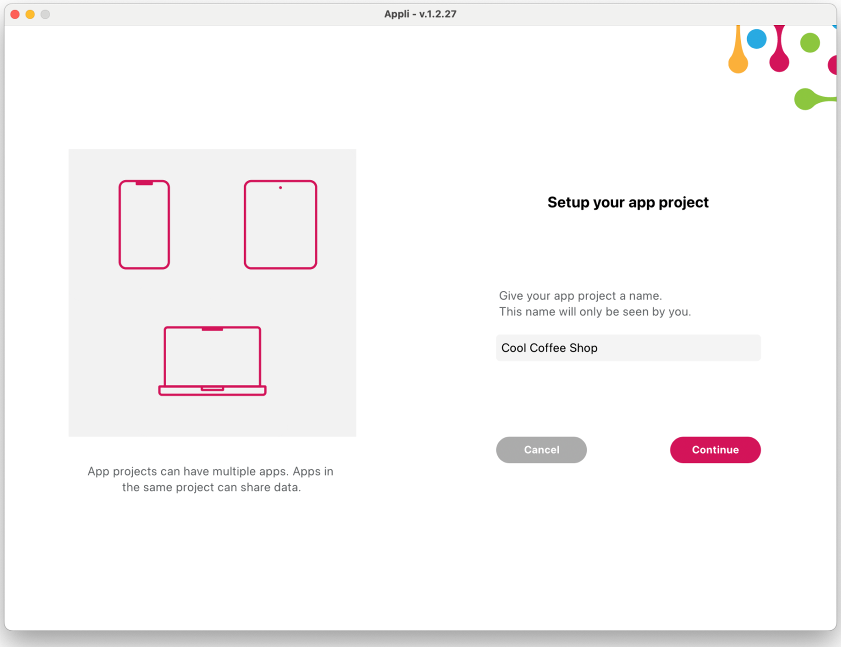


Figure 71: Project selection screen

Click on the *plus* button to create a new project named Cool Coffee Shops.



Setup your app project

Give your app project a name.
This name will only be seen by you.

Cool Coffee Shop

App projects can have multiple apps. Apps in the same project can share data.

Figure 72: Project creation

Let's create a smartphone app first. Click on the phone button to select that platform.

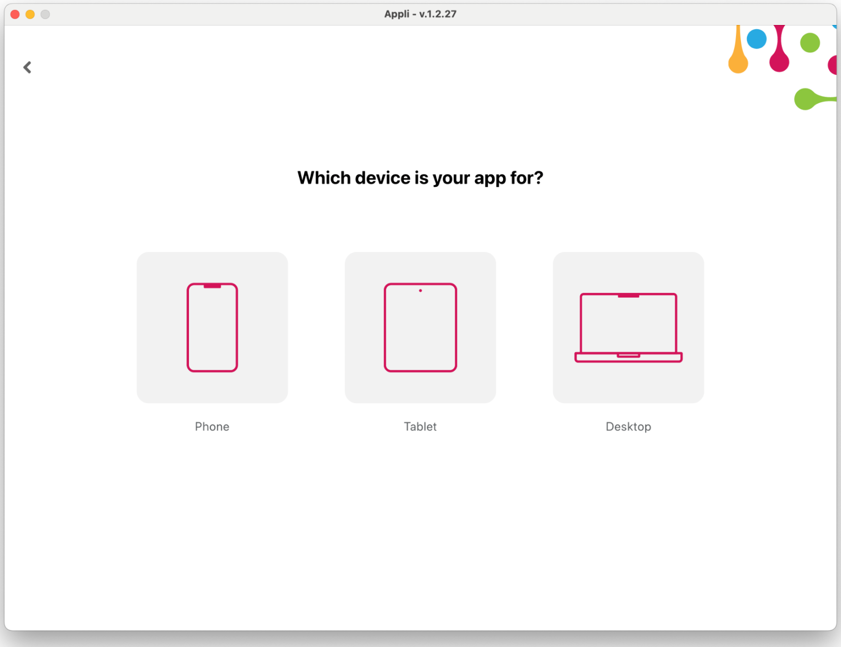
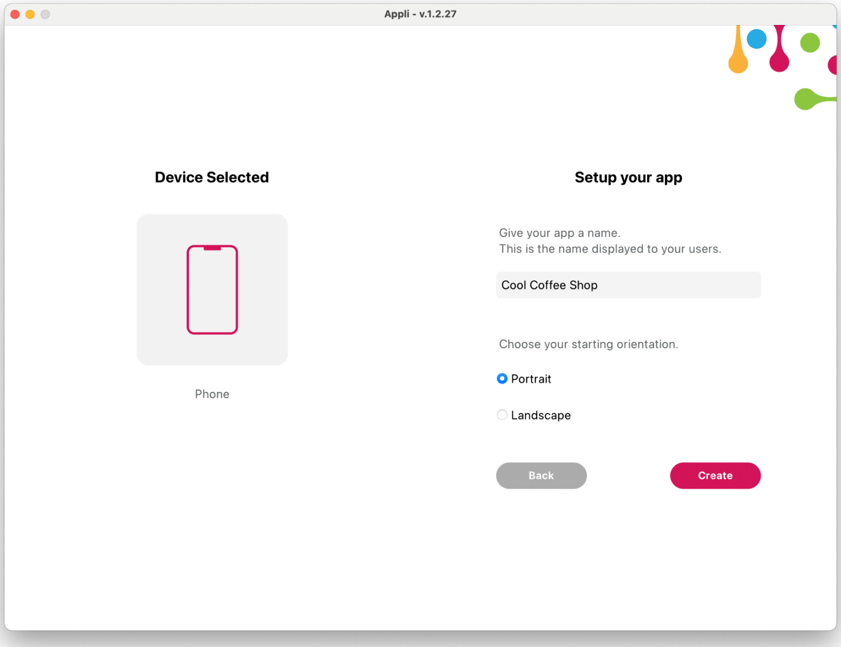


Figure 73: Platform selection

And then fill in the name for the app and select the *portrait orientation*.



Appli - v.1.2.27

Device Selected

Phone

Setup your app

Give your app a name.
This is the name displayed to your users.

Cool Coffee Shop

Choose your starting orientation.

● Portrait

○ Landscape

Figure 74: Application settings

16.2 Designing the interface

After clicking save, your new project is automatically saved to the Appli cloud and opened in the playground screen:

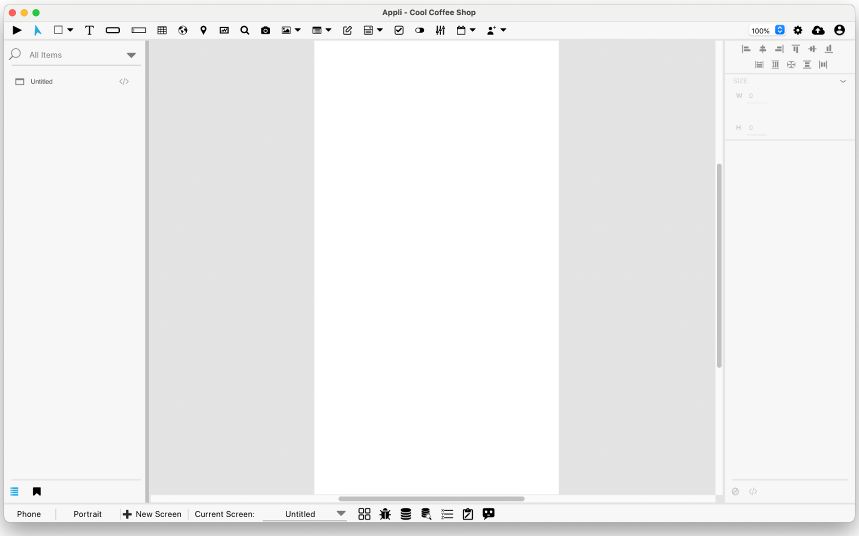


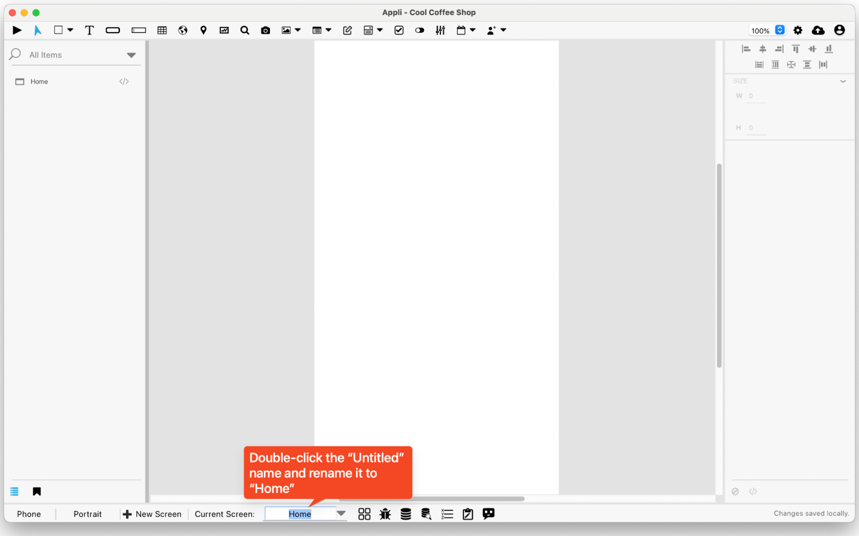
Figure 75: Empty playground

It may look empty right now, but it is actually full of possibilities.

16.2.1 Creating the home screen

Let’s begin by renaming our current screen using the controls in the footer.

Double-click the *“Untitled”* name in the footer to make it editable. Type in *Home*:



Double-click the “Untitled” name and rename it to “Home”

Figure 76: Screen rename: Home

Let’s add an image to our home screen to make it more fun. You can download this photo to your computer and then just drag and drop it into Appli. Resize the image in Appli using the corners to make it fit onto the top of our app.

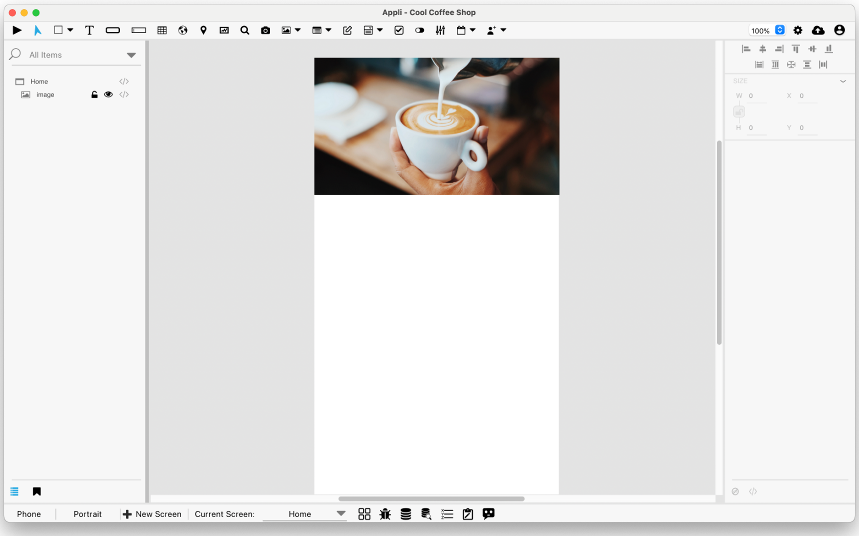
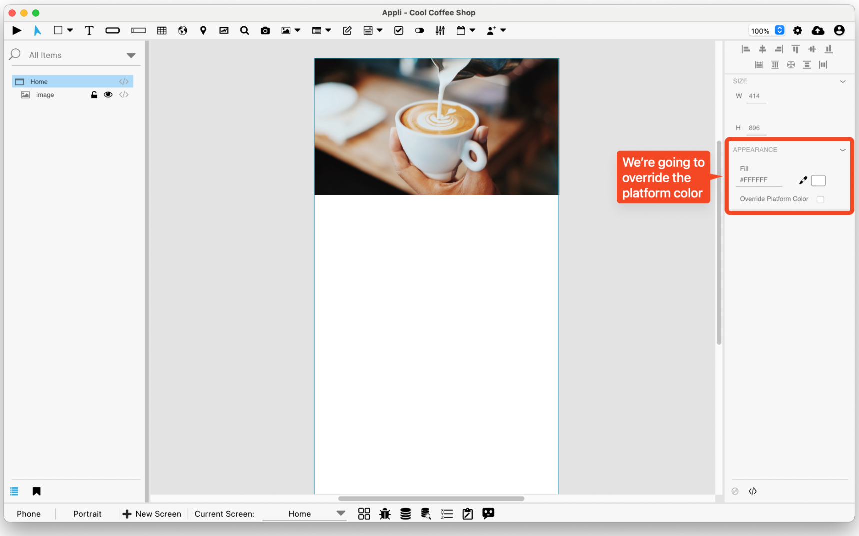


Figure 77: Header photo

Next, let's change the background color of the app to be more like coffee. Click anywhere on the white background of the playground, and let's change the `platformBackgroundColor` property:



We're going to override the platform color

W 414

H 896

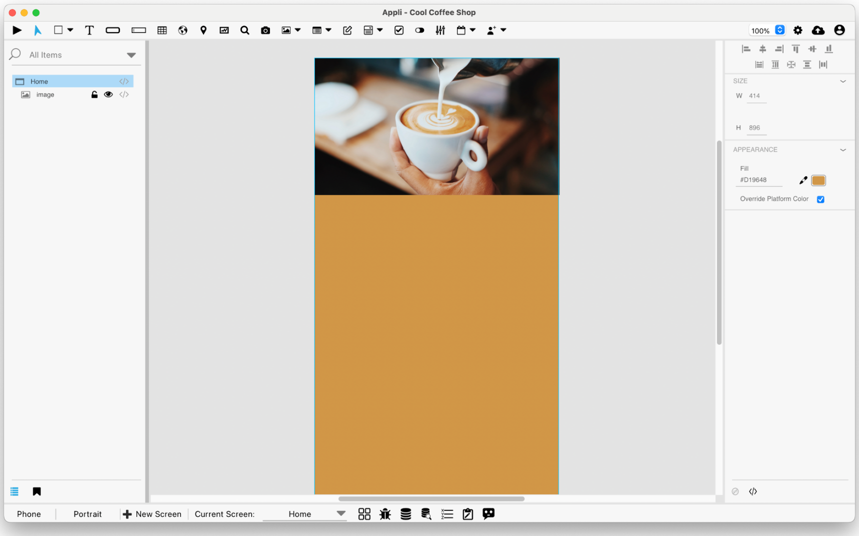
Fill

#FFFFFF

Override Platform Color □

Figure 78: Change background color

Using the color picker tool (the little eye dropper next to the *hexcoded color*) pick a nice shade of brown for the background.



W 414

H 896

Fill

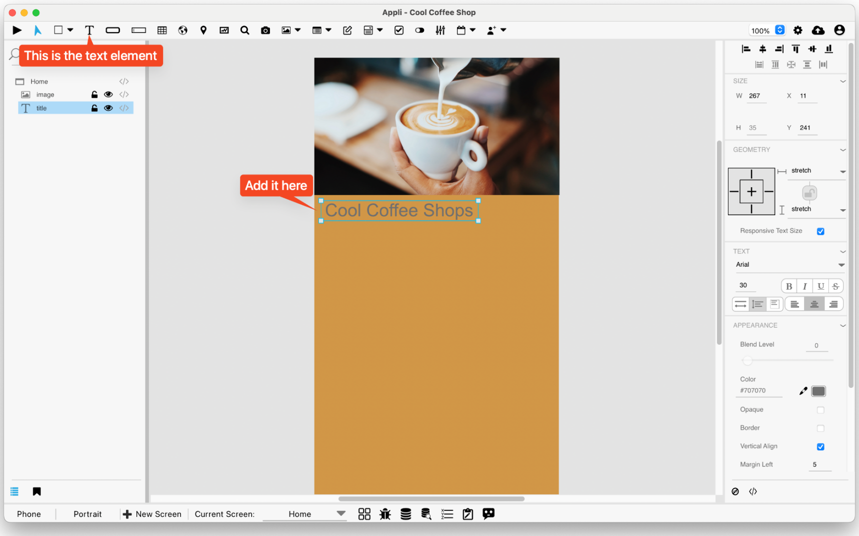
#D19648

Override Platform Color ☑

Figure 79: With new background color

Next, we're adding a little header text to ensure everyone understands what our demo app is all about.

Add a *text element* by selecting it from the tools palette, and drawing it just below our header photo. When you add a text element to a screen, you have the opportunity to change its text. Change it to *Cool Coffee Shops*.



SIZE
W 267 X 11
H 35 Y 241

GEOMETRY
stretch
stretch
Responsive Text Size ☑

TEXT
Arial
30

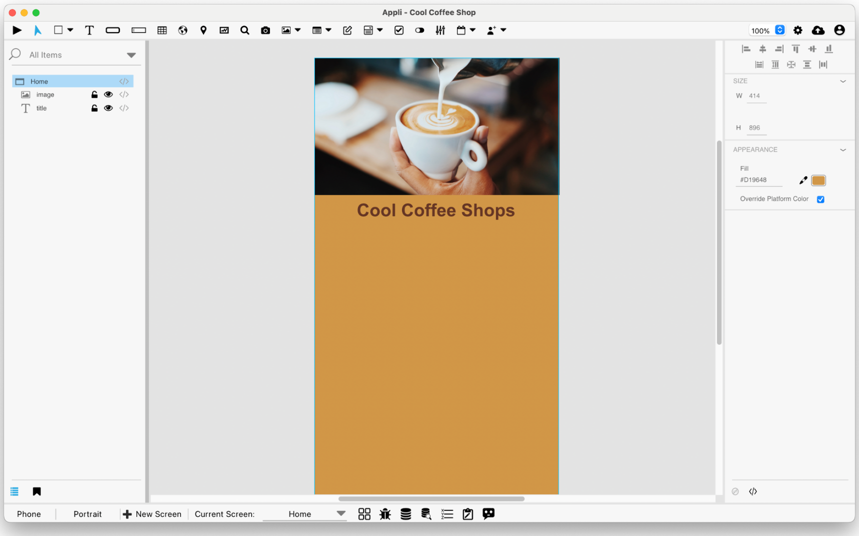
APPEARANCE
Blend Level 0
Color #707070
Opaque □
Border □
Vertical Align ☑
Margin Left 5

Figure 80: Change text color

We can dress that text element up using the property inspector. In the *Text & Appearance* sections, change the font to bold, the font size to 40, and center the text.

> Be aware that to change the text to bold, you need to double-click the text element to make it editable and select the text you want to change.

Scrolling down on the property inspector, you'll find the *textColor* property. It also has an eye dropper tool next to it. Use it to pick some darker tone in the photo.



SIZE

W 414

H 896

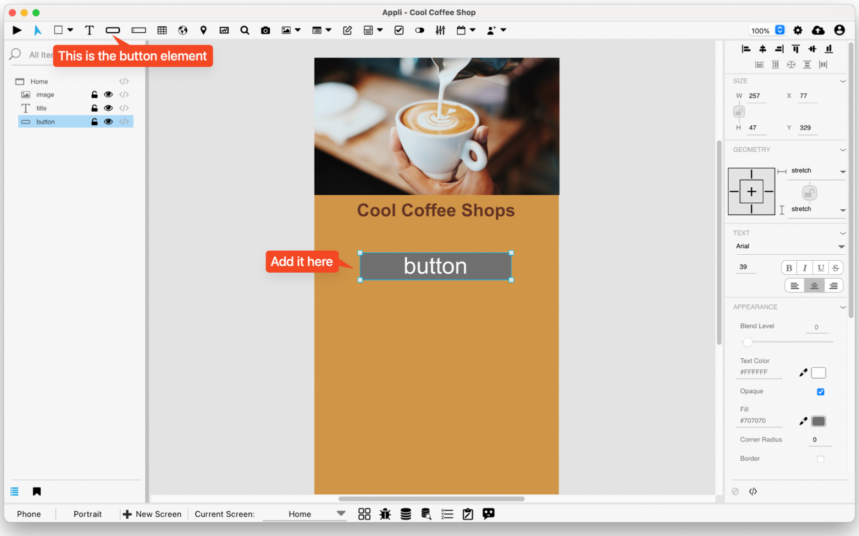
APPEARANCE

Fill #D19648

Override Platform Color ☑

Figure 81: With new text color

Add a button to the screen. That button is what the user will use to navigate to a specific coffee shop. We'll make the first one, then duplicate it to make the second one. You can name that button whatever you want, but I named it *Redemption Roasters* because that is the name of the coffee shop it will go to.



SIZE
W 257 X 77
H 47 Y 329

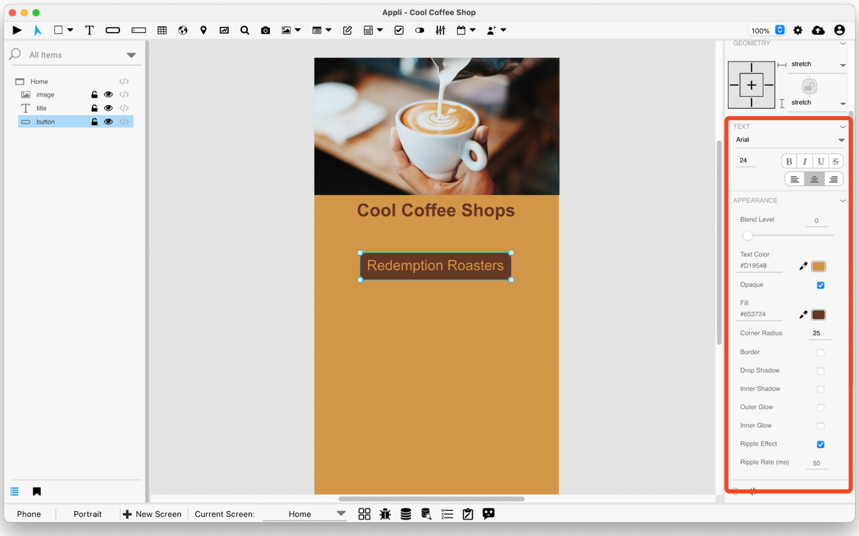
GEOMETRY
stretch
stretch

TEXT
Arial
39

APPEARANCE
Blend Level 0
Text Color #FFFFFF
Opaque ☑
Fill #707070
Corner Radius 0
Border □

Figure 82: Change button style

Much like our header text, the default appearance of the button doesn't match our application design. Let's change it using the property inspector on the right-side of the screen.



stretch
stretch
Arial
24
Blend Level 0
Text Color #D19548
Opaque ☑
Fill #653724
Corner Radius 25
Border □
Drop Shadow □
Inner Shadow □
Outer Glow □
Inner Glow □
Ripple Effect ☑
Ripple Rate (ms) 50

Figure 83: With new button style

The properties we changed:

- `Corner Radius`: 25. (be aware of that)
- `Fill Color`: #653724 (we actually used the eyedropper tool to pick the text header color).
- `Text Color`: #D19548 (Used the eyedropper to pick the screen background color).

For this demo application, we're going to list just two shops. Change the label of the button to *Redemption Roasters*. Remember to change the font size to something suitable for the button.

Now, we're ready to create our first *specific shop* screen.

16.2.2 Redemption Roasters Screen

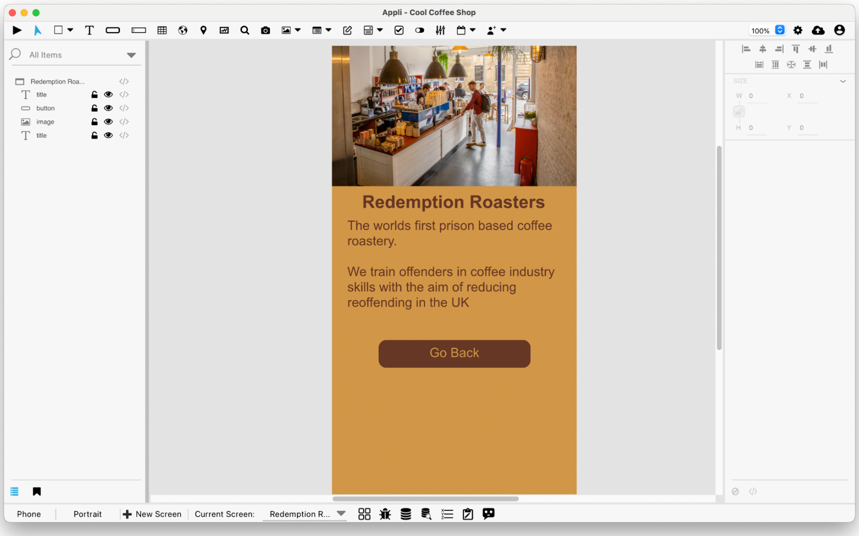
Using the controls at the footer, create a new screen named *Redemption Roasters*. You'll notice that the `backgroundColor` already matches the previous screen. That is because they're set per platform.

Our new screen will be quite simple. It will contain:

- A photo from the shop.
- Two text elements, one for the header, another for a description.
- A button to go back to the Home screen.

> Quick tip: you can copy and paste elements between screens. I assembled the screen in the screenshot below by copying the elements from the Home screen, pasting them on the Redemption Roasters screen, and altering their text.

Make your version of the screen look like this:



Redemption Roasters

The worlds first prison based coffee roastery.

We train offenders in coffee industry skills with the aim of reducing reoffending in the UK

Go Back

Figure 84: Redemption Roasters page

With those two screens in place, we're ready to add actions to those buttons and hook things up.

16.2.3 Adding an action to a button

Go back to the *Home* screen. Select the *Redemption Roasters* button. Look towards the property inspector. You'll see the *no-code button* and the *Low-Code button.* You can use them to set actions for the selected element or alter complex behaviours that are beyond what you can set using properties.

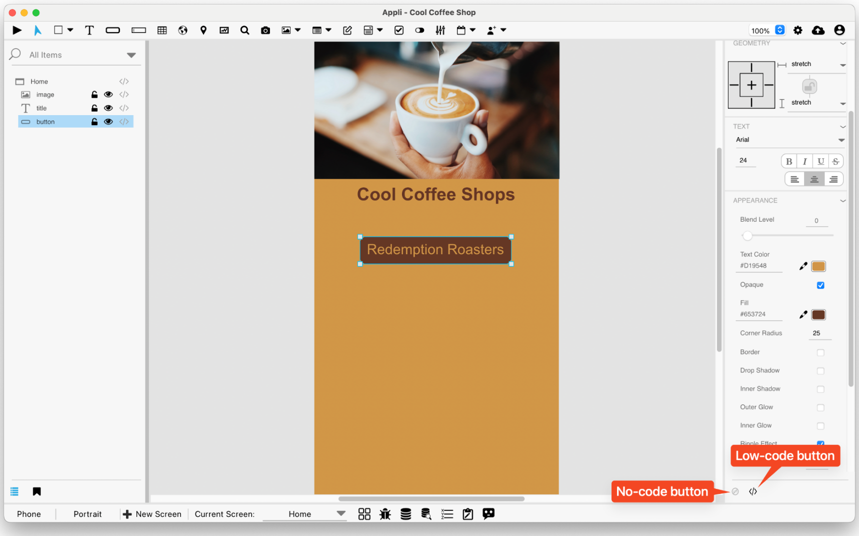


Figure 85: No-code and Low-code buttons

Buttons don't have *no-code* features, so that button is disabled. Clicking the *Low-Code button* will open the *Low-Code Action Script Editor*:

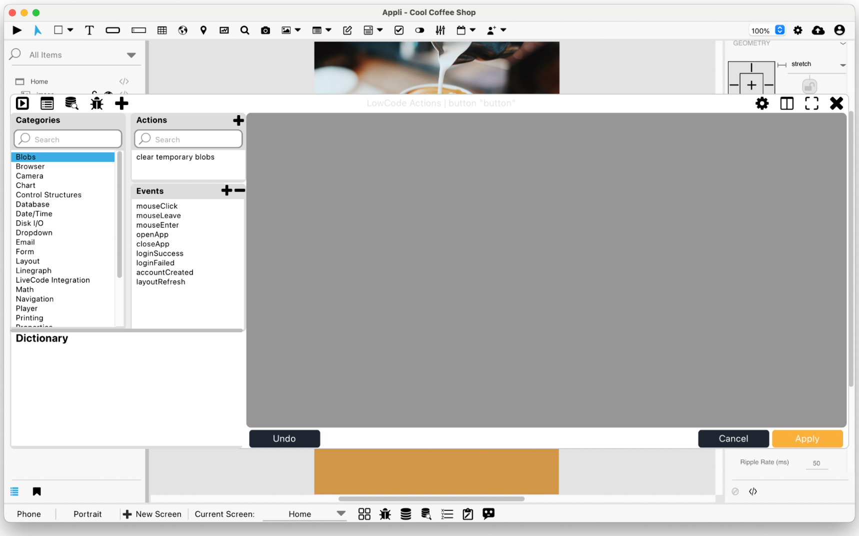
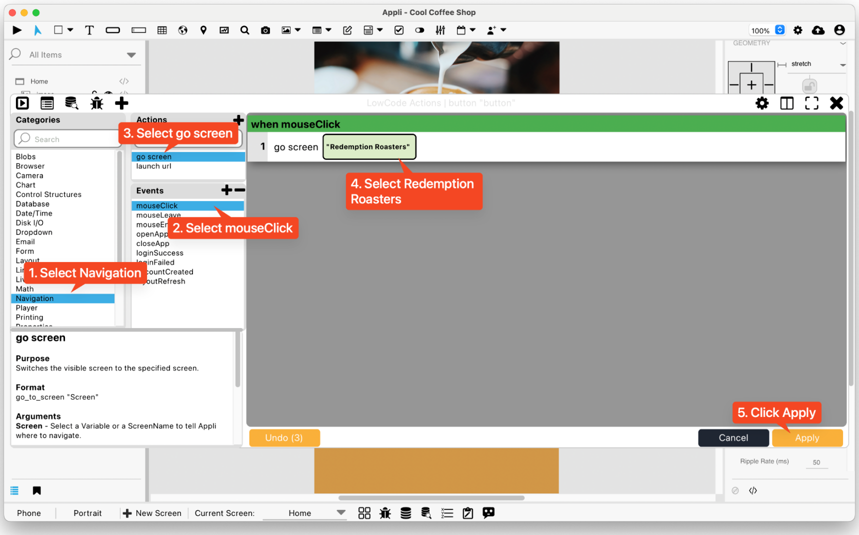


Figure 86: Low-Code interface

This editor is what we use to add actions to elements. You can learn more by reading the low-code documentation.

The action we want to use is *Go To Screen* under the *Navigation* category. Once you select it, the *action script* will be displayed in the pane on the right. Click the blue *Screen* link to select which screen you want the button to go to when clicked.



1. Select Navigation
2. Select mouseClick
3. Select go screen
4. Select Redemption Roasters
5. Click Apply

when mouseClick

```
1  go screen "Redemption Roasters"
```

go screen

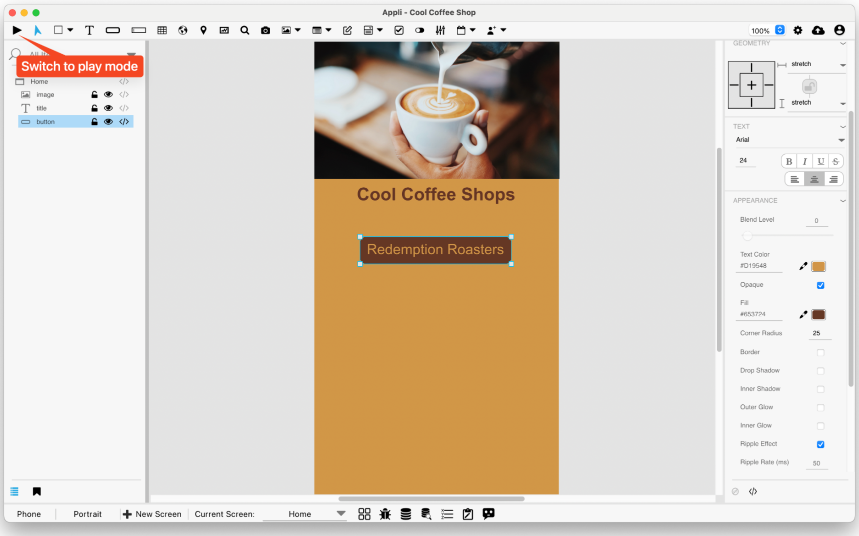
Purpose
Switches the visible screen to the specified screen.

Format
go_to_screen "Screen"

Arguments
Screen - Select a Variable or a ScreenName to tell Appli where to navigate.

Figure 87: Low-Code script for the button

Done! You can click the *Play button* at the top-right of the Tools palette to switch to play mode. This mode allows you to test out your application.



stretch
stretch
Arial
24
Blend Level 0
Text Color #D19548
Opaque ☑
Fill #653724
Corner Radius 25
Border □
Drop Shadow □
Inner Shadow □
Outer Glow □
Inner Glow □
Ripple Effect ☑
Ripple Rate (ms) 50

Figure 88: Play mode switch

Try clicking the *Redemption Roasters* button, it should navigate to the *Redemption Roasters* page.

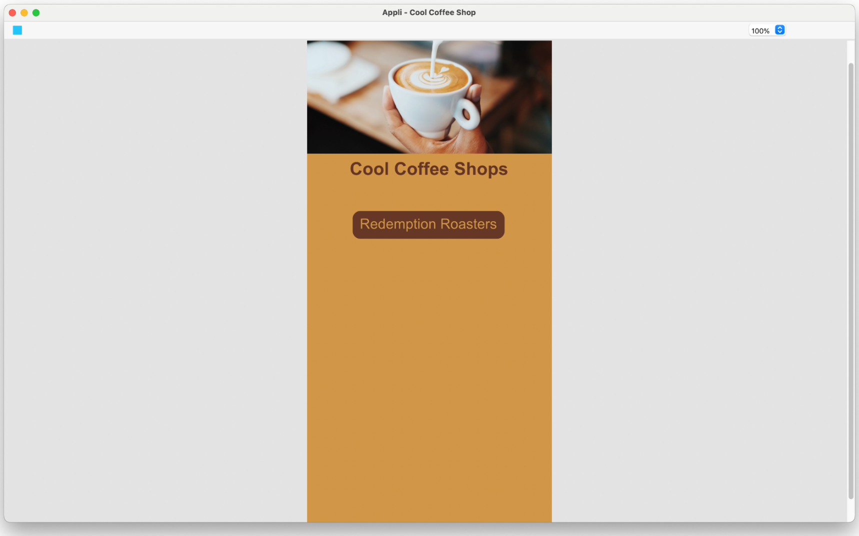


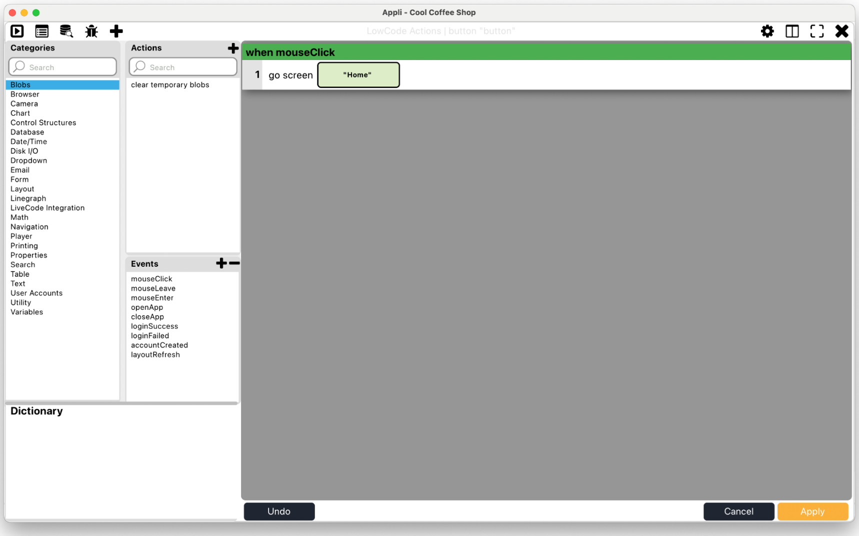
Figure 89: Interacting with the app in play mode

Once you're done with your testing, you can switch back to design mode by selecing the *Pointer tool* at the top-left side of the *Tools palette*.

16.2.4 Going back to the Home screen

Go to the *Redemption Roasters* screen. We need to wire the *Go Back* button so that it has an action to navigate back to the *Home* screen.

It is the same process as before. Select the element, click the *Low-Code button*, add a *Go To Screen* action and select the *Home* screen.



```
when mouseClick
1 go screen "Home"
```

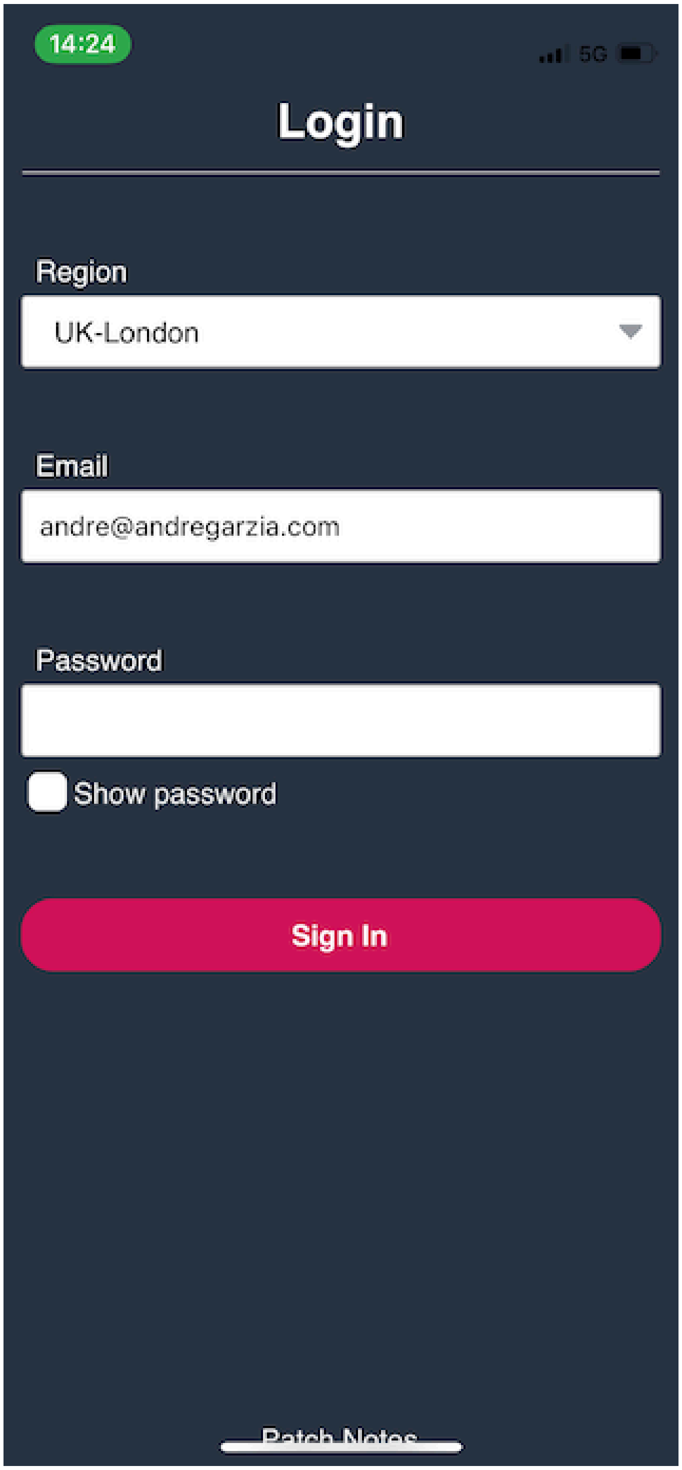
Figure 90: Low-Code for Go Back button

We're done with our little app! Read on to learn how to try it out in your mobile device.

16.3 Running on mobile

You should have the *Appli Player* installed on your mobile device for this section of the tutorial to work.

Launch the Appli Player app and enter your login details. They are the same as the ones you used for your Appli IDE login.



Region
UK-London

Email
andre@andregarzia.com

Figure 91: Player login

After logging in, you'll see the *Player Home Screen* that lists the apps *you have added to it.* It doesn't list your apps automatically. You need to click the *+ Add App* button at the bottom of the screen…

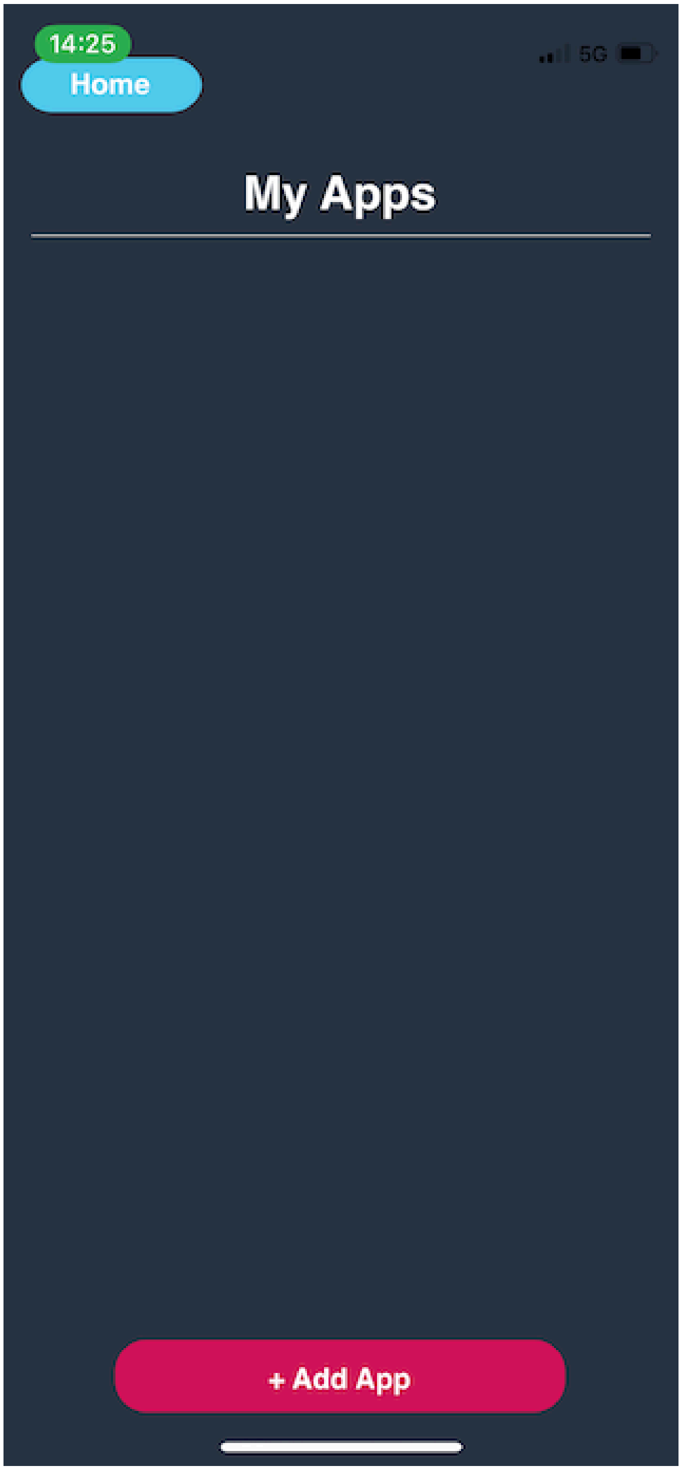
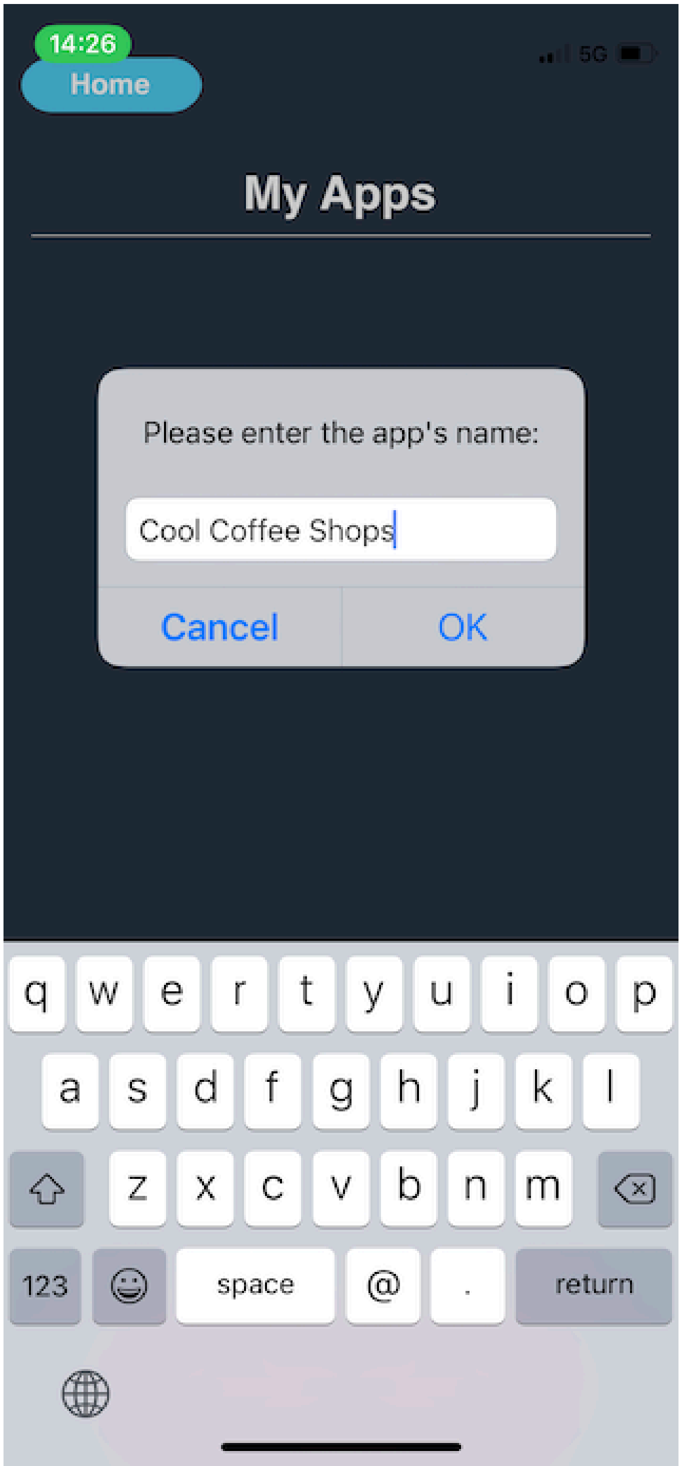


Figure 92: Adding an app

… and type in the app’s name you want to add.



Please enter the app's name:

Cool Coffee Shops

Figure 93: Adding an app

The app will be appear listed on the Home screen.

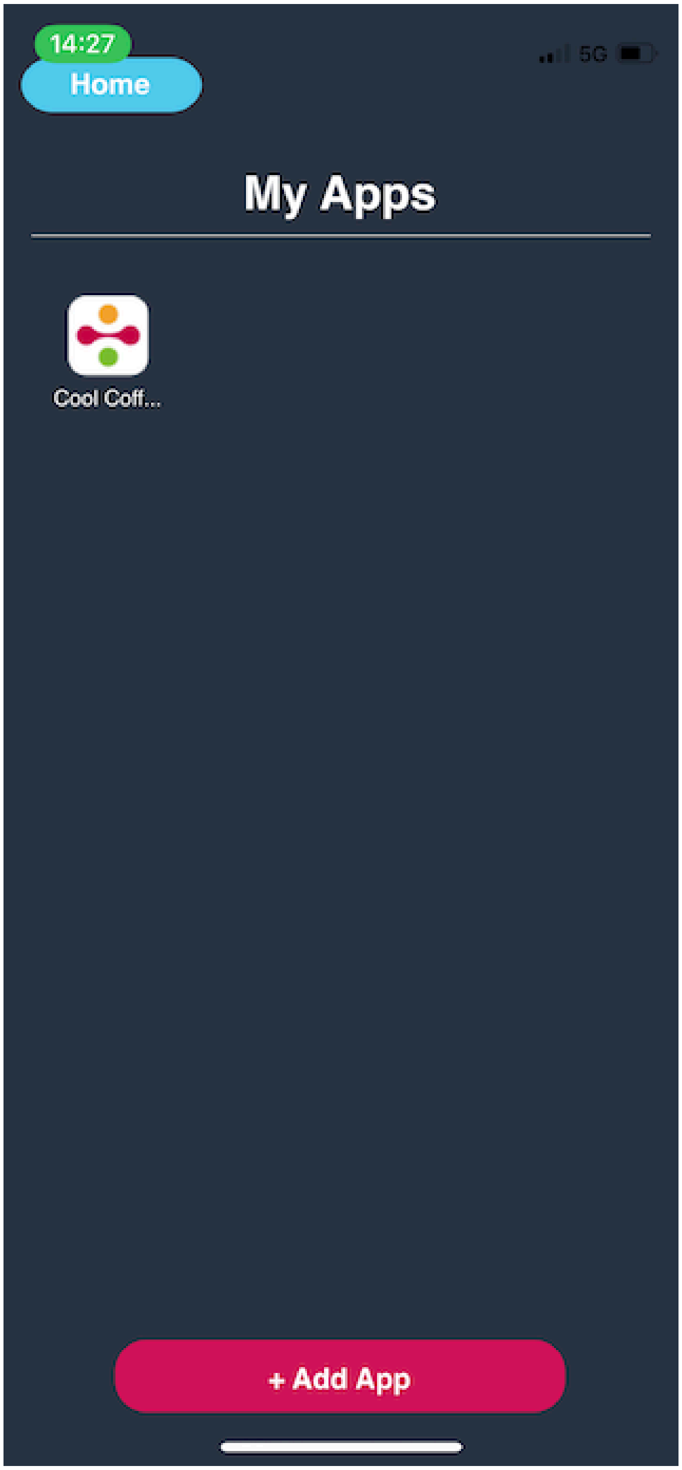


Figure 94: Adding an app

Clicking on it will launch your app. You can interact with it. There is a floating draggable round button you can use to refresh the app in case you made changes since launching it, and other controls to sign out and go back home.

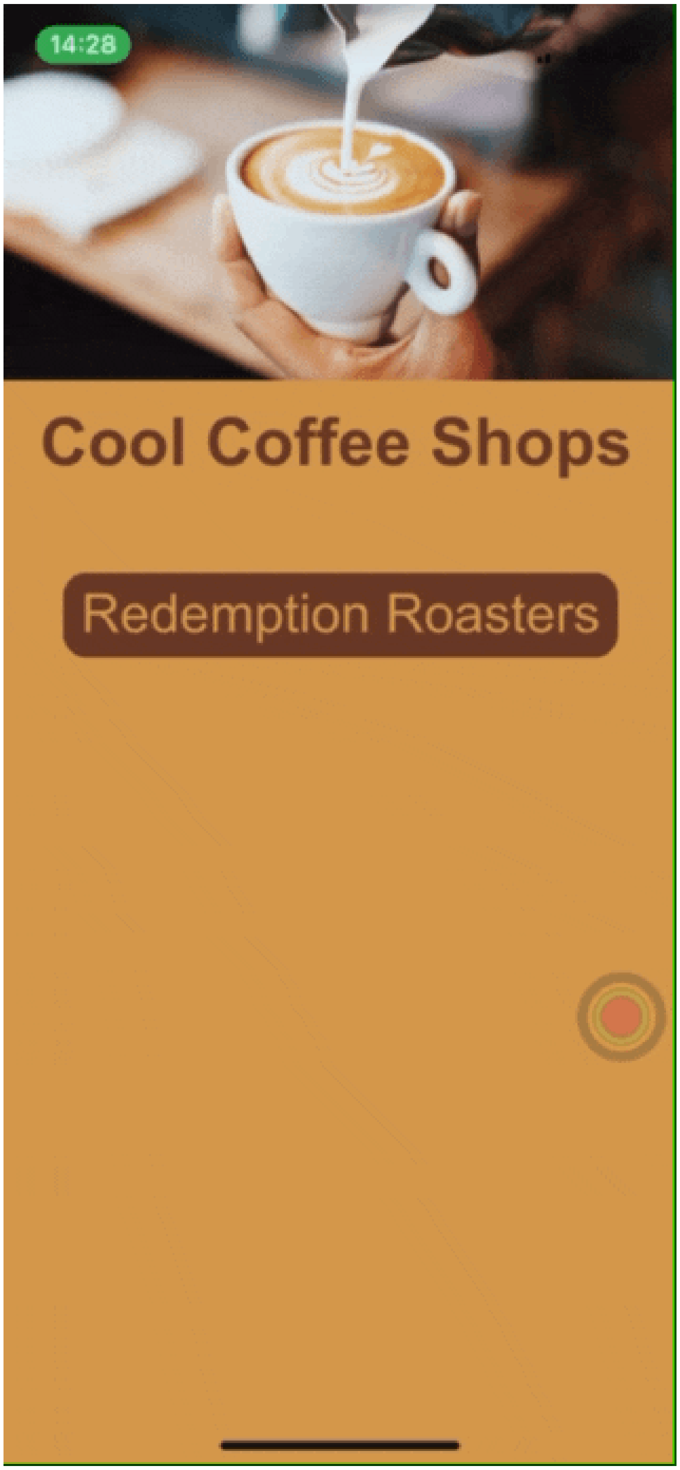


Figure 95: Adding an app

16.4 Final words and where to go next

You might have noticed that I mentioned we would add two coffee shops to this app. That second coffee shop page is your homework. Add in your favourite coffee shop, or any other shop, to the app and test it on the Appli player.

17 Tutorial: Address Book App

In this tutorial, we're going to build a minimal address book application, just enough for us to learn about how to use databases inside Appli.

Begin by creating a new project called *Address Book* and a phone application with the same name as that project in portrait orientation. Just like you did for the *Cool Coffee Shop* tutorial.

17.1 Creating the contacts table

Databases are used by apps to record and retrieve data. They are organised into collections of similar records, so a project management database might contain *projects* and also *tasks*. All records in each of these collections have the same structure as the other records in the same collection. In Appli, we call these collections _tables___.

Your application can have as many tables as it needs. For our sample address book management application, we're going to have just one table called *contacts*.

To create a new table, click the *data modeler* icon in the footer. It looks like a stack of discs. Use the *New Table* button to create a new table named *contacts*.

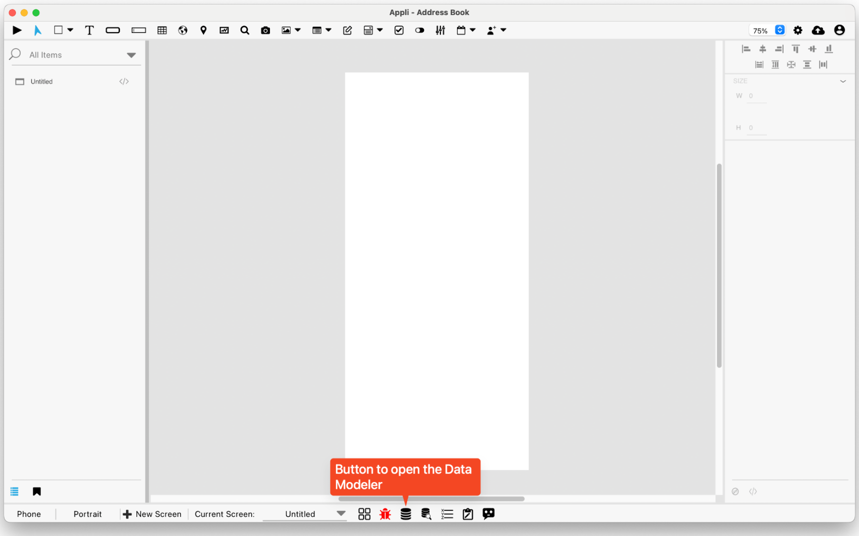


Figure 96: Opening the Data Modeler tool

All Appli projects come with to tables — one for storing binary data called *cdbBlobs* and another to store user accounts called *cdbUsers* — even if you don't plan to use those features, they are part how Appli works and will always be there for you.

Right-click the background of the *data modeler* tool to display the context menu. You'll see an *Add New Table* item, click it to add a new table.

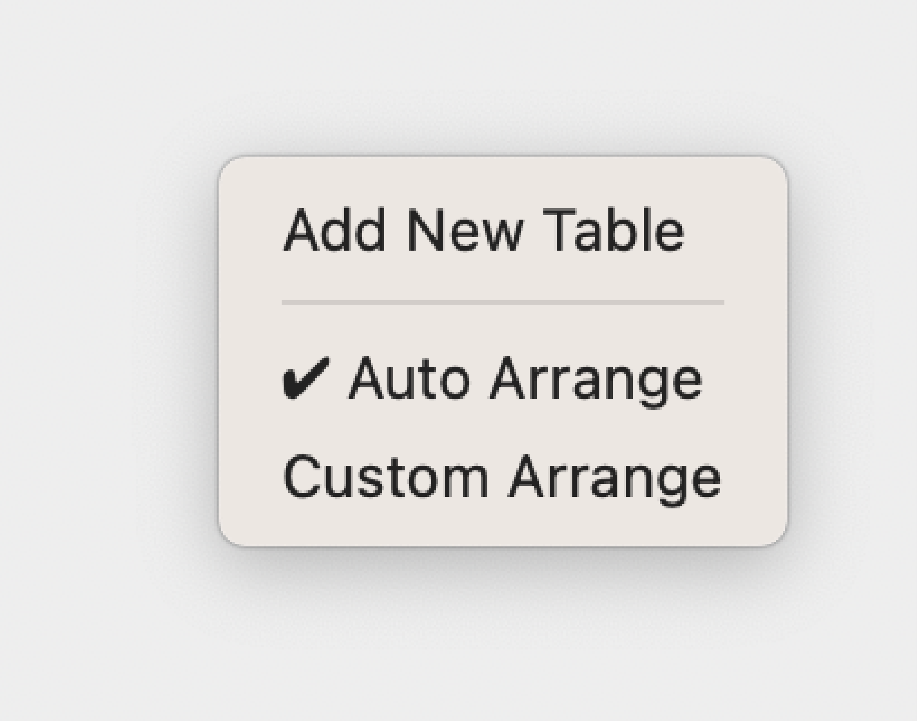
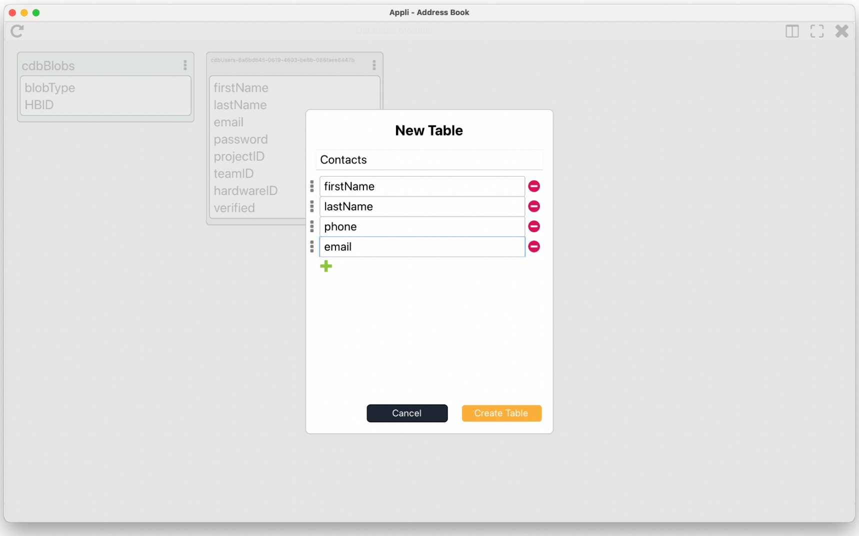


Figure 97: Context menu

A table has keys in them. Think of each key like a field in a paper form. Based on the screenshot below, add keys for `firstName`, `lastName`, `phone`, and `email`. These are the data that we're going to collect for each entry in our address book.



Contacts

firstName

lastName

phone

email

Figure 98: Table setup.

Once you click *Create Table* we're ready to move along and create a record.

17.2 Creating a form

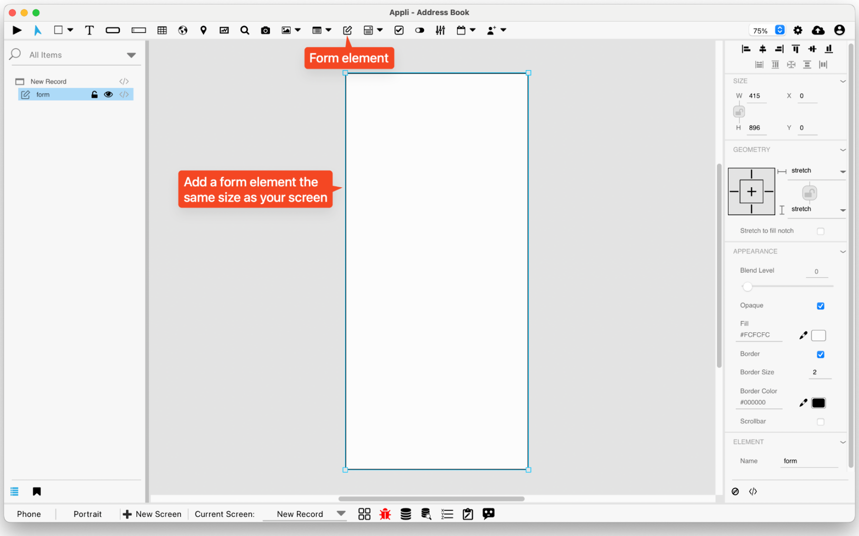
Rename the current screen to *New Contact* by clicking its name on the footer. This will help us later as we create more screens.

Our aim for this section is to create a form that enables us to add records to our *contacts* table. There are two ways of doing this. You can do it all manually by creating each field and button by yourself and connecting them with the table, it is good to learn this way because that teaches you how things work. There is an easier way this is using a tool that does the work for you and essentially goes through the same steps but in an automated way. We'll do the manual one first and then I'll show you the automated one.

17.2.1 Creating a form, manually.

To do that, we're going to use a *form element*. This element is a container element. This means we're going to place it on our screen and then we're going to place other elements inside it.

The *form element* is the one that looks like a paper with a pencil pointing at it. It can be selected with the mouse or by its keyboard shortcut SHIFT+F. Select it and draw a square the same size as the screen with it.



Form element

Add a form element the same size as your screen

SIZE
W 415 X 0
H 896 Y 0

GEOMETRY
stretch
stretch
Stretch to fill notch □

APPEARANCE
Blend Level 0
Opaque ☑
Fill #FCFCFC
Border ☑
Border Size 2
Border Color #000000
Scrollbar □

ELEMENT
Name form

Figure 99: Adding a form to the playground.

Let's add some fields to our form and some buttons. We're going to name them with the same names as the keys we created on the table. This will make it easier for us to remember their name. We'll also create two extra buttons, one for saving and another for going back to the main screen (which we'll create later).

To create a field, select the *field element* (keyboard shortcut: F) and draw each field inside the form rectangle. You should see them appear under the form in the *project browser* panel.

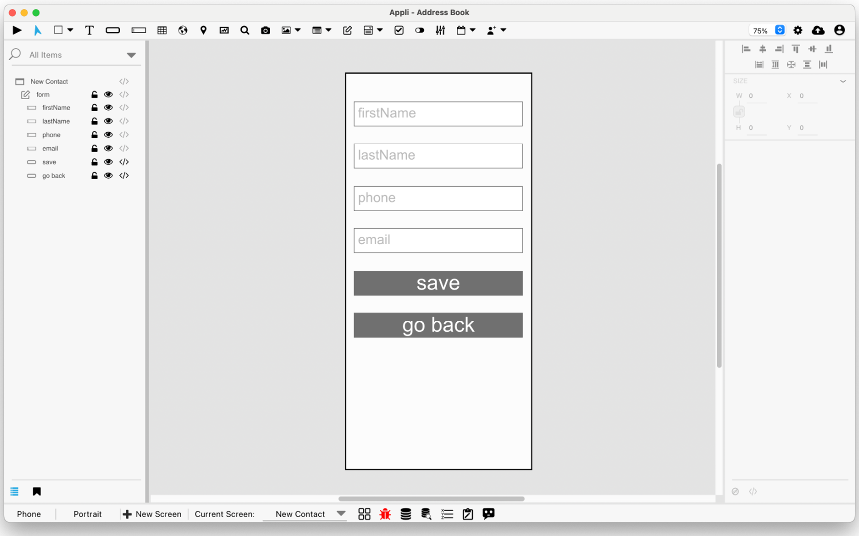
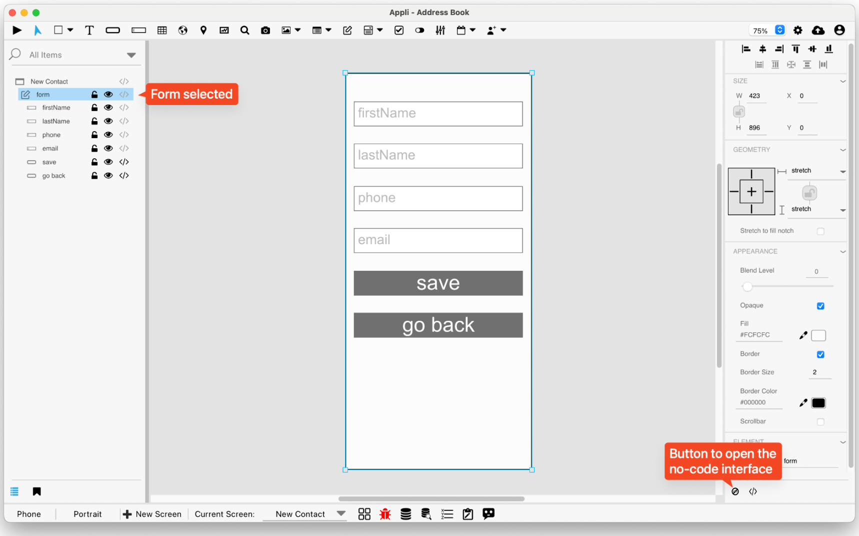


Figure 100: Adding a fields to the form.

What we're going to do next is a simple action, but one that takes a very long time with other programming tools. We're going to connect the fields we just created with keys in the *contacts* table, and we can do all that with just no-code.

Select the form by clicking it either on the playground or in the project browser. On the property inspector for the form, click the no-code button at the bottom of the interface.



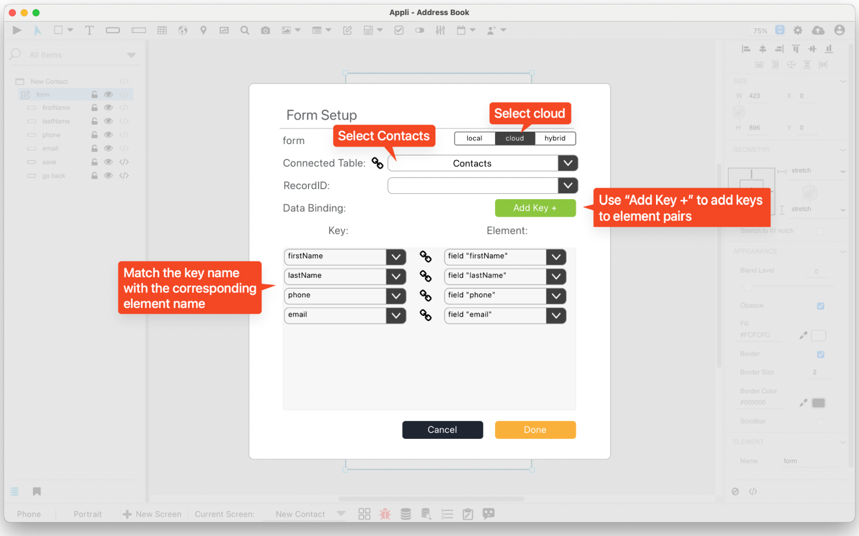
W 423 X 0
H 896 Y 0
stretch
stretch
Stretch to fill notch □
Blend Level 0
Opaque ☑
Fill #FCFCFC
Border ☑
Border Size 2
Border Color #000000
Scrollbar □
form

Figure 101: Opening the no-code interface for the form.

In the no-code interface, do the following:

1. Select *cloud*.
2. Select the *contacts* table.
3. Using the *Add Key +* button, add as many entries as necessary to cover all keys on the table.
4. For each entry, select the table key and the field of the same name to connect them.

It should look like the screenshot below after you've finished adding everything.



Appli - Address Book

Form Setup

form

○ local ● cloud ○ hybrid

Connected Table: Contacts

RecordID:

Data Binding: Add Key +

Key:	Element:
firstName	field "firstName"
lastName	field "lastName"
phone	field "phone"
email	field "email"

Cancel Done

Select cloud

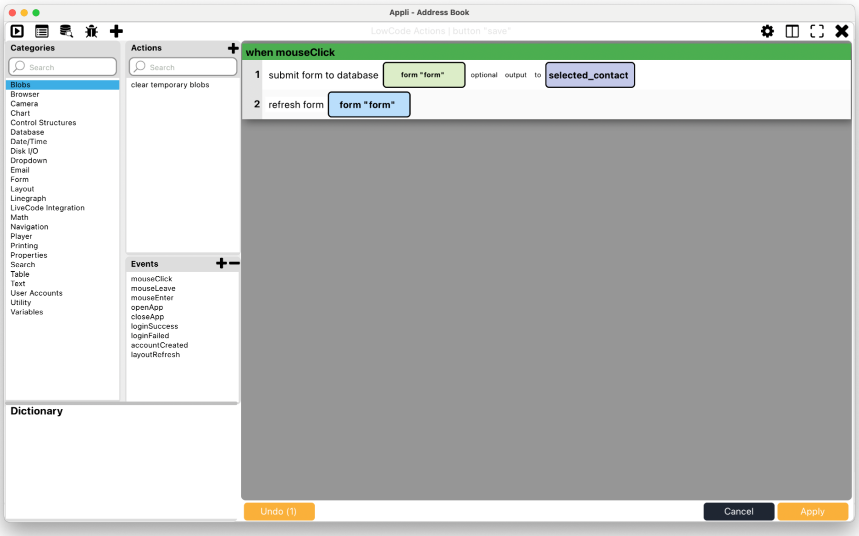
Select Contacts

Use "Add Key +" to add keys to element pairs

Match the key name with the corresponding element name

Figure 102: Binding keys and elements.

Now that the form fields are all bound to their respective keys, we can use the low-code interface to save the record to the table. Select the *Save* button element on the form, open its low-code editor to add a `mouseClick` event. Be aware that to select elements inside a form, you need to double-click them. You can also select them by clicking on them on the project browser.



```
when mouseClick
1 submit form to database form "form" optional output to selected_contact
2 refresh form form "form"
```

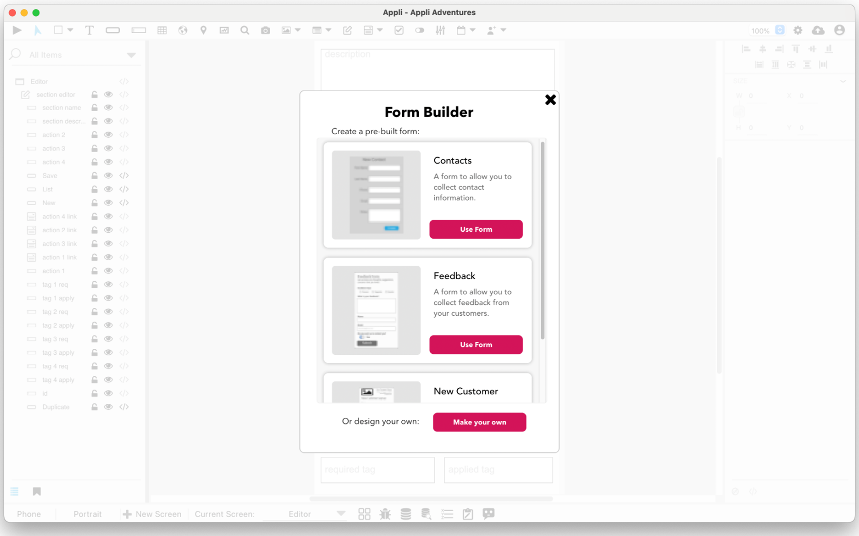
Figure 103: Low-code for the save button.

That code will send the form data to Appli and save the record. The refresh action will empty the form after the contact is saved. Both actions are under the form category. Remember to select the form element as the argument to those actions.

Go into *play mode* and add a couple of records of your own.

17.2.2 Using the form builder to create a form

This is another way to build a form, it is much faster. The form builder tool is located at the footer. It is an icon shaped like a clipboard. When you first launch it, it shows some pre-made templates you can add to your project.



Form Builder

Create a pre-built form:

Contacts

A form to allow you to collect contact information.

Feedback

A form to allow you to collect feedback from your customers.

New Customer

Or design your own:

Figure 104: Form Builder

Instead of picking any of those, we'll select *Make your own* from the bottom of the form builder window.

It will open a new editing window to specify the form name and its fields. For each field, you can pick a label and a type.

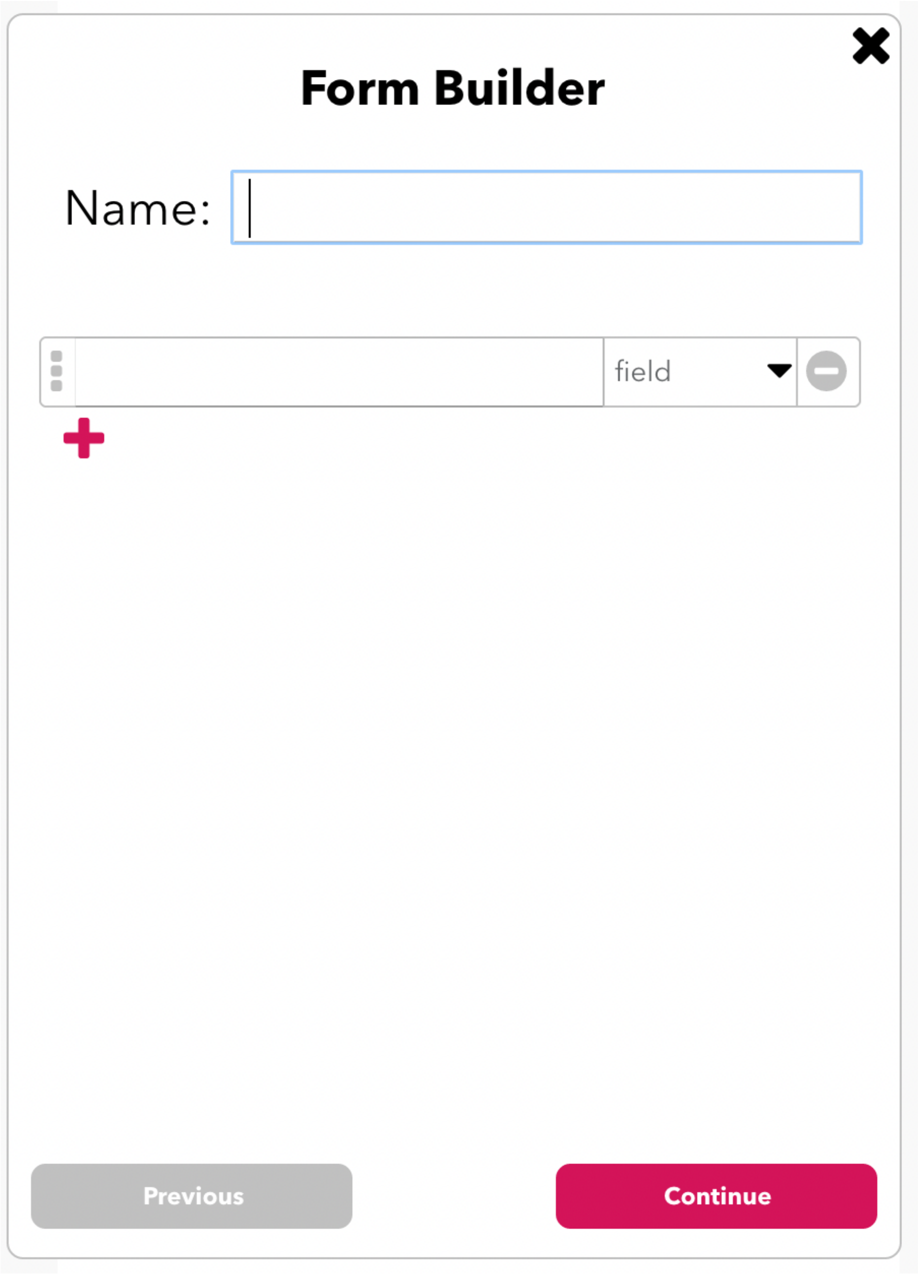
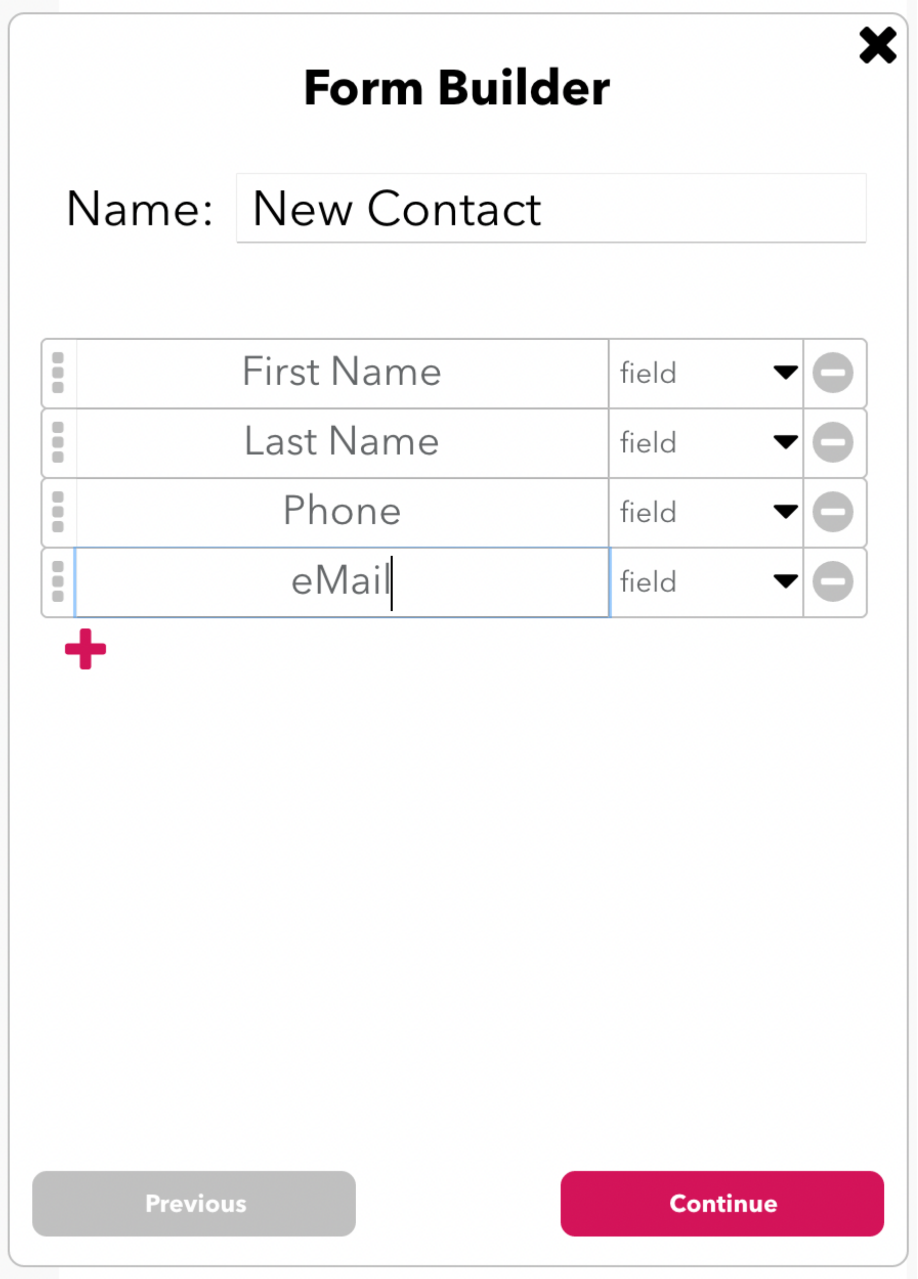


Figure 105: Editing form fields

Let's fill it in with our address book fields.

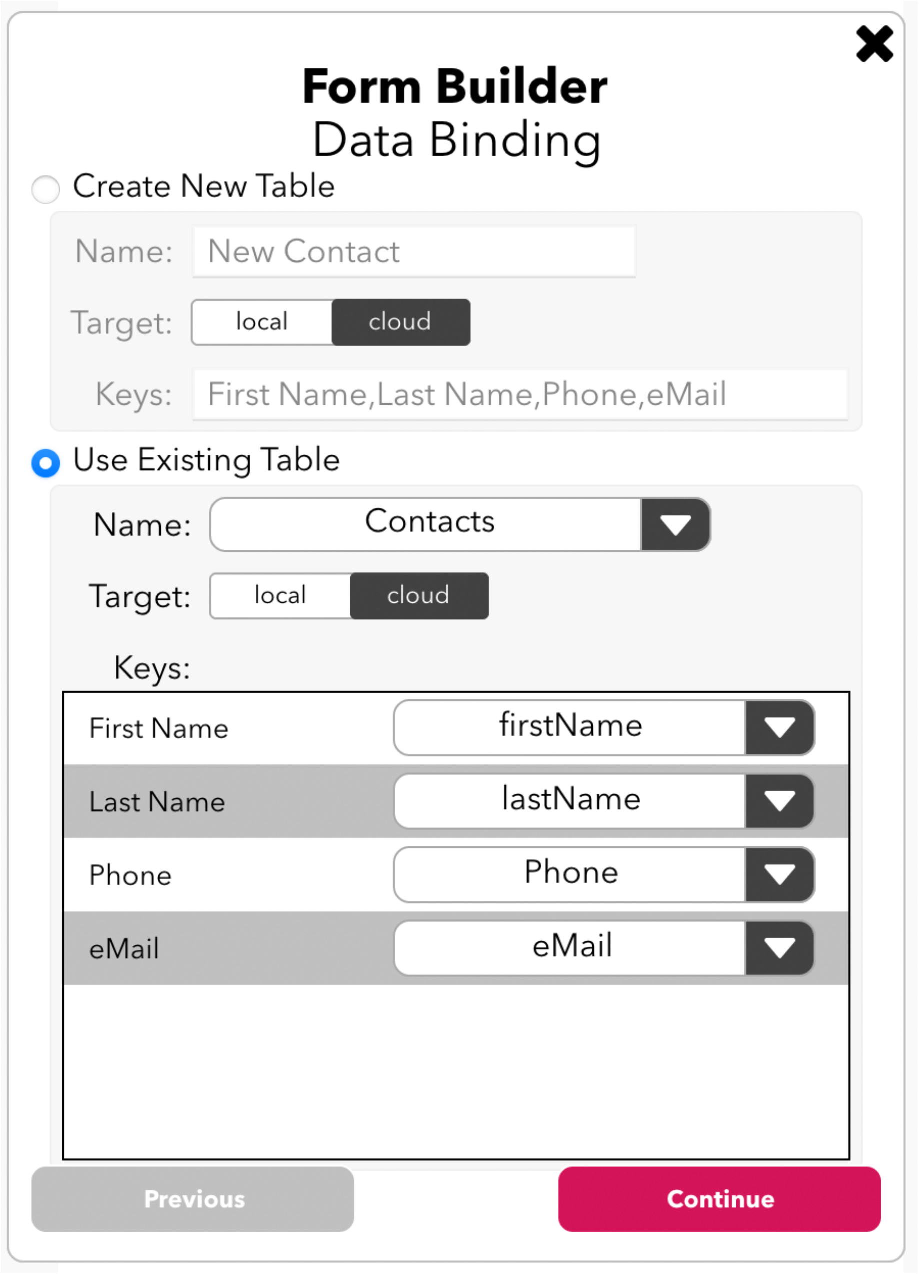

Form Builder

Name: New Contact

First Name	field	
Last Name	field	
Phone	field	
eMail	field	

Figure 106: With our fields filled in

After specifying the fields, it is time to do data binding. You can create a new table or use an existing one. Since we already have the table in place, lets select it and then use the dropdown menus next to each label to select which table key they connect to.



Form Builder

Data Binding

○ Create New Table

Name: New Contact

Target: local | cloud

Keys: First Name,Last Name,Phone,eMail

● Use Existing Table

Name: Contacts

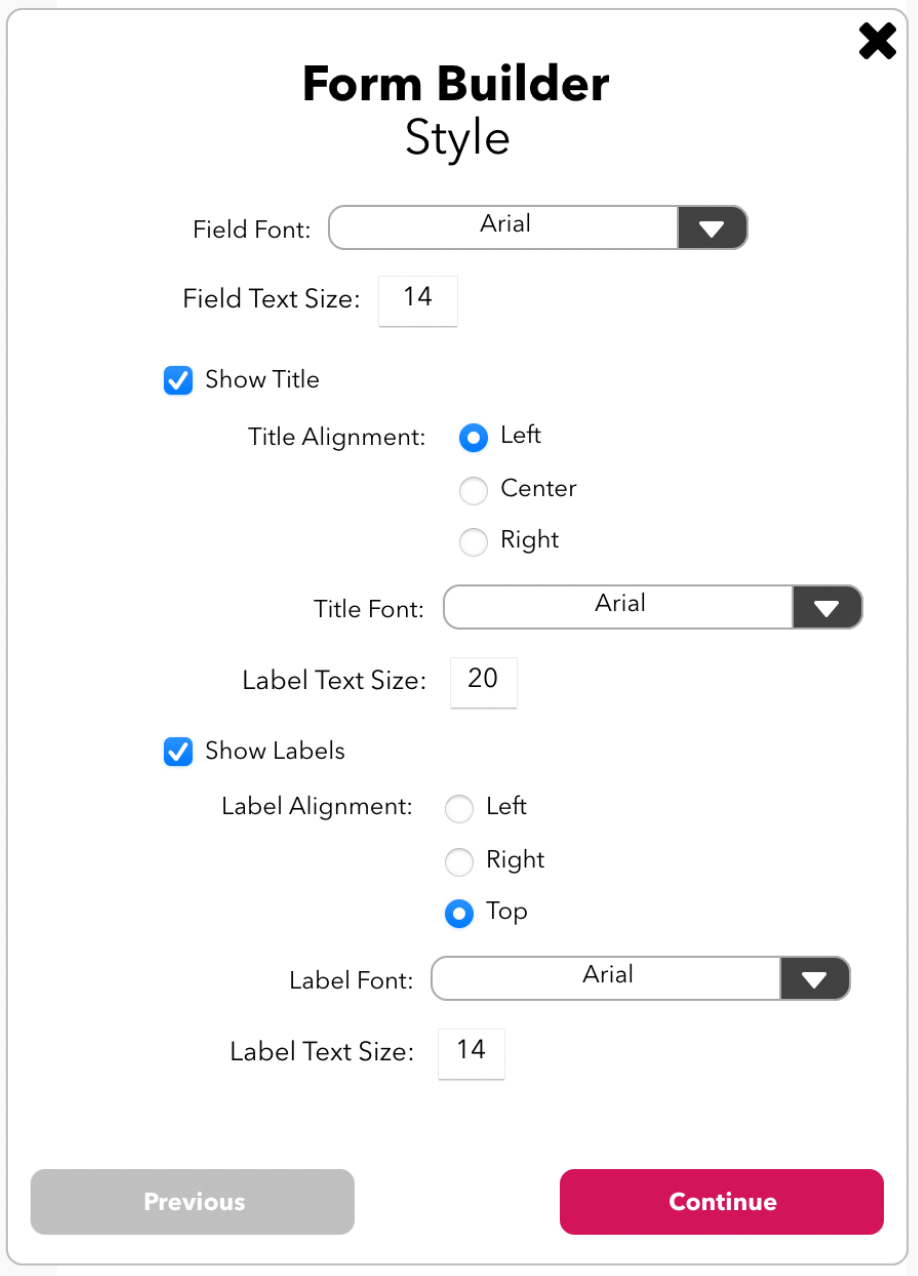
Target: local | cloud

Keys:

First Name	firstName
Last Name	lastName
Phone	Phone
eMail	eMail

Figure 107: Data Binding in the Form Builder

Next step is customising the style used for the form. That means fonts, sizes and alignment.



Field Font: Arial

Field Text Size: 14

☑ Show Title

Title Alignment: ● Left ○ Center ○ Right

Title Font: Arial

Label Text Size: 20

☑ Show Labels

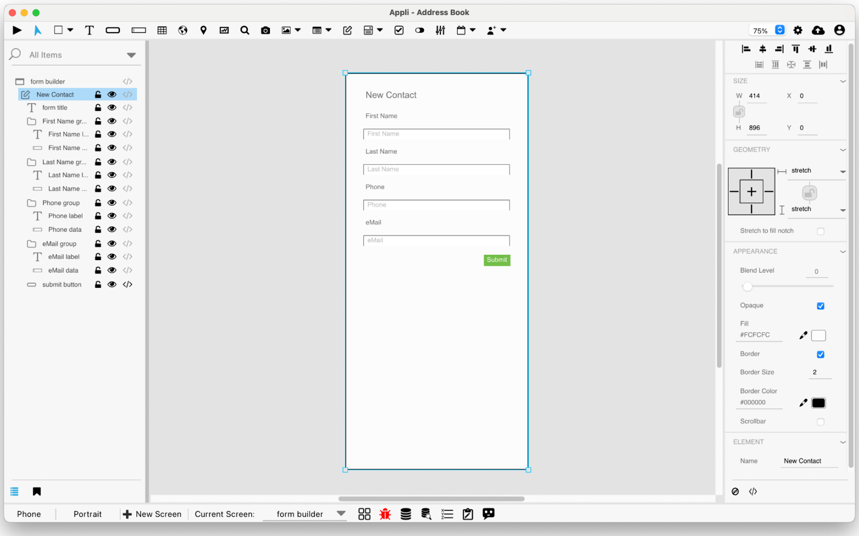
Label Alignment: ○ Left ○ Right ● Top

Label Font: Arial

Label Text Size: 14

Figure 108: Customising text styles

With that done, Appli will build the form with all the necessary elements and data bindings as if you had done it yourself.



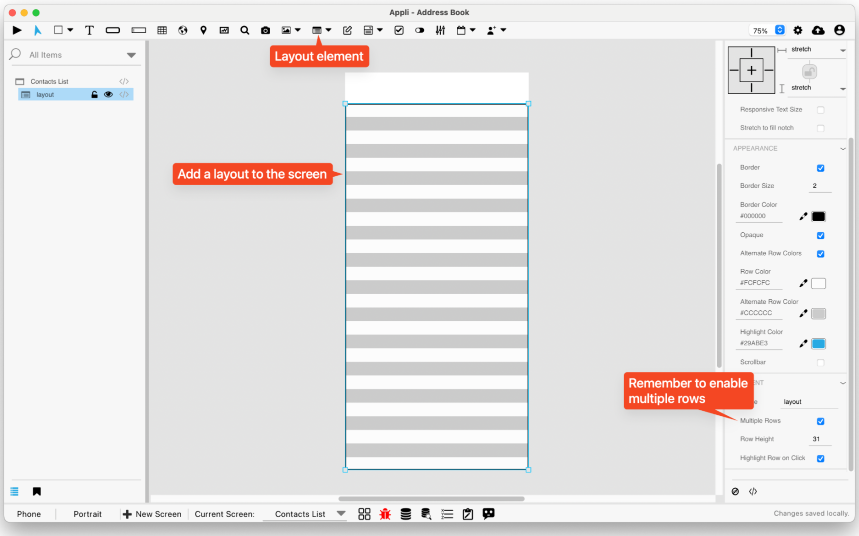
W 414 X 0
H 896 Y 0
stretch
stretch
Stretch to fill notch □
Blend Level 0
Opaque ☑
Fill #FCFCFC
Border ☑
Border Size 2
Border Color #000000
Scrollbar □
Name New Contact

Figure 109: Form build with the Form Builder

17.3 Listing contacts

Create a new screen named “Contact List”. Listing records is similar to creating records in which we’ll use a container element to connect elements inside it to the database. Here, we’re going to use a *layout element*. These elements can list records from a table and create scrollable regions in your app.

Add a *layout element* to your new screen, leaving some space at the top for other controls we’ll be adding shortly. The layout will appear like a rectangle. Using the property inspector, enable *Multiple Rows* for the layout. It will become a list of rows with alternating colors.



Layout element

Add a layout to the screen

Remember to enable multiple rows

stretch

stretch

Responsive Text Size □

Stretch to fill notch □

APPEARANCE

Border ☑

Border Size 2

Border Color #000000

Opaque ☑

Alternate Row Colors ☑

Row Color #FCFCFC

Alternate Row Color #CCCCCC

Highlight Color #29ABE3

Scrollbar □

layout

Multiple Rows ☑

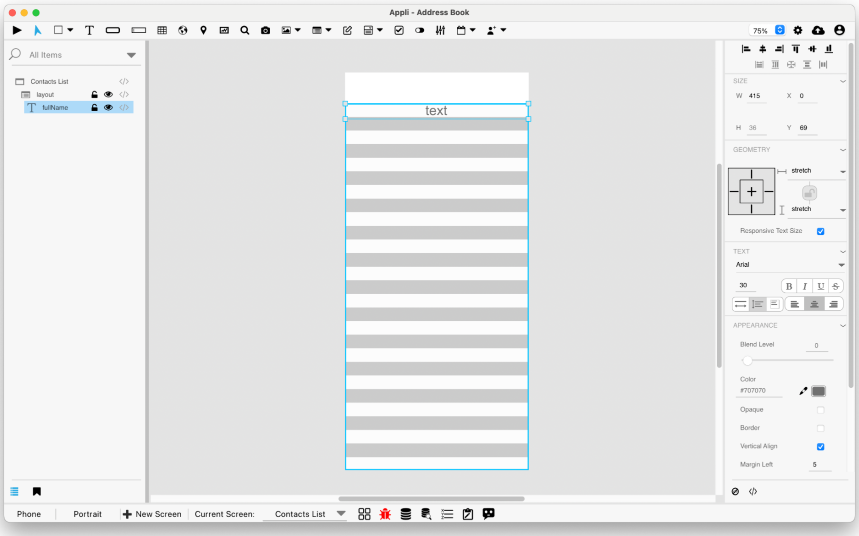
Row Height 31

Highlight Row on Click ☑

Figure 110: Adding the layout element.

When we enable *multiple rows* for a layout, Appli picks up the elements we have in it, and repeats them for each record in the table. So, if we add a *text element* to the layout, it will repeat this element for all records. This is how we create a list of records. Imagine you're just creating the first line of a list. You can make it as complex as you want with multiple elements in it if you need them. Appli will pick those elements and assemble a list using them as a model for each item in the list.

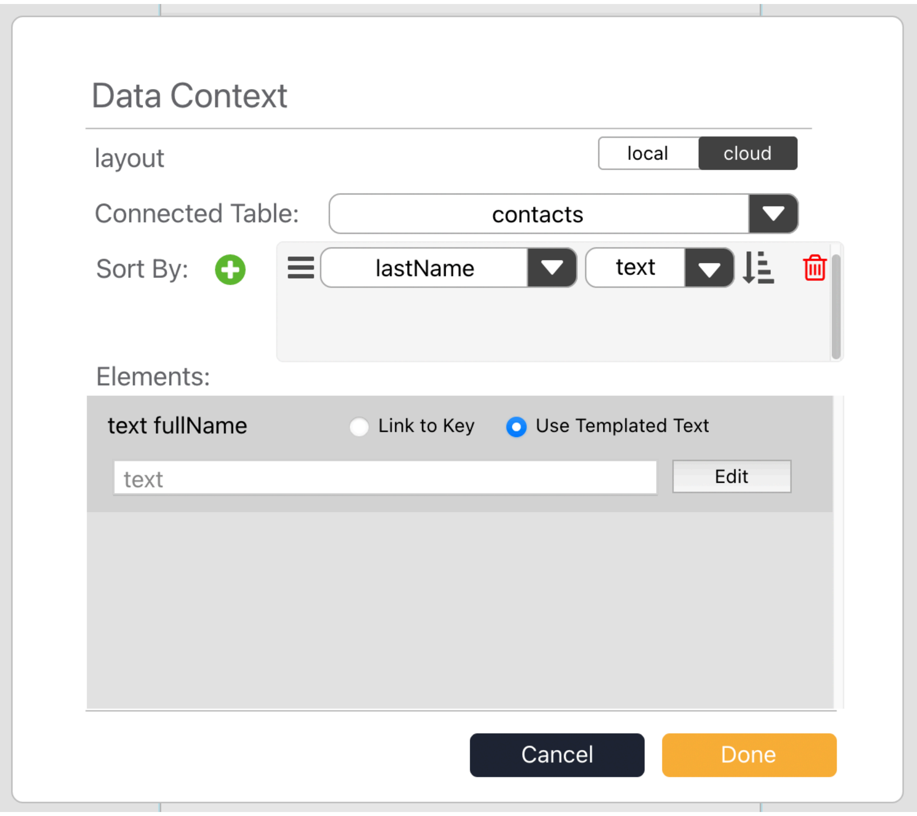
To keep things simple, we're going to just add a single *text element* to the layout. Position it so that it matches the space in the first line of the layout. If the line is too narrow, use the property inspector to make it larger (by changing the *layout row height* property). Name that text element "fullName" (notice that we don't have a full name on the table, only *firstName* and *lastName*).



W 415
X 0
H 36
Y 69
stretch
stretch
Responsive Text Size ☑
Arial
30
Blend Level 0
Color #707070
Opaque □
Border □
Vertical Align ☑
Margin Left 5

Figure 111: Adding a text element to the layout.

The text element will appear inside the layout in the project browser. Use the no-code interface for the layout to connect it to the *contacts* table.



Data Context

layout

local [✓ cloud]

Connected Table: contacts

Sort By: lastName text

Elements:

text fullName ○ Link to Key ● Use Templated Text

text

Figure 112: Initial data context setup for the layout

Select *cloud* and the *contacts* table. Sort the layout by the `lastName` field, select *text* as the ordering format and make it ascending. Don't click *done* yet, we still need to set up the elements inside the layout.

We have a single element inside the layout, a text element called `fullName`. That is why it is the only element appearing in the elements section in that dialog. If we had more elements in the layout, they would all appear inside this section.

Keep the *Use template text* radio selected. Template text is a way for us to change the content of a field by adding text of our own and interpolating that content with data from the database.

The preview shown in that section shows *text* because that is what we currently have set as the content for that field. Click *edit* to change it, a new dialog will appear.

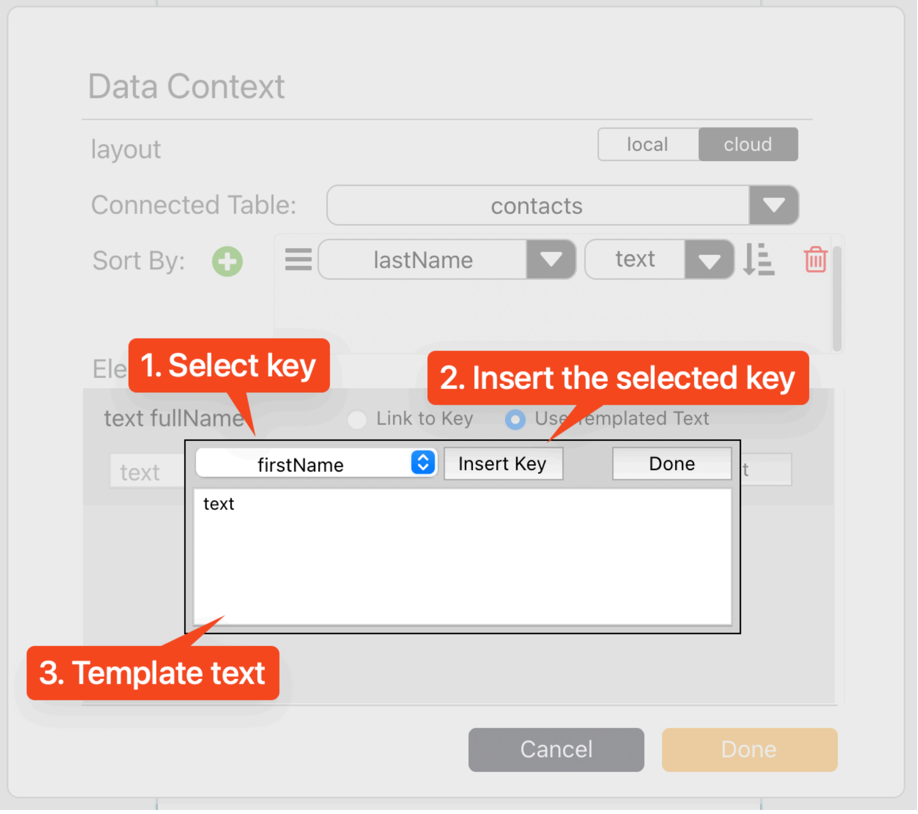
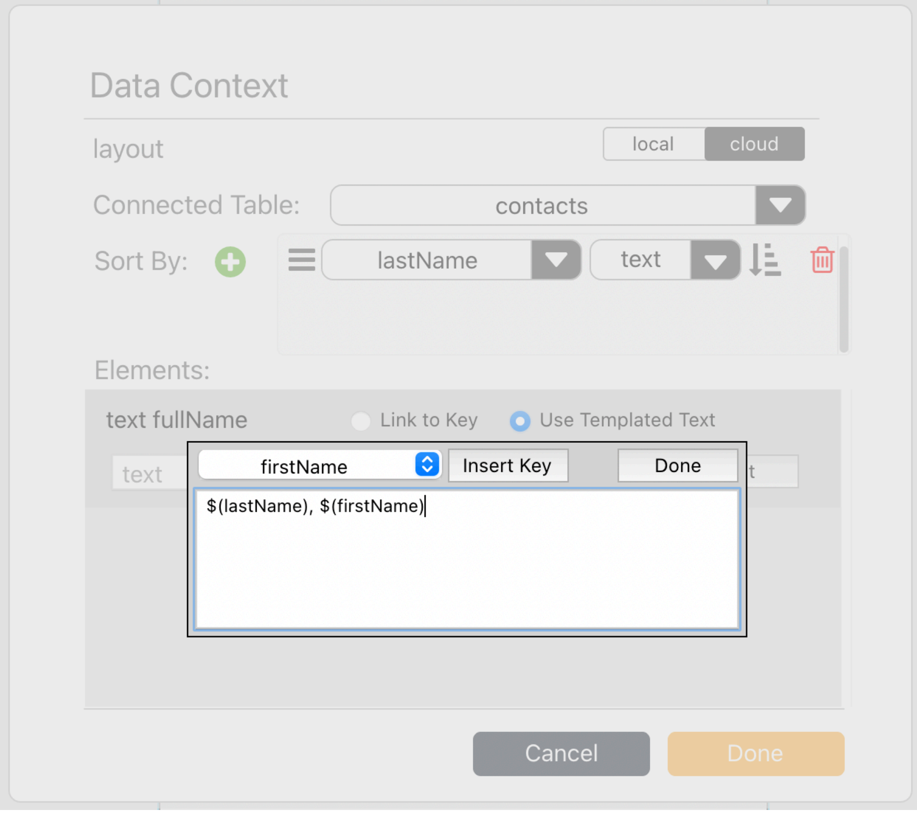


Figure 113: Template text editing dialog.

We want to list our contacts using their last name followed by a comma and then their first name (i.e. such as *Doe, John*). To do that, we can select the `lastName` key in the dropdown box, click the *Insert Key* button, edit the template text to put a comma after it, then do the same to insert the `firstName`. The final template text should look like this:



Connected Table: contacts

Sort By: lastName text

text fullName ○ Link to Key ● Use Templated Text

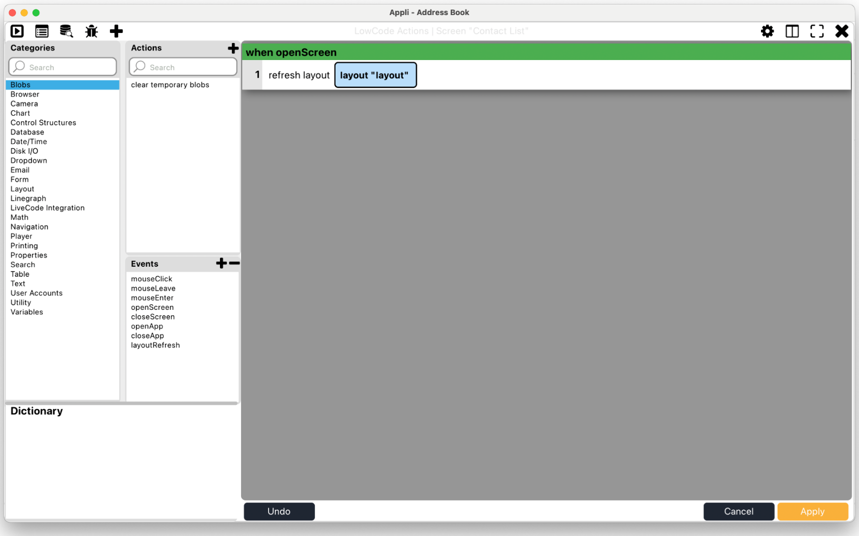
firstName

```
$(lastName), $(firstName)
```

Figure 114: Template text editing dialog.

After that, click *done* to close that template editor, and then *done* to close the data context setup.

To make sure that this layout is always up to date when we go to this screen, select the screen in the project browser and open its low-code editor. You can simply click the `</>` icon next to the screen name in the project browser.



```
when openScreen
1 refresh layout layout "layout"
```

Figure 115: Making sure the layout is up to date

Be aware that we're adding a *openScreen* event and not the *mouseClick* event like we usually do. This event will trigger every time that screen opens.

Using the pop-up menu in the footer to select screens, drag and drop the screens to reorder them so that *Contact List* is the first one.

Go into *play mode* and see your contact list working.

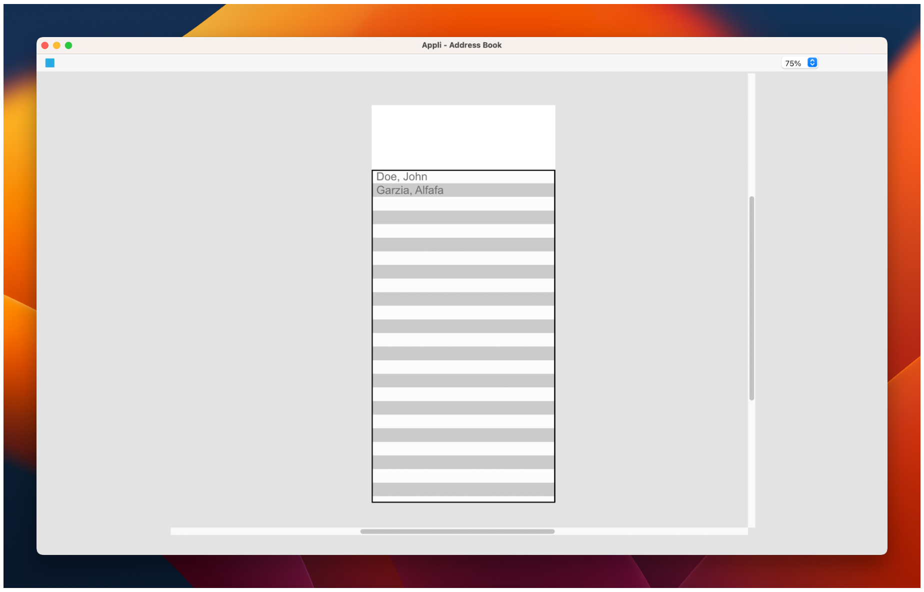
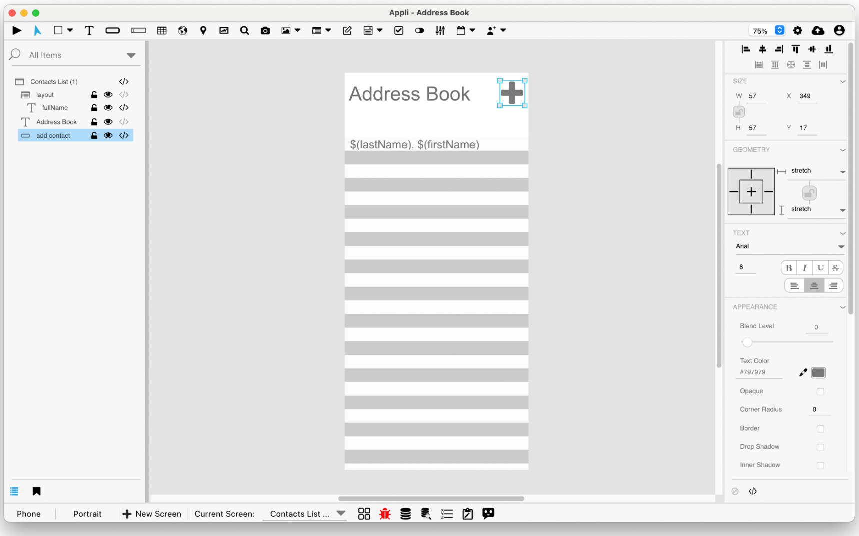


Figure 116: Sample contact list

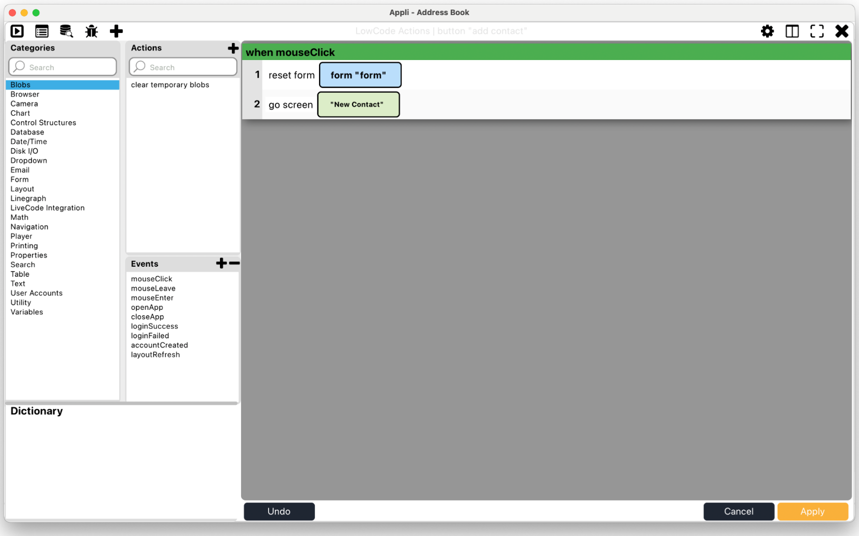
Before we move on, let's dress up this screen a bit more. Add a text element to be used as a header, and a button to add new contacts.



W 57 X 349
H 57 Y 17
stretch
stretch
Arial
8
Blend Level 0
Text Color #797979
Opaque □
Corner Radius 0
Border □
Drop Shadow □
Inner Shadow □

Figure 117: Adding more elements to the screen

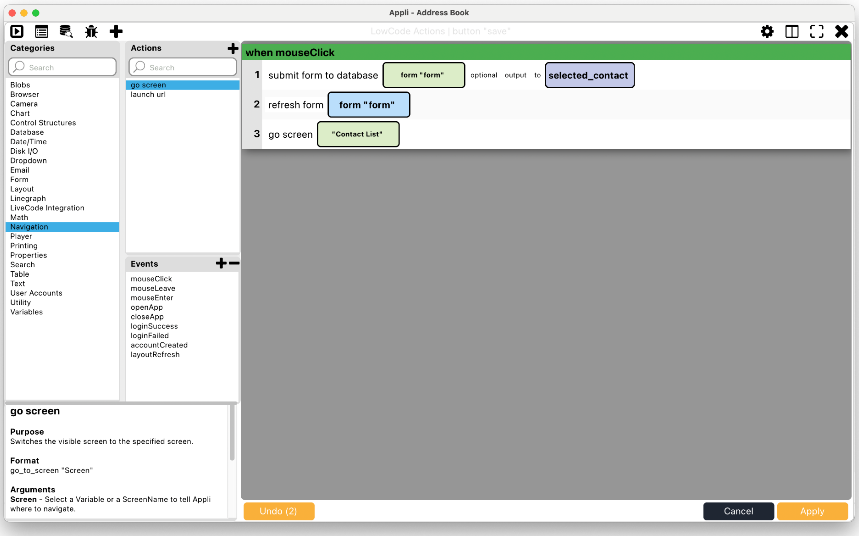
Using low-code, add a `mouseClick` to that plus button that resets the form and navigates to the *New Contact* screen.



```
when mouseClick
1 reset form form "form"
2 go screen "New Contact"
```

Figure 118: Low-code to go to the *New Contact* screen.

In the *New Contact* screen, add an extra action to the save button to go back to the *Contact List* screen. Add a low-code `mouseClick` to the *go back* button to navigate back to the *Contact List* as well.



when mouseClick

1 submit form to database form "form" optional output to selected_contact

2 refresh form form "form"

3 go screen "Contact List"

go screen

Purpose
Switches the visible screen to the specified screen.

Format
go_to_screen "Screen"

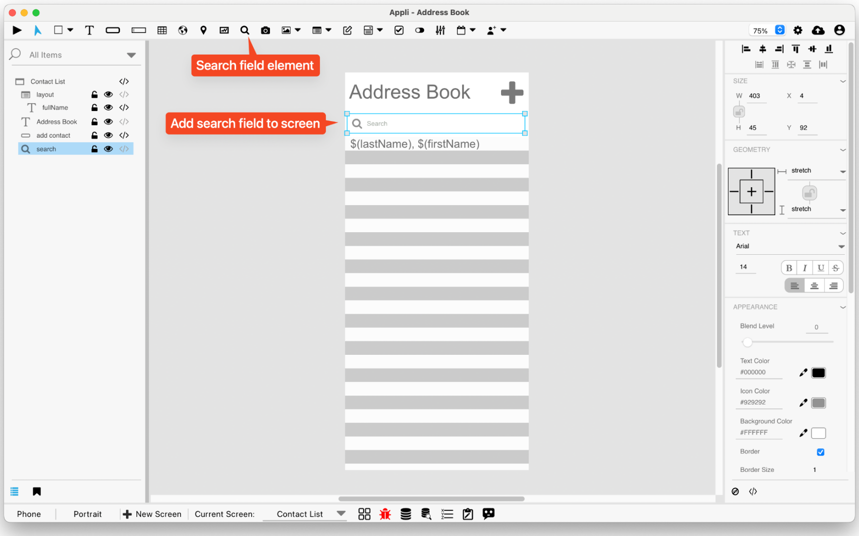
Arguments
Screen - Select a Variable or a ScreenName to tell Appli where to navigate.

Figure 119: Changes to the *save* button in the *New Contact* screen.

Go into *play mode*, use the plus button to go to the *New Contact* screen and add a new contact. It should navigate back to the *Contact List* screen after it saves your new contact and thanks to the `openScreen` action, the layout should show your recent contact there.

17.4 Searching contacts

No record list is perfect without a way of searching it. Appli helps us by having a *search field* element ready for use. Simply add it to the screen by placing it above the layout.



W 403 X 4
H 45 Y 92
stretch
stretch
Arial
14
Blend Level 0
Text Color #000000
Icon Color #929292
Background Color #FFFFFF
Border ☑
Border Size 1
Current Screen: Contact List

Figure 120: Added a search field element.

Search fields are configured using no-code. The no-code interface for a *search field element* is a *query builder*. A query is a way to ask a table to list only records that matches a certain criteria. We're going to use the query builder to ask for records that match our search terms to either the first or the last names.

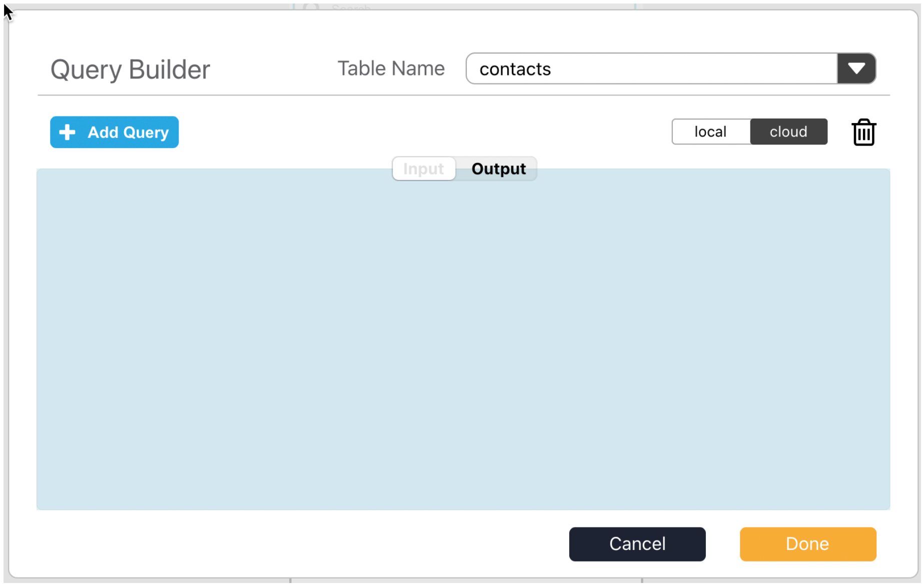


Table Name: contacts

Figure 121: Query builder for the search element.

On the query builder interface, select the *contacts* table and *cloud*.

Click *Add Query* to begin a new query. It will create a placeholder for an empty part of the query that looks like a blank rectangle. Click on it to display the controls.

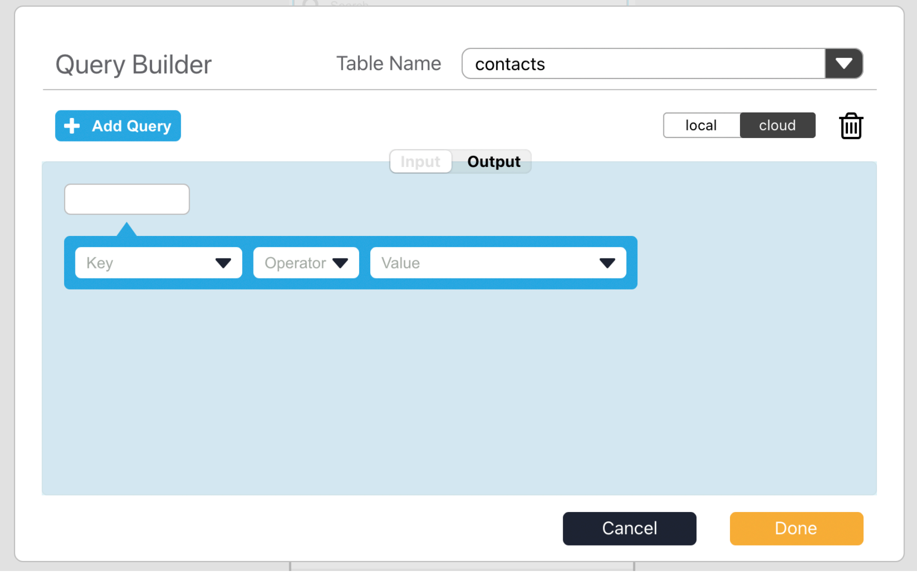


Figure 122: Building a query, opening the controls.

Queries are built by defining criteria. Expressions such as “last name should contain the search terms” and so on. To assemble that, we need to select what *table key* we want to examine, an operator (more about it in a bit), and an argument.

Select `lastName` as the key. For the operator, select `~`, that means *contains*, and for the value select *user input* which is the search terms.

It should look like this:

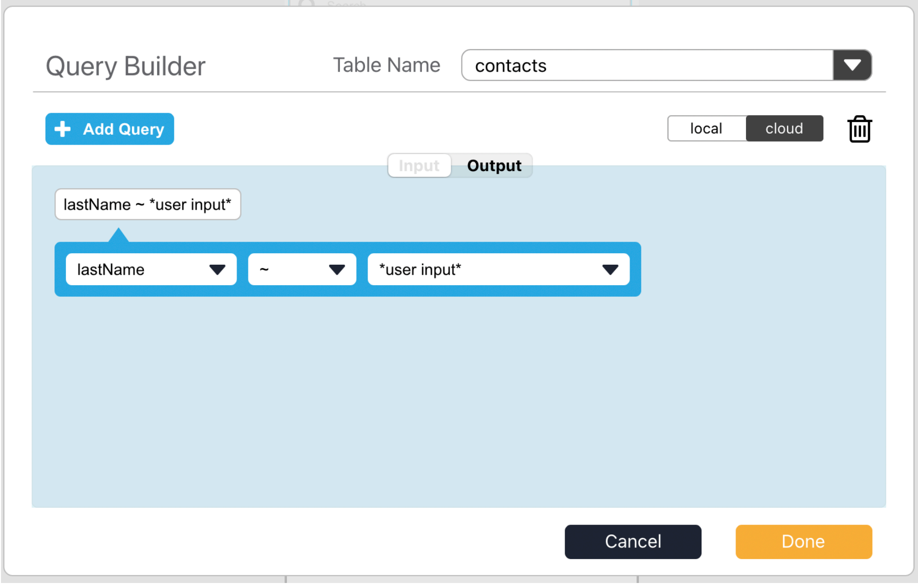


Table Name: contacts

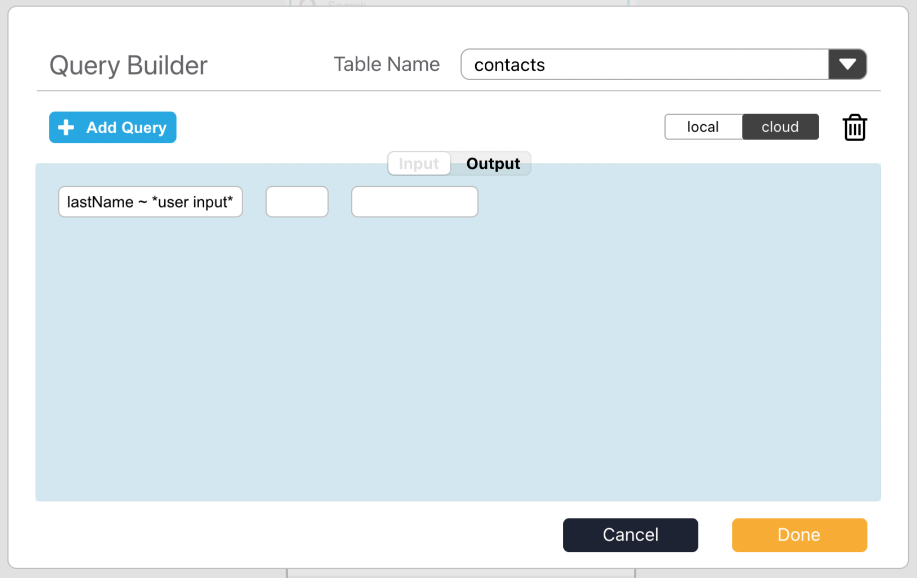
lastName ~ *user input*

lastName | ~ | *user input*

Figure 123: Building a query, opening the controls.

To learn more about each operator check the canelaDB documentation.

Click the *Add Query* again to add another criteria to our query. This time, it will create two square placeholders, one for the second criteria and one in between them to configure how they relate to one another.



contacts

lastName ~ *user input*

Figure 124: Building a query, opening the controls.

Select the `firstName` key and make it similar to the previous one by using the `~` contain operator and selecting *user input* as the value. We want values that match either `firstName` or `lastName`, so pick `OR` from the fields between the two queries.

The final query should look like the screenshot below.

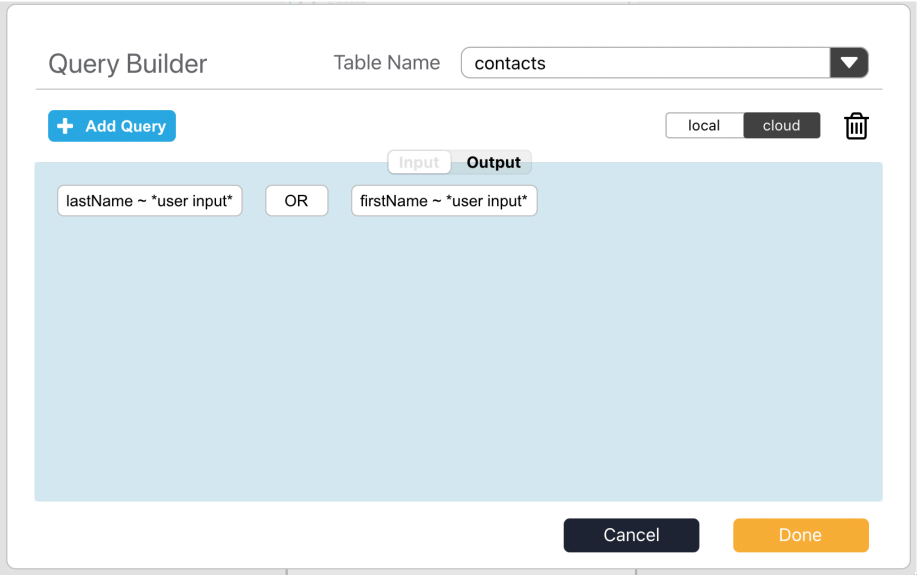

Query Builder

Table Name: contacts

lastName ~ *user input* OR firstName ~ *user input*

Figure 125: Query builder for the search element.

Switch to the *output* tab to select where to display the search results. Select the layout element we created earlier.

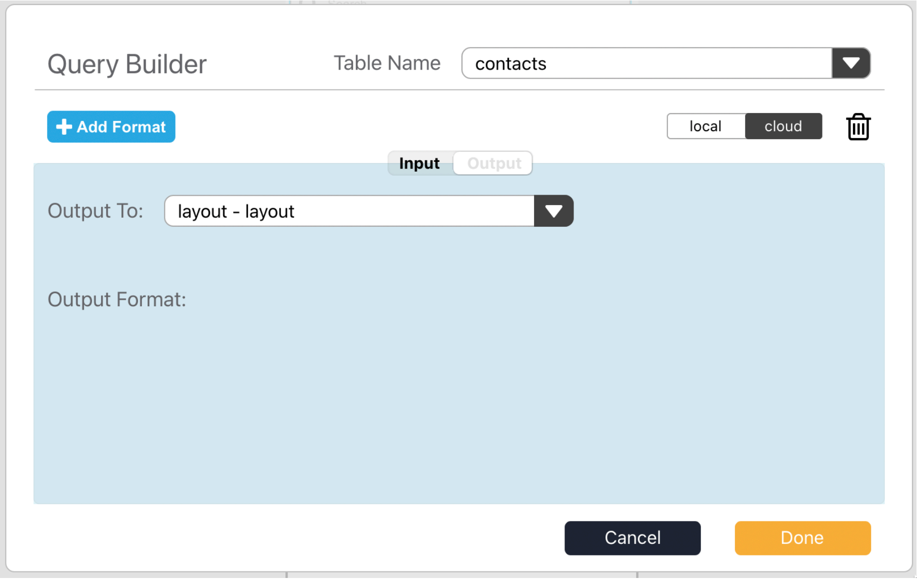

Query Builder

Table Name: contacts

Output To: layout - layout

Output Format:

Figure 126: Query builder for the search element.

Go into *play mode* and use the search element to filter the list of contacts.

17.5 Editing a contact

Our final feature for this tutorial is viewing and editing a contact. We're going to combine both features into a single screen. Create a new screen called *Edit Contact* and lay it out just like the *New Contact* screen with a form and fields for each table key plus two buttons — one for going back and another one for saving — it should look exactly like *New Contact*.

Then make the phone and email fields shorter and add a button next to each of them. Those buttons will allow us to send email and start a phone call.

Your screen should look like this:

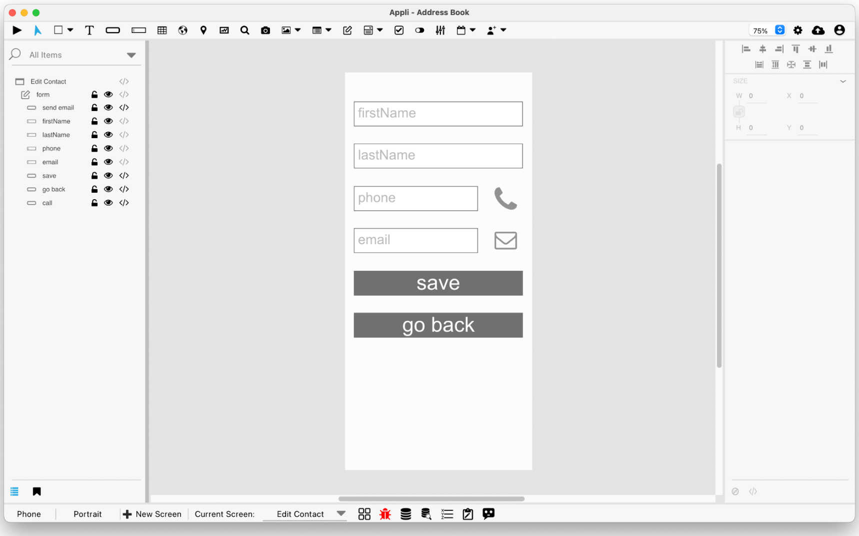


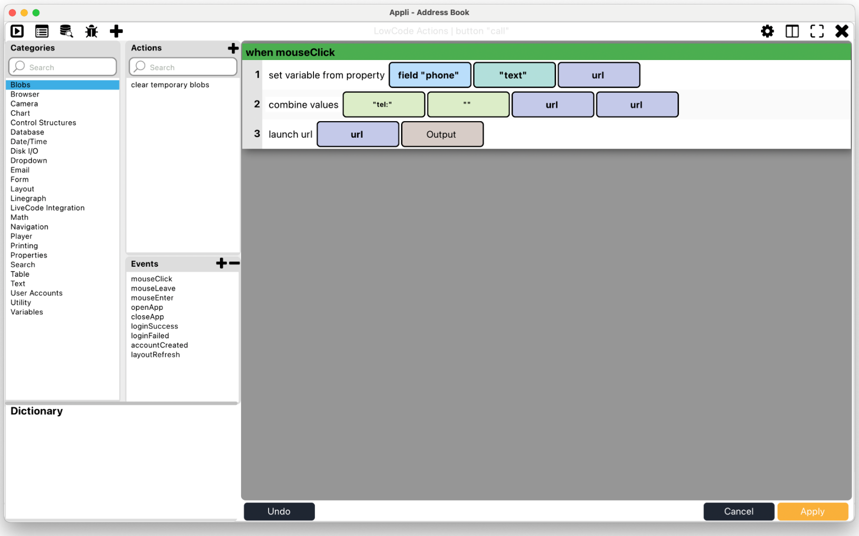
Figure 127: Contact editing screen

Bind all the elements in the form to the correct key using the form's no-code editor. The code for the *Save* button and the *Go Back* button is the same as the one you used for the *New Contact* form.

17.5.1 Making a call

On a mobile phone, you can start a call by launching an url that begins with `tel:` and contains the phone you want to call. So if you want to call *555-323-444*, you can launch the URL `tel:555323444` to open the dialer with the phone number filled in.

To do this in Appli, we need to pick the value from the *phone field element* and create a variable with `tel:` before the number, then launch a URL based on that variable. The script for the *Call button* is:



when mouseClick

1 set variable from property field "phone" "text" url

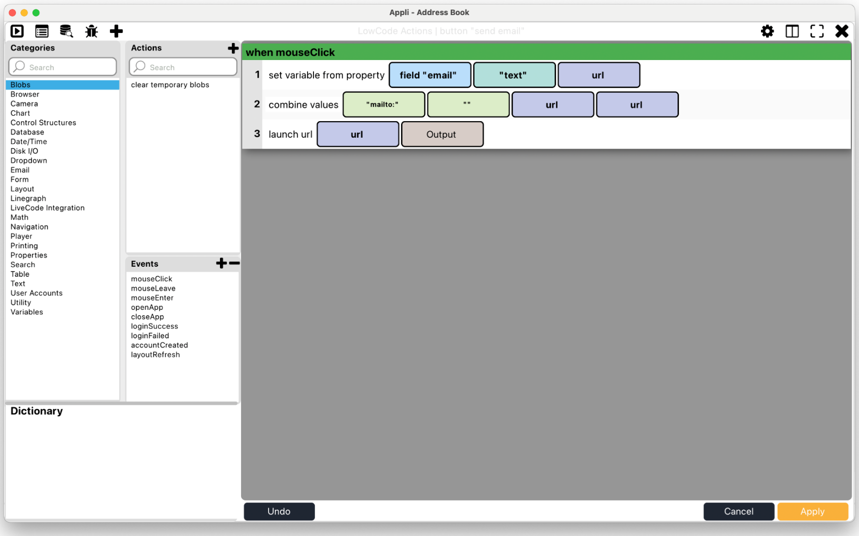
2 combine values "tel:" "" url url

3 launch url url Output

Figure 128: Low-code actions for the *call* button

17.5.2 Sending email

The code for the *email button* is very similar, but email URLs start with `mailto:` instead of `tel:`.



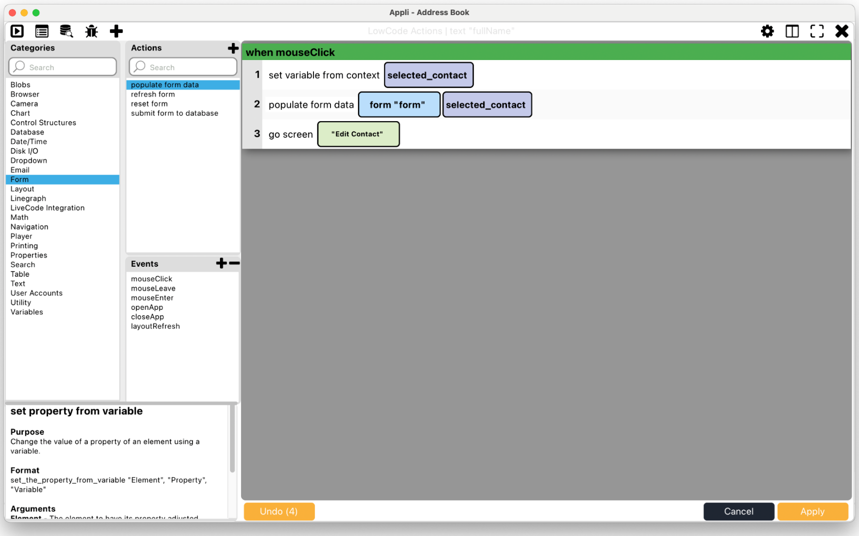
```
when mouseClick
1 set variable from property   field "email"   "text"   url
2 combine values   "mailto:"   ""   url   url
3 launch url   url   Output
```

Figure 129: Low-code for the *send email* button

> Be aware that Appli has a powerful *send email* action. You could craft a new screen with the fields needed and send email from inside your app without the need to open a third-party app like this example.

With those two buttons in place, we need to go back to the *Contact List* screen. We need to connect the clicked contact in the layout with what needs to be edited on the *Edit Contact* screen.

Select the *fullName* text element in the layout and add the following low-code:



when mouseClick

1 set variable from context selected_contact
2 populate form data form "form" selected_contact
3 go screen "Edit Contact"

set property from variable

Purpose
Change the value of a property of an element using a variable.

Format
set_the_property_from_variable "Element", "Property", "Variable"

Arguments
Element - The element to have its property adjusted

Figure 130: Low-code for *fullName* text element

There are some important concepts hidden in that simple script. First is that a *form* can have its `RecordID` property set to a specific record. When that is set, the record data is displayed on the form and saving the form updates the record in the database. Second is that a *layout* has a context which is the current clicked record. By setting a variable called `selected_contact` using the value from that form *context*, we're essentially placing the clicked contact `RecordID` into the variable, we can then use *populate form data* to set the `RecordID` property of the form in the *Edit Contact* screen to that value and load that record into it.

17.6 Next steps

Now that you're familiar with the basics of database-based workflow, take your time to explore Appli's built in help in the low-code editor and check out the other powerful actions in the database category.

18 Tutorial: Understanding Layouts

In the database tutorial, we didn't explore layouts much, even though we made use of one to list records. Layouts are a flexible container that can be used to:

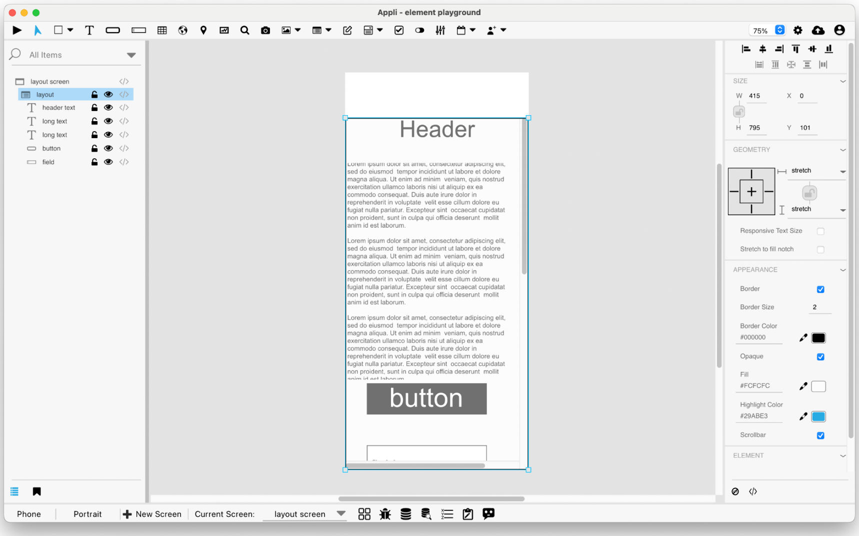
- Create scrollable regions in your app.
- List a single record.
- List multiple records.

Mastering layouts is an essential skill to be an effective developer, especially if you are working on mobile layouts. Building a complex UI on a small screen becomes much easier once you factor layouts into your app.

Using a layout is the most common way to display database records. While using them to create larger than screen sections is good, understanding how to display records with them is the most important aspect of this chapter.

18.1 Layouts are containers

A Layout is a container for other elements. After placing it on the playground, one should place elements inside it.



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

W 415 X 0
H 795 Y 101
stretch
stretch
Responsive Text Size □
Stretch to fill notch □
Border ☑
Border Size 2
Border Color #000000
Opaque ☑
Fill #FCFCFC
Highlight Color #29ABE3
Scrollbar ☑

Figure 131: Sample layout

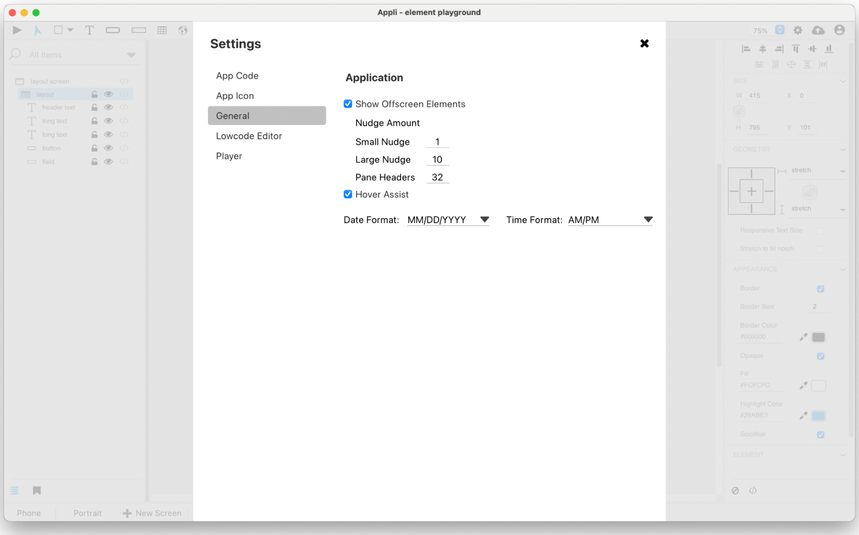
Above you can see a simple layout with text elements inside it. Notice that the layout is selected in the playground. You can see the blue rectangle showing the layout boundaries and the elements contained inside the form are shown under it in the project browser.

18.2 Adding elements to a layout

To add an element to a layout, either draw the element inside the layout or move it into the layout boundaries. You can also drag an element in the project browser and place it under the layout.

18.3 Creating a scrollable region

Temporarily resize the layout to span the entire region you want to scroll. Appli IDE needs to be configured to *show offscreen elements* or you won't be able to see the parts of the layout that are offscreen. You can toggle the *show offscreen elements* preference by using the `o` keyboard shortcut.



Application

☑ Show Offscreen Elements

Nudge Amount

Small Nudge 1

Large Nudge 10

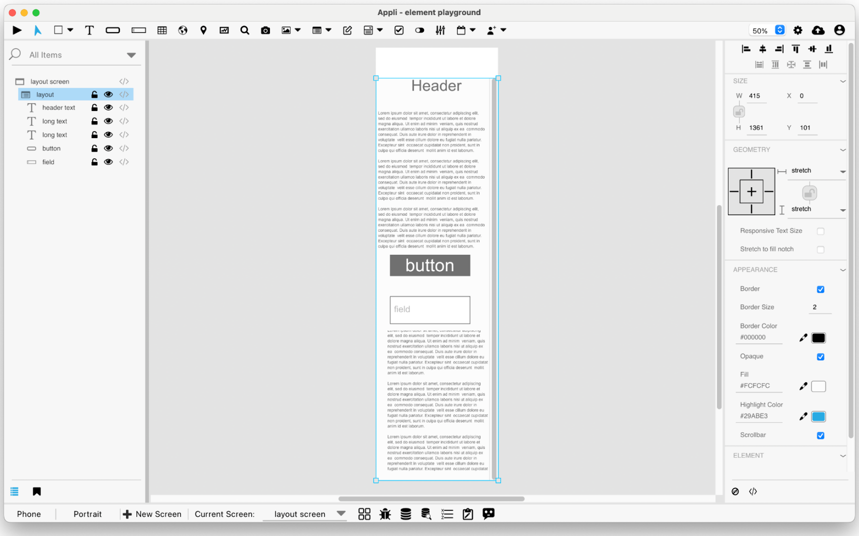
Pane Headers 32

☑ Hover Assist

Date Format: MM/DD/YYYY   Time Format: AM/PM

Figure 132: Set show offscreen elements to true

Add the elements inside the layout as if the screen was wider or taller than it is.



W 415 X 0
H 1361 Y 101
stretch
stretch
Responsive Text Size ☐
Stretch to fill notch ☐
Border ☑
Border Size 2
Border Color #000000
Opaque ☑
Fill #FCFCFC
Highlight Color #29ABE3
Scrollbar ☑
Current Screen: layout screen

Figure 133: Sample layout with tall layout

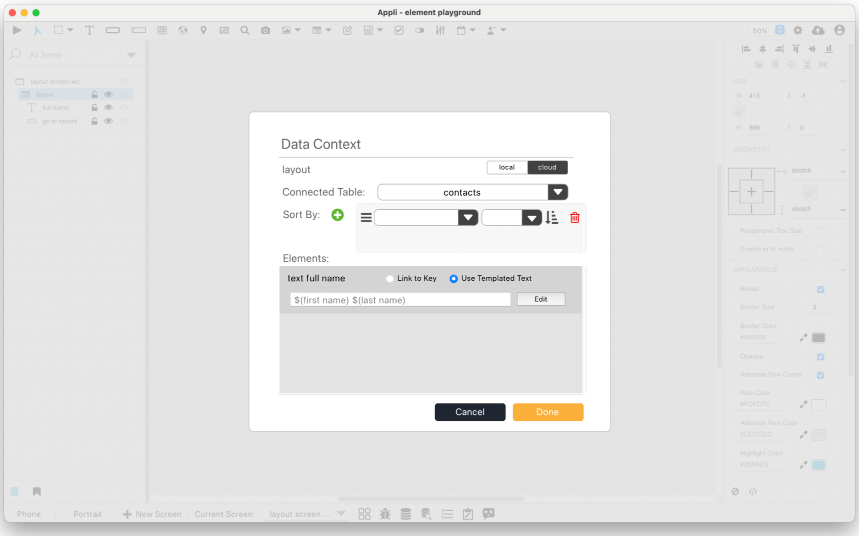
Once you're happy with it, resize the layout back to the size you need. Since the layout has elements placed outside its final rectangle, the user will be able to scroll it and access those elements.

> A key UX challenge is making sure the user knows that they can scroll that region to access more elements. An easy to remember guideline is trying to make the layout the root element of the screen. This way, the whole screen scrolls. That is easier to discover than some small scrolling region inside a larger screen. Remember to place any header or navigation bar outside the layout or you may end up scrolling them out of view.

Remember to set the `scrollbar` property of the layout to true. You'll need to take the scrollbar width into account when designing your screen. If you make an element span the whole width of the layout and then add a scrollbar to it, Appli will also add a horizontal scrollbar to the user interface since some pixels of that element are being clipped by the vertical scrollbar.

18.4 Layouts are database aware

Besides being able to create scrollable regions, layouts can be linked to a database table (just like forms). Use the *no-code interface* to configure a layout and link it to a table.



Data Context

layout ○ local ● cloud

Connected Table: contacts

Sort By:

Elements:

text full name ○ Link to Key ● Use Templated Text

$(first name) $(last name)

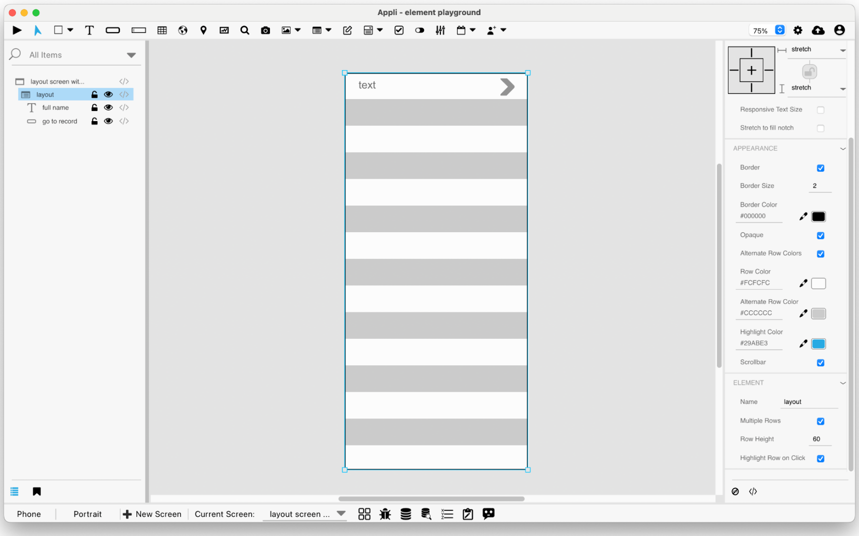
Figure 134: Configuring data context for a layout

Select the table you want to link to the layout and its location. Once that is done, you can further configure the layout by adding elements to it and binding them to keys in the database.

18.5 Showing multiple records

The most common use of a database-aware layout is to show a list of records. After linking the layout to a table, it is time to add elements to it.

The objective is to create a *template for the rows of records*. What you create is a single row representing one record. Appli will repeat that row to build a list of records.



stretch
stretch
Responsive Text Size ☐
Stretch to fill notch ☐

APPEARANCE
Border ☑
Border Size 2
Border Color #000000
Opaque ☑
Alternate Row Colors ☑
Row Color #FCFCFC
Alternate Row Color #CCCCCC
Highlight Color #29ABE3
Scrollbar ☑

ELEMENT
Name layout
Multiple Rows ☑
Row Height 60
Highlight Row on Click ☑

Figure 135: A row template

You need to set the `Multiple Rows` property of the layout to true to list multiple records. Once you do, Appli will automatically set it to alternate row colors. New properties related to the *row height* and *row colors* will be available. Configure them to match your UI.

18.6 Working with template text

Template text is a flexible way of configuring how a *text element* that belongs to a layout will show its data. The *no-code interface* for the layout will display the elements inside it.

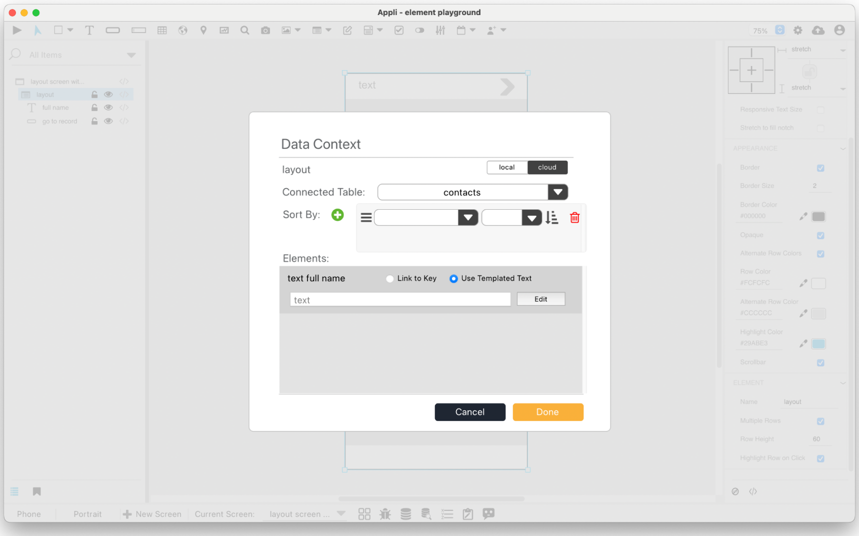


Figure 136: Configuring elements

Selecting *template text* and clicking *edit* opens the template text editor.

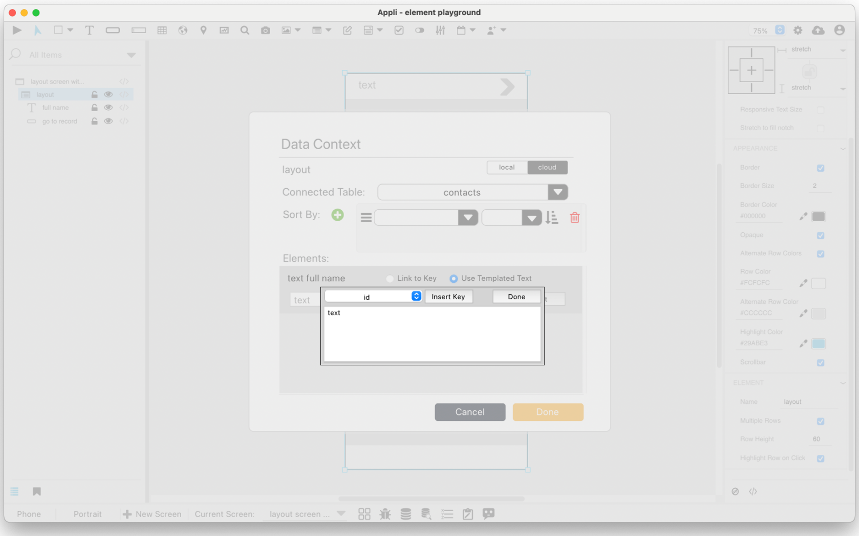
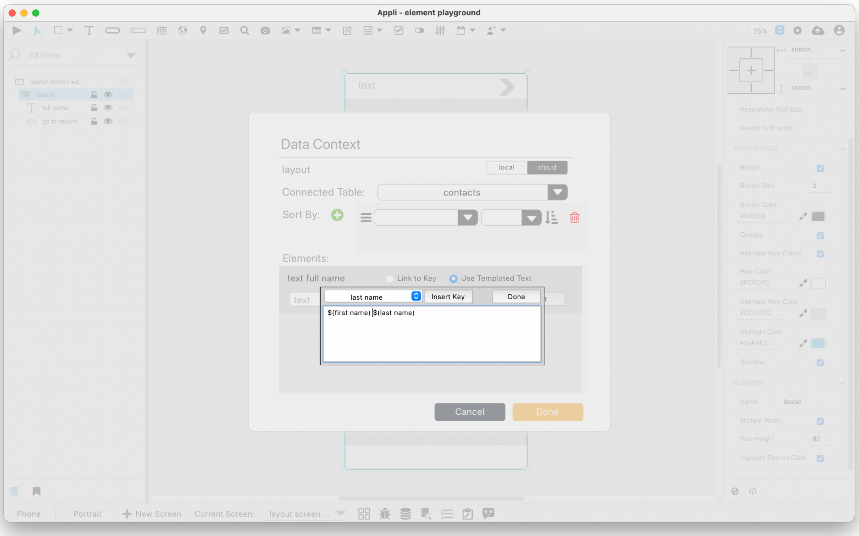


Figure 137: Template text editor



Data Context

layout ○ local ● cloud

Connected Table: contacts

Sort By:

Elements:

text full name ○ Link to Key ● Use Templated Text

last name | Insert Key | Done

$(first name) $(last name)

Figure 138: Template text editor showing configured text element

When you run the app, it will repeat itself, listing the records and the template text will be replaced with the values from the table.

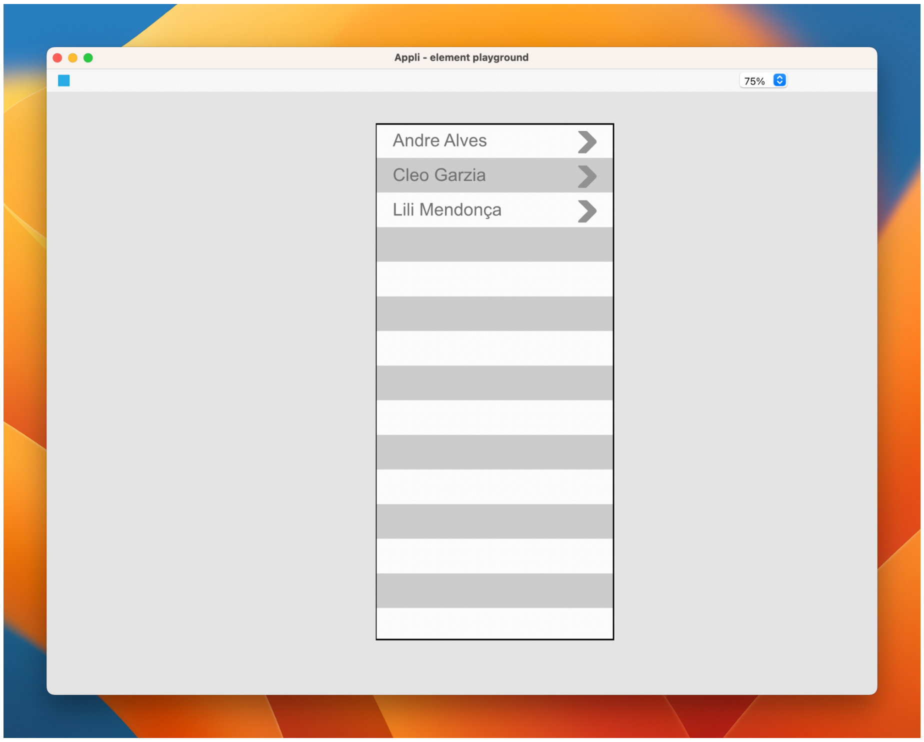


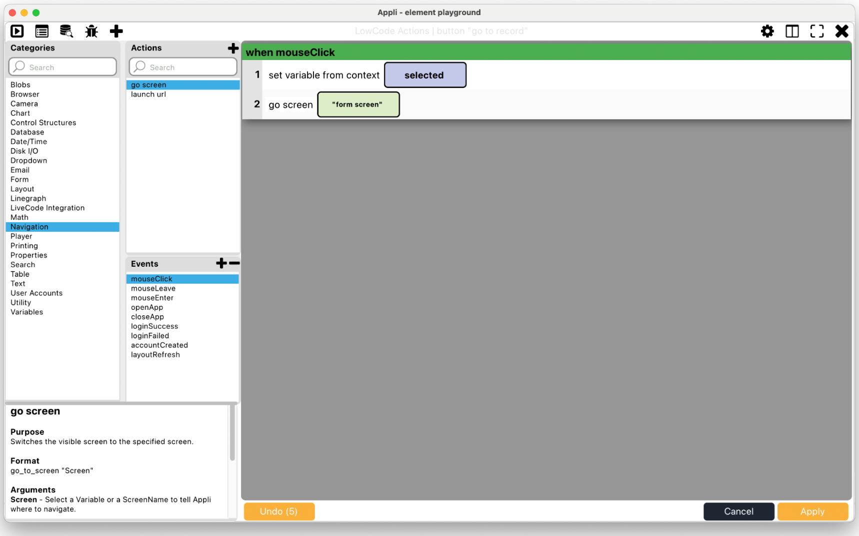
Figure 139: A list of records

18.7 Selecting a record

Use the `Set Variable From Context` action to set a variable to the `RecordID` of the selected record. Use the `mouseClick` event of some element in your row template to call that action. You can use that variable later to populate a form or some other workflow.

Consider the *row template* we've shown above with a field and a chevron pointing to the right. The user would click that button to select a contact.

The best way to select a record is to create a variable that points to the selected record, such as shown in this *low-code* script for the chevron button:



when mouseClick

1 set variable from context selected

2 go screen "form screen"

go screen

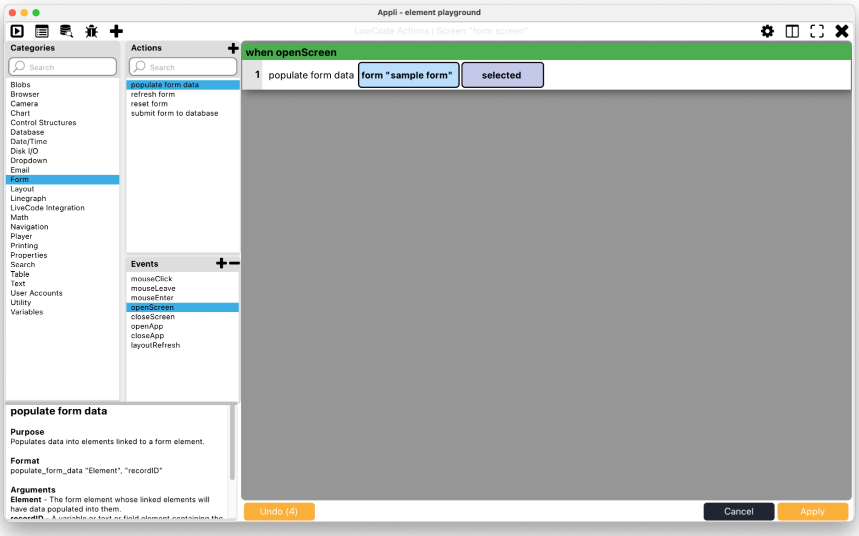
Purpose
Switches the visible screen to the specified screen.

Format
go_to_screen "Screen"

Arguments
Screen - Select a Variable or a ScreenName to tell Appli where to navigate.

Figure 140: Selecting a record

You can then use the variable you set on another screen to populate a form or some other action. Here is an example with a form. This script goes into the screen script:



when openScreen

1 populate form data form "sample form" selected

populate form data

Purpose
Populates data into elements linked to a form element.

Format
populate_form_data "Element", "recordID"

Arguments
Element - The form element whose linked elements will have data populated into them.
recordID - A variable or text or field element containing the

Figure 141: Using a variable with a record to populate a form

18.8 Showing a single record

That is usually the responsibility of forms, but it can be done with a layout as well. Just set the `RecordID` property of the layout to the record you want.

19 Tutorial: Understanding Forms

In the database tutorial, we used a form but didn't dive deeply into them. Forms are a crucial part of any database-enabled app and having a full understanding of them will make development much easier.

While layouts are often used to show multiple records, forms are used to edit a single record or insert a new record.

19.1 Forms are containers

A Form is a container for other elements. After placing it on the playground, one should place elements inside it.

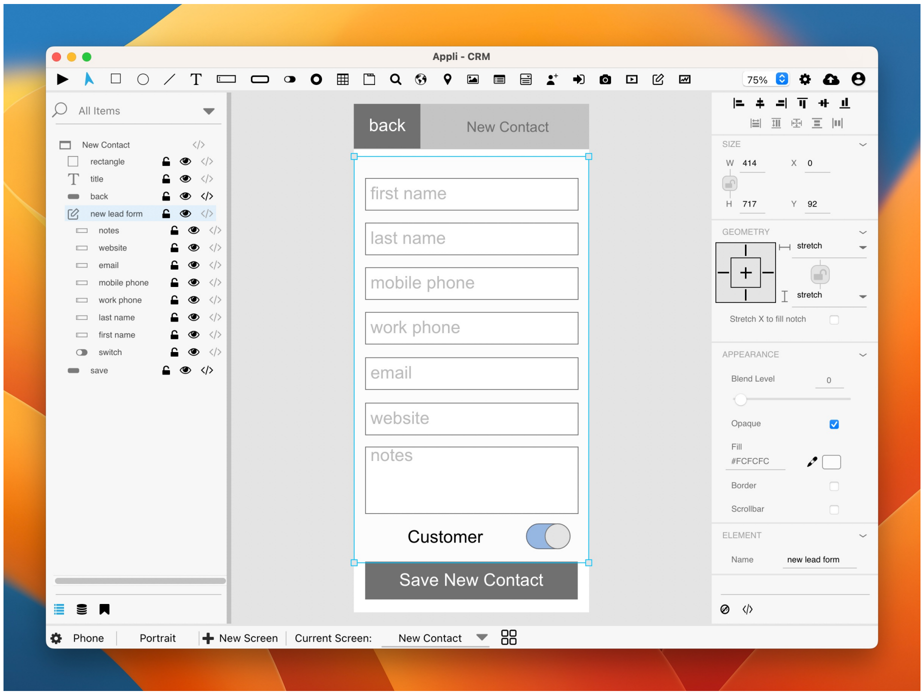


Figure 142: Sample form

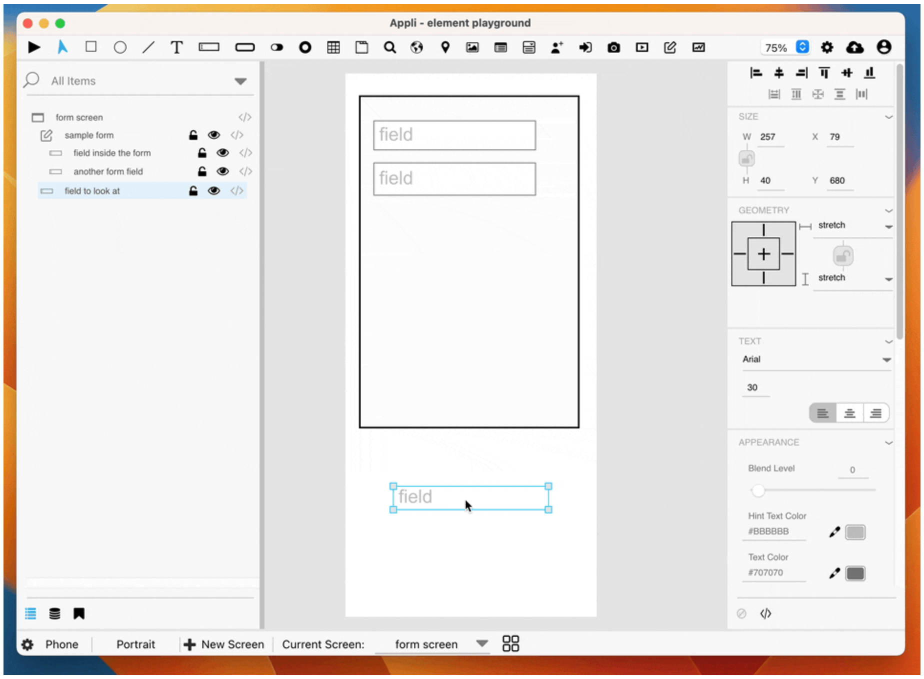
Above you can see a simple form with fields and a switch element inside it. Notice that the form is selected in the playground. You can see the blue rectangle showing the form boundaries and in the project browser the elements contained inside the form are shown under it.

19.2 Adding elements to a form

To add an element to a form, either draw the element inside the form or move it into the form boundaries. You can also drag an element in the project browser and place it under the form.

Forms can use the following elements as inputs: fields, dropdowns, switches, and radios.

19.2.1 Invisible elements inside forms



W 257
X 79
H 40
Y 680
stretch
stretch
Arial
30
Blend Level 0
Hint Text Color #BBBBBB
Text Color #707070

Figure 143: Invisible field inside a form

Dragging a field in the project browser into a form will make that field invisible if it is outside the boundaries of the form.

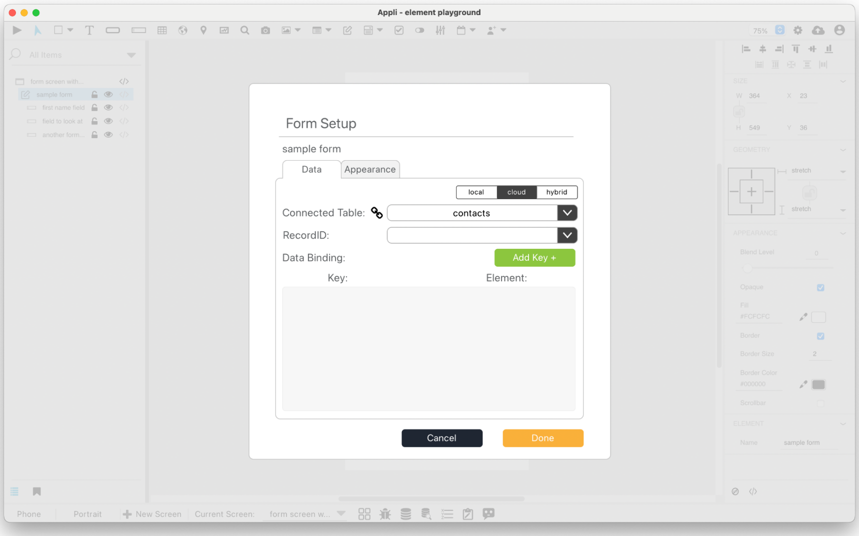
Having an invisible field is quite useful as it can be used as a hidden data container for storing values for internal use without cluttering the UI the user sees.

Just be careful that if you move that field in the playground, it might exit the form if its final place is outside the form rectangle. Remember to use the project browser to place it inside the form again if you moved it.

> If you want to have an invisible element in a form and don't want to move it outside the form's boundaries, you can use the *eye* icon on the *project browser* to toggle the element visibility.

19.3 Forms are database aware

Like layouts, a form is a database-aware container. Use the *no-code interface* to configure it and bind it to a database table.



Form Setup

sample form

Data | Appearance

Connected Table: contacts

RecordID:

Data Binding:

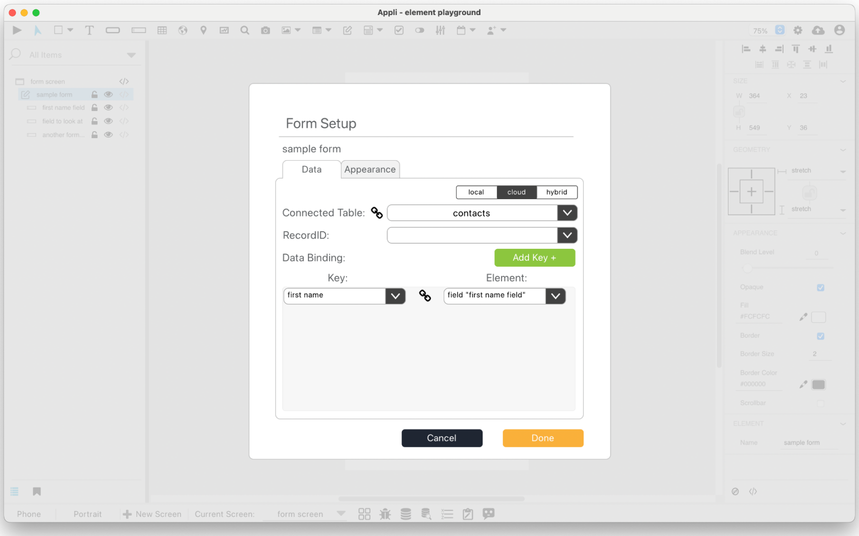
Key: Element:

Figure 144: Configuring a form

Select the context for the database (local or cloud) and the table you want to connect to that form.

19.4 Binding elements to data

Once a form is connected to a table, the elements inside it can be linked to keys in that table. In the forms *no-code interface*, click the *Add Key +* button to create a new element to key relationship.



Form Setup

sample form

Connected Table: contacts

RecordID:

Data Binding:

Key:	Element:
first name	field "first name field"

Figure 145: Configuring element to key relationship

19.5 Inserting new records

If a form is not showing an existing record, then it can insert a new record into the table. The bound elements will provide the values for the new record. Any key in the table that has no corresponding element will be empty.

There is a whole category of actions to handle forms. To insert a new record, use *Submit Form To The Database* action. That action is normally placed inside a button labeled *Save* or something similar.

19.6 Editing records

Forms are normally used to edit records. Use the *populate form data* to load a record into the form. You'll need to pass a form element and a record as arguments for that action. The record usually comes from a layout with a list of records, but it can also come from the database itself if you store them in a record.

You can also use any of the *set property* actions to set the RecordID property of the form to the selected record, but *populate form data* is easier.

> *populate form data* will set the `RecordID` property of the form and load the data into it.

Once the `RecordID` is set, submitting the form to the database will update that record instead of saving a new record. If you want to insert a record instead of updating one, you'll need to clear the `RecordID` property first.

Usually, selecting a record and editing it are actions that happen on different screens. A common workflow is to use *set variable from context* in a screen with a layout followed by *go to screen* to go to the screen with the form. On the `openScreen` event for the screen itself, a call to *populate form data* uses the variable set in the previous event to load data into the form.

> If a form has its `RecordID` set, you'll be able to see it in the *no-code* interface.

20 Tutorial: User Management

Some apps may need multi-user workflows, fortunately Appli provides a robust set of elements and actions that enables the developer to build their own user management system suitable for the needs of their application.

Each project comes with a table to hold user accounts called `cdbUsers`. To access that table open the *data modeler* and you'll see the table that begins with `cdbUser`. Appli automatically adds a unique identifier suffix to that table, so each version of `cdbUser` has a slightly different name.

> **Note:** You probably won't need to deal with that table manually, but it is good to know how to access it. The main reason to open this table is for debugging user accounts while in development.

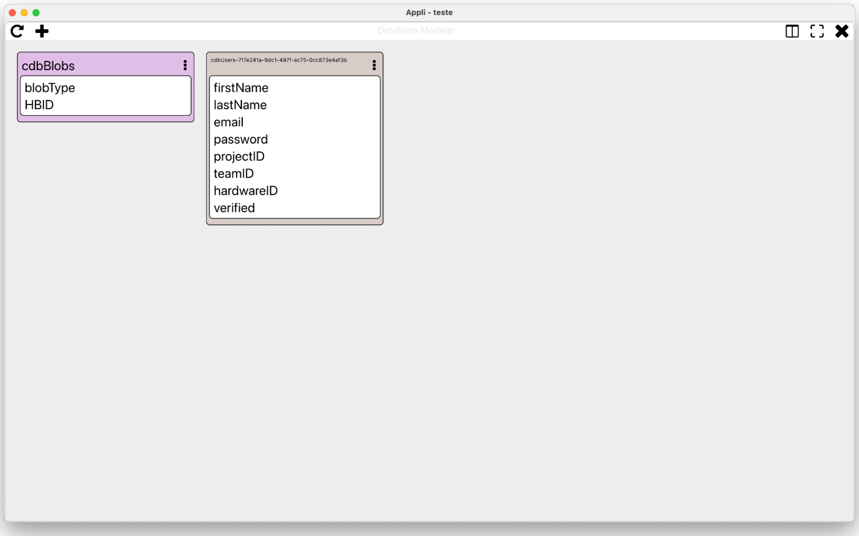


Figure 146: How to view the `cdbUser` table

20.1 Creating users

There are two ways to create a user, one is using the ready-made `Create Account` element, and the other is rolling your own form and using the `Create Account` action.

When a user is created, a record for that user is added to the `cdbUsers` table.

20.2 Login and logout

Use either the `Login` element or `Get Auth`, and `Logout` actions to manage which user is logged into the application.

20.3 When to roll your own user management

If your application will handle many users, it might be best to leverage the actions and elements outlined above to roll your own user management workflow. You can even create your own user permissions table and roll different user levels. Appli does not assume a workflow for you, but gives you the tools to roll something bespoke to your needs.

Even if your application is going to be used by multiple users, you might not need to use user management. Such control over user workflows matters only when you need to keep track of which user is doing the actions in your application.

For example, a field data-collection app used by volunteers in a community project might not need to know which data is coming from which volunteers and completely ignore all the user management features.

On the other hand, a company-wide CRM will surely need complex user management features.

21 Tutorial: LiveCode Integration

Even though we are updating Appli collection of low code actions, there are times when you might need features we haven't yet covered. In these cases, Appli allows you to create *custom code* using LiveCode language.

The *custom code* action allows you to write LiveCode scripts that interact with Appli's variables and elements. You can also access tables using LiveCloud API which is the database technology behind Appli.

In this brief tutorial, we're going to build a small application to display cool facts from history. We'll use a LiveCode based script to access a Web API that provides us with facts and then use Appli to present those facts to the user. By the end of this tutorial, you'll know how to leverage LiveCode integration and how to pass data back and forth between Appli and LiveCode.

> The same sample app can be built using Appli without relying on LiveCode integration by using the `rest api call` action. I'll show how to replace the LiveCode integration version with the pure Appli version at the end.

For this tutorial we'll access a Web API called *On this day* which is provided by Wikipedia. This API received a *day* and a *month* and returns cool historical facts that happened on the same day and month.

You can see it right now on your web browser. Just navigate to:

`https://api.wikimedia.org/feed/v1/wikipedia/en/onthisday/all/04/03` and you'll see a JSON response that contains historical events that happened on the 3rd of April. The last two numbers there are month and day.

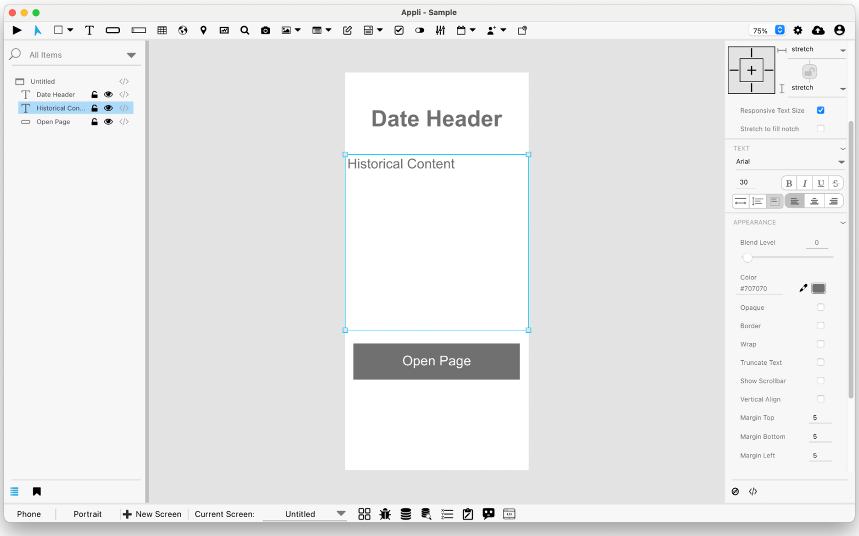
JSON might seem strange if you never seen it before, but it is an easy format for computers to understand. Our sample app will pick the current day and month, make a request to that API, and then show the results to the user.

The *On this day* API returns a collection of items divided into three categories: selected, births, and deaths. So for any combination of day and month, it will tell us selected historical events, notable births and deaths as well.

Our sample app will only show the selected events. We'll build it in a way that it shows the first historical event in the returned collection (the API might return more than one event).

21.1 Building the user interface

We'll use two fields, one for the data and another one for the historical fact. We'll also add a button to open the Wikipedia page with the relevant entry on the user's default browser. This is how our user interface looks:



stretch
stretch
Responsive Text Size ☑
Stretch to fill notch □
TEXT
Arial
30
APPEARANCE
Blend Level 0
Color #707070
Opaque □
Border □
Wrap □
Truncate Text □
Show Scrollbar □
Vertical Align □
Margin Top 5
Margin Bottom 5
Margin Left 5

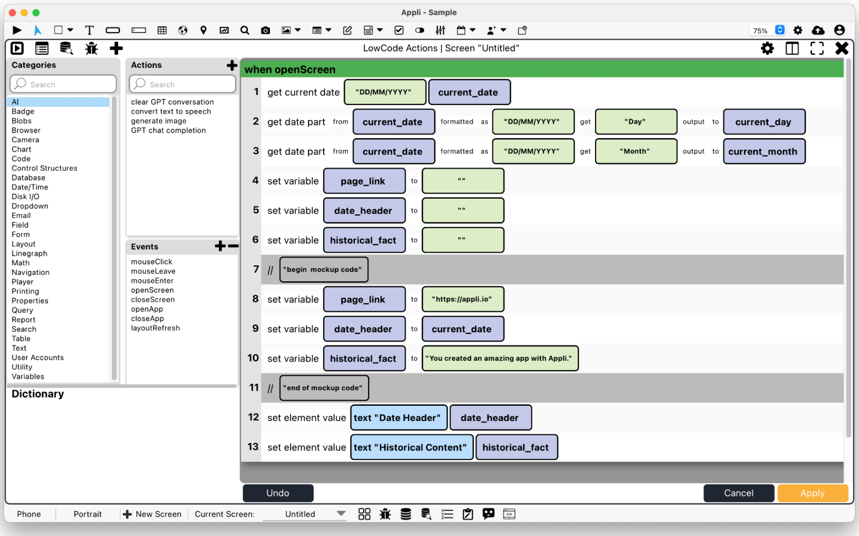
Figure 147: User Interface

- A text element called *Date Header*.
- A text element called *Historical Content*.
- A button element called *Open Page*.

21.2 Initial setup

Before we work with LiveCode custom code, we'll create a mock version of our app just to illustrate how it should work. Later, we'll replace the mock part of it with the actual request.

The objective is to display a historical fact when the application launches, so our low code actions will go on the `OpenScreen` event for the screen. That event is triggered as soon as the screen is displayed.



```
when openScreen
get current date "DD/MM/YYYY" current_date
get date part from current_date formatted as "DD/MM/YYYY" get "Day" output to current_day
get date part from current_date formatted as "DD/MM/YYYY" get "Month" output to current_month
set variable page_link to ""
set variable date_header to ""
set variable historical_fact to ""
// "begin mockup code"
set variable page_link to "https://appli.io"
set variable date_header to current_date
set variable historical_fact to "You created an amazing app with Appli."
// "end of mockup code"
set element value text "Date Header" date_header
set element value text "Historical Content" historical_fact
```

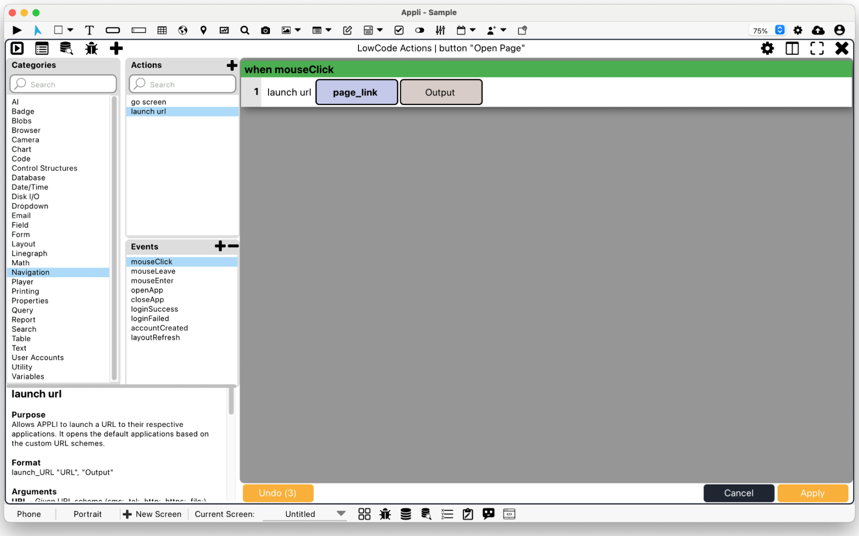
Figure 148: Script

Our script picks the current date and month and set two variables: `current_day` and `current_month`. Then it sets the `page_link`, `historical_fact`, and `date_header` variables to empty. These variables are going to be used in the custom code script to return data back to Appli.

Our mock (which is just jargon for fake) code comes afterwards and fills the variables `page_link` `date_header` and `historical_fact` with hardcoded values.

After that we use those variables to insert data into the elements in the user interface.

The button action just launches the browser pointed at the content of `page_link`.



when mouseClick

```
launch url  page_link  Output
```

launch url

Purpose
Allows APPLI to launch a URL to their respective applications. It opens the default applications based on the custom URL schemes.

Format
launch_URL "URL", "Output"

Arguments
URL - Given URL scheme (sms:, tel:, http:, https:, file:)

Figure 149: Button script

Switching to play mode, we can see our app working even if the mocked data. Time to make the actual request work and make our app real.

21.3 Using the custom code editor

The custom code editor lives in the footer. It is the icon with the clipboard and angle brackets. Click it to open the interface.

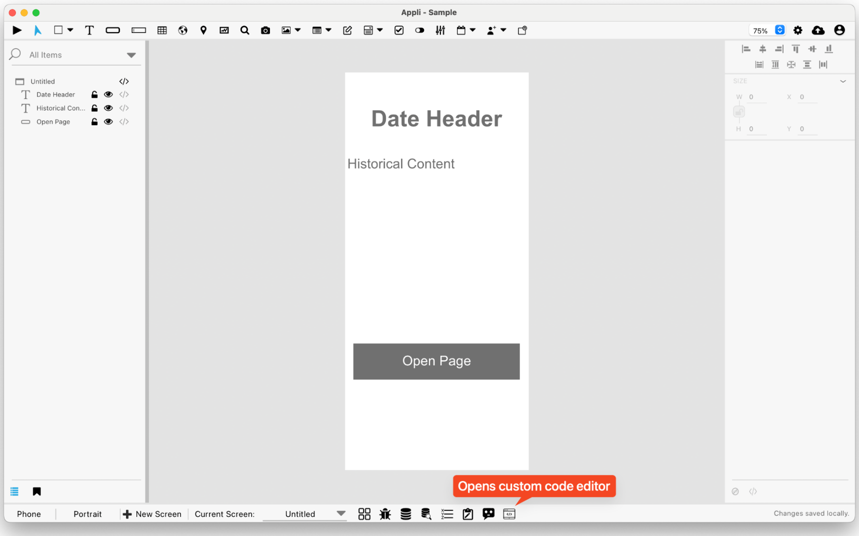


Figure 150: Button to launch the custom code editor

You can have as many custom code scripts as you want. You can give them names and then from Appli you can trigger them by the name you set.

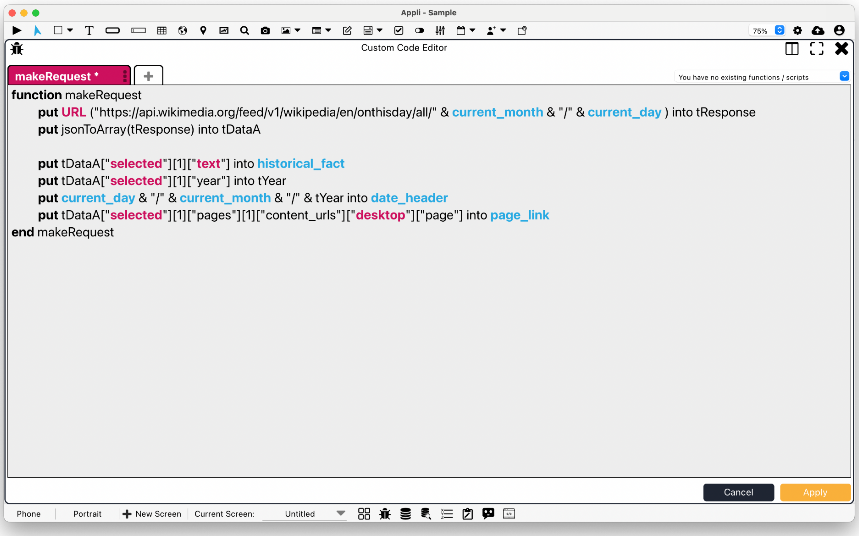
Appli will execute the first function found in the custom code script. You can have more than one handler there, but the first one is the one that is executed.

There will be only one function in our script and we'll call it `makeRequest`. We'll use the same name for our custom code. Click the plus icon on the tab bar to add a new script. Name it `makeRequest`. To rename the custom code, double click on the name on its tab.

This is our LiveCode script:

```
function makeRequest
  put URL ("https://api.wikimedia.org/feed/v1/wikipedia/en/onthisday/all/" & current_month &
  put jsonToArray(tResponse) into tDataA

  put tDataA["selected"][1]["text"] into historical_fact
  put tDataA["selected"][1]["year"] into tYear
  put current_day & "/" & current_month & "/" & tYear into date_header
  put tDataA["selected"][1]["pages"][1]["content_urls"]["desktop"]["page"] into page_link
end makeRequest
```

> Explaining LiveCode code is beyond the scope of this documentation, but you can read more about it on their own user guide.



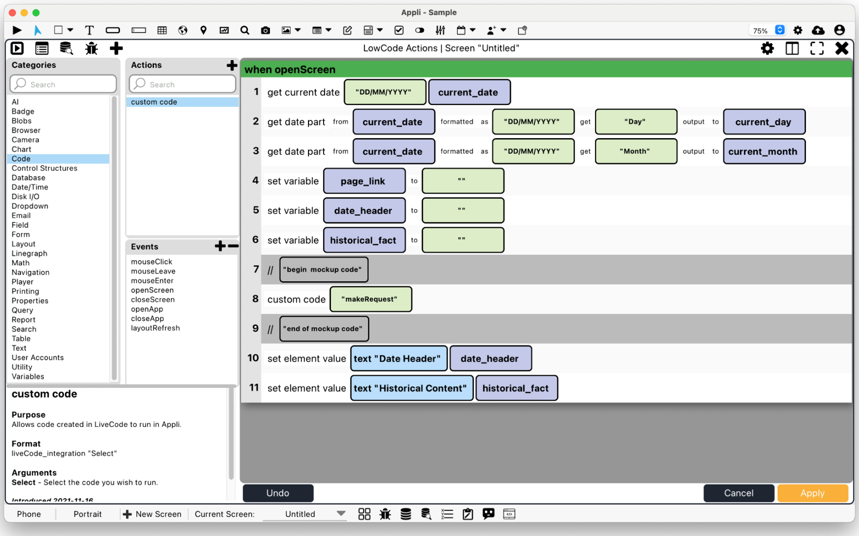
```
function makeRequest
    put URL ("https://api.wikimedia.org/feed/v1/wikipedia/en/onthisday/all/" & current_month & "/" & current_day ) into tResponse
    put jsonToArray(tResponse) into tDataA

    put tDataA["selected"][1]["text"] into historical_fact
    put tDataA["selected"][1]["year"] into tYear
    put current_day & "/" & current_month & "/" & tYear into date_header
    put tDataA["selected"][1]["pages"][1]["content_urls"]["desktop"]["page"] into page_link
end makeRequest
```

Figure 151: Script as seen in Appli

Those among you that know both LiveCode and Appli might be looking funny at the script. Yes, you're seeing it right, you can access and use Appli's variables as if they were LiveCode variables. Our backend system takes care of it. You can read from them and write to them and Appli will make sure it works.

21.4 Calling the custom code from Appli

So we can replace all that mock section in `OpenScreen` with a call to `code` and configure the parameter to the name of our function `makeRequest`.



```
when openScreen
1  get current date "DD/MM/YYYY" current_date
2  get date part from current_date formatted as "DD/MM/YYYY" get "Day" output to current_day
3  get date part from current_date formatted as "DD/MM/YYYY" get "Month" output to current_month
4  set variable page_link to ""
5  set variable date_header to ""
6  set variable historical_fact to ""
7  // "begin  mockup code"
8  custom code "makeRequest"
9  // "end of mockup code"
10 set element value text "Date Header" date_header
11 set element value text "Historical Content" historical_fact
```

custom code

Purpose
Allows code created in LiveCode to run in Appli.

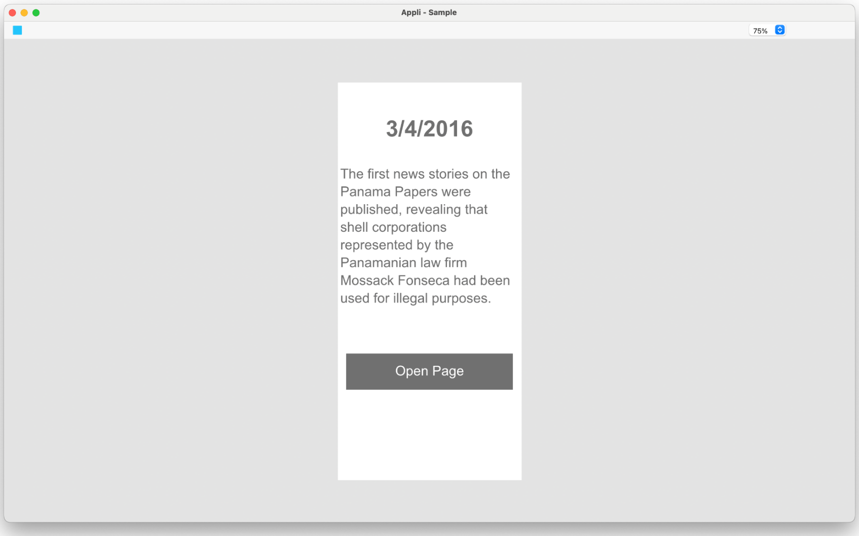
Format
liveCode_integration "Select"

Arguments
Select - Select the code you wish to run.

Introduced 2021-11-16

Figure 152: New version of the script

If you switch to play mode now, you'll see real historical facts.



3/4/2016

The first news stories on the Panama Papers were published, revealing that shell corporations represented by the Panamanian law firm Mossack Fonseca had been used for illegal purposes.

Figure 153: Running with real historical facts

21.5 A pure Appli version

With the introduction of `rest api call` in Appli 1.3.11, you can now make Web requests from low-code. Let's build a version of this app that uses that instead of LiveCode integration.

Before we begin, change the name of the `Untitled` screen to `LiveCode Integration Screen`. Using the contextual menu (right click) on the project browser, click the screen name and choose *Duplicate Screen*:

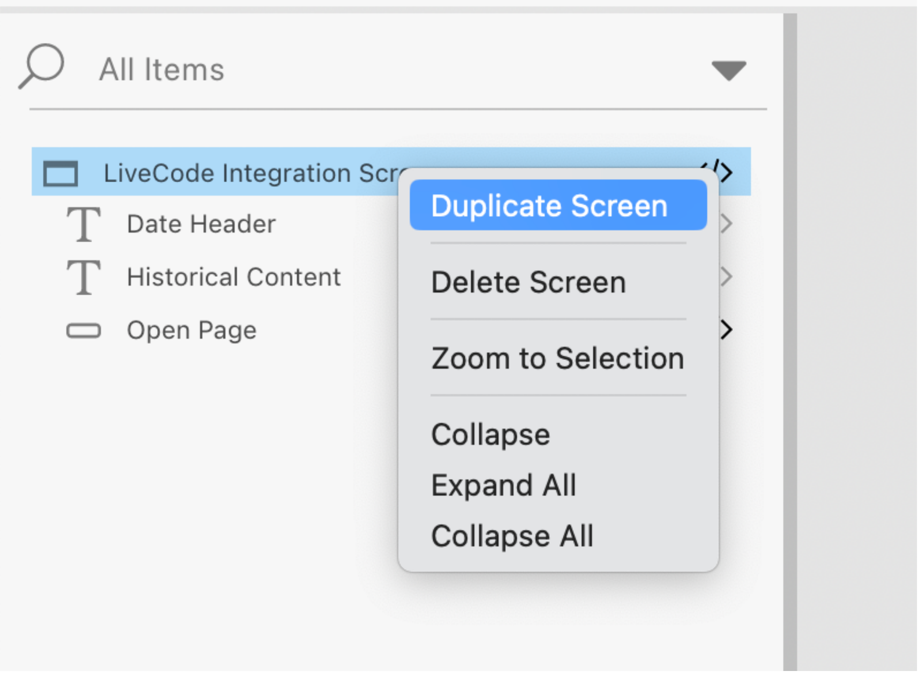


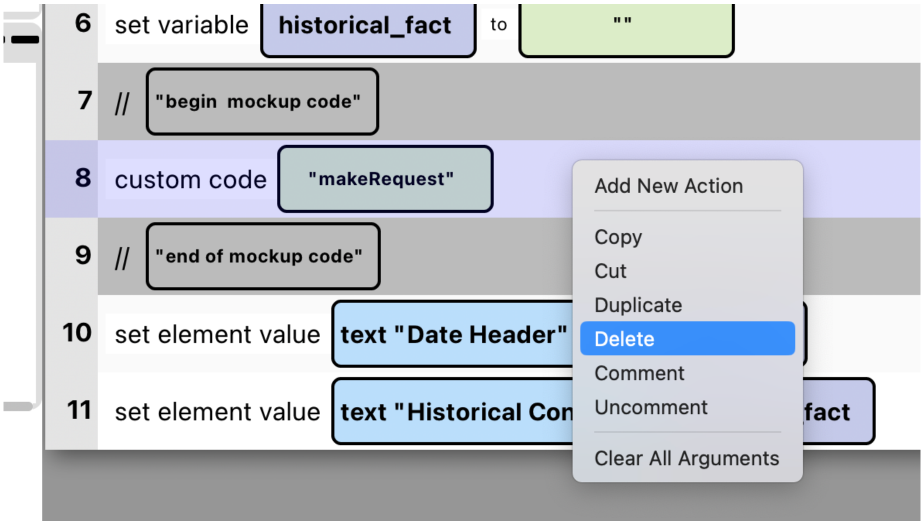
Figure 154: Duplicate Screen contextual menu

Appli will ask for the name for the new screen. Call it `Pure Appli Screen`. After Appli creates the screen, it will automatically switch to the new screen.

Remember those *Mockup code* comments in the `OpenScreen` event? We'll change the code between them. They shouldn't be there anyway since we replaced the mocked version with a real version, we just didn't update the comments.

We'll make heavy use of contextual menus to edit that script. You can right click a line to delete it and also to add a new action below it.

Let's delete that `custom code` call.

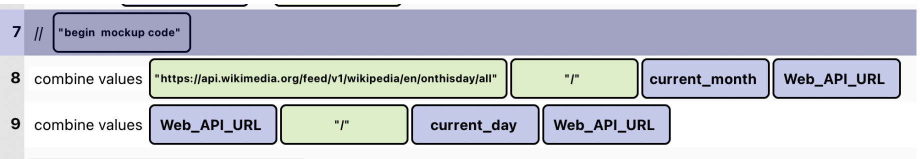


```
set variable historical_fact to ""
// "begin  mockup code"
custom code "makeRequest"
// "end of mockup code"
set element value text "Date Header"
set element value text "Historical Con                fact
```

Figure 155: Delete custom code

And using the `Add Action` menu item, add a new action bellow the *Begin mockup code* comment. When you do that, it adds an input field below the action. You can type part of the action there and select it from a popup list. It is a very fast way to add actions to your script. Faster than selecting them from the categories.

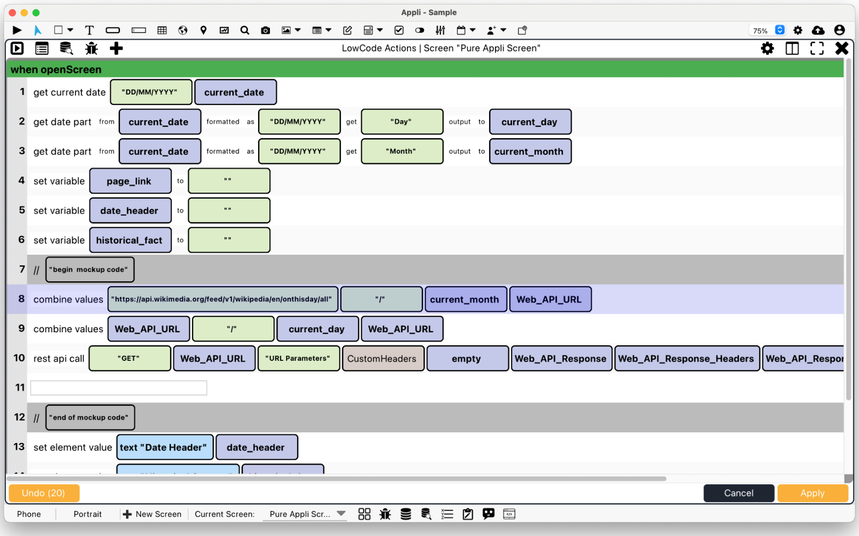
Lets add two `Combine values` actions. We'll use them to assemble the URL for the Web API call:



```
// "begin  mockup code"
combine values "https://api.wikimedia.org/feed/v1/wikipedia/en/onthisday/all" "/" current_month Web_API_URL
combine values Web_API_URL "/" current_day Web_API_URL
```

Figure 156: Assembling the URL

After that, we add a `rest api call` action. That action has lots of parameters.



```
when openScreen
get current date "DD/MM/YYYY" current_date
get date part from current_date formatted as "DD/MM/YYYY" get "Day" output to current_day
get date part from current_date formatted as "DD/MM/YYYY" get "Month" output to current_month
set variable page_link to ""
set variable date_header to ""
set variable historical_fact to ""
// "begin mockup code"
combine values "https://api.wikimedia.org/feed/v1/wikipedia/en/onthisday/all" "/" current_month Web_API_URL
combine values Web_API_URL "/" current_day Web_API_URL
rest api call "GET" Web_API_URL "URL Parameters" CustomHeaders empty Web_API_Response Web_API_Response_Headers Web_API_Respor

// "end of mockup code"
set element value text "Date Header" date_header
```

Figure 157: rest api call

That's mostly because some parameters are *configuration for the request*, that means they are used to tell Appli what kind of request it is and what the format for the data being sent is. It also needs to specify which data is being sent and which variables will be used for the data coming back.

Data for a web request is transmitted using different formats. It can be sent as the body of a request, part of the address for the request, or even sent as HTTP headers. Some Web APIs require you to send data using a combination of these formats. It is common for example to send authentication HTTP headers and JSON request bodies. The `rest api call` action allows you to specify all of that, but it also requires you to specify all of that. That means you'll spend some time configuring the action.

Here is a list of the parameters to use:

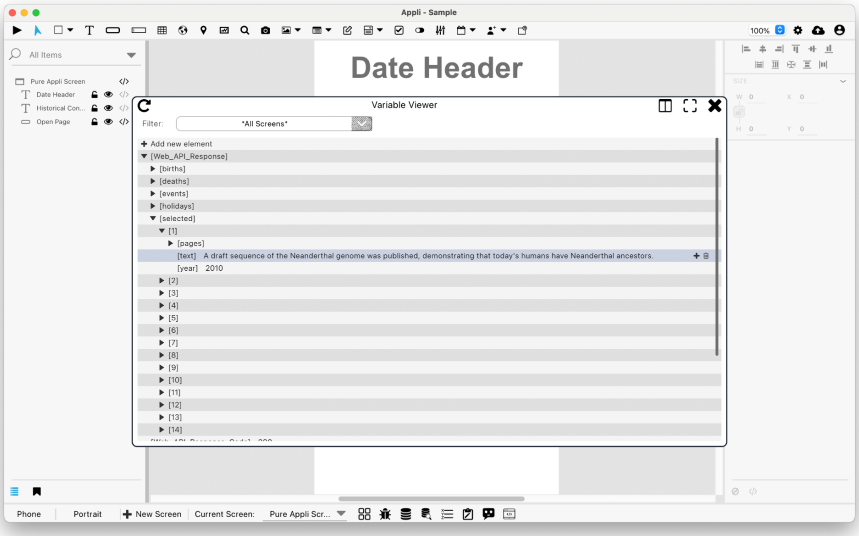
- `GET`: Thats the request method we're using.
- `Web_API_URL`: That's the endpoint URL, we assembled it using the combine actions above.
- `URL Parameters`: We're not really using URL Parameters but we need to select something, so that will do.
- `Custom Headers`: Leave it untouched, it is optional and we're not using it.
- `empty`: At the moment, the `RequestBody` parameter is required. We're not sending any data because all we need for this Web API is specified in the URL. Unfortunately, we can't leave that argument empty. So we just

create a new variable named `empty` that has no content.

- `Web_API_Response`: A new variable that will contain the response data from the Web API request
- `Web_API_Response_Headers`: A new variable that will contain the response HTTP headers from the Web API request.
- `Web_API_Response_Code`: A new variable that will contain the response status code from the HTTP request.

We're not really interested in the last two variables, we'll not use them but they are required to complete the action.

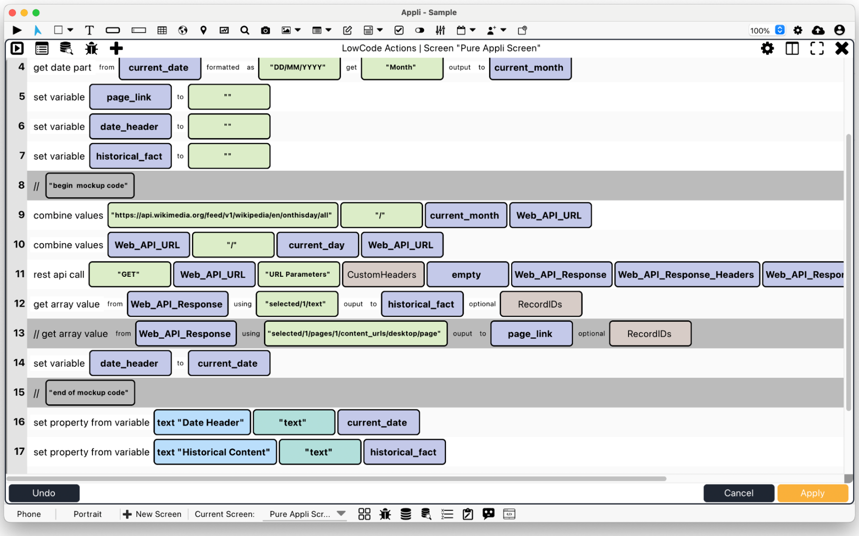
At this point, you can switch to *Play mode* and run the code. It won't work, as in you won't see any fact but the request will be made and the variables will be filled in with data. After switching to play mode, wait for the request to happen and switch back to *Pointer mode* and open the *Variable Viewer* (it is the icon on the footer with X,Y,Z).



```
+ Add new element
▼ [Web_API_Response]
  ▶ [births]
  ▶ [deaths]
  ▶ [events]
  ▶ [holidays]
  ▼ [selected]
    ▼ [1]
      ▶ [pages]
        [text]   A draft sequence of the Neanderthal genome was published, demonstrating that today's humans have Neanderthal ancestors.
        [year]   2010
    ▶ [2]
    ▶ [3]
    ▶ [4]
    ▶ [5]
    ▶ [6]
    ▶ [7]
    ▶ [8]
    ▶ [9]
    ▶ [10]
    ▶ [11]
    ▶ [12]
    ▶ [13]
    ▶ [14]
```

Figure 158: Variable viewer showing the responde of a Web API call

By checking the content of the `Web_API_Response` array, we can figure out how to fill the `historical_fact`, `date_header`, and `page_link variables`.



```
get date part from current_date formatted as "DD/MM/YYYY" get "Month" output to current_month
set variable page_link to ""
set variable date_header to ""
set variable historical_fact to ""
// "begin  mockup code"
combine values "https://api.wikimedia.org/feed/v1/wikipedia/en/onthisday/all" "/" current_month Web_API_URL
combine values Web_API_URL "/" current_day Web_API_URL
rest api call "GET" Web_API_URL "URL Parameters" CustomHeaders empty Web_API_Response Web_API_Response_Headers Web_API_Respor
get array value from Web_API_Response using "selected/1/text" ouput to historical_fact optional RecordIDs
// get array value from Web_API_Response using "selected/1/pages/1/content_urls/desktop/page" ouput to page_link optional RecordIDs
set variable date_header to current_date
// "end of mockup code"
set property from variable text "Date Header" "text" current_date
set property from variable text "Historical Content" "text" historical_fact
```

Figure 159: Final version of the script

Once the script is done, you have a full historical fact gathering app built with Appli.

21.6 Final words and next steps

LiveCode integration allows you to make more complex apps by leveraging the ease of use of Appli development workflow combined with the powerful features of LiveCode, thus being able to create apps that wouldn't be possible by just using low code and no code.

By using LiveCloud APIs, you can read and write to tables in Appli. Some next steps you might want to attempt on your own are:

- Can you write the historical facts to a table instead of variables?
- Can you display more than one fact using a layout?

22 Tutorial: Web Request

22.1 An introduction to Web APIs

Much of our day to day digital lives are powered by Web APIs. They provide the infrastructure that allows most web services and apps to work. In essence, they provide a way for an app or service to access the features provided by a server.

> For example: When you use Appli to save a database record to the cloud, Appli makes a Web API request to our cloud servers to store the data.

Most Web APIs are powered by the same technologies that make the web work. They are accessible via URLs and the HTTP protocol. The most common format used to exchange data with a server is called JSON.

JSON might seem strange if you never seen it before, but it is an easy format for computers to understand. Our sample app will pick the current day and month, make a request to that API, and then show the results to the user.

In a nutshell, a very reductive nutshell, what happens is that the developer packs some instructions into a network request to a server and the server replies with the answers to those instructions. On public facing APIs, most of the features that are exposed to developers are querying features, things that make it easier to get data from a database.

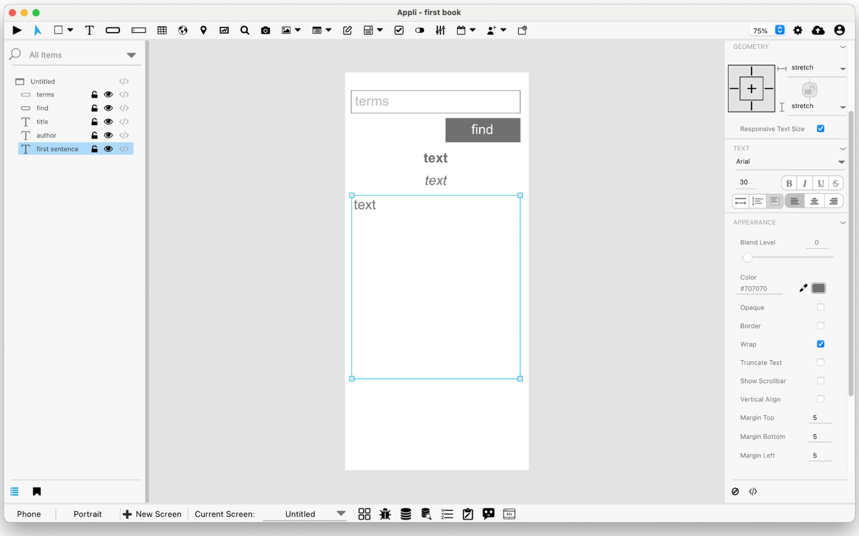
22.2 Open Library API

The Open Library API is an API provided by The Internet Archive to search their extensive catalog of media. You can think of it as a massive library catalog encompassing books, games, and other media types you could find in a library. The API is divided into different sets depending on what features you need for your app. In our case we'll use their Search API to get metadata about books (and other media) depending on search terms provided by the user.

22.3 Our sample app

Instead of creating a full feature library browsing app, something that is doable but beyond the scope of a simple tutorial on Web Requests, we'll create a very simple app. A search box for the user to fill in with their query terms, and a display that will show the first match returned from the *Search API*. The API can (and will) return thousands of matches for a given search. For the sake of simplicity we'll just show one.

22.3.1 Creating the UI



stretch
stretch
Responsive Text Size ☑
Arial
30
Blend Level 0
Color #707070
Opaque □
Border □
Wrap ☑
Truncate Text □
Show Scrollbar □
Vertical Align □
Margin Top 5
Margin Bottom 5
Margin Left 5

Figure 160: Sample Application UI

The UI consists of:

- A *field element* named *terms*.
- A *button element* named *find*.
- Three *text elements* named *title* for the book title, *author* for the book author, and *first sentence* to display the first sentence in the book.

Notice that you need to tweak the *first sentence* appearance to display multiple lines and wrap the text (the properties are shown in the screenshot).

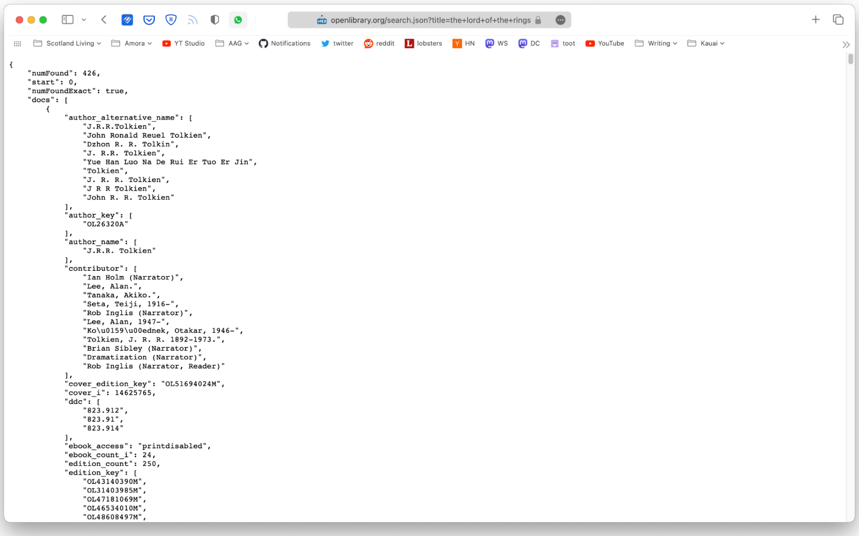
22.3.2 Implementing the find book button

The script we'll use for the find book button will display a *loading dialog* so the user knows the application is busy doing the request, make the web request and extract the data we need from the API response, and finally fill the text elements with the data and hide the *loading dialog*.

The *search API* works with URL parameters. URL parameters (also known as *query strings*) are a way to pass key and value pairs in a Web Address. The format for a key and value pair is `<key>=<value>` and each pair is separated from the others with an ampersand. The address part is separated from the parameters part using a question mark. So for searching the Internet Archive catalog, for *The Lord Of The Rings*, we'd make a request like for:

`https://openlibrary.org/search.json?title=the+lord+of+the+rings`

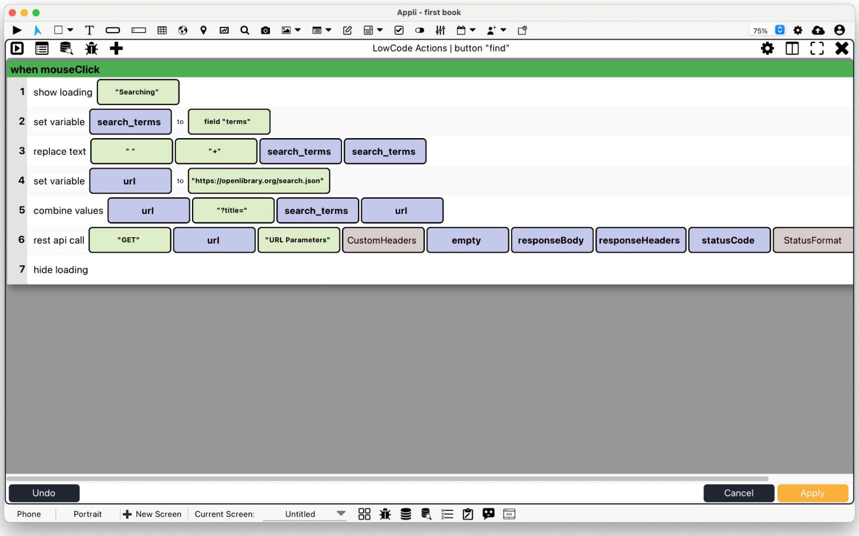
Notice that spaces in parameters are replaced with plus signs. If you open that URL in your browser, you'll see the JSON response containing all the results for that query.



```
{
    "numFound": 426,
    "start": 0,
    "numFoundExact": true,
    "docs": [
        {
            "author_alternative_name": [
                "J.R.R.Tolkien",
                "John Ronald Reuel Tolkien",
                "Dzhon R. R. Tolkin",
                "J. R.R. Tolkien",
                "Yue Han Luo Na De Rui Er Tuo Er Jin",
                "Tolkien",
                "J. R. R. Tolkien",
                "J R R Tolkien",
                "John R. R. Tolkien"
            ],
            "author_key": [
                "OL26320A"
            ],
            "author_name": [
                "J.R.R. Tolkien"
            ],
            "contributor": [
                "Ian Holm (Narrator)",
                "Lee, Alan.",
                "Tanaka, Akiko.",
                "Seta, Teiji, 1916-",
                "Rob Inglis (Narrator)",
                "Lee, Alan, 1947-",
                "Ko\u0159\u00ednek, Otakar, 1946-",
                "Tolkien, J. R. R. 1892-1973.",
                "Brian Sibley (Narrator)",
                "Dramatization (Narrator)",
                "Rob Inglis (Narrator, Reader)"
            ],
            "cover_edition_key": "OL51694024M",
            "cover_i": 14625765,
            "ddc": [
                "823.912",
                "823.91",
                "823.914"
            ],
            "ebook_access": "printdisabled",
            "ebook_count_i": 24,
            "edition_count": 250,
            "edition_key": [
                "OL43140390M",
                "OL31403985M",
                "OL47181069M",
                "OL46534010M",
                "OL48608497M",
```

Figure 161: Screenshot showing search results in JSON

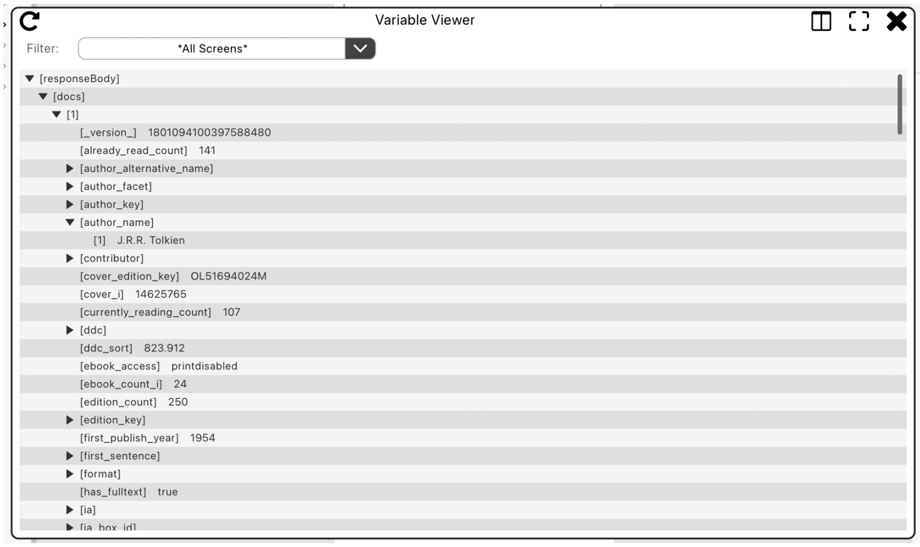
Before creating the full script for the find button, let's create a partial version just to show how the same search results will appear inside Appli. Add this script to your *find button*:



```
when mouseClick
1 show loading "Searching"
2 set variable search_terms to field "terms"
3 replace text " " "+" search_terms search_terms
4 set variable url to "https://openlibrary.org/search.json"
5 combine values url "?title=" search_terms url
6 rest api call "GET" url "URL Parameters" CustomHeaders empty responseBody responseHeaders statusCode StatusFormat
7 hide loading
```

Figure 162: Partial script for the find button

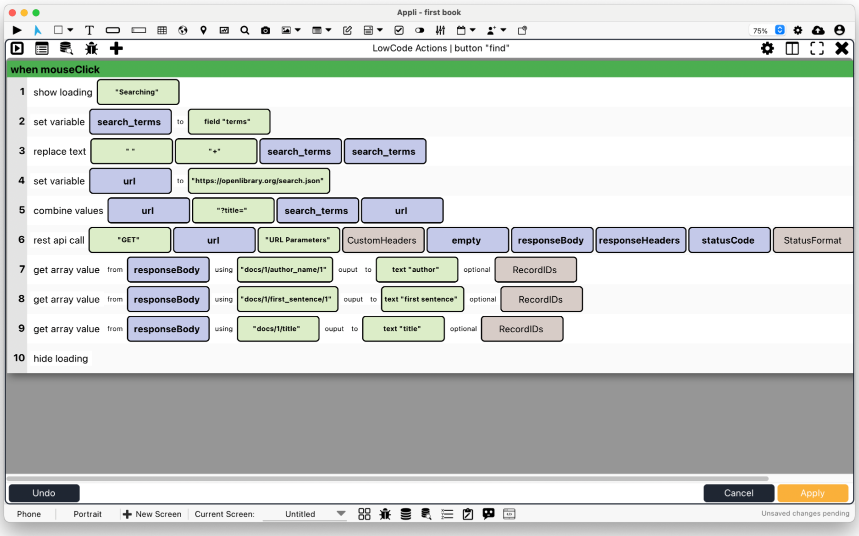
That script will execute the search and store the response in a variable called *responseBody*. If you switch into play mode and type *"the lord of the rings"* and press the *find button*, the app will execute the search and stop. Using the *variable viewer* you can see how the JSON response is represented inside an Appli array.



```
▼ [responseBody]
  ▼ [docs]
    ▼ [1]
        [_version_]   1801094100397588480
        [already_read_count]   141
      ▶ [author_alternative_name]
      ▶ [author_facet]
      ▶ [author_key]
      ▼ [author_name]
          [1]   J.R.R. Tolkien
      ▶ [contributor]
        [cover_edition_key]   OL51694024M
        [cover_i]   14625765
        [currently_reading_count]   107
      ▶ [ddc]
        [ddc_sort]   823.912
        [ebook_access]   printdisabled
        [ebook_count_i]   24
        [edition_count]   250
      ▶ [edition_key]
        [first_publish_year]   1954
      ▶ [first_sentence]
      ▶ [format]
        [has_fulltext]   true
      ▶ [ia]
      ▶ [ia_box_id]
```

Figure 163: Variable viewer showing the *responseBody* array

It is quite useful to execute partial scripts like this and then inspect the response using the variable viewer. It makes it easier to find out which array elements you need to assemble your user interface. Let's complete the script now.

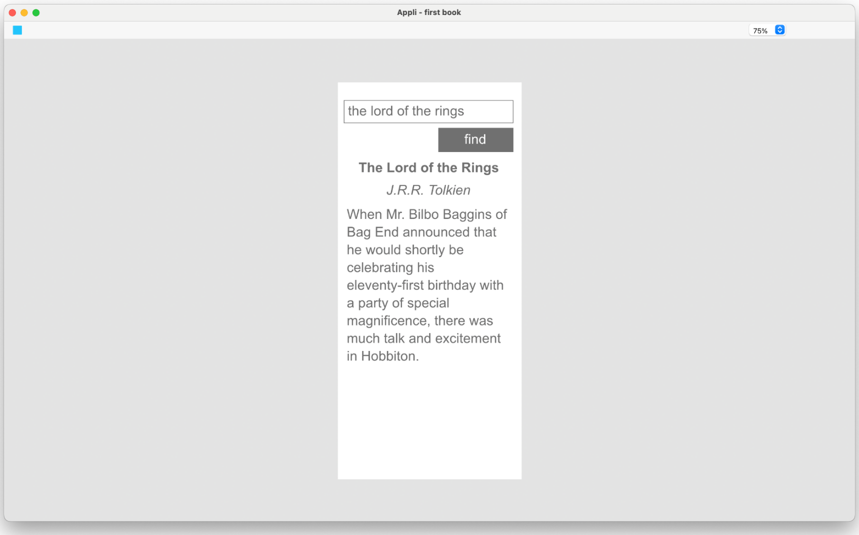


```
when mouseClick
1  show loading "Searching"
2  set variable search_terms to field "terms"
3  replace text " " "+" search_terms search_terms
4  set variable url to "https://openlibrary.org/search.json"
5  combine values url "?title=" search_terms url
6  rest api call "GET" url "URL Parameters" CustomHeaders empty responseBody responseHeaders statusCode StatusFormat
7  get array value from responseBody using "docs/1/author_name/1" ouput to text "author" optional RecordIDs
8  get array value from responseBody using "docs/1/first_sentence/1" ouput to text "first sentence" optional RecordIDs
9  get array value from responseBody using "docs/1/title" ouput to text "title" optional RecordIDs
10 hide loading
```

Figure 164: Script for the find button

> Check out the documentation for Get Array Value to learn more about how that action access deep array elements using *path notation*.

Using three *Get Array Value* actions, we can extract the title, author, and first sentence from the *responseBody* array and add them to the elements in the user interface.



the lord of the rings

The Lord of the Rings

J.R.R. Tolkien

When Mr. Bilbo Baggins of Bag End announced that he would shortly be celebrating his eleventy-first birthday with a party of special magnificence, there was much talk and excitement in Hobbiton.

Figure 165: App running

22.4 Final Words

This tutorial shows how to do a simple web request. To be an effective developer using Web API servers, you need to understand how Web Requests work. The best resource for this is The Web Mechanics section of MDN Webdocs. In there you'll find how web requests work and how URLs are assembled and much more.

If you find yourself being challenged with some Web API server you want to interface with but doesn't know exactly how to do it, reach out to us in our forum. Web Requests are an advanced topic and it is ok to ask for help when you need it.

23 Tutorial: Relational Data

Relational data happens when some information you're storing in a table is linked to another records in another table. A good example is how comments on a blog post can be linked to the post itself and to the author of the comment, like shown in the screenshot below:

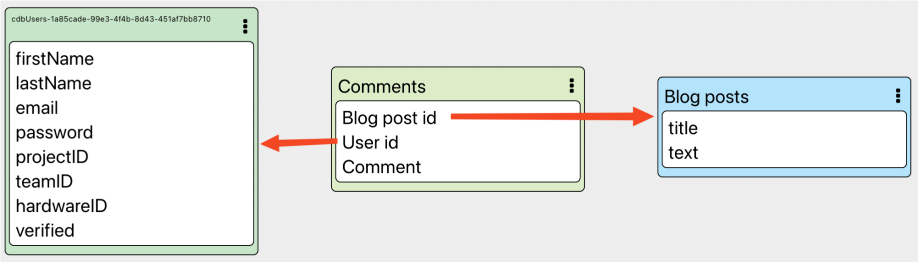


Figure 166: A sample set of tables showing the links between the keys and tables.

The comments table has three keys (I drew the arrows showing the links between tables):

- The comment text.
- A key holding the ID of the user who posted the comment, that one matches a record ID from the users table.
- A key holding the ID of the original blog post the user is commenting on, this ID matches the record ID of a record in the blog posts table.

> Remember that records always have a Record ID. The ID key does not appear in the Data Modeler screen but you can see them in the Data Viewer.

For this tutorial, we'll create a simple demo to show book quotes. This demo will have two screens with one layout on each screen. On the first screen, the user will select the book and that will navigate to the second screen where a layout will show quotes from that book.

We'll not spend too long on how to create layouts and interfaces on this tutorial, you need to go through the database tutorial first if you don't remember how to do it.

23.1 Creating the tables

Let's create two tables: Book and Quote. Since this is just a demo we'll make the tables as simple as possible. For the Book table we're just having a single key to hold the book title. In the Quote table we'll have a key to hold the quote text and one to hold a Record ID from the Book table.

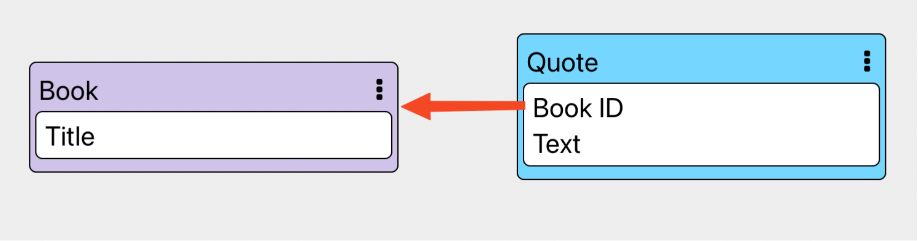


Figure 167: A simple way to store book quotes.

23.2 Adding data

Instead of creating an interface to add data, we'll just use the Data Viewer to insert it. Just double-click on a line and start typing. Appli will create the record for you. Notice that your IDs will not match the ones in the screenshot.

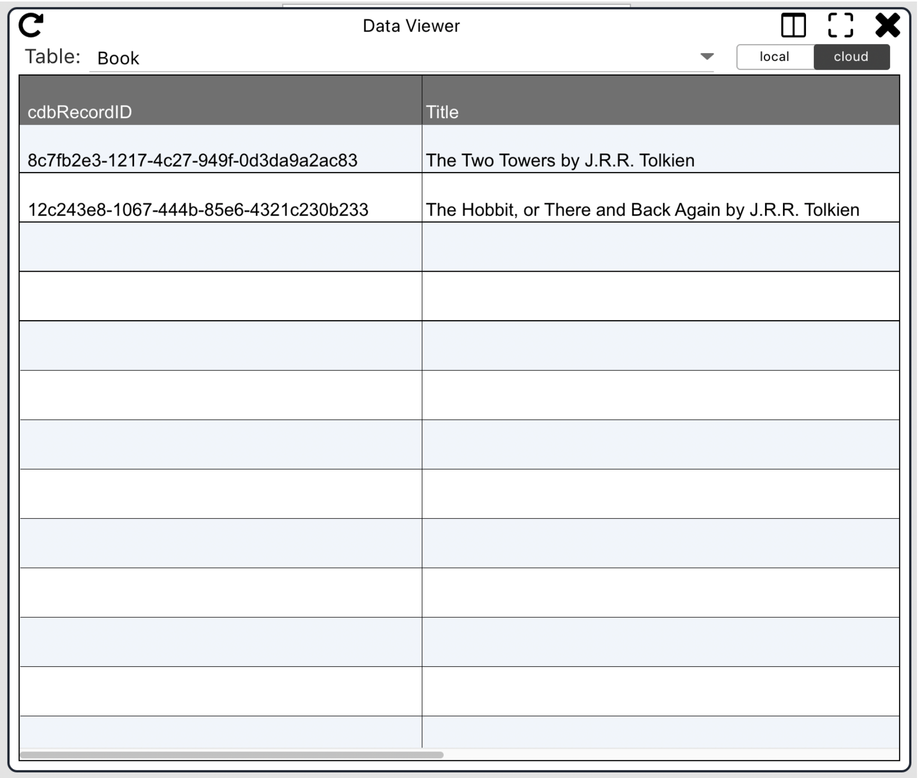


Table: Book

cdbRecordID	Title
8c7fb2e3-1217-4c27-949f-0d3da9a2ac83	The Two Towers by J.R.R. Tolkien
12c243e8-1067-444b-85e6-4321c230b233	The Hobbit, or There and Back Again by J.R.R. Tolkien

Figure 168: Book table with data.

You can use the context menu on a Record ID to copy the ID and paste it on another table.

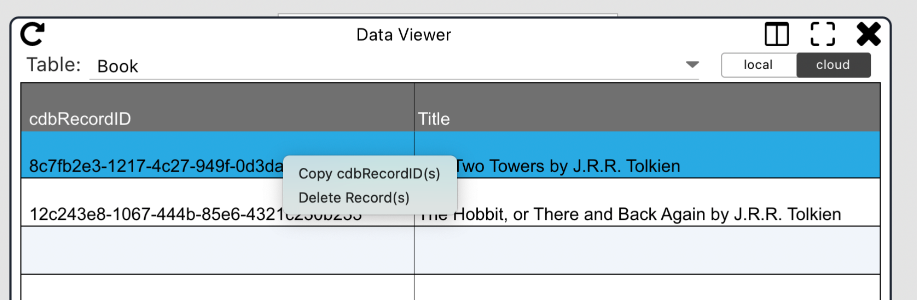


Table: Book

cdbRecordID	Title
8c7fb2e3-1217-4c27-949f-0d3da	Two Towers by J.R.R. Tolkien
12c243e8-1067-444b-85e6-4321c230b233	The Hobbit, or There and Back Again by J.R.R. Tolkien

Figure 169: Copying an ID.

Then on the Quote table, you can paste that ID into the Book ID field for a new record and then add the quote you want. For this tutorial you can make the quotes up, I copied them from the Internet.

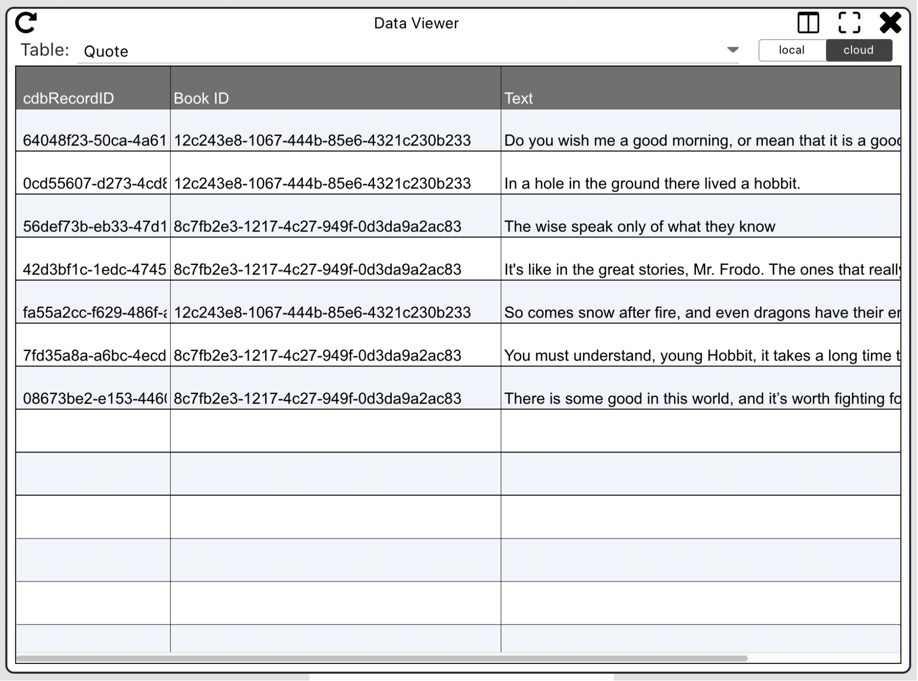


Table: Quote

cdbRecordID	Book ID	Text
64048f23-50ca-4a61	12c243e8-1067-444b-85e6-4321c230b233	Do you wish me a good morning, or mean that it is a good
0cd55607-d273-4cd8	12c243e8-1067-444b-85e6-4321c230b233	In a hole in the ground there lived a hobbit.
56def73b-eb33-47d1	8c7fb2e3-1217-4c27-949f-0d3da9a2ac83	The wise speak only of what they know
42d3bf1c-1edc-4745	8c7fb2e3-1217-4c27-949f-0d3da9a2ac83	It's like in the great stories, Mr. Frodo. The ones that really
fa55a2cc-f629-486f-a	12c243e8-1067-444b-85e6-4321c230b233	So comes snow after fire, and even dragons have their en
7fd35a8a-a6bc-4ecd	8c7fb2e3-1217-4c27-949f-0d3da9a2ac83	You must understand, young Hobbit, it takes a long time t
08673be2-e153-4460	8c7fb2e3-1217-4c27-949f-0d3da9a2ac83	There is some good in this world, and it’s worth fighting fo

Figure 170: Data for the Quote table.

Make sure the Book IDs match Record IDs from the Book table, that is how we link them.

23.3 Creating the books screen

Add a layout and a text element to the layout. Link them to the Book table using no-code, exactly how you learned to do in the database tutorial.

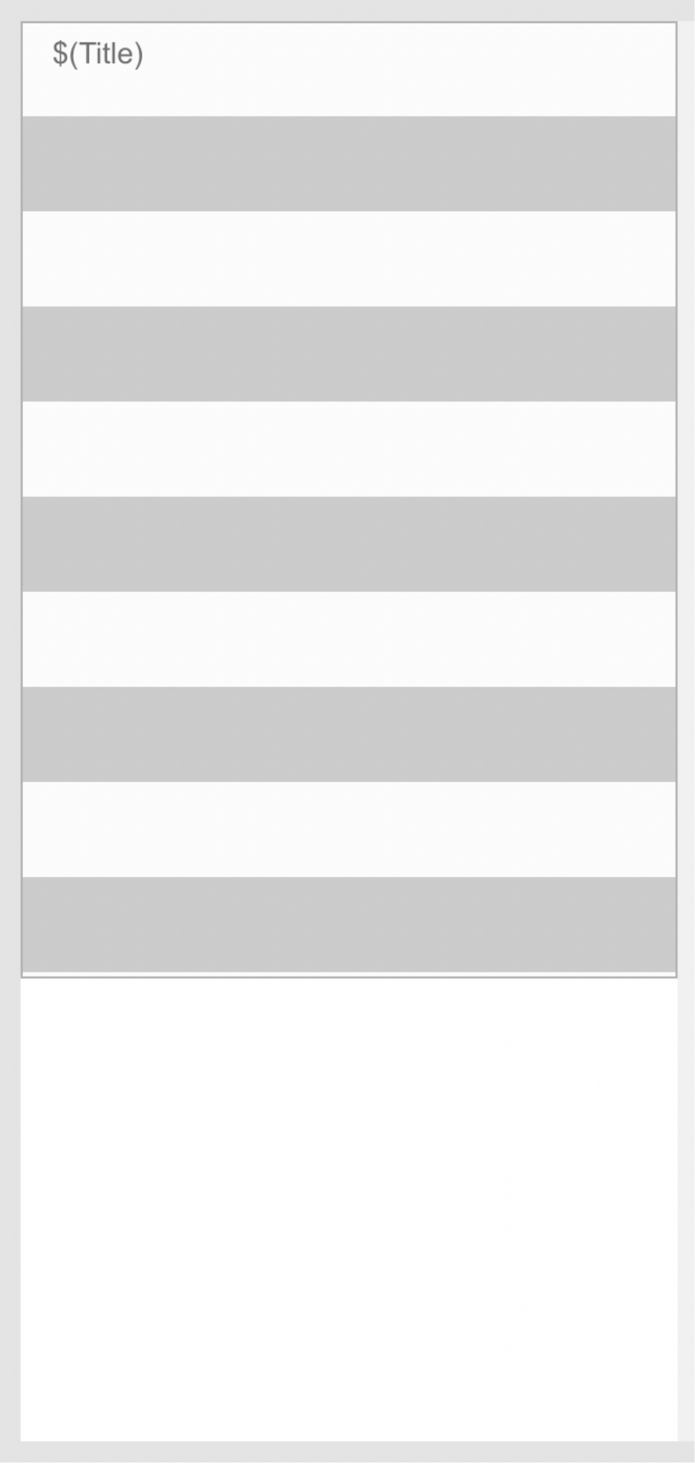
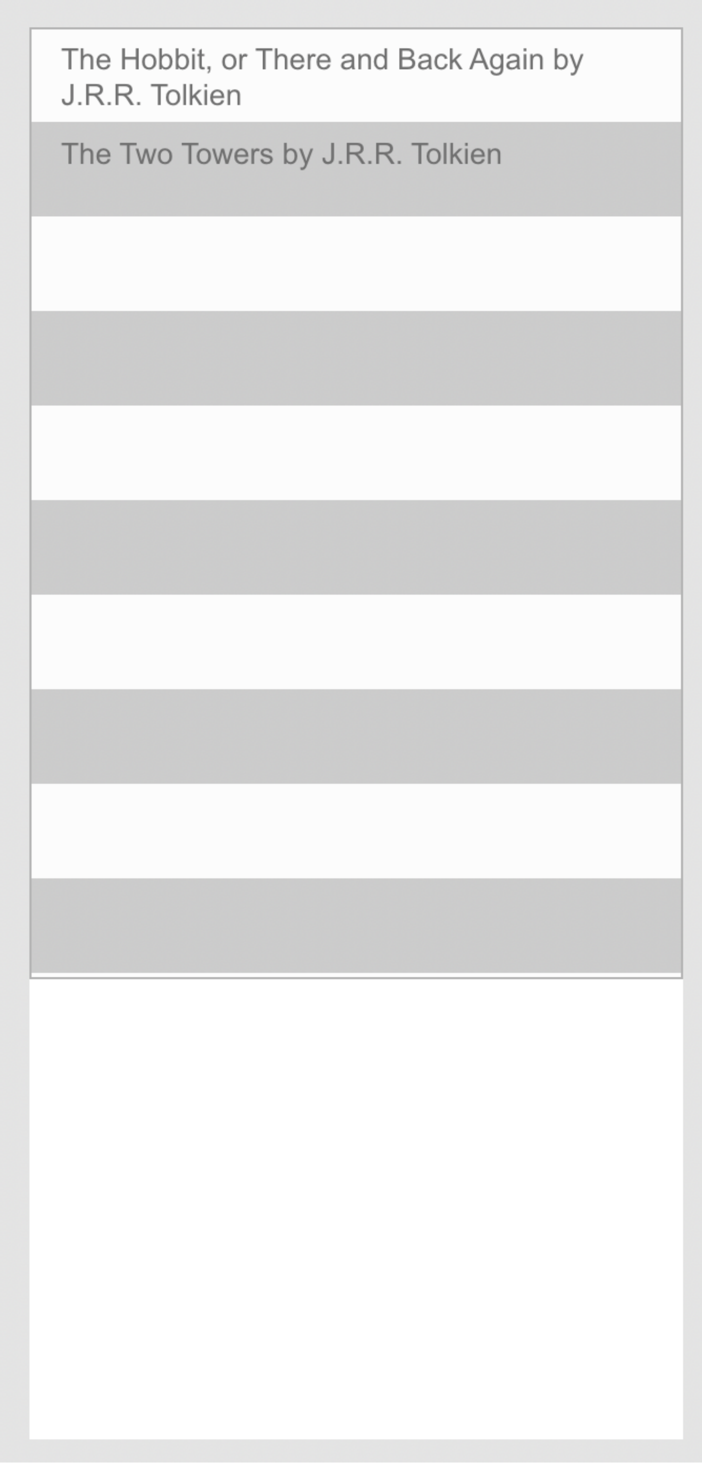


Figure 171: Layout with a text element.

When you switch to play mode, you should see a list of the books you added to the database.

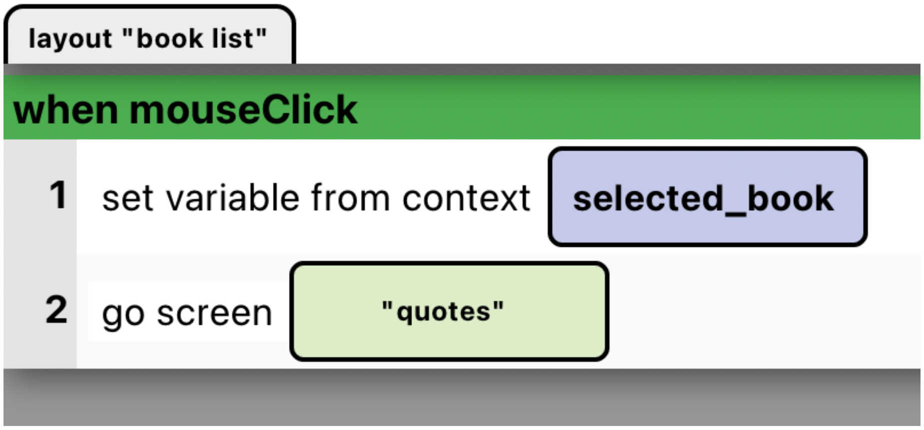


The Hobbit, or There and Back Again by J.R.R. Tolkien

The Two Towers by J.R.R. Tolkien

Figure 172: Showing all books.

The low-code actions for this screen will be very simple. We'll simply store the ID for the clicked book in a variable and move to the quotes screen.



layout "book list"

when mouseClick

1 set variable from context selected_book

2 go screen "quotes"

Figure 173: Actions for the layout.

23.4 Creating the quotes screen

This is where the magic will happen. Make it just like the previous screen, just a layout and a text element.

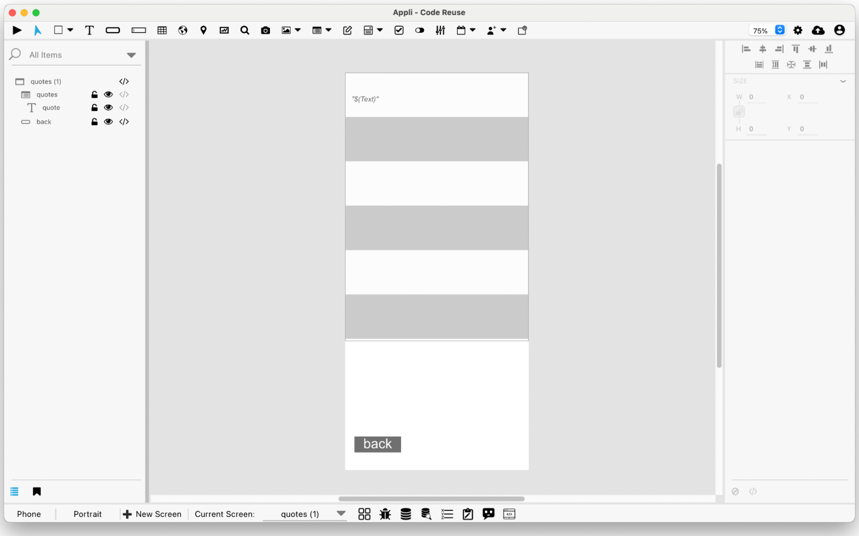
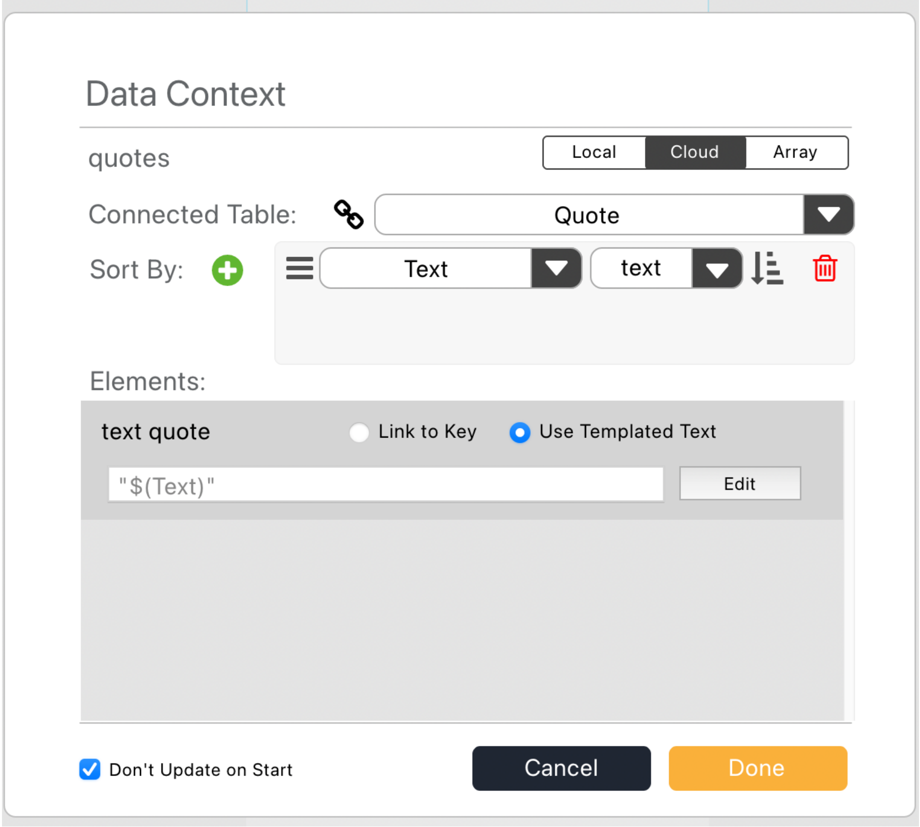


Figure 174: Interface for the quotes screen.

Using no-code, link that layout to the quotes table but make sure you check the *don't update on start* checkbox. If you don't do it, the screen will show all quotes when it opens instead of just the ones from the book you selected. If you check this and switch to play mode, the layout will be empty. This is intentional, we don't want the layout to fetch records by itself, we want to tell it what records to display and we'll do it by adding actions to the *openScreen* event of this screen.



Data Context

quotes — Local | **Cloud** | Array

Connected Table: Quote

Sort By: Text, text

Elements:

text quote ○ Link to Key ● Use Templated Text

"$(Text)"

☑ Don't Update on Start

Figure 175: No-code configuration for the quotes layout.

23.5 How to query a table based on data from another table

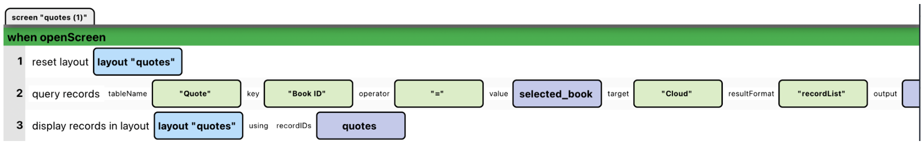
The *openScreen* event is the first event to trigger when we navigate to a screen, so it is the best place to put our code. Our first step is to *Reset Layout* making sure it is empty. If we don't do it, the layout will show the previous results in it for a split second when the screen opens.

To get the quotes for the `selected_book`, we'll use the *Query Records* action.

This action allows us to filter a table for records matching a criteria we specify. We'll configure that action to use the following arguments:

- tableName: Quote
- key: Book ID. That means we'll filter the table based on the values of Book ID.
- operator: Equals (=). There are many operators, we can do partial matches, check numerical values, they are very powerful. For this sample, we'll use the *equals* operator which will check if the Book ID is an exact match to a value we specify.
- value: `selected_book`. That is the value we're matching against. That means we'll filter the records from this table with values that are an exact match to the selected book.
- target: cloud.
- resultFormat: recordList. This means the result of this action will be a list of records.
- output: `quotes`. We'll create a new variable called Quotes to hold the list of matching records.

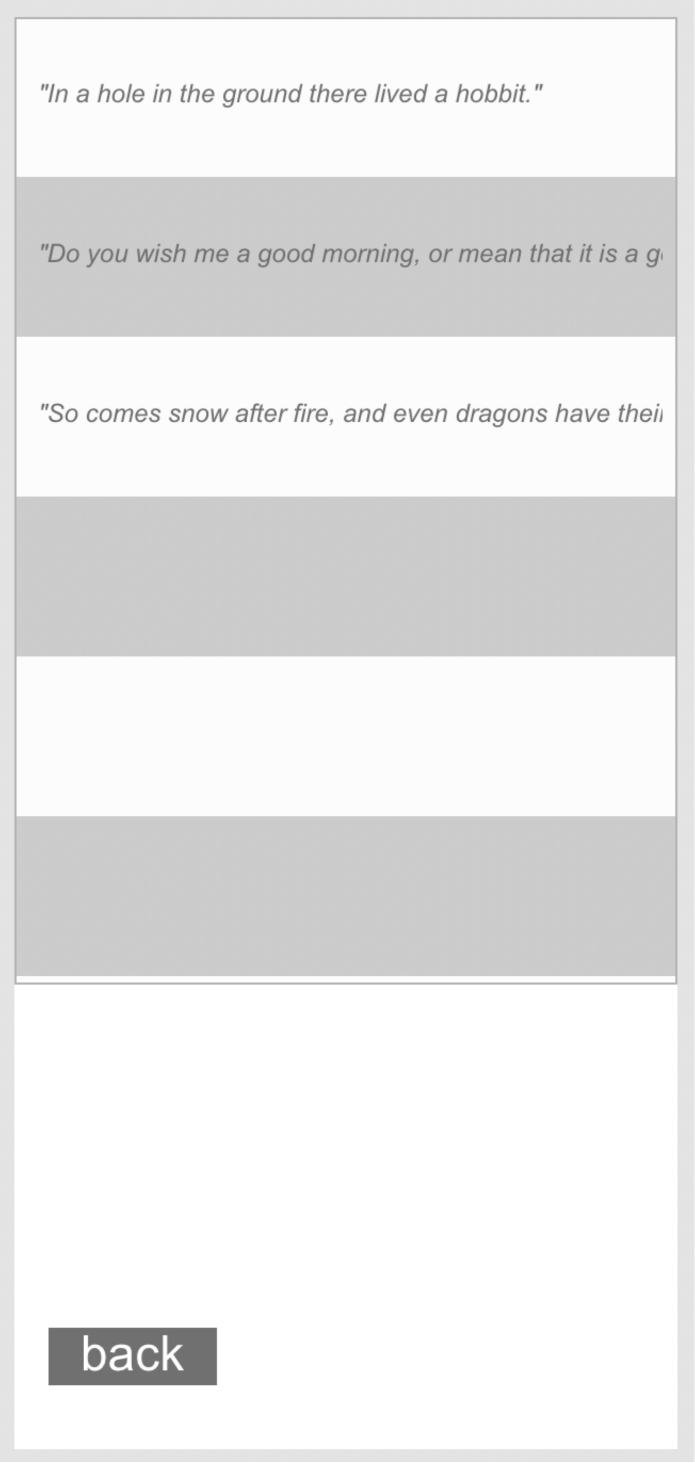
The result from executing this action is that we'll populate the `quotes` variable with a list of matching records. The next step is to tell the layout to display the records from the list, you can do that with the `Display Records In Layout` action. Just select which layout to use and pass the variable `quotes` as the recordIDs argument.



```
screen "quotes (1)"
when openScreen
1 reset layout layout "quotes"
2 query records tableName "Quote" key "Book ID" operator "=" value selected_book target "Cloud" resultFormat "recordList" output
3 display records in layout layout "quotes" using recordIDs quotes
```

Figure 176: Actions for the openScreen event.

Now if you move back to the first screen and switch into play mode, you can select a book and see the quotes in the quotes screen.



"In a hole in the ground there lived a hobbit."

"Do you wish me a good morning, or mean that it is a g

"So comes snow after fire, and even dragons have thei

Figure 177: Quotes for the selected book.

23.6 Next steps

Check out the database actions, they are very powerful and allow you to do complex data manipulation with very few actions. You might also want to check our advance topics playlist on YouTube.

24 Tutorial: Code Reuse

If you've done the other tutorials, you'll be very familiar with events at this point. Specially events such as *openScreen* and *mouseClick* both of which we used in previous chapters. As you develop more advanced apps in Appli, you might start expeciencing some code duplication when you need to have the same sequence of events happen at different moments of your app, such as refreshing a layout which needs to happen both when the screen opens and after a database manipulation. Let me expand on that for a moment.

Consider a screen with a layout that shows multiple records but not all the records in a database. Something like showing quotes for a given book. This is the sample demo we build in the relational data tutorial. The layout data is assembled by a combination of `query records` action to find the quotes for a book and `display records in layout` to load the records into the layout.

> This tutorial builds upon the knowledge and sample from relational data tutorial. You need to complete that before going through the remaining of this content.

That works well for that sample because we hardcoded the database by doing manual data insertions. Data doesn't change so one the app queries the database for quotes, it doesn't need to do it again unless the user selects a different book.

24.1 Adding quotes to the database

Now, let's add a form to the quotes screen and use no-code to bind it to the quote table, as seen in the screenshot below:

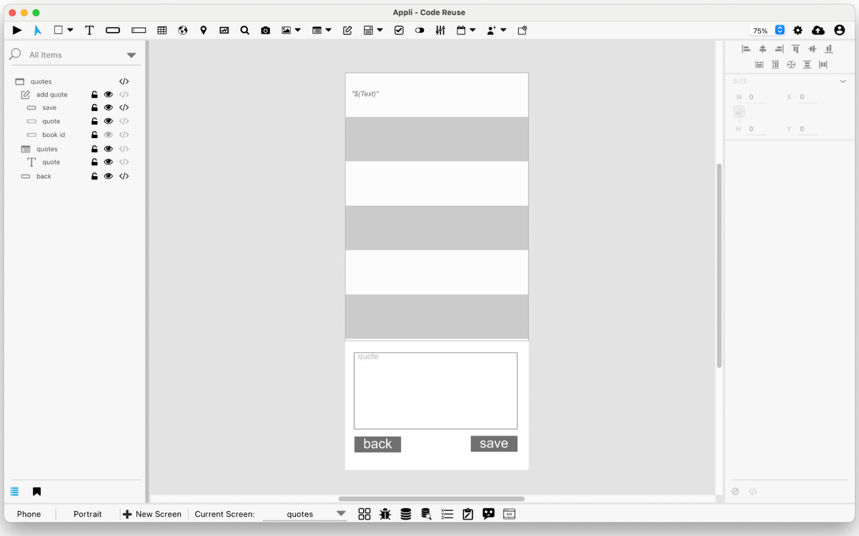


Figure 178: Form for adding quotes.

You might be wondering, considering that the quote table has a *Book ID* field and our form only contains one text element for the quote text, how that ID is handled. It is very simple, we have a hidden text field that we bound to that key using no-code. This is how the form looks if we make that field visible:

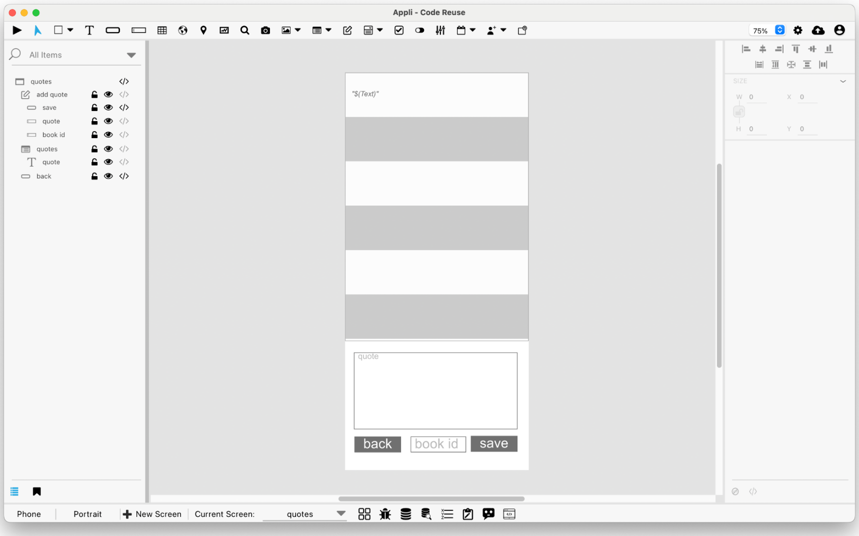
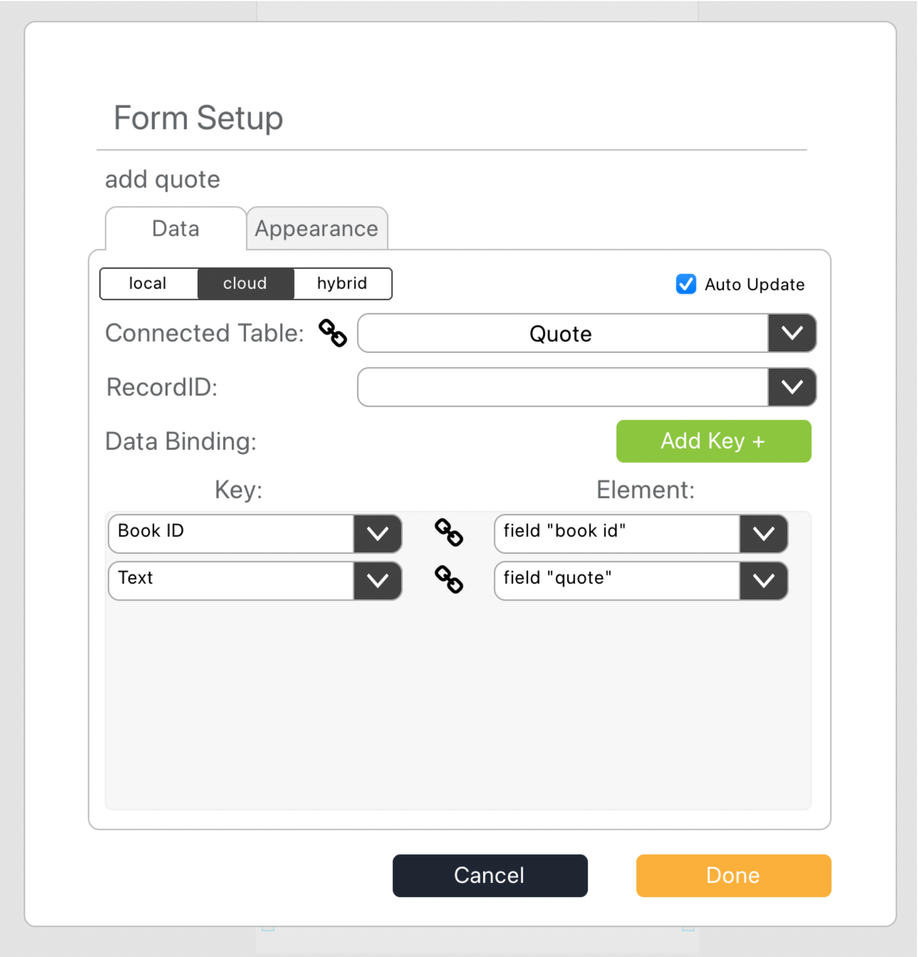


Figure 179: Form with the Book ID field visible.

> Look at the screenshot above the last one to see how the *eye icon* next to the *book id field* is disabled thus making it invisible.

The no-code configuration for that form looks like this:



Form Setup

add quote

local [✓ cloud] hybrid

☑ Auto Update

Connected Table: Quote

RecordID:

Data Binding:

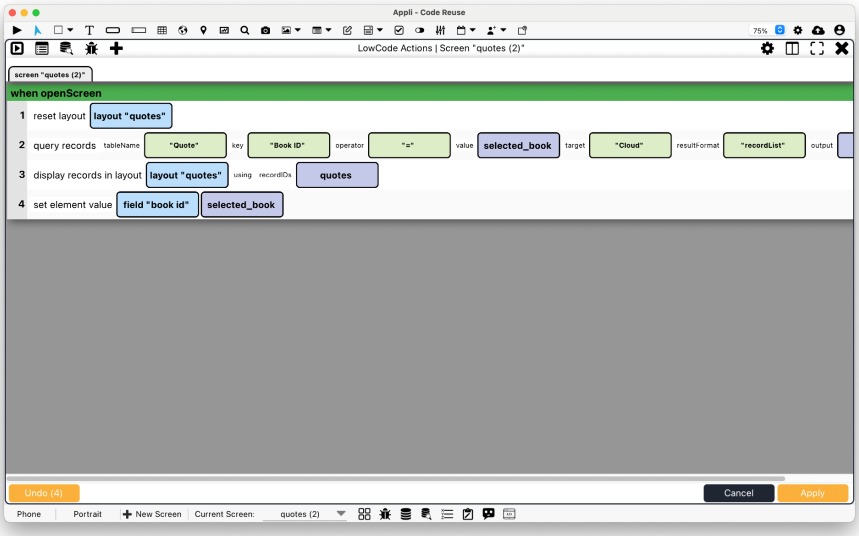
Key:	Element:
Book ID	field "book id"
Text	field "quote"

Figure 180: No-code configuration for the add quote form.

We need to change the *openScreen* event to add the value to the *book id* field. Just add the `set element value` action to the bottom of the *openScreen* event:

```
set element value <book id field> <selected_book>
```

In the end it will look like this:

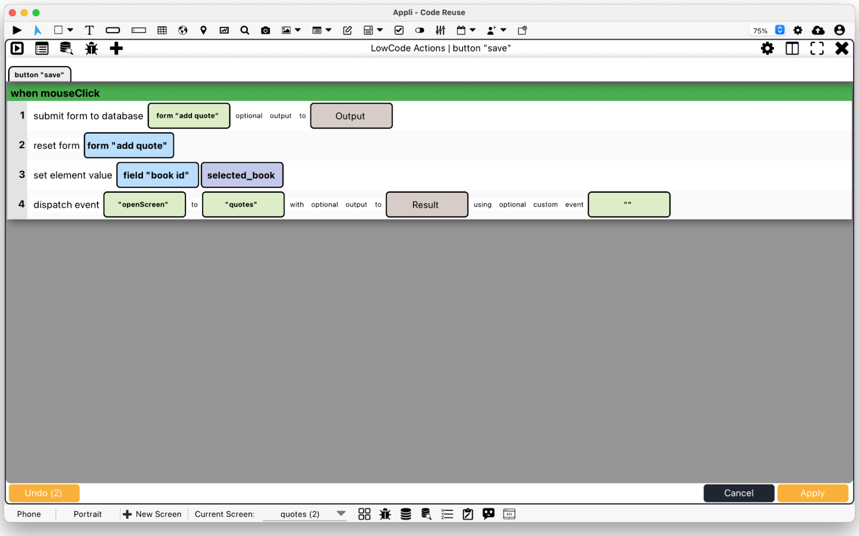


```
when openScreen
1 reset layout  layout "quotes"
2 query records  tableName "Quote"  key "Book ID"  operator "="  value selected_book  target "Cloud"  resultFormat "recordList"  output
3 display records in layout  layout "quotes"  using  recordIDs quotes
4 set element value  field "book id"  selected_book
```

Figure 181: Script with "set element value".

Before I show you the script for that *save button*, let's stop for a second and reason about what is happening once you save a new quote. When you do it, the database changes and a new record is added with a *book id* that matches the current selected book. But the query to figure out the quotes was executed during *openScreen* which means it is not going to run again after you press save. The layout will not automatically update to display your new quote even though the user will expect it to happen.

One solution is to copy and paste the `query records` and `display records in layout` actions from *openScreen* into the bottom of the actions for the *save button*, but that meant we'd have the exact same code in two places and that add burden to maintenance. Avoiding code duplication keeps your application smaller and more nimble which leads to easier support and development cycles. Another solution to cause a layout refresh is to use `dispatch` to execute *openScreen* again.



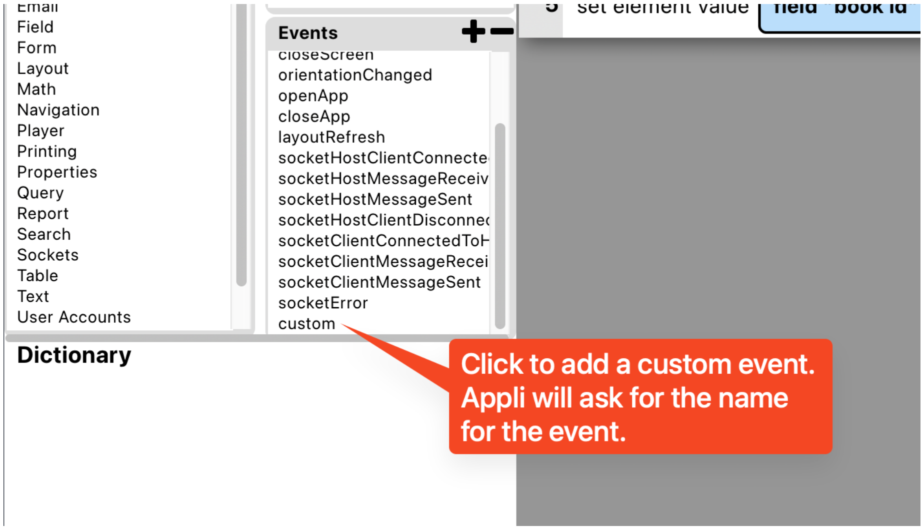
```
when mouseClick
1 submit form to database  form "add quote"  optional output to  Output
2 reset form  form "add quote"
3 set element value  field "book id"  selected_book
4 dispatch event  "openScreen"  to  "quotes"  with optional output to  Result  using optional custom event  ""
```

Figure 182: Dispatching "openScreen" to trigger a layout refresh.

Even though that solves our problem, it creates a bit of friction because unless you remember what happens on *openScreen*, you won't really understand why you're executing it again after saving a record to the database. Right now, that need is clear in your head because you just implemented *openScreen* and the contents are flresh in your head, months from now when you're doing some maintenance work this memory might not be as fresh. That is when *custom events* are useful. You can define your own event with a descriptive name that makes sense for your app and put actions in there to be triggered by other part of your app using `dispatch`.

24.2 Adding a custom event

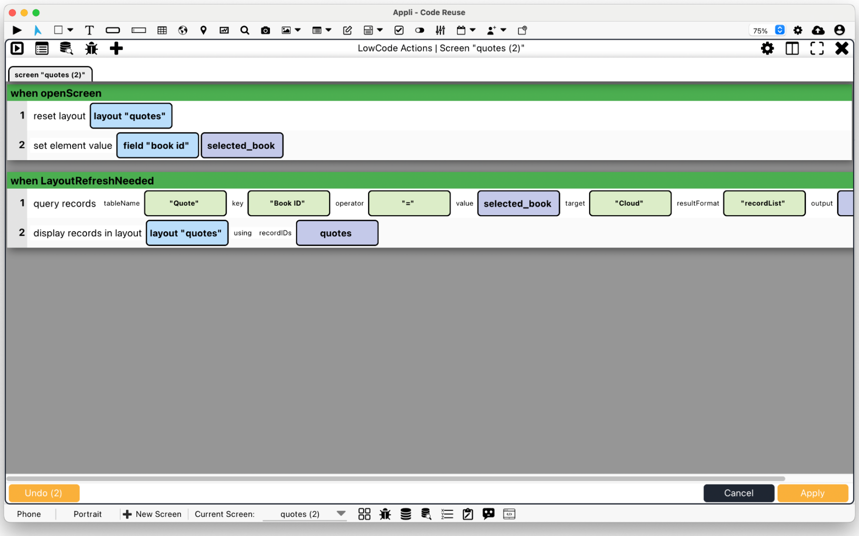
Let's add a custom event to the quotes screen. Open the quotes screen low-code editor and double-click *custom* in the events list. Appli will ask you for a name for the event, use something descriptive such as *LayoutRefreshNeeded*.



Click to add a custom event. Appli will ask for the name for the event.

Figure 183: Custom event in the events list.

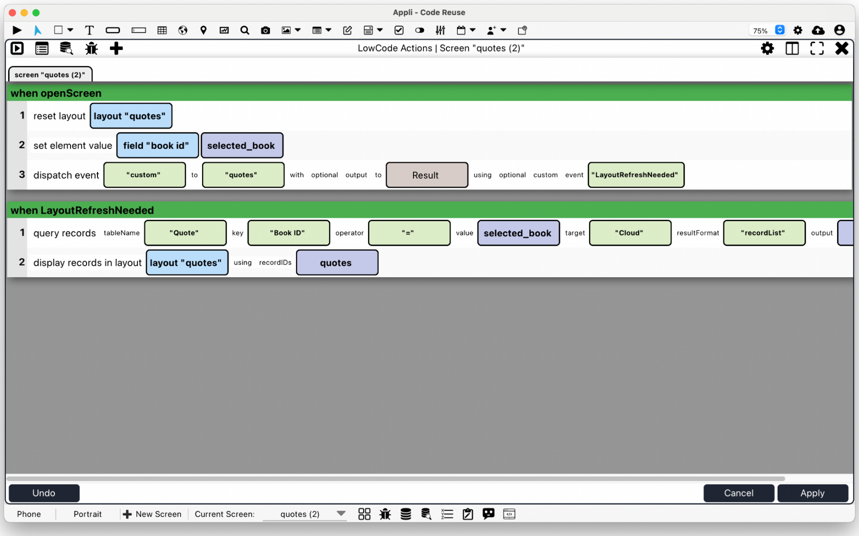
Appli will then add an empty event with that name to the low-code editor. Select and move the `query records` and `display records in layout` actions to the new event.



```
when openScreen
1  reset layout  layout "quotes"
2  set element value  field "book id"  selected_book

when LayoutRefreshNeeded
1  query records  tableName "Quote"  key "Book ID"  operator "="  value selected_book  target "Cloud"  resultFormat "recordList"  output
2  display records in layout  layout "quotes"  using  recordIDs  quotes
```

Figure 184: Custom event.

After that, you can use `dispatch` to trigger the *LayoutRefreshNeeded* event. Just select *custom* as the event type and add *LayoutRefreshNeeded* as the custom event type argument at the end of the action. You'll need to add it both to the *openScreen* event for the screen and the *mouseClick* event from the save button since those are the two places where the layout needs to be refreshed.



when openScreen

1 reset layout layout "quotes"

2 set element value field "book id" selected_book

3 dispatch event "custom" to "quotes" with optional output to Result using optional custom event "LayoutRefreshNeeded"

when LayoutRefreshNeeded

1 query records tableName "Quote" key "Book ID" operator "=" value selected_book target "Cloud" resultFormat "recordList" output

2 display records in layout layout "quotes" using recordIDs quotes

Figure 185: Now with the "dispatch" action

25 Final words

Custom events are very powerful. They allow you to make your code more expressive and meaningful by assignging descriptive names to what would just be a collection of actinons. They also allow you to group often reused actions into a single place making maintenance easier.

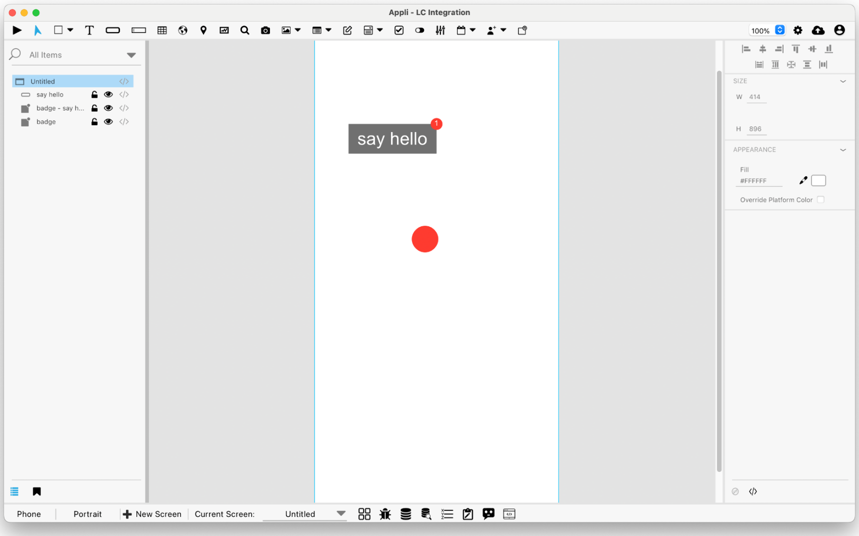
26 Elements

- Badge
- Browser
- Button
- Camera
- Chart
- Checkbox
- Create Account
- Date

- Dropdown
- Field
- Form
- Graphic
- Image
- Layout
- Login
- Map
- Media
- Query
- Radio Group
- Report
- Search Field
- Slider
- Switch
- Tab Menu
- Table
- Text
- Time

27 Element: Badge

This element is a notification or status badge. It can be attached to other elements by clicking on the element, or created as a standalone element. The badge can have text, numbers, icons, or nothing inside of it. It can be connected to a variable so that it is updated whenever the screen is changed.



W 414

H 896

Fill #FFFFFF

Override Platform Color □

Current Screen: Untitled

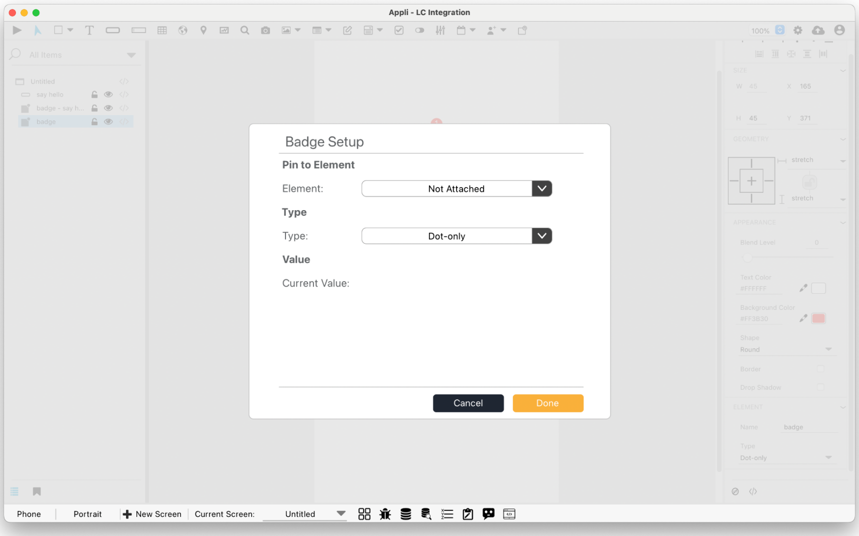
Figure 186: Two Badges, one pinned to a button and another just an unnumbered red dot

27.1 Events

Switches have a mouse related events that can be configured using low-code.

27.2 No Code

Badges can be pinned to elements and have different visual presentations. Both can be configured using the no code interface.



Badge Setup

Pin to Element

Element: Not Attached

Type

Type: Dot-only

Value

Current Value:

Figure 187: Badge Setup Screen

You can also pin badges to elements by clicking the badge element in the toolbar and then clicking the element you want to pin the badge to.

27.3 Actions

Badges have their own action category. Learn more about them here.

27.4 Properties

27.4.1 Size Section

This section is used to configure the badge size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.

Property	Description
height	The value in pixels representing the distance between the bottom side of the element and its top side.

27.4.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

27.4.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the badge.

Property	Description
textFace	The font used in the element's text.
textSize	The text size used in the element's text.
textStyle	The style used in the element's text. It can be bold, italic, underlined, and strikethrough.

27.4.4 Appearance Section

This section is used to configure how the badge looks.

Property	Description
blendLevel	How transparent the element is.
textColor	the color of the text in the element.
backgroundColor	The color used for the element's background.
shape	sets the shape of the element.

Property	Description
showBorder	Turns the visibility of the element's borders on or off.
dropShadow	The drop shadow for the element.

27.4.5 Element Section

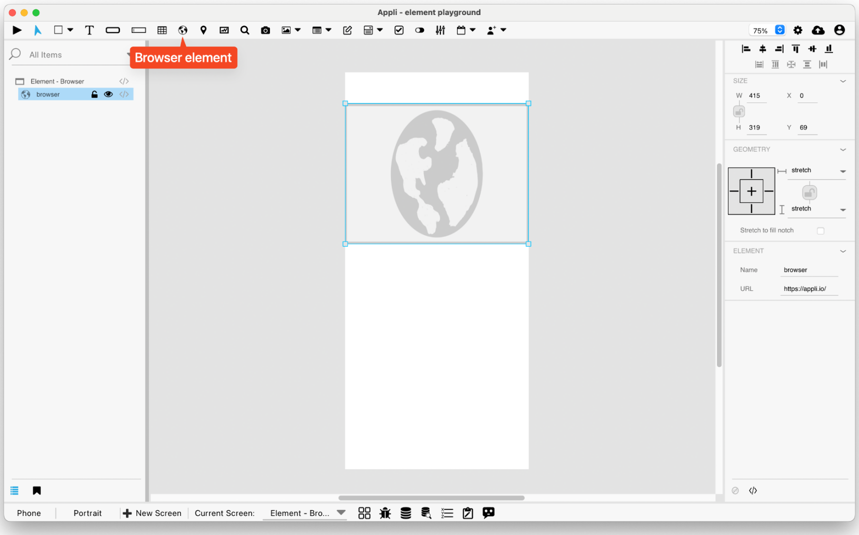
This section is contains properties that are specific to elements of type badge.

Property	Description
name	The name of the element. Displayed in the project browser.
Type	sets the Type of the element.

28 Element: Browser

This element is a WebView that allows the developer to display content from a Web page inside their application.

The browser URL can be set using the property inspector or using the low-code action *Browser* → *Set Browser URL*.



W 415
X 0
H 319
Y 69

stretch
stretch
Stretch to fill notch □

Name browser
URL https://appli.io/

Current Screen: Element - Bro...

Figure 188: Sample browser

Browser elements will only load the web address in play mode; they remain with just a placeholder globe in edit mode.

28.1 Properties

28.1.1 Size Section

This section is used to configure the browser size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

28.1.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

28.1.3 Element Section

This section is contains properties that are specific to elements of type browser.

Property	Description
name	The name of the element. Displayed in the project browser.
url	A URL. Also known as a Web Address, it points to a network accessible resource.

29 Element: Button

This element is a clickable button whose behavior can be configured using low-code ActionScript.

Buttons are among the most useful and common elements in an app.

29.1 Events

Buttons have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks or touches the button.

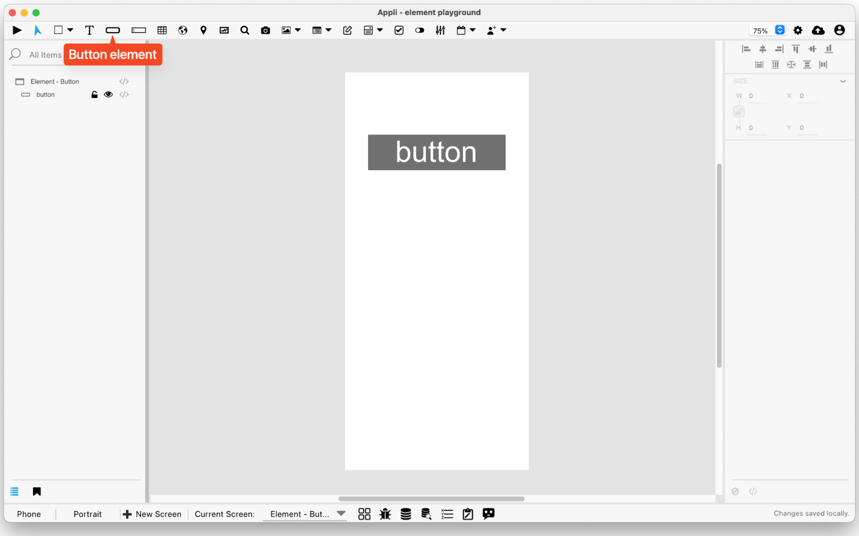


Figure 189: Sample Button

29.2 Properties

29.2.1 Size Section

This section is used to configure the button size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.

Property	Description
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

29.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

29.2.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the button.

Property	Description
textFace	The font used in the element's text.
textSize	The text size used in the element's text.
textStyle	The style used in the element's text. It can be bold, italic, underlined, and strikethrough.
textAlign	The text alignment used for the element's text. It can be aligned to the left, right, or center.

29.2.4 Appearance Section

This section is used to configure how the button looks.

Property	Description
blendLevel	How transparent the element is.
foregroundColor	The color used in text.
showBackground	If the element background should be opaque or transparent.
backgroundColor	The color used for the element's background.
roundRadius	How round the corners of the element are.
showBorder	Turns the visibility of the element's borders on or off.
dropShadow	The drop shadow for the element.
innerShadow	The inner shadow for the element.
outerGlow	The outer glow for the element.
innerGlow	The inner glow for the element.
lockIconRatio	Locks the placement of the icon in relationship with the dimensions of the element.
lineSize	the size of a line in pixels.
borderColor	the color used for the border of the element.

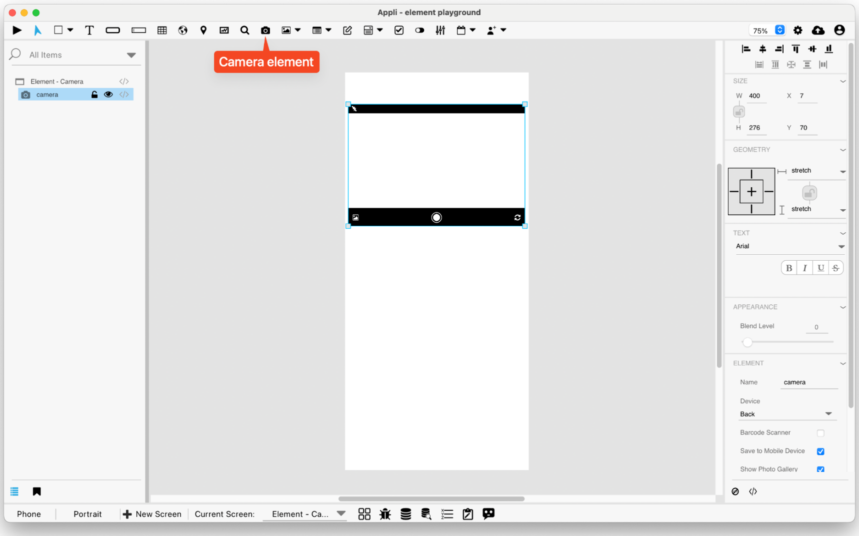
29.2.5 Element Section

This section is contains properties that are specific to elements of type button.

Property	Description
name	The name of the element. Displayed in the project browser.
label	The text label for the element.
layout	Element-specific property that configures how it should be displayed.
lowCode	ActionScript that controls the behavior of the element.

30 Element: Camera

This element displays the content of a camera. The camera can be a built-in camera or some attached webcam. It can also be used to scan barcodes and QR codes.



SIZE
W 400 X 7
H 276 Y 70

GEOMETRY
stretch
stretch

TEXT
Arial

APPEARANCE
Blend Level 0

ELEMENT
Name camera
Device Back
Barcode Scanner ☐
Save to Mobile Device ☑
Show Photo Gallery ☑

Figure 190: Sample Camera

The camera element shows as a placeholder while in edit mode and will only turn on the camera or scanner when in play mode.

30.1 Extra configuration

Using the no-code properties, the developer can configure which element receives the output from the camera element.

30.2 Properties

30.2.1 Size Section

This section is used to configure the camera size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.

Property	Description
height	The value in pixels representing the distance between the bottom side of the element and its top side.

30.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

30.2.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the camera.

Property	Description
textFace	The font used in the element's text.
textStyle	The style used in the element's text. It can be bold, italic, underlined, and strikethrough.

30.2.4 Appearance Section

This section is used to configure how the camera looks.

Property	Description
blendLevel	How transparent the element is.

30.2.5 Element Section

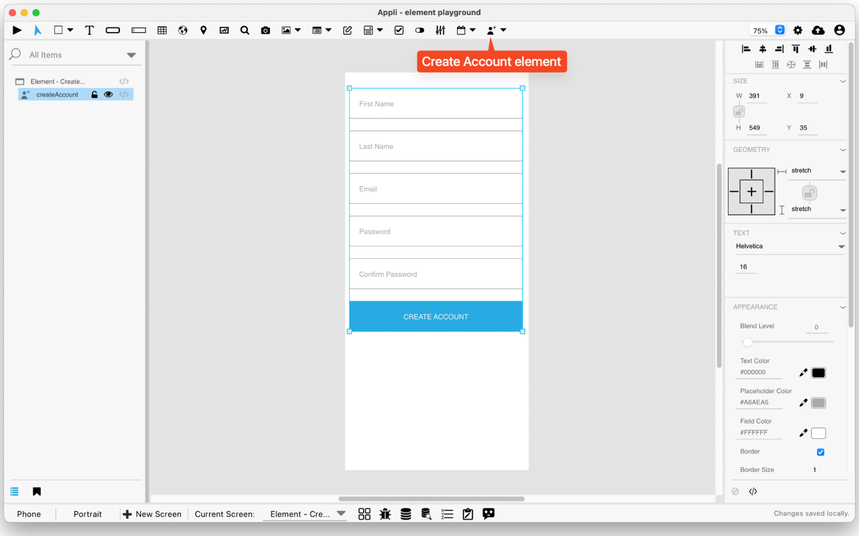
This section is contains properties that are specific to elements of type camera.

Property	Description
name	The name of the element. Displayed in the project browser.
device	selects which camera device should be used for the camera element.
barcodeScanner	toggles if the camera should behave like a barcode scanner or not.
flashMode	configures the behavior of the flash for the camera element.
focusMode	configures the focus for the camera element.

31 Element: Create Account

This element is a form that creates accounts. It has all the necessary fields and buttons for account creation already in place.

The fields are: first name, last name, email, password, and a password confirmation dialog. If your needs are different, you can roll your own form very easily.


Create Account element

SIZE
W 391 X 9
H 549 Y 35

GEOMETRY
stretch
stretch

TEXT
Helvetica
16

APPEARANCE
Blend Level 0
Text Color #000000
Placeholder Color #A6AEA5
Field Color #FFFFFF
Border ☑
Border Size 1

Changes saved locally.

Figure 191: Sample “Create Account” element

31.1 User management in Appli

Every Appli project has its own database. Besides the tables created by the developer, all projects have a `cdbUser` table that is used to keep user accounts.

Use elements such as `Create Account`, `Login`, and the actions under the `User Accounts` category to craft your own interface to handle user management workflows.

The User Management Tutorial explores the topic in more depth and is recommended reading.

31.2 Events

Create account forms have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks the *Create Account* button.

31.3 Properties

31.3.1 Size Section

This section is used to configure the create-account size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

31.3.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.

Property	Description
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

31.3.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the create-account.

Property	Description
textFace	The font used in the element's text.
textSize	The text size used in the element's text.

31.3.4 Appearance Section

This section is used to configure how the create-account looks.

Property	Description
blendLevel	How transparent the element is.
gap	the distance between various parts of an element. For example, in a field form it is the distance between fields.
fieldHeight	how tall the field is in pixels.
textColor	the color of the text in the element.
placeholderColor	The color used for placeholder text.
fieldColor	The color used for the background of a field.
showBorder	Turns the visibility of the element's borders on or off.
lineSize	the size of a line in pixels.
lineColor	the color of the line.
buttonHeight	how tall is the button.
buttonColor	the color for a button.
buttonLabelColor	the color used for the text label in a button.

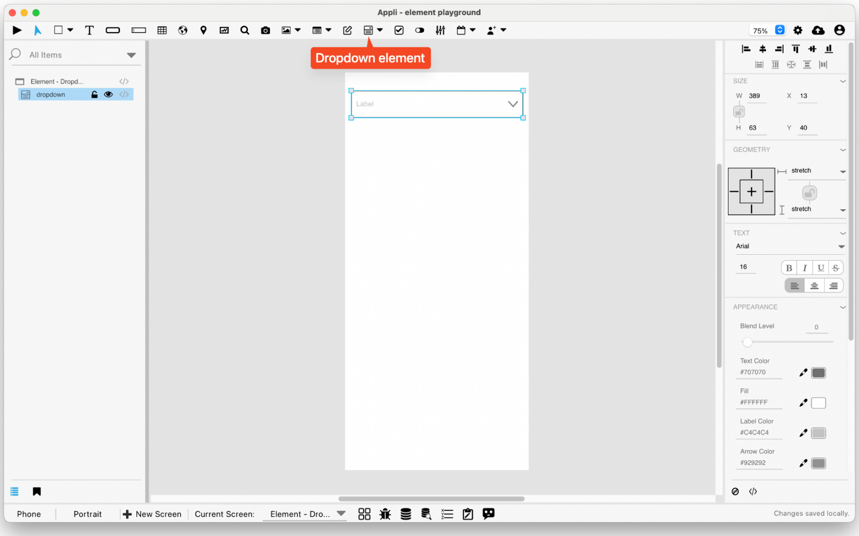
31.3.5 Element Section

This section is contains properties that are specific to elements of type create-account.

Property	Description
name	The name of the element. Displayed in the project browser.
lowCode	ActionScript that controls the behavior of the element.

32 Element: Dropdown

This element holds a menu. Clicking the element causes a scrolling list of options to open. Selecting an item will close back the dropdown menu.



W 389
X 13
H 63
Y 40
stretch
stretch
Arial
16
Blend Level 0
Text Color #707070
Fill #FFFFFF
Label Color #C4C4C4
Arrow Color #929292

Figure 192: Sample Dropdown

32.1 Events

Dropdowns have a *mouseUp* event that can be configured using low-code. This event is triggered when the user selects one of the available options.

32.2 Low code

When changing the `options` property of a dropdown menu using lowcode, you must use double pipes `||` to seperate the label of the option and the value of the option.

```
LABEL 1||VALUE 1
LABEL 2||VALUE 2
```

If the value of an option is empty, the label is also used as the value. You must use the Refresh Dropdown function to update the changes.

32.3 Properties

32.3.1 Size Section

This section is used to configure the dropdown size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

32.3.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

32.3.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the dropdown.

Property	Description
textFace	The font used in the element's text.

Property	Description
textSize	The text size used in the element's text.
textStyle	The style used in the element's text. It can be bold, italic, underlined, and strikethrough.
textAlign	The text alignment used for the element's text. It can be aligned to the left, right, or center.

32.3.4 Appearance Section

This section is used to configure how the dropdown looks.

Property	Description
blendLevel	How transparent the element is.
textColor	the color of the text in the element.
backgroundColor	The color used for the element's background.
labelColor	The color used for the text in the label of the element.
arrowColor	The color used for the arrow inside the element.
showBorder	Turns the visibility of the element's borders on or off.
lineSize	the size of a line in pixels.
lineColor	the color of the line.

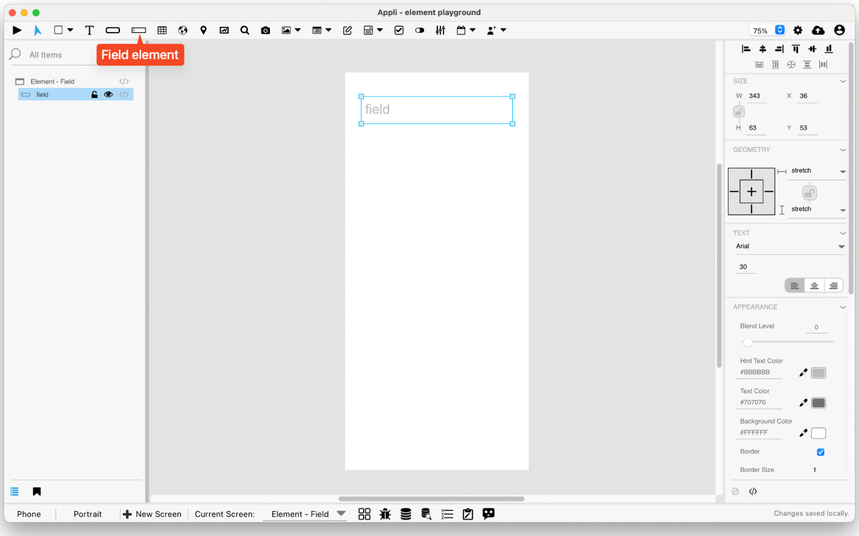
32.3.5 Element Section

This section is contains properties that are specific to elements of type dropdown.

Property	Description
name	The name of the element. Displayed in the project browser.
options	A list of options used by the element.
optionsToDisplay	How many options should be displayed by the element. If there are more options available than the value of this property, the element will display a scrollbar.
label	The text label for the element.
lowCode	ActionScript that controls the behavior of the element.

33 Element: Field

This element allows the user to enter or edit text.



W 343
X 36
H 63
Y 53
stretch
stretch
Arial
30
Blend Level 0
Hint Text Color #BBBBBB
Text Color #707070
Background Color #FFFFFF
Border ☑
Border Size 1

Figure 193: Sample Field

33.1 Events

Fields have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks or touches the field.

33.2 Properties

33.2.1 Size Section

This section is used to configure the field size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.

Property	Description
height	The value in pixels representing the distance between the bottom side of the element and its top side.

33.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

33.2.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the field.

Property	Description
textFace	The font used in the element's text.
textSize	The text size used in the element's text.
textAlign	The text alignment used for the element's text. It can be aligned to the left, right, or center.

33.2.4 Appearance Section

This section is used to configure how the field looks.

Property	Description
blendLevel	How transparent the element is.
hintTextColor	the color used by the hint text.
textColor	the color of the text in the element.
backgroundColor	The color used for the element's background.

Property	Description
showBorder	Turns the visibility of the element's borders on or off.
lineSize	the size of a line in pixels.
lineColor	the color of the line.

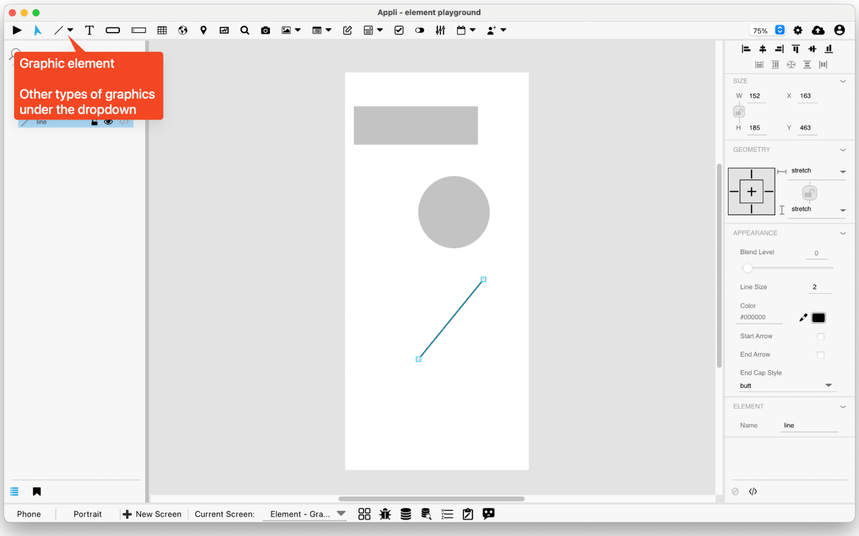
33.2.5 Element Section

This section is contains properties that are specific to elements of type field.

Property	Description
name	The name of the element. Displayed in the project browser.
hintText	the hint text.
useNative	Replaces the element with the native version of that element for the running platform. Used by text fields.
multiLine	Configures the text field to accept multiline text instead of single line input.
passwordField	marks the field as a password entry. This masks the input so that it is not readable.
keyboardType	configure what kind of keyboard is to be used when entering data in the element.
autoCorrectionType	configures if the element should use auto-correction features.
autoCapitalizationType	configures if the element should use auto-capitalize it's data.
lowCode	ActionScript that controls the behavior of the element.

34 Element: Graphic

This element represents a graphic on the screen. Appli has distinct controls on the toolbar to make the most common graphics: rectangles, ellipses, and lines.



Graphic element

Other types of graphics under the dropdown

SIZE
W 152 X 163
H 185 Y 463

GEOMETRY
stretch
stretch

APPEARANCE
Blend Level 0
Line Size 2
Color #000000
Start Arrow □
End Arrow □
End Cap Style butt

ELEMENT
Name line

Figure 194: Sample Graphic

34.1 Events

Graphics have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks or touches the graphic.

34.2 Properties

34.2.1 Size Section

This section is used to configure the graphic size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

34.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

34.2.3 Appearance Section

This section is used to configure how the graphic looks.

Property	Description
blendLevel	How transparent the element is.
opaque	if the element is opaque or if it's background is transparent.
foregroundColor	The color used in text.
showBackground	If the element background should be opaque or transparent.
backgroundColor	The color used for the element's background.
roundRadius	How round the corners of the element are.
showBorder	Turns the visibility of the element's borders on or off.
dropShadow	The drop shadow for the element.
innerShadow	The inner shadow for the element.
outerGlow	The outer glow for the element.
innerGlow	The inner glow for the element.
lockRatio	Locks the aspect ratio of the element.
fillGradient	The gradient color used to fill the element.

34.2.4 Element Section

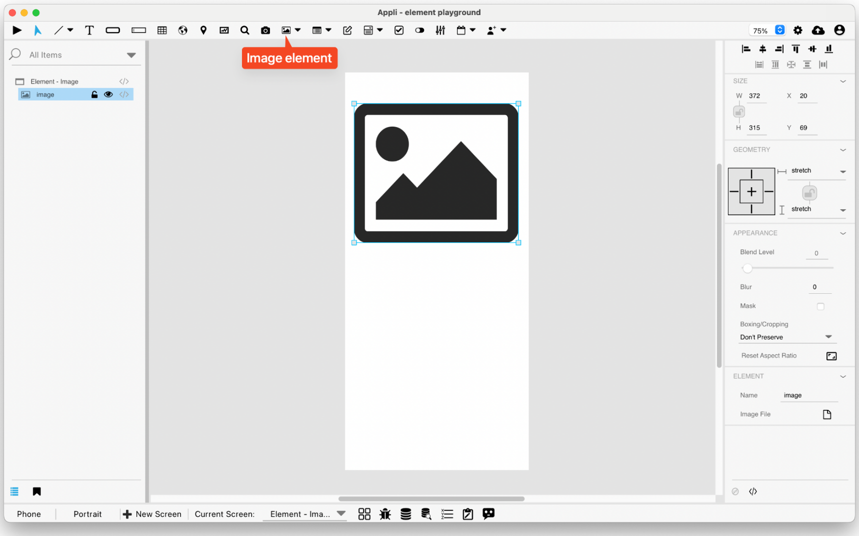
This section is contains properties that are specific to elements of type graphic.

Property	Description
name	The name of the element. Displayed in the project browser.
lowCode	ActionScript that controls the behavior of the element.

35 Element: Image

This element contains an image.

Some properties are only available if you apply a mask to the image.



SIZE

W 372 X 20

H 315 Y 69

GEOMETRY

stretch

stretch

APPEARANCE

Blend Level 0

Blur 0

Mask □

Boxing/Cropping

Don't Preserve

Reset Aspect Ratio

ELEMENT

Name image

Image File

Figure 195: Sample Image

35.1 Events

Images have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks or touches the image.

35.2 Properties

35.2.1 Size Section

This section is used to configure the image size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.

Property	Description
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

35.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

35.2.3 Appearance Section

This section is used to configure how the image looks.

Property	Description
blendLevel	How transparent the element is.
blur	how much blur is applied to the element content.
useMask	if a mask should be applied to the element.
shapeStyle	The shape to be used for masking the element.
lineSize	the size of a line in pixels.
lineColor	the color of the line.
preserveAspectRatio	If the element should preserve the aspect ratio of it's data.

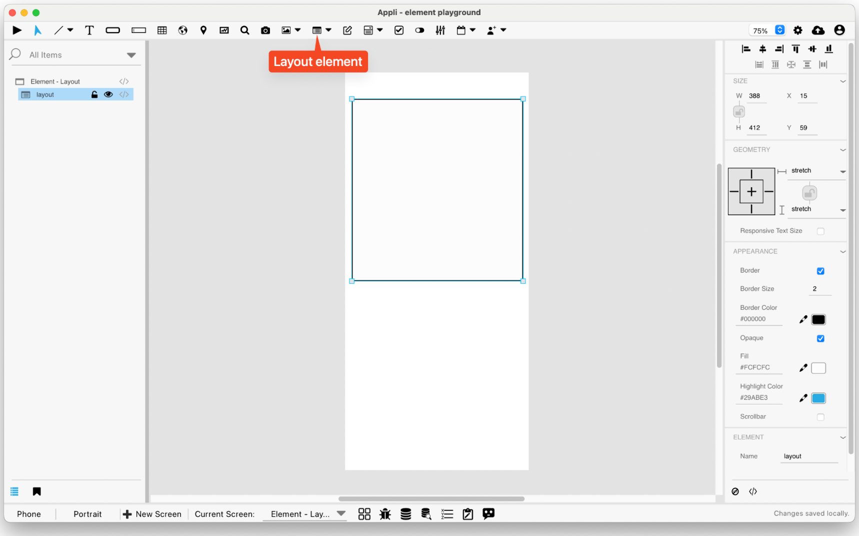
35.2.4 Element Section

This section is contains properties that are specific to elements of type image.

Property	Description
name	The name of the element. Displayed in the project browser.
imageFile	the file containing the image to be used in the element.
lowCode	ActionScript that controls the behavior of the element.

36 Element: Layout

This element collects other elements as a group.



W 388 X 15
H 412 Y 59

stretch
stretch
Responsive Text Size □

Border ☑
Border Size 2
Border Color #000000
Opaque ☑
Fill #FCFCFC
Highlight Color #29ABE3
Scrollbar □

Name layout

Figure 196: Sample Layout

> Be aware that a layout element is empty when first drawn on the screen. You need to place the elements you want inside it.

36.1 No code

There are extra options available for Layout elements that can be configured using *no code*.

One can use records on a database to configure the elements inside a Layout.

36.2 Actions

These are the *low code* actions available for Layout elements.

36.2.1 Refresh the layout

Used to refresh the records of a Layout element.

> Example: after changing the records in a database, you can use this action to update the elements inside the Layout to reflect the current database state.

36.2.2 Set variable from Context

Each element inside a Layout has a *Context* (i.e. a reference to the *RecordID*). Using this action, one can update a variable based on a value from that context.

36.3 Events

Layouts have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks or touches the layout.

36.4 Properties

36.4.1 Size Section

This section is used to configure the layout size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

36.4.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.

Property	Description
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

36.4.3 Appearance Section

This section is used to configure how the layout looks.

Property	Description
showBorder	Turns the visibility of the element's borders on or off.
borderColor	the color used for the border of the element.
lineSize	the size of a line in pixels.
opaque	if the element is opaque or if it's background is transparent.
backgroundColor	The color used for the element's background.
showScrollbar	If the scrollbar should be visible or not.

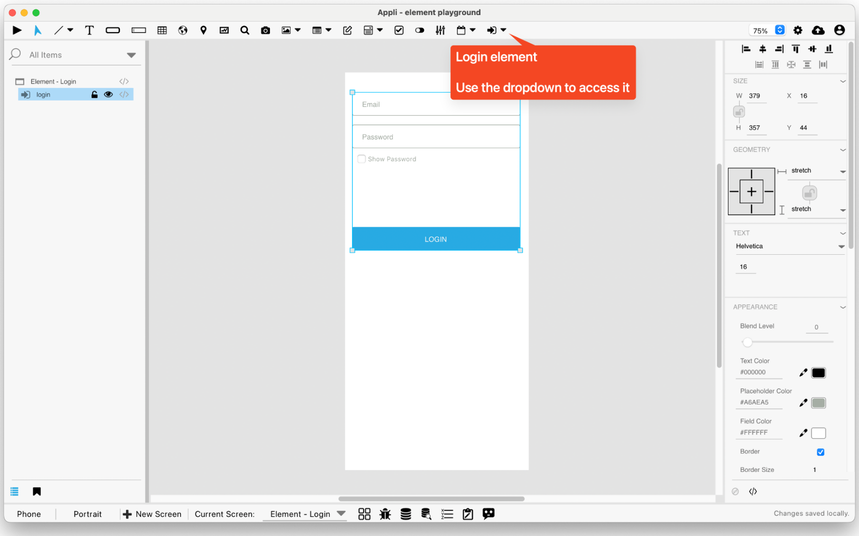
36.4.4 Element Section

This section is contains properties that are specific to elements of type layout.

Property	Description
name	The name of the element. Displayed in the project browser.
multipleRows	If the element should contain multiple rows.
bottomMargin	the value in pixels of the bottom margin.

37 Element: Login

This element represents a login form. It contains the most common fields that are needed to perform a login operation: email and password.



Login element

Use the dropdown to access it

SIZE
W 379 X 16
H 357 Y 44

GEOMETRY
stretch
stretch

TEXT
Helvetica
16

APPEARANCE
Blend Level 0
Text Color #000000
Placeholder Color #A6AEA5
Field Color #FFFFFF
Border ☑
Border Size 1

Figure 197: Sample Login

37.1 User management in Appli

Every Appli project has its own database. Besides the tables created by the developer, all projects have a `cdbUser` table that is used to keep user accounts.

Use elements such as `Create Account`, `Login`, and the actions under the `User Accounts` category to craft your own interface to handle user management workflows.

The User Management Tutorial explores the topic in more depth and is recommended reading.

37.2 Events

Login elements have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks or touches the Login button.

37.3 Properties

37.3.1 Size Section

This section is used to configure the login size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

37.3.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

37.3.3 Appearance Section

This section is used to configure how the login looks.

Property	Description
blendLevel	How transparent the element is.
fieldHeight	how tall the field is in pixels.
textColor	the color of the text in the element.
placeholderColor	The color used for placeholder text.
fieldColor	The color used for the background of a field.
showBorder	Turns the visibility of the element's borders on or off.
lineSize	the size of a line in pixels.
borderColor	the color used for the border of the element.
buttonHeight	how tall is the button.

Property	Description
buttonColor	the color for a button.
buttonLabelColor	the color used for the text label in a button.

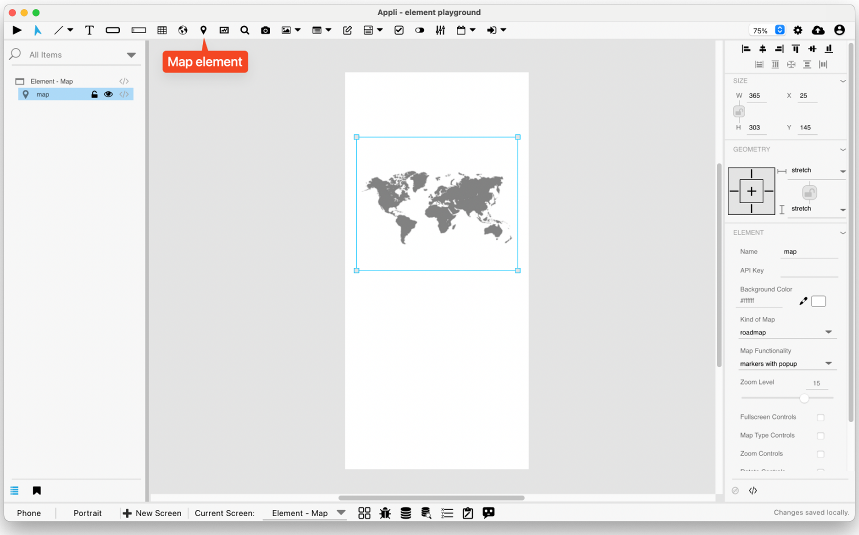
37.3.4 Element Section

This section is contains properties that are specific to elements of type login.

Property	Description
name	The name of the element. Displayed in the project browser.

38 Element: Map

This element contains a map with markers.



SIZE
W 365 X 25
H 303 Y 145

GEOMETRY
stretch
stretch

ELEMENT
Name: map
API Key:
Background Color: #ffffff
Kind of Map: roadmap
Map Functionality: markers with popup
Zoom Level: 15
Fullscreen Controls □
Map Type Controls □
Zoom Controls □

Figure 198: Sample Map

Attention: You need a Google Maps API Key to use the map element. Check this blog post to see how to obtain one.

38.1 Events

Maps have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks or touches the map.

38.2 Properties

38.2.1 Size Section

This section is used to configure the map size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

38.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

38.2.3 Element Section

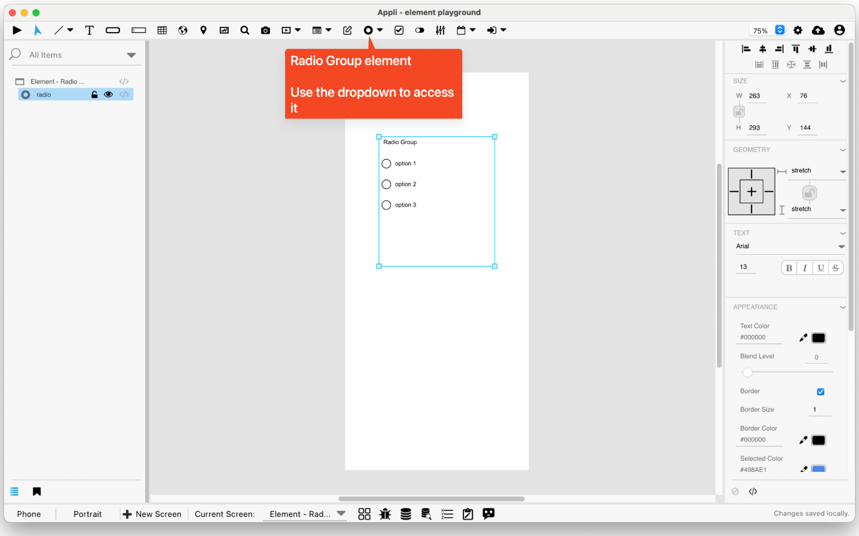
This section is contains properties that are specific to elements of type map.

Property	Description
name	The name of the element. Displayed in the project browser.

Property	Description
Markers	a collection of geolocation markers to be added to a Map element.
APIKey	A Google Maps API Key.
lowCode	ActionScript that controls the behavior of the element.

39 Element: Radio Group

This element is a group of radio buttons. The user can select only one radio button in a group.



Radio Group element

Use the dropdown to access it

W 263 X 76

H 293 Y 144

stretch

stretch

Arial

13

Text Color #000000

Blend Level 0

Border ☑

Border Size 1

Border Color #000000

Selected Color #498AE1

Figure 199: Sample Radio Group

39.1 Events

Radio groups have a *mouseUp* event that can be configured using low-code. This event is triggered when the user selects one of the options.

39.2 Properties

39.2.1 Size Section

This section is used to configure the radio-group size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

39.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

39.2.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the radio-group.

Property	Description
textFace	The font used in the element's text.
textSize	The text size used in the element's text.
textStyle	The style used in the element's text. It can be bold, italic, underlined, and strikethrough.

39.2.4 Appearance Section

This section is used to configure how the radio-group looks.

Property	Description
textColor	the color of the text in the element.
blendLevel	How transparent the element is.
hiliteColor	The color of the hilited state of the element.

39.2.5 Element Section

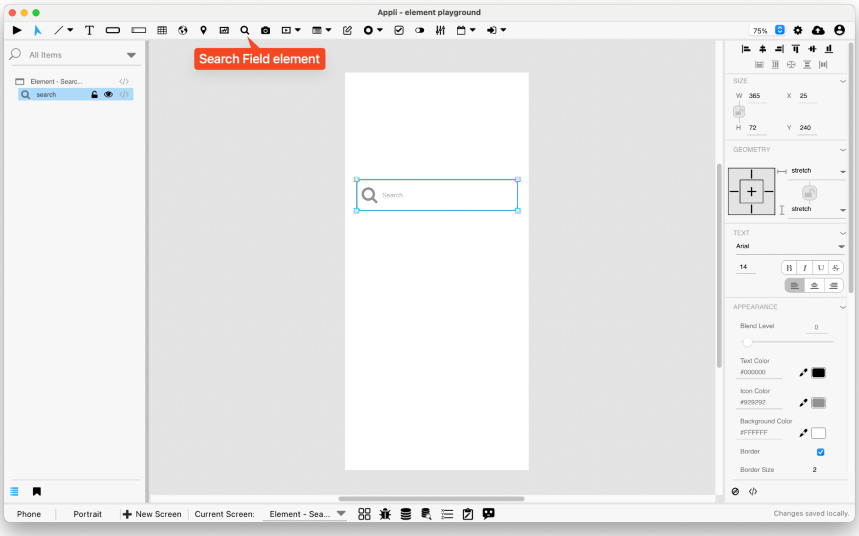
This section is contains properties that are specific to elements of type radio-group.

Property	Description
name	The name of the element. Displayed in the project browser.
useIcons	if the element should use icons.
options	A list of options used by the element.
label	The text label for the element.
showLabel	configures if the label is visible.
orientation	The element's orientation.
lowCode	ActionScript that controls the behavior of the element.

40 Element: Search Field

This element represents a search box.

The element has two labels, one is used for the focused state and the other for the unfocused state.



SIZE
W 365 X 25
H 72 Y 240

GEOMETRY
stretch
stretch

TEXT
Arial
14

APPEARANCE
Blend Level 0
Text Color #000000
Icon Color #929292
Background Color #FFFFFF
Border ☑
Border Size 2

Figure 200: Sample Search Field

40.1 Events

Search elements have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks or touches the magnifying glass button inside the search element.

40.2 Properties

40.2.1 Size Section

This section is used to configure the search-field size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

40.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

40.2.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the search-field.

Property	Description
textFace	The font used in the element's text.
textSize	The text size used in the element's text.
textStyle	The style used in the element's text. It can be bold, italic, underlined, and strikethrough.
textAlign	The text alignment used for the element's text. It can be aligned to the left, right, or center.

40.2.4 Appearance Section

This section is used to configure how the search-field looks.

Property	Description
blendLevel	How transparent the element is.
textColor	the color of the text in the element.
iconColor	the color of the icon.
backgroundColor	The color used for the element's background.
showBorder	Turns the visibility of the element's borders on or off.
lineSize	the size of a line in pixels.
lineColor	the color of the line.

Property	Description
showLabel1	for elements with more than one label, this toggles the visibility of the first label.
label1Color	for elements with more than one label, this specifies the color for the first label.
showLabel2	for elements with more than one label, this toggles the visibility of the second label.
label2Color	for elements with more than one label, this specifies the color for the second label.

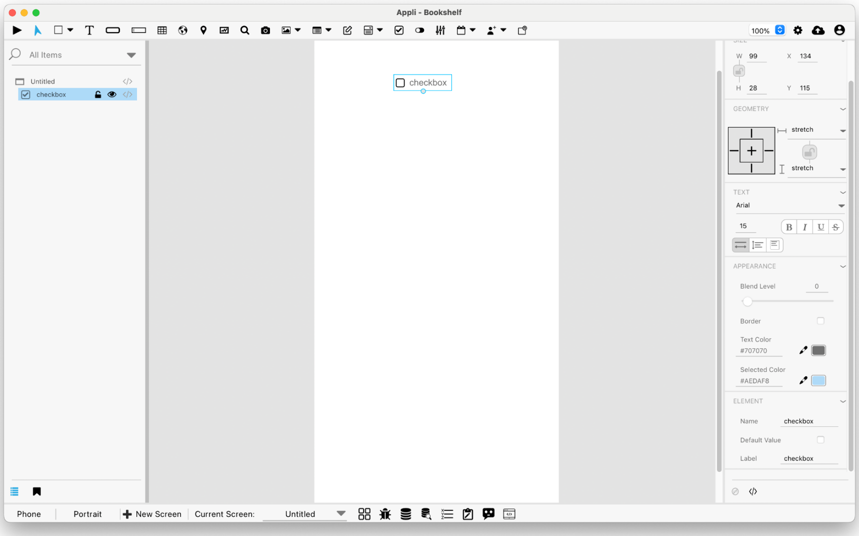
40.2.5 Element Section

This section is contains properties that are specific to elements of type searchfield.

Property	Description
name	The name of the element. Displayed in the project browser.
label1	for elements with more than one label, this specifies the content for the first label.
label2	for elements with more than one label, this specifies the content for the second label.
iconPlacement	how the icon should be placed inside the element.
lowCode	ActionScript that controls the behavior of the element.

41 Element: Checkbox

This element is a checkbox that can be toggled on or off.



W 99 X 134
H 28 Y 115

GEOMETRY
stretch
stretch

TEXT
Arial
15

APPEARANCE
Blend Level 0
Border □
Text Color #707070
Selected Color #AEDAF8

ELEMENT
Name checkbox
Default Value □
Label checkbox

Figure 201: Sample Switch

41.1 Events

Switches have a *mouseUp* event that can be configured using low-code. This event is triggered when the user toggles the checkbox.

Appli treats the checked and unchecked state of a checkbox as true or false value.

41.2 Properties

41.2.1 Size Section

This section is used to configure the checkbox size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.

Property	Description
height	The value in pixels representing the distance between the bottom side of the element and its top side.

41.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

41.2.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the checkbox.

Property	Description
textFace	The font used in the element's text.
textSize	The text size used in the element's text.
textStyle	The style used in the element's text. It can be bold, italic, underlined, and strikethrough.

41.2.4 Appearance Section

This section is used to configure how the checkbox looks.

Property	Description
blendLevel	How transparent the element is.
showBorder	Turns the visibility of the element's borders on or off.
textColor	the color of the text in the element.
hiliteColor	The color of the hilited state of the element.

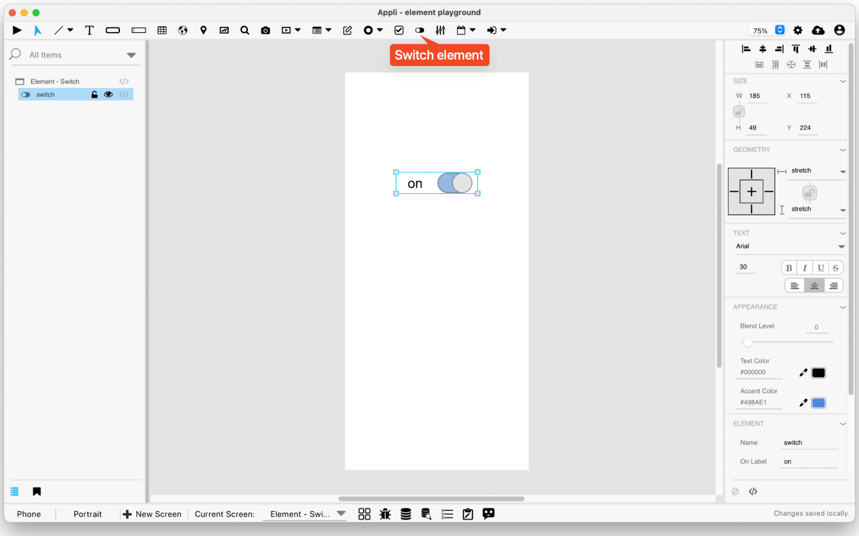
41.2.5 Element Section

This section is contains properties that are specific to elements of type checkbox.

Property	Description
name	The name of the element. Displayed in the project browser.
defaultValue	Default Value
label	The text label for the element.
lowCode	ActionScript that controls the behavior of the element.

42 Element: Switch

This element is a switch that can be toggled on or off.



W 185
X 115
H 49
Y 224
stretch
stretch
Arial
30
Blend Level 0
Text Color #000000
Accent Color #498AE1
Name switch
On Label on
Current Screen: Element - Swi...

Figure 202: Sample Switch

42.1 Events

Switches have a *mouseUp* event that can be configured using low-code. This event is triggered when the user toggles the switch.

Appli treats the on and off state of a switch a true or false value.

42.2 Properties

42.2.1 Size Section

This section is used to configure the switch size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

42.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

42.2.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the switch.

Property	Description
textFace	The font used in the element’s text.
textSize	The text size used in the element’s text.
textStyle	The style used in the element’s text. It can be bold, italic, underlined, and strikethrough.

42.2.4 Appearance Section

This section is used to configure how the switch looks.

Property	Description
blendLevel	How transparent the element is.
textColor	the color of the text in the element.
hiliteColor	The color of the hilited state of the element.

42.2.5 Element Section

This section is contains properties that are specific to elements of type switch.

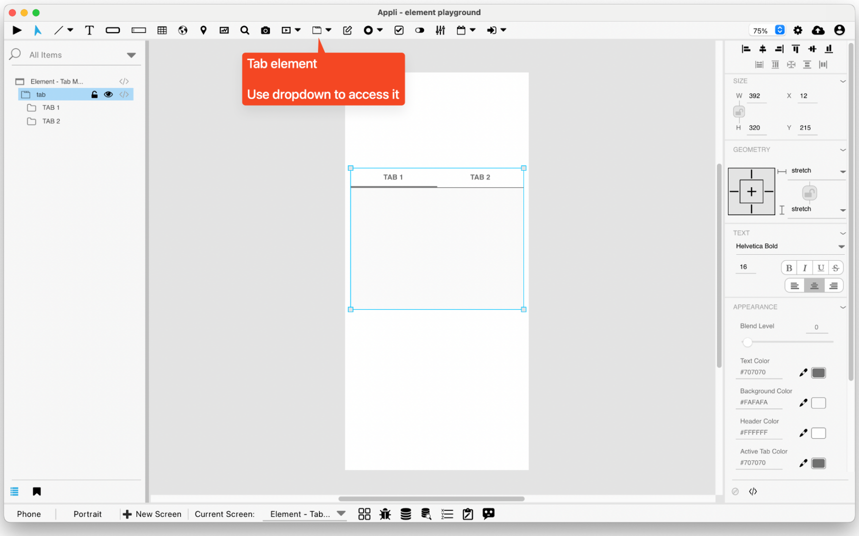
Property	Description
name	The name of the element. Displayed in the project browser.
trueLabel	the text for the true state of the element.
falseLabel	the text for the false state of the element.
value	if the element should show a value.
showLabel	configures if the label is visible.
lowCode	ActionScript that controls the behavior of the element.

43 Element: Tab Menu

This element is a tab menu. It can contain multiple tabs each with their own collection of elements.

Each tab contains their own elements. To edit a specific tab, double-click the tab you want to change or change the `selectedTab` property using the property inspector. Then add elements to it by dragging and dropping them on top of the tab.

To change a tab name, select the tab using the `selectedTab` property and change the `tabName` property. That property always display the name of the selected tab.



Tab element

Use dropdown to access it

SIZE
W 392 X 12
H 320 Y 215

GEOMETRY
stretch
stretch

TEXT
Helvetica Bold
16

APPEARANCE
Blend Level 0
Text Color #707070
Background Color #FAFAFA
Header Color #FFFFFF
Active Tab Color #707070

Figure 203: Sample Tab Menu

43.1 Events

Tab menus have a *mouseUp* event that can be configured using low-code. This event is triggered when the user changes the selected tab.

43.2 Properties

43.2.1 Size Section

This section is used to configure the tab-menu size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

43.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

43.2.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the tab-menu.

Property	Description
textFace	The font used in the element's text.
textSize	The text size used in the element's text.
textStyle	The style used in the element's text. It can be bold, italic, underlined, and strikethrough.
textAlign	The text alignment used for the element's text. It can be aligned to the left, right, or center.

43.2.4 Appearance Section

This section is used to configure how the tab-menu looks.

Property	Description
headerTextColor	the color used by the header text.
backgroundColor	The color used for the element's background.
headerColor	the color used by the header background.
blendLevel	How transparent the element is.
activeTabColor	the color used to hilite which tab is selected.

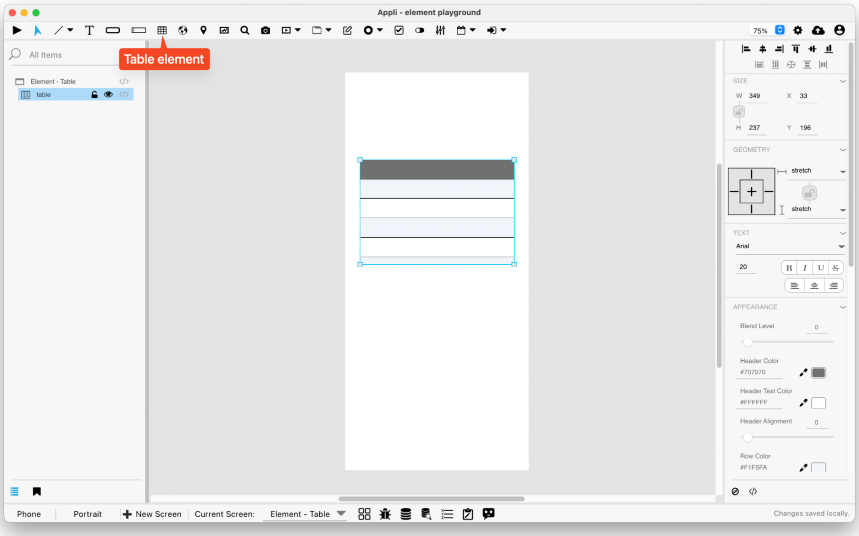
43.2.5 Element Section

This section is contains properties that are specific to elements of type tab-menu.

Property	Description
name	The name of the element. Displayed in the project browser.
numberOfTabs	how many tabs the tab menu element contains.
selectedTab	which of tabs is the active tab.
tabName	the name of the selected tab.

44 Element: Table

This element contains a table of records. Tables can be used to display and edit records. The `canEditData` property configures if the table is read-only or not.



Appli - element playground

W 349
X 33
H 237
Y 196

stretch
stretch

Arial
20

Blend Level 0
Header Color #707070
Header Text Color #FFFFFF
Header Alignment 0
Row Color #F1F5FA

Changes saved locally.

Figure 204: Sample Table

44.1 Table Setup

Use *no-code* to configure the table. Tables can be local, cloud, of hybrid. Using the *no-code* select which table and keys should be displayed on the table.

44.2 Actions

These are the *low code* actions available for table elements.

44.2.1 Refresh the Table

Used to refresh the records of a table element.

Example: after changing the records in a database, you can use this action to update the elements inside the table to reflect the current database state.

44.2.2 Count Displayed Records

Counts the amount of records in the table and place that value in a variable or another element.

44.3 Events

Tab menus have a *mouseUp* event that can be configured using low-code. This event is triggered when the user changes the selected tab.

44.4 Properties

44.4.1 Size Section

This section is used to configure the table size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

44.4.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

44.4.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the table.

Property	Description
textFace	The font used in the element's text.
textSize	The text size used in the element's text.
textStyle	The style used in the element's text. It can be bold, italic, underlined, and strikethrough.
textAlign	The text alignment used for the element's text. It can be aligned to the left, right, or center.

44.4.4 Appearance Section

This section is used to configure how the table looks.

Property	Description
blendLevel	How transparent the element is.
headerColor	the color used by the header background.
headerTextColor	the color used by the header text.
headerVerticalAlign	the vertical alignment of the header.
rowColor1	For elements with alternating row colors, this is the color for the odd rows.
rowColor2	For elements with alternating row colors, this is the color for the even rows.
rowHeight	the height for the row.
rowVerticalAlign	the vertical alignment for the row.
bodyTextColor	the color used by the body text.
columnAlign	the alignment for the column.
showBorder	Turns the visibility of the element's borders on or off.
lineColor	the color of the line.
hiliteColor	The color of the hilited state of the element.
useRowColorForEditCell	if the element should use the row color for the edit cell background.
editCellTextColor	the color used by the text inside an edit cell.
editCellHiliteColor	the hilite color used by edit cells.

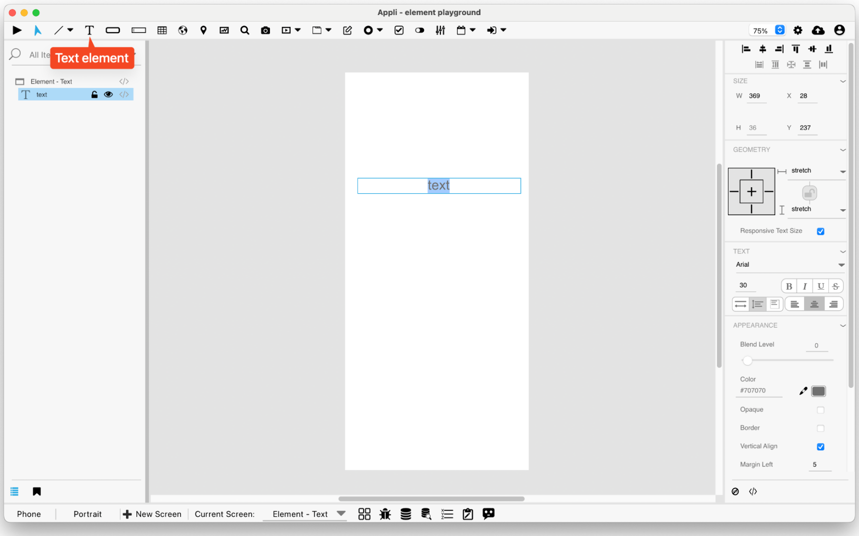
44.4.5 Element Section

This section is contains properties that are specific to elements of type table.

Property	Description
name	The name of the element. Displayed in the project browser.
canEditData	if the data shown in the element is editable.
adjustColumnsOnPlayer	if the columns should be adjusted depending on the player size.

45 Element: Text

This element contains non-editable text.



W 369 X 28
H 36 Y 237
stretch
stretch
Responsive Text Size ☑
Arial
30
Blend Level 0
Color #707070
Opaque □
Border □
Vertical Align ☑
Margin Left 5

Figure 205: Sample Text

45.1 Text Setup

The contents of a text element can be set from a database record. Use the *no-code* interface to select the table and recordID used for the text element.

45.2 Actions

These are the *low code* actions available for text elements.

45.2.1 Format The Text

This action can be used to format the text. The current available format is *USD Currency*. The formatted text can be placed in a variable or inside an element.

45.3 Events

Text elements have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks or touches the element.

45.4 Properties

45.4.1 Size Section

This section is used to configure the text size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

45.4.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

45.4.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the text.

Property	Description
textFont	The font used in the element's text.
textSize	The text size used in the element's text.
textStyle	The style used in the element's text. It can be bold, italic, underlined, and strikethrough.
textAlign	The text alignment used for the element's text. It can be aligned to the left, right, or center.

45.4.4 Appearance Section

This section is used to configure how the text looks.

Property	Description
lockedText	Lock Text (prevents modification).
dynamicTextSize	changes the size of the fontface used for the text depending on the dimensions of the element.
dynamicHeight	change the height of the element based on the text content.
dynamicWidth	change the width of the element based on the text content.
dontWrap	prevents the text content from wrapping.
truncateWithEllipsis	truncates the text and places an ellipsis character at the end.
showScrollbar	If the scrollbar should be visible or not.
verticalAlign	if the element should be aligned vertically.
leftMargin	the value in pixels for the left margin.
rightMargin	the value in pixels for the right margin.
fixedLineHeight	if the element should use a fixed line height.
textHeight	the line height in pixels.
blendLevel	How transparent the element is.
textColor	the color of the text in the element.
opaque	if the element is opaque or if it's background is transparent.
showBorder	Turns the visibility of the element's borders on or off.

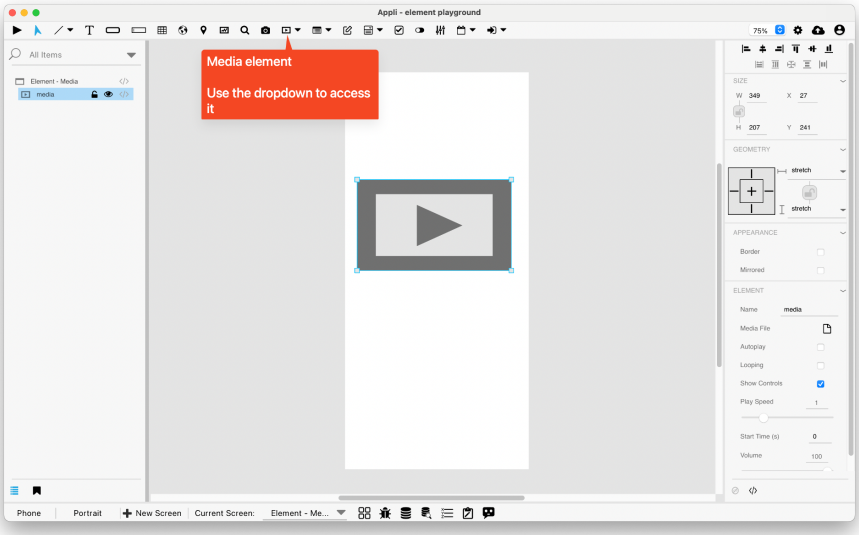
45.4.5 Element Section

This section is contains properties that are specific to elements of type text.

Property	Description
name	The name of the element. Displayed in the project browser.

46 Element: Media

This element displays content from a media file.



Media element

Use the dropdown to access it

SIZE
W 349 X 27
H 207 Y 241

GEOMETRY
stretch
stretch

APPEARANCE
Border ☐
Mirrored ☐

ELEMENT
Name media
Media File
Autoplay ☐
Looping ☐
Show Controls ☑
Play Speed 1
Start Time (s) 0
Volume 100

Figure 206: Sample Media

46.1 Events

Media elements have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks or touches the element.

46.2 Properties

46.2.1 Size Section

This section is used to configure the media size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.

Property	Description
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

46.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

46.2.3 Appearance Section

This section is used to configure how the media looks.

Property	Description
showBorder	Turns the visibility of the element's borders on or off.
mirrored	If the content of the element should be mirrored (flipped horizontally).

46.2.4 Element Section

This section is contains properties that are specific to elements of type media.

Property	Description
name	The name of the element. Displayed in the project browser.
mediaFile	The file used by the media element.

Property	Description
autoPlay	if the video should play automatically.
looping	if the video should restart automatically.
showController	if the video element should show user-accessible controls.

47 Element: Form

This element collects other elements as part of a form.

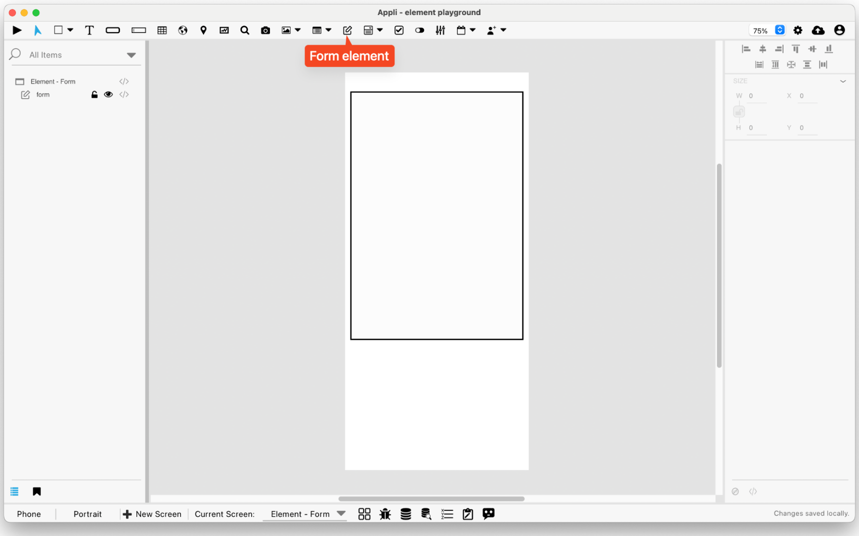


Figure 207: Sample Form

> Be aware that a form element is empty when first drawn on the screen. You need to place the elements you want inside it or use the form builder tool.

47.1 No code

Forms can be connected to Tables using *no-code*. This way they can be used to create or edit records.

47.2 Low code

There is a property that can be set using *low-code* called `dataRecordID`. Use the `set_property_from_variable` action to set this property and connect a form to a specific record in a table.

47.3 Actions

These are the *low code* actions available for Form elements.

47.3.1 Refresh the form

If the form is tied to a record in a table, this can be used to refresh the elements inside the form and update them to the new state of the record.

47.3.2 Submit Form To Database

Saves the data from the form to the associated table.

47.4 Events

Forms have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks or touches the element.

47.5 Properties

47.5.1 Size Section

This section is used to configure the form size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

47.5.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

47.5.3 Appearance Section

This section is used to configure how the form looks.

Property	Description
showBorder	Turns the visibility of the element's borders on or off.
borderColor	the color used for the border of the element.
lineSize	the size of a line in pixels.
opaque	if the element is opaque or if it's background is transparent.
backgroundColor	The color used for the element's background.

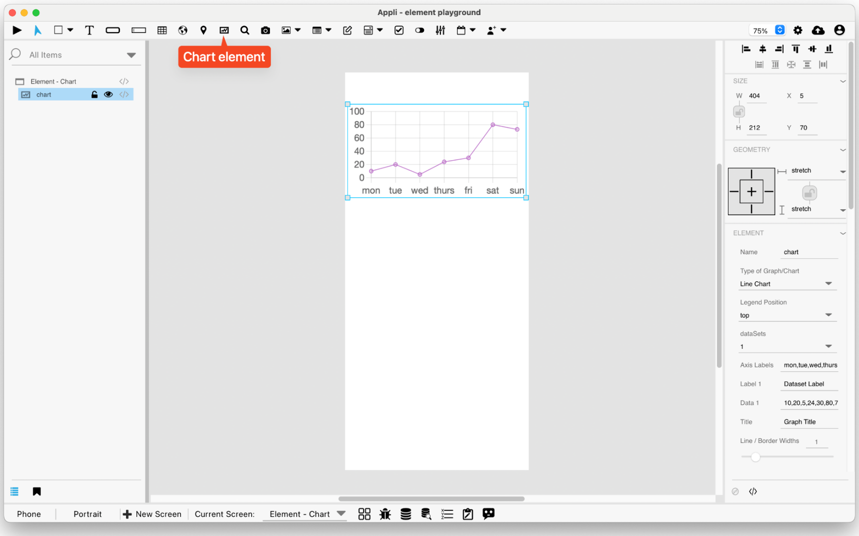
47.5.4 Element Section

This section is contains properties that are specific to elements of type form.

Property	Description
name	The name of the element. Displayed in the project browser.

48 Element: Chart

This element represents a chart on the screen.



W 404 X 5

H 212 Y 70

stretch

stretch

Name: chart

Type of Graph/Chart: Line Chart

Legend Position: top

dataSets: 1

Axis Labels: mon,tue,wed,thurs

Label 1: Dataset Label

Data 1: 10,20,5,24,30,80,7

Title: Graph Title

Line / Border Widths: 1

Figure 208: Sample Chart

48.1 Events

Charts have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks or touches the graphic.

48.2 Properties

48.2.1 Size Section

This section is used to configure the chart size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

48.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

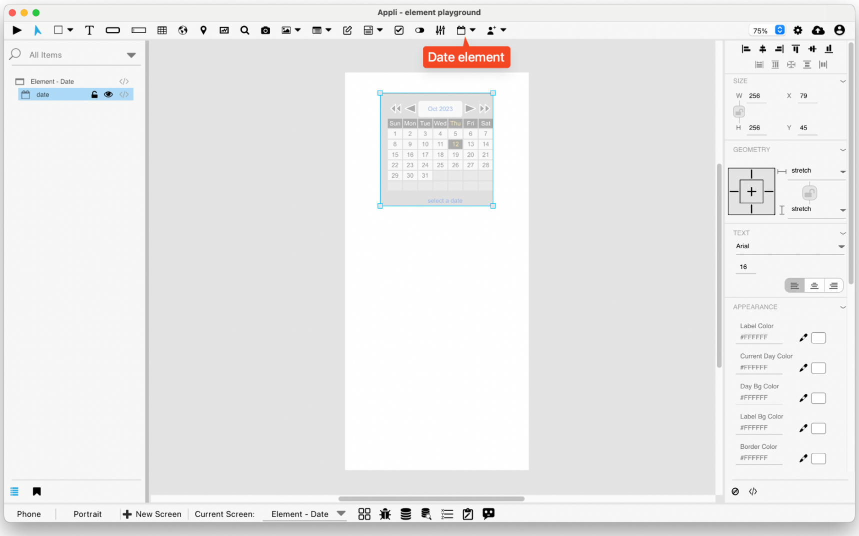
48.2.3 Element Section

This section is contains properties that are specific to elements of type chart.

Property	Description
name	The name of the element. Displayed in the project browser.
showLine	toggles the display of the line in a graph.
showXLines	Toggles the display of the vertical background lines in a graph.
showYLines	Toggles the display of the horizontal background lines in a graph.
opaque	if the element is opaque or if it's background is transparent.
makerColor	sets the color for markers in a graph.
markerShape	Configures which shape to use for markers in a graph.
axisData	The data to be used by the axis in a graph.
markerData	the data used to plot markers on a graph.
lowCode	ActionScript that controls the behavior of the element.

49 Element: Date

This element allows the user to display or pick a date. Use the property inspector in the playground to configure your time picker and use the date/time actions to show and hide them.



SIZE
W 256 X 79
H 256 Y 45

GEOMETRY
stretch
stretch

TEXT
Arial
16

APPEARANCE
Label Color #FFFFFF
Current Day Color #FFFFFF
Day Bg Color #FFFFFF
Label Bg Color #FFFFFF
Border Color #FFFFFF

Figure 209: Sample Date element

49.1 Events

Date elements have a *mouseUp* event that can be configured using low-code.

49.2 Properties

49.2.1 Size Section

This section is used to configure the date size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

49.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.

49.2.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the date.

Property	Description
textFace	The font used in the element's text.
textSize	The text size used in the element's text.
textAlign	The text alignment used for the element's text. It can be aligned to the left, right, or center.

49.2.4 Appearance Section

This section is used to configure how the date looks.

Property	Description
colorActive	Label Color
currentActiveVar	Current Day Color
colorBack	Day Bg Color
colorBackVar	Label Bg Color
borderColor	the color used for the border of the element.
colorEmpty	Blank Color
colorFore	Top Label Color
colorForeVar	Day Color
colorHilite	Highlight Color
colorWidgetBack	Bg Color
borderRoundRadius	Border Round Radius
borderWidth	Border Size

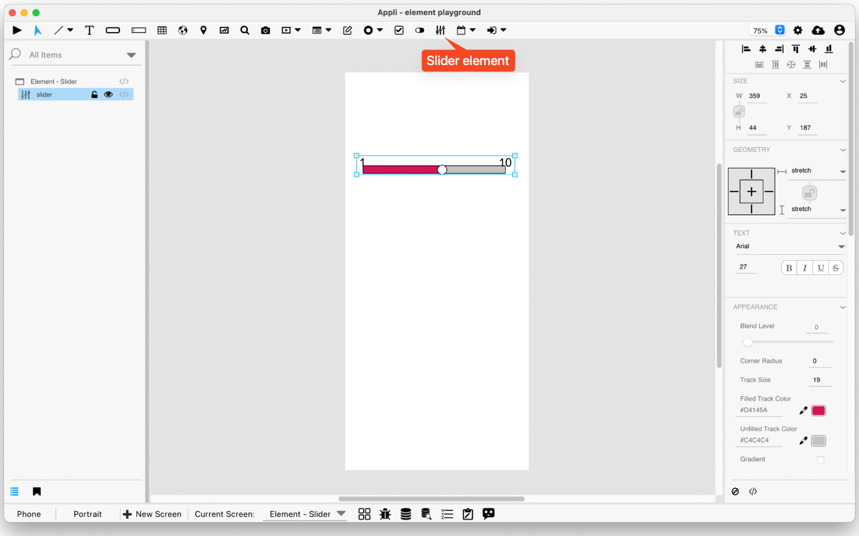
49.2.5 Element Section

This section is contains properties that are specific to elements of type date.

Property	Description
name	The name of the element. Displayed in the project browser.
textSize	The text size used in the element’s text.
dateFormats	Date Formats
hidePicker	Hide Picker
lowCode	ActionScript that controls the behavior of the element.

50 Element: Slider

This element allows the user to enter a value in a range. It shows a slider from minimum to maximum value.



W 359
X 25
H 44
Y 187
stretch
stretch
Arial
27
Blend Level 0
Corner Radius 0
Track Size 19
Filled Track Color #D4145A
Unfilled Track Color #C4C4C4
Gradient □
Current Screen: Element - Slider

Figure 210: Sample Slider

50.1 Events

Sliders have a mouse and touch related events that can be configured using low-code. They also have a `sliderValueChanged` event that is triggered when the value changes.

50.2 Properties

50.2.1 Size Section

This section is used to configure the slider size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

50.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.

50.2.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the slider.

Property	Description
textFace	The font used in the element's text.
textSize	The text size used in the element's text.
textAlign	The text alignment used for the element's text. It can be aligned to the left, right, or center.

50.2.4 Appearance Section

This section is used to configure how the slider looks.

Property	Description
blendLevel	How transparent the element is.
roundRadius	How round the corners of the element are.
trackSize	Track Size
minTrackColor	Filled Track Color
maxTrackColor	Unfilled Track Color
gradient	Gradient
trackBorder	Track Border
trackBorderSize	Track Border Size
trackBorderColor	Track Border Color
showThumb	Show Thumb
thumbColor	Thumb Color
thumbBorderSize	Thumb Border Size
thumbBorderColor	Thumb Border Color
labelType	Label Type
labelPosition	Label Position
foregroundColor	The color used in text.
autoSize	Sets size to automatic

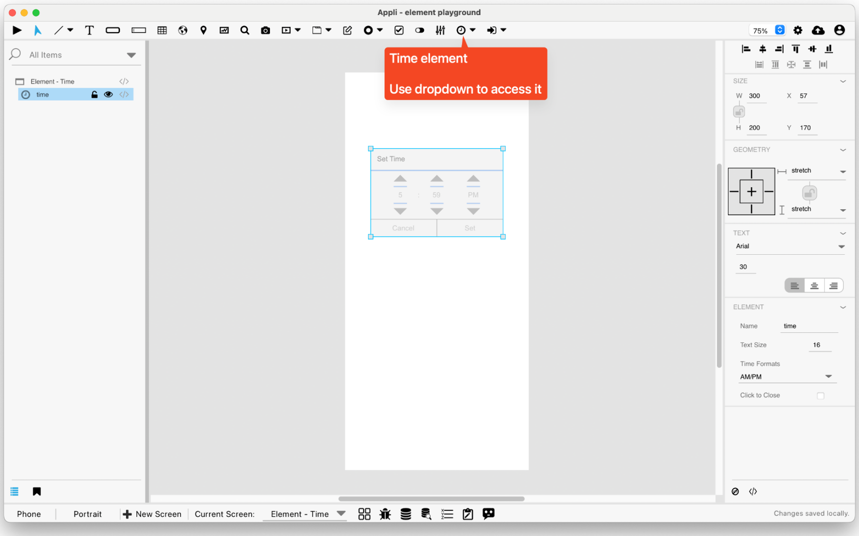
50.2.5 Element Section

This section is contains properties that are specific to elements of type slider.

Property	Description
name	The name of the element. Displayed in the project browser.
sliderOrientation	Slider Orientation
currentValue	Current Value
minValue	Minimum Value
maxValue	Maximum Value
minValueText	Min Value Text
maxValueText	Max Value Text
step	Step Size
showIncrements	Show Increments

51 Element: Time

This element allows the user to display or pick a time. Use the property inspector in the playground to configure your time picker and use the date/time actions to show and hide them.



Time element

Use dropdown to access it

SIZE

W 300 X 57

H 200 Y 170

GEOMETRY

stretch

stretch

TEXT

Arial

30

ELEMENT

Name time

Text Size 16

Time Formats AM/PM

Click to Close □

Figure 211: Sample Time

51.1 Events

Time elements have a *mouseUp* event that can be configured using low-code.

51.2 Properties

51.2.1 Size Section

This section is used to configure the time size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

51.2.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.

51.2.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the time.

Property	Description
textFace	The font used in the element's text.
textSize	The text size used in the element's text.
textAlign	The text alignment used for the element's text. It can be aligned to the left, right, or center.

51.2.4 Element Section

This section is contains properties that are specific to elements of type time.

Property	Description
name	The name of the element. Displayed in the project browser.
textSize	The text size used in the element's text.
timeFormats	Time Formats
hidePicker	Hide Picker
lowCode	ActionScript that controls the behavior of the element.

52 Element: Query

This element collects field elements used as inputs for generating a query.

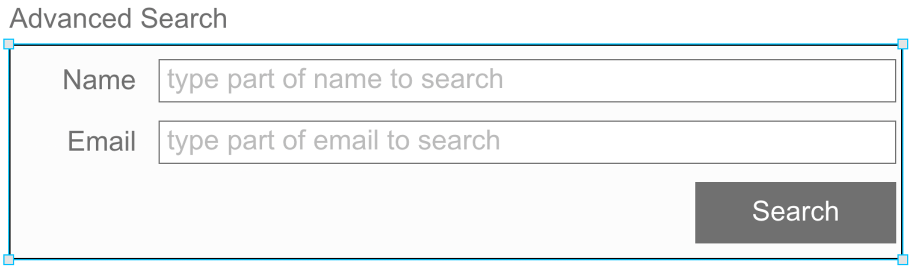
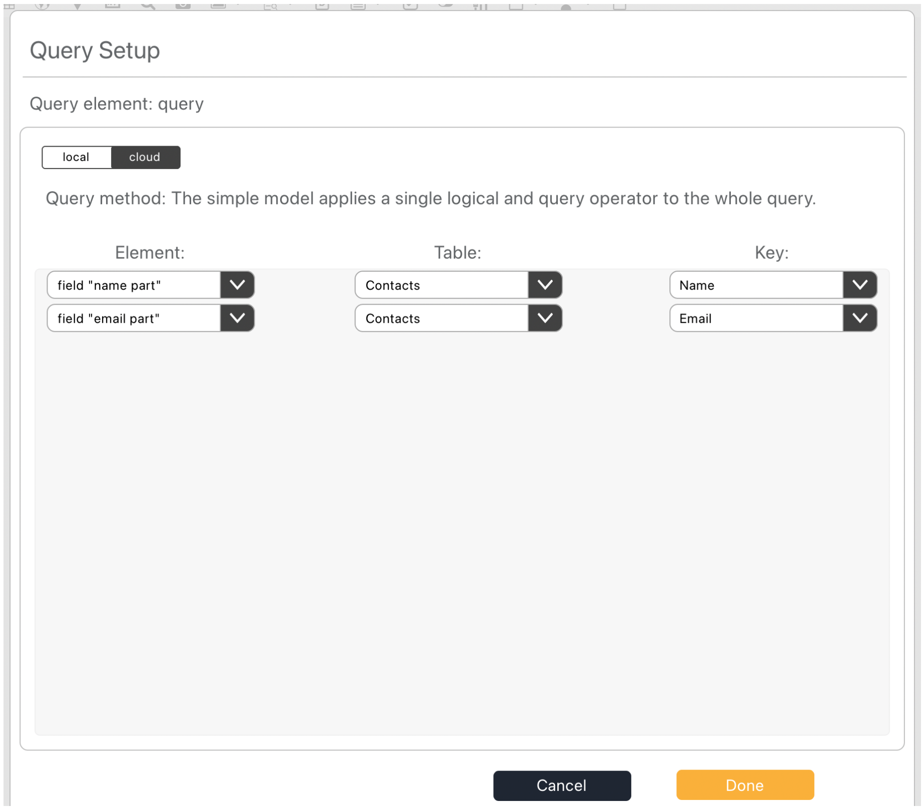


Figure 212: Sample query element

Like a layout or form, query elements are containers for other elements. Using the *no-code* interface, the developer can use the elements contained in the query element to configure a query to a table.



Query Setup

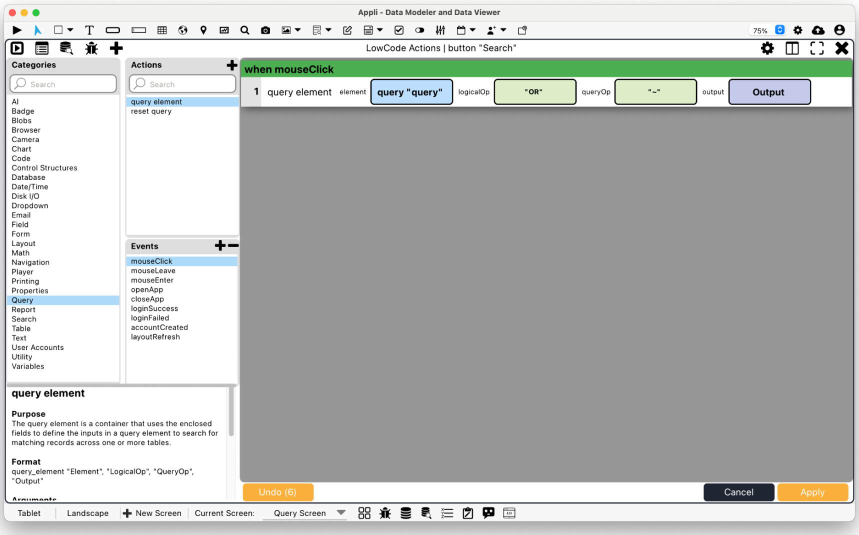
Query element: query

Query method: The simple model applies a single logical and query operator to the whole query.

Element:	Table:	Key:
field "name part"	Contacts	Name
field "email part"	Contacts	Email

Figure 213: Example of no code interface for the query element above

Use the actions in the *query category* after configuring the query element using no code to perform searches on a table.



when mouseClick

1 query element element query "query" logicalOp "OR" queryOp "~" output Output

query element

Purpose
The query element is a container that uses the enclosed fields to define the inputs in a query element to search for matching records across one or more tables.

Format
query_element "Element", "LogicalOp", "QueryOp", "Output"

Figure 214: Example of action to search a contacts table

52.1 Properties

52.1.1 Size Section

This section is used to configure the query size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

52.1.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

52.1.3 Appearance Section

This section is used to configure how the query looks.

Property	Description
blendLevel	How transparent the element is.
opaque	if the element is opaque or if it's background is transparent.
backgroundColor	The color used for the element's background.
showBorder	Turns the visibility of the element's borders on or off.
lineSize	the size of a line in pixels.
borderColor	the color used for the border of the element.
showScrollbar	If the scrollbar should be visible or not.

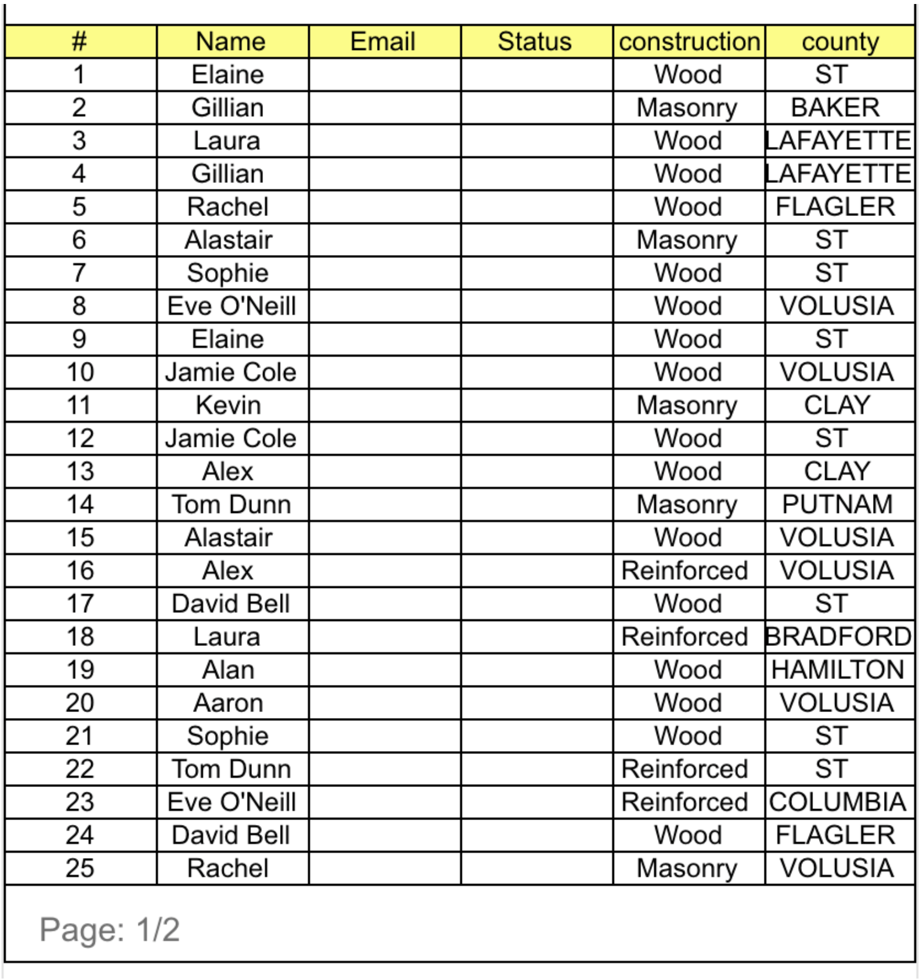
52.1.4 Element Section

This section is contains properties that are specific to elements of type query.

Property	Description
name	The name of the element. Displayed in the project browser.

53 Element: Report

This element contains a report.



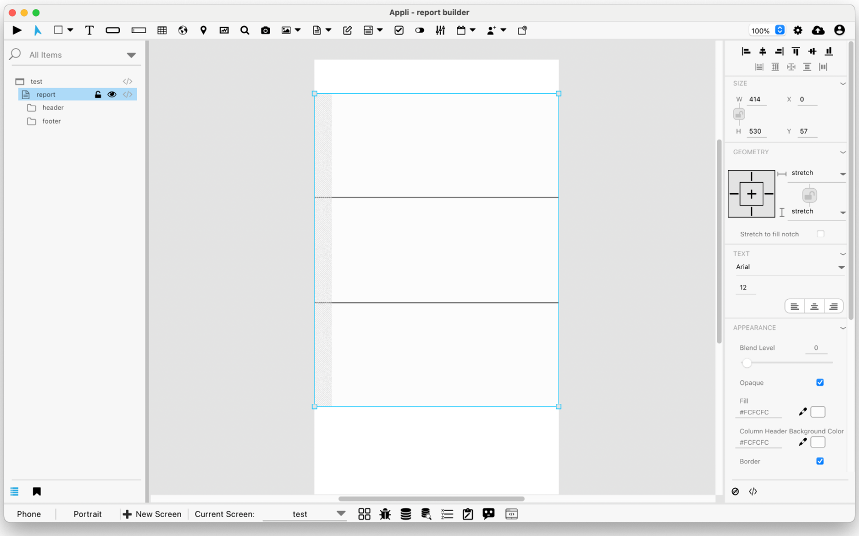
#	Name	Email	Status	construction	county
1	Elaine			Wood	ST
2	Gillian			Masonry	BAKER
3	Laura			Wood	LAFAYETTE
4	Gillian			Wood	LAFAYETTE
5	Rachel			Wood	FLAGLER
6	Alastair			Masonry	ST
7	Sophie			Wood	ST
8	Eve O'Neill			Wood	VOLUSIA
9	Elaine			Wood	ST
10	Jamie Cole			Wood	VOLUSIA
11	Kevin			Masonry	CLAY
12	Jamie Cole			Wood	ST
13	Alex			Wood	CLAY
14	Tom Dunn			Masonry	PUTNAM
15	Alastair			Wood	VOLUSIA
16	Alex			Reinforced	VOLUSIA
17	David Bell			Wood	ST
18	Laura			Reinforced	BRADFORD
19	Alan			Wood	HAMILTON
20	Aaron			Wood	VOLUSIA
21	Sophie			Wood	ST
22	Tom Dunn			Reinforced	ST
23	Eve O'Neill			Reinforced	COLUMBIA
24	David Bell			Wood	FLAGLER
25	Rachel			Masonry	VOLUSIA

Page: 1/2

Figure 215: Sample Report

> Be aware that a layout element is empty when first drawn on the screen. You need to place the elements you want inside it.

A report element is a container element that by default contains two groups inside it: *header* and *footer*. A report is composed of a header, the report, and a footer.



W 414 X 0
H 530 Y 57
stretch
stretch
Stretch to fill notch □
Arial
12
Blend Level 0
Opaque ☑
Fill #FCFCFC
Column Header Background Color #FCFCFC
Border ☑

Figure 216: A Report element in the builder

53.1 No code

There **necessary** options available for Report elements that **must** be configured using *no code*.

53.2 Events

Maps have a *mouseUp* event that can be configured using low-code. This event is triggered when the user clicks or touches the map.

53.3 Actions

There are the *low code* actions available for Report elements. Check the Actions: Report documentation for more information.

53.4 Properties

53.4.1 Size Section

This section is used to configure the report size and position.

Property	Description
top	The value in pixels representing how far the element is from the top of the window.

Property	Description
left	The value in pixels representing how distant element is from the left side of the window.
width	The value in pixels representing the distance between the left side of the element and its right side.
height	The value in pixels representing the distance between the bottom side of the element and its top side.

53.4.2 Geometry Section

Use the properties in this section to configure the responsive design behaviors for the element.

Property	Description
lockAspectRatio	Locks the aspect ratio of the element.
responsiveX	how the element resizes responding to screen changes in the X axis.
responsiveY	how the element resizes responding to screen changes in the Y axis.
allowUnderNotch	Stretch to fill notch

53.4.3 Text Section

The properties in this section are used together with those in the appearance section to configure how the element is displayed on the screen. This section deals with the properties related to text used by the report.

Property	Description
textFace	The font used in the element's text.
textSize	The text size used in the element's text.
textStyle	The style used in the element's text. It can be bold, italic, underlined, and strikethrough.
textAlign	The text alignment used for the element's text. It can be aligned to the left, right, or center.

53.4.4 Appearance Section

This section is used to configure how the report looks.

Property	Description
blendLevel	How transparent the element is.
showBackground	If the element background should be opaque or transparent.
backgroundColor	The color used for the element's background.
showBorder	Turns the visibility of the element's borders on or off.
borderWidth	Border Size
borderColor	the color used for the border of the element.
showCellLines	Cell Lines
cellLineSize	Cell Line Size
cellLineColor	Cell Line Color

53.4.5 Element Section

This section is contains properties that are specific to elements of type report.

Property	Description
name	The name of the element. Displayed in the project browser.
showHeader	Header
showFooter	Footer
showLineNumberColumn	Line Number Column
lineNumberColumnName	Column Name
lowCode	ActionScript that controls the behavior of the element.

54 Actions: Layout

54.1 display records in layout

Displays the specified records in a layout element.

Argument	Description
Sort	An optional argument that determines if the layout element's no-code sort is applied to the displayed records. The default value is 'true'.
Element	The layout element to display the records in.

Argument	Description
RecordIDs	A variable containing the recordIDs of the records to be displayed. (line-delimited)

54.2 get property from layout row

Retrieves the value of a property of an element inside of a layout element.

Argument	Description
Element	The element to have its property value returned.
Property	The attribute whose value will be returned.
RecordID	A variable containing the recordID of the layout row containing the element.
Output	The variable where the results will go.

54.3 refresh layout

Refreshes the records of a layout element.

Argument	Description
Element	The layout element to refresh.

54.4 reset layout

Clears all data and the recordIDs associated with a layout.

Argument	Description
Element	The layout element to reset.

54.5 select record in layout

Selects the specified record in a layout element.

Argument	Description
Element	The layout element containing the record to be selected.
RecordID	A variable containing the recordID of the record to be selected.

54.6 set property in layout row

Change the value of a property of an element inside of a layout with multiple rows enabled.

Argument	Description
Element	The element to have its property adjusted.
Property	The attribute that is to be adjusted.
Value	A variable or input that will be applied to the selected element.
RecordID	A variable containing the recordID of the layout row containing the element.

54.7 set property in row from variable

Set a property of an element in all rows of a layout element from a variable. This action can only be used in the ‘layout refresh’ event.

Argument	Description
Variable	The variable to apply to the property of the selected element.
Element	The element to have its property adjusted.
Property	The property that is to be adjusted.

54.8 set variable from context

Set a variable’s value based on the clicked element’s context (i.e., the recordID). This only works for elements inside of Layouts.

Argument	Description
Variable	The variable where the context’s value should be stored.

54.9 set variable from row data

Set a variable to a key value of the current row. This action can only be used in the ‘layout refresh’ event.

Argument	Description
Variable	The variable to set the value of.
TableName	The table containing the specified key.
Key	The key of the current row’s record to place into the variable.

54.10 unhighlight layout

Unhighlights the highlighted row of a layout element.

Argument	Description
Element	The layout element to unhighlight.

55 Actions: Date/Time

55.1 add value to date

Outputs the sum of a value and a date(represented as seconds) into the selected format.

Argument	Description
Output	The location of the output of both the value and the date (variable or text or field element).
OutputFormat	The desired format for the sum of the value and the date in the formats provided by the date element.
Date	The location where the date in seconds is located (variable or date element).
DateFormat	The format of the inputted date.
Value	The value added to the date in seconds.
ValueType	The type of value being added to the date in either: day, month, or year.

55.2 age to birth date

A rough calculation of the birth date from an age value.

Argument	Description
age	The age to be converted to a birth date. The date should be followed by the letter ‘m’ for months or a number for years. A decimal value will provide a higher level of accuracy.
Output	The location where the result will go (variable, text or field element).

55.3 bind date picker

Binds a date element to a field element.

Argument	Description
FieldFormat	The format of the date from the field.
Date	The date element to bind to a field element.
Field	The field element to bind to a date element.

55.4 bind time picker

Binds a time element to a field element.

Argument	Description
FieldFormat	The format of the time from the field.
Time	The time element to bind to a field element.
Field	The field element to bind to a time element.

55.5 check date range

Determines if the date falls within the specified range. Range includes the start date and end date (inclusively bound).

Argument	Description
RangeFormat	The format of the start and end dates.
Output	An output statement specifying whether the Date is ‘within’, ‘below’, or ‘above’ the range.
Date	The location of the date to check whether it falls within the start date and end date (variable, text, field element, or datepicker).
InputFormat	The format of the inputted date.
StartDate	The start date of the range.
EndDate	The end date of the range.

55.6 compare dates

Determines if the date is before or after the date being compared to.

Argument	Description
Output	An output statement specifying whether the Date ‘Occurs Before’ or ‘Occurs After’ the compared date. Otherwise they are the ‘Same Date’.
Date	The date being compared with (variable, text, field element, or from date picker).
DateFormat	The format of the date being compared with.
ComparedDate	The date being compared to (variable, text, field element, or from date picker).
ComparedDateFormat	The format of the other date.

55.7 convert date

Converts date into the selected format.

Argument	Description
Input	The location of the date that will be converted (variable or text or field element).
InputFormat	The current format of the date.
OutputFormat	The desired format for the date.
Output	The location where the converted date will go (variable or text or field element).

55.8 convert datetime

Converts the date and time into the selected format.

Argument	Description
OutputTimeFormat	The desired format for the time.
Output	The location where the converted date and time will go (variable or text or field element).
Input	The location of the date and time that will be converted (variable or text or field element).
InputDateFormat	The current format of the date.
InputTimeFormat	The current format of the time.
OutputDateFormat	The desired format for the date.

55.9 convert month

Converts a line-delimited list of months to the specified format.

Argument	Description
Output	The location where the converted month or list of months will go (variable or text or field element).
Input	The variable or text or field element that contains a line-delimited list of month names or numbers.
Format	The desired format for the month(s).

55.10 convert time

Converts time into the selected format.

Argument	Description
Input	The location of the time that will be converted (variable or text or field element).
InputFormat	The current format of the time.
Output	The location where the converted time will go (variable or text or field element).
OutputFormat	The desired format for the time.

55.11 date difference

Get the difference between two dates or times in multiple formats.

Argument	Description
Element1	The first date in the comparison (variable or field element).
Element2	The second date in the comparison (variable or field element).
Format	The format of the resulting comparison (verbose, seconds, minutes, hours, days, weeks, months, years). Verbose will provide a value like: 2 years, 3 months, 7 days, 15 hours. The other values will return the raw value of each format, unrelated to each other. They may contain non-integer values.

Argument	Description
Output	The location where the result will go (variable or text or field element).

55.12 find day of week

Outputs the day of the week based on the date given.

Argument	Description
Date	The location where the date is located (input, variable or text or field element).
DateFormat	The format of the date.
Output	The output location of the day of the week based on the given date (variable or text or field element).
Abbreviation	Optional parameter to abbreviate the outputted day of the week.

55.13 find relative date

Finds the first or last day of the month given a date.

Argument	Description
Output	The location to store the first or last date of the month (variable, text, or field element).
Date	The location where the date in seconds is located (variable, field, text, or date element).
DateFormat	The format of the inputted date.
DatePosition	Specify whether to return the first or last day of the month.
OutputFormat	The format of the first or last date of the month.

55.14 get current date

Retrieves the current date in the selected format.

Argument	Description
DateFormat	The desired format for the date.

Argument	Description
Output	The location where the date will go (variable or text or field element).

55.15 get current date and time

Retrieves the current date and time in the selected format.

Argument	Description
Output	The location where the date and time will go (variable or text or field element).
DateFormat	The desired format for the date.
TimeFormat	The desired format for the time.

55.16 get current time

Retrieves the current time in the selected format.

Argument	Description
Format	The desired format for the time.
Output	The location where the formatted time will go (variable or text or field element).

55.17 get date

Retrieves the date value from a date element and convert it into the selected date format.

Argument	Description
OutputFormat	The desired format for the date.
Input	The location of the seconds that will be converted to a date format.
Output	The location where the converted seconds will go (variable or text or field element).

55.18 get date format

Gets a format from a list of date formats.

Argument	Description
DateFormat	List of date formats.
Output	The location of where the date format will go (variable).

55.19 get date part

Retrieves the day, month, or year component from the given date.

Argument	Description
Date	The location where the date is located (variable, field, text, or date element).
DateFormat	The format of the inputted date.
DatePart	The date component to be retrieved.
Output	The location to store the outputted date component (variable, text, or field element).

55.20 get time

Retrieves the time value from a time element and convert it into the selected time format.

Argument	Description
OutputFormat	The desired format for the time.
Input	The location of the time that will be converted to the specified time format.
Output	The location where the converted seconds will go (variable or text or field element).

55.21 hide picker

Hides the date or time picker.

Argument	Description
Picker	Shows the date or time picker.

55.22 set date

Sets the selected date element to the provided date.

Argument	Description
Element	The date element to modify.
Date	The specific date used to modify the date element.

55.23 set date part

Modify a date by setting the day, month, or year.

Argument	Description
OutputFormat	The format of the outputted date after setting the date component.
Output	The location to store the modifed date (variable or text or field element).
Date	The location where the date in seconds is located (variable, field, text, or date element).
DateFormat	The format of the inputted date.
DatePart	The date component to be modified.
Value	The value to change the day, month, or year.

55.24 show picker

Shows the date or time picker.

Argument	Description
Picker	Shows the date or time picker.

55.25 subtract value from date

Outputs the subtraction of a value from a date(represented as seconds) into the selected format.

Argument	Description
Output	The location of the outputted difference of the value and the date (variable or text or field element).
OutputFormat	The desired format for the output.
Date	The location where the date in seconds is located (variable or date element).
DateFormat	The format of the date input.

Argument	Description
Value	The value subtracted from the date in seconds.
ValueType	The type of value being subtracted from the date in either: day, month, or year.

55.26 toggle picker

Toggles the visability of the time or date picker.

Argument	Description
Picker	The date or time picker to toggle

55.27 validate date

Determines if the date is valid and returns an output of true or false.

Argument	Description
Output	A binary 'true' or 'false' statement specifying whether the date is valid.
Date	The date being validated(variable or text or field element).
DateFormat	The format of the date being validated.

56 Actions: Table

56.1 display records in table

Displays the specified records in a table element.

Argument	Description
Element	The table element to display the records in.
RecordIDs	A variable containing the recordIDs of the records to be displayed. (line-delimited)

56.2 get all selected recordIDs

Retreives the recordID of all the selected rows in a table element.

Argument	Description
Element	The table element with the selected rows.
Output	The location where the recordIDs will go (variable or text or field element).

56.3 get last selected recordID

Retreives the recordID of the last row selected in a table element.

Argument	Description
Element	The table element with the selected row.
Output	The location where the recordID will go (variable or text or field element).

56.4 refresh table

Refreshes the records of a table element.

Argument	Description
MaintainHighlight	Whether to maintain the record(s) highlighted after a refresh (True/False).
Element	The table element to refresh.
MaintainScroll	Whether to maintain the current scroll after a refresh (True/False).

56.5 reset table

Clears all data and the recordIDs associated with a table.

Argument	Description
Element	The table element to reset.

56.6 set table columns

Displays the specified columns in a table element.

Argument	Description
Element	The table element to display the columns in.

Argument	Description
Columns	A key name(s) of the column(s) to be displayed. (comma-delimited)

56.7 sort table

Sort a table by column and sort direction.

Argument	Description
Element	The table element to refresh.
Column	Choose which column to sort.
Direction	Sort ascending or descending.
SortType	You can sort by text, numeric, or dateTime.

57 Actions: Form

57.1 populate form data

Populates data into elements linked to a form element.

Argument	Description
Element	The form element whose linked elements will have data populated into them.
recordID	A variable or text or field element containing the ID of a record from the form element's linked table.

57.2 refresh form

Refreshes a form. If there is a table and record associated with the form, it will be set in the form. Otherwise, it will be emptied.

Argument	Description
Element	The form element to refresh.

57.3 reset form

Clears the table's RecordID and clears all data in the form.

Argument	Description
Element	The form element to reset.

57.4 set form recordID

Sets the recordID that a form is connected to.

Argument	Description
Element	The form element to refresh.
recordID	The recordID to be connect to.

57.5 set form table

Sets the table that a form is connected to.

Argument	Description
Element	The form element to refresh.
TableName	The table to be connect to.

57.6 submit form to database

Submits the data from the form element to the database.

Argument	Description
Form	The name of the form element.
Output	The optional output variable will contain the recordID of the new or updated record.

58 Actions: Navigation

58.1 go screen

Switches the visible screen to the specified screen.

Argument	Description
Screen	Select a Variable or a ScreenName to tell Appli where to navigate.Variable: The variable may contain either a screen ID (UUID) or a screen name.ScreenName: Select a screen from a list of screen names.

58.2 launch url

Allows APPLI to launch a URL to their respective applications. It opens the default applications based on the custom URL schemes. .

Argument	Description
Output	Optional status message that checks if an application exists to handle the URL scheme.
URLScheme	Given URL scheme (sms:, tel:, http:, https:, file:) that is handled by the default app (variable, input, text or field element).An example of an URL scheme to open the phone app would be tel:7141234567.http: and https: URLs, which launch in the default browser application if available.file: URLs, which open the file in the associated application if available.tel: URLs, which open the dialer with a given phone number.Other examples:On macOS you can open the system preferences and jump to the Privacy tab.launch url”x-apple.systempreferences:com.apple.preference.security?Privacy_Automatio mobile you can send an SMS:launch url(“sms:”tMobile”?body=“tMessage)Useful links that provide other examples:https://x-callback-url.com/https://en.wikipedia.org/wiki/List_of_URI_schemesLaunching URLs will take your user to another app if the app is installed.
URL	The URL to be launched from the URL scheme.

59 Actions: Math

59.1 Random Number

Returns a random integer.

Argument	Description
Output	The location where the result will go (variable or text or field element).

Argument	Description
lowerNumber	An integer or variable containing an integer to be the lowest possible random number.
upperNumber	An integer or variable containing an integer to be the highest possible random number.

59.2 add 1 to variable

Adds 1 to a variable.

Argument	Description
Variable	The variable to which 1 is added. An empty variable implies a value of 0.

59.3 get difference

Calculates the difference between two numbers.

Argument	Description
Number1	A number or variable containing a number to subtract from.
Number2	A number or variable containing a number to subtract by.
Output	The location where the subtraction of two numbers will be stored (variable).
ErrorHandling	The action to take if an error occurs.

59.4 get division

Calculates the division of two numbers.

Argument	Description
Dividend	A number or variable containing a number to be the dividend (number that is being divided; ‘Dividend’/‘Divisor’).
Divisor	A number or variable containing a number to be the divisor (number that is being divided by).
Output	The location where the division of the two numbers will be stored (variable).

Argument	Description
ErrorHandling	[Optional] The action to take if an error occurs. Shows error in a popup error as a default action.

59.5 get power

Calculates the number raised to a power.

Argument	Description
Base	A number or variable containing a number to take the exponent of.
Exponent	A number or variable containing a number representing what number to raise the base number by.
Output	The location where the number raised by the power will be stored (variable).
ErrorHandling	The action to take if an error occurs.

59.6 get product

Calculates the product of two numbers.

Argument	Description
Multiplier	A number or variable containing a number to be multiplied.
Multiplicand	A number or variable containing a number to be multiplied.
Output	The location where the product of the two numbers will be stored (variable).
ErrorHandling	The action to take if an error occurs.

59.7 get root

Calculates the ‘nth’ root of a number.

Argument	Description
Radicand	A number or variable containing a number to take the root of.
Degree	A number or variable containing a number to take the root by.

Argument	Description
Output	The location where the nth root of a number will be stored (variable).
ErrorHandling	The action to take if an error occurs.

59.8 get sum

Calculates the sum of two numbers.

Argument	Description
Number1	A number or variable containing a number to be summed.
Number2	A number or variable containing a number to be summed.
Output	The location where the sum of the two numbers will be stored (variable).
ErrorHandling	The action to take if an error occurs.

59.9 round to number

Rounds off a value to the nearest whole number.

Argument	Description
Output	The location where the rounded number will be stored.
Number	The value to round to the nearest number.
Precision	The number of decimal places.

60 Actions: Chart

60.1 set chart axis

Change the axis (labels) of a chart.

Argument	Description
Element	The chart to have its axis set.
Axis Labels	The axis labels to apply to the selected chart, in comma-delimited format

60.2 set chart data

Change the data of a chart.

Argument	Description
Element	The chart to have its data set.
Value	The data to apply to the selected chart, in comma-delimited format

60.3 set chart data from table

Change the data of a chart.

Argument	Description
Target	Either local data or cloud data
Sort	Sort Methodology for the Axis Key
Element	The chart to have its data set.
TableName	The Table to pull data from.
Axis Key	The key to pull the axis label from.
Data Key	The key to pull the data from.

61 Actions: Disk I/O

61.1 ask for file path

Asks the user to select a desired filepath. A filepath contains a filename and its folder path.

Argument	Description
FilePathVariable	The variable that will contain the filepath. It will contain 'Cancel' if the user cancelled the selection.
DefaultFileName	Pass an optional default name and location of the folder whose contents will be shown in the selection dialog.

61.2 ask for folder

Asks the user to select a folder.

Argument	Description
FolderSelectionVariable	The variable that will contain the path of the selected folder. 'Cancel' will be output to the variable if the user cancel's the selection dialog. The returned folder path INCLUDES the trailing slash (/).

61.3 create a folder

Create a folder inside the user's 'documents' folder.

Argument	Description
FolderName	The name of the folder to be created in the current user's 'documents' folder

61.4 export as CSV

Exports the provided column headers and row data as a CSV file. On desktop, the file will be saved to the selected location. On mobile, the file will be saved to the app's documents folder.

Argument	Description
ColumnHeaders	A comma-delimited list of headers for each column of the CSV file. If the value contains a comma or new line, the value must be enclosed in a pair of quotation marks. If the value contains a quotation mark, use double quotation marks.Example Header Data:Book Title,Author,Quote

Argument	Description
RowData	A line-delimited list where each line is a row of the CSV file. Each line contains a comma-delimited list of column values. If the value contains a comma or new line, the value must be enclosed in a pair of quotation marks. If the value contains a quotation mark, use double quotation marks.Example Row Data:The Fellowship of the Ring,J.R.R. Tolkien,“ “ “Not all those who wander are lost.” “ “”Oh, the Places You’ll Go!“,Dr. Seuss,” “ “Kid, you’ll move mountains.” “ “The Collected Poems of Emily Dickinson,Emily Dickinson,” “ “That it will never come againIs what makes life so sweet.” “”

61.5 list files

Lists the files in a specified folder inside the current user’s ‘Documents’ folder.

Argument	Description
FolderName	The name of the folder inside the current user’s ‘Documents’ folder to list the files from.
VariableName	The variable to store the file list output into.

61.6 load a text file

Load a text file from specified folder inside current user’s ‘Documents’ folder.

Argument	Description
VariableName	The variable to load the text file into
FolderName	The name of the folder to load from, inside current user’s ‘Documents’ folder
FileName	Name of the text file to load

61.7 load binary file

Load a binary file from specified folder inside current user’s ‘Documents’ folder.

Argument	Description
VariableName	The variable to load the binary file into
FolderName	The name of the folder to load from, inside current user's 'Documents' folder
FileName	Name of the binary file to load

61.8 load file from path

Load a binary file from specified folder inside current user's 'Documents' folder.

Argument	Description
Mode	Set the mode of the load operation. Binary or text. The default is Binary, and usually works for text as well.
FilePath	The full path of the file to load.
Variable	The variable into which to load the file.

61.9 save a binary file

Save a binary file into specified folder inside current user's 'Documents' folder.

Argument	Description
VariableName	The variable to save into a binary file
FolderName	The name of the folder to save into, inside current user's 'Documents' folder
FileName	Name of the binary file to save

61.10 save a text file

Save a text file into specified folder inside current user's 'Documents' folder.

Argument	Description
VariableName	The variable to save into a text file
FolderName	The name of the folder to save into, inside current user's 'Documents' folder
FileName	Name of the text file to save

61.11 save file to path

Save a file to a precise path on the device's local file system. Include the file name and extension in the path.

Argument	Description
Mode	Set the mode of the save operation. Binary or text. The default is Binary, and usually works fine for text as well.
Data	The variable where the data to be saved is stored.
FilePath	The variable where the full file path is stored.

61.12 select file

Ask the user to select one or more files.

Argument	Description
FileNameVariable	The variable that will contain only the file name of the selected file. Or a list of filenames, if multiple files are selected
SelectedFileVariable	The variable that will contain a line delimited list of selected files
AllowMultipleFiles	Set to true to allow for selection of multiple files

61.13 set working folder

Set the working folder (inside the user's 'documents' folder).

Argument	Description
FolderName	The value to set variable to (input or another variable)

62 Actions: Browser

62.1 set browser url

Updates the URL displayed by the browser element.

Argument	Description
Element	The browser element to set the URL.

Argument	Description
URL	The URL to be displayed. The form must include https:// as a prefix the the rest of the URL. Input may come from a variable, element, or raw input.

63 Actions: Dropdown

63.1 get dropdown records list

Retrieves a line-delimited list of record IDs of the options in a dropdown element that has been connected to a database table.

Argument	Description
Element	A dropdown element connected to a database table whose list of record IDs will be returned.
Output	The location where the result will go (variable or text or field element).

63.2 refresh dropdown

Refreshes the options of a dropdown element that has been linked to a table.

Argument	Description
Element	The dropdown element to refresh.

64 Actions: Badge

64.1 refresh badge

Refreshes a badge's value, based on when.

Argument	Description
Element	The badge element to refresh.

65 Actions: Field

65.1 focus field

Focuses the specified field.

Argument	Description
Field	The field to be focused.

66 Actions: Query

66.1 query element

The query element is a container that uses the enclosed fields to define the inputs in a query element to search for matching records across one or more tables.

Argument	Description
Element	The query element whose field elements will be used as inputs for the query.
LogicalOp	The logical operator ‘and/or’ to be used for the query.
QueryOp	The query operator ‘equals, is in, after, etc’ to be used to compare inputs for the query.
Output	The recordIDs returned from the query may be placed in a variable, text, or field element.

66.2 reset query

Clears the field data in the query.

Argument	Description
Element	The form element to reset.

67 Actions: Printing

67.1 add to print job

Adds the specified element’s rectangular boundary of a given screen to the print job as a new page. This action must be called after the ‘start print process’ action and before the ‘end print process’ action.

Argument	Description
Screen	The screen to be printed.

Argument	Description
Element	An element whose rectangular boundary specifies the area of the screen to be printed.

67.2 end print process

Ends the print process and sends the open print job to the printer or saves it as a PDF, as specified by the ‘start print process’ action called prior to this action. This action must be called after the ‘start print process’ action.

67.3 get print setting

Retrieves the value of the specified print setting. Print margins are returned in inches.

Argument	Description
Setting	The print setting whose value will be returned.
Output	The location where the result will go (variable or field or text element).

67.4 print screen

Prints a single page of the specified element’s rectangular boundary of a given screen. This is only available for MacOS and Windows platforms.

Argument	Description
Screen	The screen to be printed.
Element	An element whose rectangular boundary specifies the area of the screen to be printed.

67.5 print text

Prints all text in a field or text element.

Argument	Description
Element	The field or text element to have its text printed.

67.6 print to pdf

Prints the specified element's rectangular boundary of a given screen to a single-page pdf file.

Argument	Description
Filename	The filename of the pdf to be saved.
Screen	The screen to be printed.
Element	An element whose rectangular boundary specifies the area of the screen to be printed.

67.7 set print margins

Specify the margin widths of the page(s) when printed. This action must be called before the 'start print process' action.

Argument	Description
Left	Width of the left margin in inches.
Top	Width of the top margin in inches.
Right	Width of the right margin in inches.
Bottom	Width of the bottom margin in inches.

67.8 set print orientation

Specify the orientation of the page(s) when printed. This action must be called before the 'start print process' action.

Argument	Description
Orientation	The orientation of the printed content.

67.9 start print process

Starts a print job to send to a printer (desktop only) or save as a PDF.

Argument	Description
Filename	The filename of the PDF to be saved if the 'output' argument is set to 'PDF'. If this argument is empty, a default name with the app's name and timestamp will be used.

Argument	Description
Output	Where the print job goes, either sent to a printer (desktop only) or saved as a PDF.
Result	The action taken by the user when presented with the print dialog if the 'output' argument is set to 'printer'. 'True' will be returned if the user proceeds with the print job. Otherwise 'false' will be returned.

68 Actions: Email

68.1 send email

Send an email.

Argument	Description
EmailBodyTEXT	The plain text body of the email. This will be used only if EmailBodyHTML is empty.
ToEmail	The TO address'.You can seperate multiple addresses using a comma.
FromEmail	The FROM address
Subject	The subject of the email.
EmailBodyHTML	The HTML Body of the email.

69 Actions: Text

69.1 format text

Transforms text to a specified format.

Argument	Description
Output	The location where the formatted text will be stored (variable or text element).
Text	A variable containing the text to be formatted.
Format	The desired format for the text.

69.2 refresh text

Refreshes the contents of a text element that is connected to a table and record.

Argument	Description
Element	The text element to refresh.

69.3 set text record ID

Sets the record ID that a text element is connected to if a table has been linked.

Argument	Description
Element	The text element whose record ID will be set.
RecordID	The record ID that the selected text element will be connected to.

70 Actions: Properties

70.1 get property

Retrieves the value of a property of an element.

Argument	Description
Output	The location where the results will go (variable or text or field element).
Element	The element to have its property value returned.
Property	The attribute whose value will be returned.

70.2 match property

Matches a property to another property with an optional numeric offset. Note on properties made up of comma delimited numbers: The Offset is added to each number in the comma delimied set.

Argument	Description
Offset	The value by which to offset the final target value. Only works on numeric properties, or properties with comma delimited numbers. Accepts negative and positive numbers.
TargetElement	The element to have its property replaced.
TargetProperty	The property to be replaced

Argument	Description
SourceElement	The element to have its property matched.
SourceProperty	The property to match

70.3 set property

Change the value of a property of an element.

Argument	Description
Value	A variable, element, or input that will be applied to the selected element.
Element	The element to have its property adjusted.
Property	The attribute that is to be adjusted.

70.4 set property from variable

Change the value of a property of an element using a variable.

Argument	Description
Variable	The variable to apply to the property of the selected element.
Element	The element to have its property adjusted.
Property	The attribute that is to be adjusted.

70.5 set variable from property

Set a variable's value based on an element's property.

Argument	Description
Variable	The variable where the property's value should be stored.
Element	The element to have its property read.
Property	The property to be read.

71 Actions: Utility

71.1 answer dialog

Displays a dialog box with a given message and 1-3 choices as buttons.

Argument	Description
interfaceTextColor	The color of the header and buttons (variable or input). If the argument is empty, the default dialog text color in the player settings will be used.
iconID	The ID of the blob that will be displayed as an icon on the answer dialog. If the argument is empty, the icon set in the player settings will be used.
Title	The text displayed in the header (variable or input).
Choice1	A variable or input to be displayed as a button in the dialog box.
Choice2	A variable or input to be displayed as a button in the dialog box.
Output	The location where the selection will go (variable or text or field element).
Message	A variable or input to be displayed in the dialog box.
Choice3	A variable or input to be displayed as a button in the dialog box.
interfaceColor	The color of the header and buttons (variable or input). If the argument is empty, the default dialog interface color in the player settings will be used.

71.2 blank action

This is an action you can type into to quickly filter from the list of all Appli actions. Clicking or pressing enter in the auto-completion field turns the blank action into the selected action. Any remaining blank actions get deleted when you press 'Apply' .

71.3 check internet status

Checks the status of the internet connection, and returns 'true' or 'false'.

Argument	Description
Output	The location where the internet connection status will go (variable or element)

71.4 combine multiple values

Combines two or more values together and places the result in a Variable or Text element.

Argument	Description
Value6	A variable, input, text or field element to be combined.
Value4	A variable, input, text or field element to be combined.
Separator	A separator placed between the values being combined. Use 'LF' for a line feed.
Value1	A variable, input, text or field element to be combined.
Value3	A variable, input, text or field element to be combined.
Output	The location where the combine value will be stored (variable, element).
Value2	A variable, input, text or field element to be combined.
Value5	A variable, input, text or field element to be combined.

71.5 combine values

Combines two values (input, variable, text or field element) and stores the result in a specified variable.

Argument	Description
Value1	A variable, input, text or field element to be combined.
Separator	A separator placed between the two values being combined. Use 'LF' for a line feed.
Value2	A variable, input, text or field element to be combined.
Output	The location where the combination of the two values will be stored (variable, element).

71.6 comment

Add a helpful comment to your code.

Argument	Description
TheComment	The comment string to display

71.7 compose text message

Compose a text to send to the recipient with the given text body.

Argument	Description
Recipient	The recipient of the text'.You can seperate multiple addresses using a comma.
Text	The text to send to the recipient

71.8 count down timer

Start a countdown on the selected element until it reaches 0.

Argument	Description
Element	The field or text element to count down.

71.9 create UUID

Create a random UUID (v4).

Argument	Description
Variable	The variable that will store the UUID

71.10 dispatch event

Have Appli process an event anywhere in the current app.

Argument	Description
Event	Select an event to dispatch.
Target	A target screen or element that contains the event you want to activate.
Result	Optional output to a variable or element containing details about how the dispatch was processed: handled, unhandled, or passed.
CustomEvent	The custom event name to be dispatched.

71.11 email pdf

Emails the specified element's rectangular boundary of a given screen as a pdf file.

Argument	Description
Subject	The subject of the email.
Body	The body of the email.
Screen	The screen to be emailed.
Element	An element whose rectangular boundary specifies the area of the screen to be emailed as a pdf.
Recipient	The email where the pdf will be sent.
FromEmail	The email that will be displayed as the sender's email.

71.12 filter duplicate lines

Removes duplicate lines from a list.

Argument	Description
Input	The variable or text or field element that contains a line-delimited list.
Output	The location to store the filtered list (variable or text or field element).

71.13 filter lines

Filter lines from a list.

Argument	Description
ListToFilter	A variable that contains a line-delimited list.
Operator	Which operator (with or without) to filter the lines by.
Filter	A variable that contains a line-delimited list.
Output	The variable to store the filtered list.

71.14 get character count

Retrieves the number of characters in the specified text.

Argument	Description
Input	The variable or field or text element whose characters will be counted.
Output	The location where the result will go (variable or field or text element).

71.15 get element value

Retrieves the value of the specified element.

Argument	Description
Element	The element (dropdown, field, radio, slider, switch, date, time, checkbox, or text) whose value will be retrieved.
Output	The location where the result will go (variable or dropdown, field, radio, slider, switch, date, time, checkbox, or text element).

71.16 get orientation

Retrieves the current orientation of the screen (portrait or landscape).

Argument	Description
Output	The location where the result will go (variable or text or field element).

71.17 get platform

Retrieves the specific platform of the current device (Windows, MacOS, iOS, Android).

Argument	Description
Output	The location where the result will go (variable).

71.18 get platform color

Retrieves the color of the platform’s background.

Argument	Description
Output	The location where the result will go (variable or field or text element).

71.19 get playground info

Get useful information about the state of Appli's playground. lastMouseClickLoc, lastClickLoc H, lastClickLoc V, lastMouseLoc, lastClickLine, playgroundWidth, playgroundHeight.

Argument	Description
VariableName	The array variable that will contain the playground information keys.

71.20 get screen rect

Retrieves the dimensions of the screen's rectangle of the current device. If there are multiple screens, the rectangle of the screen containing the app's center location is returned. Results are output in the format width,height in pixels.

Argument	Description
Output	The location where the result will go (variable or field or text element).

71.21 get tax

Retrieves the tax rate of a given location.

Argument	Description
Address	A variable containing the address of the location.
City	A variable containing the city of the location.
Zip	A variable containing the zip of the location.
Output	The location where the result will go (variable or text or field element).

71.22 get text offset

Retrieves the character position of the first occurrence of the text to find in the text to search.

Argument	Description
Output	The location where the result will go (variable or field or text element).
TextToFind	The input, variable, or field or text element whose text you want to find.
TextToSearch	The variable or field or text element whose text will be searched.

71.23 get version

Retrieves the current version of Appli IDE or Appli Player the device is running.

Argument	Description
Output	The location where the result will go (variable or text element).

71.24 hide element

Hide an element.

Argument	Description
Element	The element to hide

71.25 hide loading

Hides the loading dialog box.

71.26 lock screen

Locks the screen to prevent actions from being visible on the screen.

71.27 play audio clip

Plays a sound from a file. Supported formats are WAV & MP3.

Argument	Description
AudioFilePath	The path to the audio file to play.

71.28 remove whitespace

Removes leading and trailing whitespace from the specified input.

Argument	Description
Input	A variable or a text or field element whose contents will have its leading and trailing whitespace removed.
Output	A variable or a text or field element where the result will go.

71.29 replace text

Replaces the specified text from the provided source with other text and stores the result in a variable or text or field elenent.

Argument	Description
textToReplace	A variable, input, text or field element containing the piece of text that will be replaced.
replacementText	A variable, input, text or field element containing the replacement text. Use 'LF' for a line feed.
originalText	A variable, input, text or field element containing the original text that will be modified.
Output	The location where the modified text will be stored (variable, text or field element).

71.30 reset element

Resets the specified element to its default state. Supported elements: field, radio, search, dropdown, login, create account, image.

Argument	Description
Element	The element to reset.

71.31 set element value

Sets the value of the specified element.

Argument	Description
Element	The element (dropdown, field, radio, slider, switch, date, time, checkbox, or text) whose value will be set.
Value	The value the specified element will store (input, variable or dropdown, field, radio, slider, switch or text element).

71.32 set item separator

Set the item separator for this code block.

Argument	Description
Separator	Specify the string character(s) that will act as an item separator within (this) event's code

71.33 set platform color

Sets the color of the platform's background.

Argument	Description
Color	The color or a variable storing a color (RGB or hex code) to set the platform's background to.

71.34 show element

Show an element.

Argument	Description
Element	The element to hide

71.35 show loading

Displays a dialog box with a given message and a loading spinner.

Argument	Description
Message	A variable or input to be displayed in the dialog box.

71.36 sort lines

Sorts lines from a list.

Argument	Description
Input	The variable or text or field element that contains a line-delimited list.
Direction	Which direction (ascending or descending) to sort the lines by.
SortType	The type of data stored in each line to sort by.
Output	The location to store the sorted list (variable or text or field element).

71.37 stop count down timer

Stops all countdown timers.

71.38 timer log

Record timestamped logs to time how long things take.

Argument	Description
Label	Label/name for this timer instance
Action	Choose whether to start or stop the timer
TimerLogVariable	Variable to store the timer log entries
LogString	Additional text to include in the log entry

71.39 toggle password

Show or hide password input in a field element. The “password” property must be on for this action to occur.

Argument	Description
Element	The field element to have its password input shown or hidden.

71.40 unlock screen

Unlocks and renders the updated screen.

71.41 wait

Wait for a specific amount of time, in milliseconds. (The wait is non-blocking).

Argument	Description
Number	Specify the number of milliseconds to wait

72 Actions: Variables

72.1 JSON to array

Convert a JSON string into an array. Ensure the input JSON is properly formatted.

Argument	Description
Error	A variable to store an error, should one occur.
Variable	The variable containing JSON to convert to an array
Output	The variable where the resulting array will be placed.

72.2 array to JSON

Turn an array variable to JSON. Does not support binary data. Base64Encode your binary keys first.

Argument	Description
Error	A variable to store an error, should one occur.
Variable	The variable to be processed.
Output	The variable where the JSON will be placed.
ForceRootObjectType	Force root element value to be ‘Object’ or ‘String’.
Pretty	Whether to format the JSON to be more human readable.

72.3 clear variable

Set variable to an empty value.

Argument	Description
VariableName	The variable to set empty.

72.4 count indexes in array

Gets the number of index keys present in an array.

Argument	Description
Path	An optional variable, element, or input that consists of a series of slash delimited indexes to access an array.
Variable	The variable to have its indexes counted.
Output	The location where the number of indexes will go (variable or text or field element).

72.5 count lines in variable

Gets the number of lines in a variable.

Argument	Description
Variable	A variable where the lines will be counted.
Output	The location where the number of lines will go (variable or text or field element).

72.6 delete array element

Create a path to access the value of nested elements.

Argument	Description
Value	Select a variable that contains the element or value to delete.
Array	Select an array to modify.
Path	A series of slash delimited elements to remove data from an array.Example:Original array:myUserData/email/kayla@acme.commyUserData/zipCode/98027Prov ‘zipCode’ for the path, we are left with the following in our array:myUserData/email/kayla@acme.com

72.7 delete empty variables

Deletes all empty Appli variables. Useful in clearing out unused variables. Variables which your app populates will reappear.

72.8 find value in array

Find a specific value of a key inside the records of an array. This returns the first value that is found, so it works on arrays where the values are unique to each record.

Argument	Description
Output	Where do you want to save the result?
Key	The key to query in the array
Operator	The comparison operator for comparing values in the table against the input value.
KeyToReturn	The key of the found record to return
Array	Select an array to search
Value	The value to compare the array record key against
ReturnAs	If found, how should the result be returned?

72.9 get array value

Create a path to access the value of nested elements.

Argument	Description
Array	Select an array to parse.
RecordIDs	Optional input from a variable, element, or raw input that consists of line or comma delimited list where each line or item is a valid cdbRecordID of a record in the specified table. Use "*" to specify all records or read an array from source other than a table.
Path	A variable, element, or input that consists of a series of slash delimited elements to access a value of an array.
Output	The returned value can be placed in a variable or an element.Example array:[6ea1f2aa-b7af-426e-879e-3c2caa54fa7a][email][mark@acme.org]Path: emailOutput: mark@acme.org

72.10 get item from variable

Gets the specified item from a variable.

Argument	Description
Variable	A variable where the item will be retrieved from.
ItemNumber	The item number to retrieve from the variable.
Output	The location where the item will go (variable or text or field element).
Separator	The character(s) to use as the item separator for this operation. The default is a comma

72.11 get key from variable

Gets the specified key from a variable. Also works if the key is an array.

Argument	Description
Output	The location where the value of the key will go (variable or text or field element).
Variable	A variable where the key will be retrieved from.
Key	A variable or input specifying which key to get.

72.12 get line from variable

Gets the specified line from a variable.

Argument	Description
Output	The location where the line will go (variable or text or field element).
Variable	A variable where the line will be retrieved from.
LineNumber	A variable or input specifying which line to get.

72.13 get list of keys

Get a list of keys of an array variable.

Argument	Description
Variable	The array variable whose keys are returned.

Argument	Description
KeysList	The variable where the list of keys will be placed.

72.14 process variable

Performs an algorithmic operation on the content of a variable. Base64Encode & Base64Decode, Compress & Decompress, BinaryEncode & BinaryDecode functions are available.

Argument	Description
Variable	The variable to be processed.
Process	The function to process the variable.
Output	The variable where the result will be placed.
Error	A variable to store an error, should one occur.

72.15 reindex array

Reindexes an array to have sequential numeric indexes without gaps.

Argument	Description
ArrayVariable	The array variable to reindex

72.16 set array

Useful for creating or modifying an array with a new key = value.

Argument	Description
Value	The value will replace an existing value or can be used to create new values. The input may come from a variable, an element, or raw input.Examples: keys valueOld value: userDetails/email/ = tom@acme.comNew value: userDetails/email/ = andre@acme.com keys valueOld value: userDetails/zipCode/ = 92592New value: userDetails/zipCode/ = 98027
Array	Select an array to create or modify.
Path	A series of slash delimited keys in an array. The input may come from a variable, an element, or raw input.

72.17 set variable

Set a variable based on input, variable, or an element.

Argument	Description
Variable	The variable to set the value of.
Value	The value to set variable to (input, variable, text/field element)

73 Actions: User Accounts

73.1 create account

Creates an account for a user. Takes in their name, email, and password.

Argument	Description
ConfirmPassword	(Optional) field to confirm the password matches.
Verified	Users that are verified will not receive a verification email.
FirstName	The location of the user's first name. (Element or Variable).
LastName	The location of the user's last name.
Email	The location of the user's email.
Password	The location of the user's password.

73.2 delete account

Deletes a user account.

Argument	Description
PasswordRequired	Whether a password is required for user account deletion or not. If this argument is set to 'false', the 'password' argument can be set to empty.
Email	The email of the user's account which will be deleted.
Password	The password of the user's account which will be deleted.

73.3 get auth

Authenticate the user using email address & password.

Argument	Description
UserEmail	The user’s email.
UserPassword	The user’s password.

73.4 get auth status

Get the current auth status of CanelaDB.It returns the authetication status from either the local or cloud database as either true or false.

Argument	Description
Output	The location where the auth_status will go (variable or element).
Target	The type of authetication that will occur (local or cloud).

73.5 get logged in user

Get the email address of the logged in user.

Argument	Description
Output	The location where the logged in user’s e-mail address will go (variable or element)

73.6 get user first name

Get the first name of the logged in user.

Argument	Description
Output	The location where the logged in user’s first name will go (variable or element)

73.7 logout

Logs out the current user.

73.8 read user record

Reads the user record from the cdbUsers table of the user with the email provided.

Argument	Description
UserEmail	The email of the user whose record will be read.
Output	The location where the results will go (variable or element).

73.9 resend verification email

Sends a verification email to an existing user who has not been verified.

Argument	Description
Email	The email of the user's account where the verification email will be sent to.
Password	The password of the user's account where the verification email will be sent to.

73.10 update user record

Updates a record of an existing user in the 'cdbUsers' table.

Argument	Description
Data	An array containing key-value pairs of the data to be updated.
Email	The email of the user's account whose record will be updated.
Password	The password of the user's account whose record will be updated.
PasswordRequired	Whether a password is required or not for the user record to be updated. If this argument is set to 'false', the 'password' argument can be set to empty.

74 Actions: Blobs

74.1 batch create blob

Creates one or more blobs containing the data of the specified files.

Argument	Description
Output	Where the resulting array that contains the blobs' record IDs of uploaded files will be stored.Example Output:[1][01234567-abcd-8901-efgh-2345678901ij][2][99999999-efgh-1231-aaaa-234567890142]
Data	A variable storing an array that contains the filepath and type of each data file to be uploaded.Example Array:[1][blobData][/Users/username/Desktop/image.png][1][blobType][png]
Target	The place to create the blob(s) (local or cloud).

74.2 batch delete blob

Deletes one or more blob records.

Argument	Description
Data	A variable storing an array that contains the record IDs of the blobs to be deleted.Example Array:[01234567-abcd-8901-efgh-2345678901ij][][99999999-efgh-1231-aaaa-234567890142][]
Target	The place to delete the blob(s) (local or cloud).

74.3 batch read blob

Retrieves the path(s) where the specified blob records can be accessed.

Argument	Description
Output	Where the resulting array that contains the blobs' filepaths will be stored.Example Output:[01234567-abcd-8901-efgh-2345678901ij][/Appli/data/config/CanelaDB/blobs/user@email.com/Proje abcd-8901-efgh-2345678901ij.png][99999999-efgh-1231-aaaa-234567890142][/Appli/data/config/CanelaDB/blobs/user@email.com/Proj efgh-1231-aaaa-234567890142.mp4]

Argument	Description
Data	A variable storing an array that contains the record IDs of the blobs to be read.Example Array:[01234567-abcd-8901-efgh-2345678901ij][][99999999-efgh-1231-aaaa-234567890142][]
Target	The place to read the blob(s) (local or cloud).

74.4 batch update blob

Updates one or more blobs and replaces the existing data with that of the specified files.

Argument	Description
Data	A variable storing an array that contains the blob record ID whose data will be replaced and the filepath of the new file to be uploaded.Example Array:[01234567-abcd-8901-efgh-2345678901ij][/Users/username/Desktop/newImage.png][99999999-efgh-1231-aaaa-234567890142][/Users/username/Desktop/newVideo.mp4]
Target	The place to update the blob(s) (local or cloud).

74.5 clear temporary blobs

Clears the temporary blobs for this project.

74.6 create blob

Creates a blob containing the data in the specified file. You can specify a filepath using a variable, or allow selection of file by the user.

Argument	Description
BlobID	Where the resulting blobID should be be placed.
DataFilePath	The filepath to where the data for the blob can be found. THe path can be in a variable or can be selected by the user.
Target	The place to create the blob (local or cloud).

74.7 delete blob

Deletes a blob from your app's blob table. This can not be undone.

Argument	Description
Output	Where should the result of the delete function be placed
BlobID	The ID of the blob to be deleted.
Target	The place to delete the blob (local or cloud).

74.8 read blob

Returns the URL where the specified data blob is accessible.

Argument	Description
BlobID	Where the data for the blob can be found
Target	The place to access the blob (local or cloud).
OutputFormat	Specify whether the blob's URL or DATA is returned.
Output	Where the resulting blobID should be be placed.

74.9 update blob

The update blob action will replace an existing blob with an updated version.

Argument	Description
DataFilePath	Where the new data for the blob can be found.
BlobID	The ID of the blob to be updated.
Target	The place to access the records (local or cloud).
Output	The to place the result of the update.

75 Actions: Database

75.1 append to value

Appends data to the specified key's value of a given table.

Argument	Description
Target	The place to access the records (local or cloud).
RecordID	The ID of the specified record (Either a variable or a value).
Key	The key whose value will be appended to.

Argument	Description
Value	The data appended to the selected key's value.
TableName	The table where the specified key is located.
Delimiter	An optional space or command separating the appended value.
DelaySend	Optional argument to delay the action if the target is 'cloud'. Use 'Flush the Cache' to process delayed transactions.

75.2 batch query records

Searches the specified table and returns the records that match the provided queries.

Argument	Description
Output	The location where the results will go (variable).
TableName	The table whose records will be queried.
Query	A single or set of queries used to search the table for matching records.
Target	The place to access the records (local or cloud).
ResultFormat	The format of the query's result, either a line-delimited list (recordList) or an array containing the records' data (recordData).

75.3 calc update record

Performs one or more calculations on an existing record with the specified key(s), value(s), and operation and updates the record with the results.

Argument	Description
DelaySend	Optional argument to delay the action if the target is 'cloud'. Use 'Flush the Cache' to process delayed transactions.
Data	An array containing key-value pairs where the key belongs to the specified table and the value is the numeric value to be used in the calculation.

Argument	Description
Operator	The mathematical operator used in the calculation (add, subtract, multiply, divide).
Target	The place to update the record (local or cloud).
TableName	The table of the record to be updated.
RecordID	The ID of the record to be updated.
Output	The location where the results will go (variable).

75.4 count displayed records

Counts the number of records in a table element or layout element.

Argument	Description
Element	The table element or layout element whose records will be counted.
Output	The location where the results will go (variable or text or field element).

75.5 count records

This function counts the number of records (either cloud or local) in a given table.

Argument	Description
Output	The location where the results will go (variable or element).
TableName	The table where the specified records are located.
Target	The place to access the records (local or cloud).

75.6 create record

Creates a new record in the specified table with the provided array.

Argument	Description
Output	The location where the created record's ID will be stored (variable).

Argument	Description
TableName	The table where the new record will be created.
Data	An array containing key-value pairs.
Target	The place to create the record (local or cloud).
DelaySend	Optional argument to delay the action if the target is ‘cloud’. Use ‘Flush the Cache’ to process delayed transactions.

75.7 delete keys

Delete the specified keys from a record’s contents.

Argument	Description
TableName	The table where the specified records are located.
Keys	A selection of keys (fields) used to make a record.
RecordIDs	A line delimited list where each line is the cdbRecordID of a record in the specified table. Can use “*” to specify all records.
Target	The place to access the records (local or cloud).

75.8 delete record

Removes one or more records in a table.

Argument	Description
TableName	The table whose record(s) will be deleted.
RecordID	The recordID(s) of the record(s) to delete in the specified table. Can use “*” to specify all records.
Target	The place to access the records (local or cloud).
DelaySend	Optional argument to delay the action if the target is ‘cloud’. Use ‘Flush the Cache’ to process delayed transactions.

75.9 flush cache

Flushes the cache where cloud actions with a delaySend of ‘true’ are stored.

75.10 get result

Get the result of the last CanelaDB call.

Argument	Description
WhichResult	Select the kind of result you need. (Boolean) - if ‘completed’ or empty is passed.(String) - if ‘context’, ‘response’, ‘date’, or ‘log’ is passed.(Array) - if ‘recent’ is passed.
OutputVariable	Which of the results do you want? See dictionary panel for more

75.11 get table id

Get and return the ID of the given table name.

Argument	Description
TableName	The name of the table whose ID you wish to know
Output	The location where the results will go (variable or element).

75.12 get table keys

Get a list of the keys that make up the database table.

Argument	Description
OutputFormat	The format returned can be either a line-delimited list or a comma-delimited list.
TableName	Select the name of the table whose keys you wish to know.
Output	The location where the results will go (variable or element).

75.13 list records

Get a line delimited list of values from the specified key(s) for all the records in the specified table. If multiple keys are specified, the values of the keys will be comma-delimited.

Argument	Description
TableName	The table where the desired record values are located.
Target	The place to access the records (local or cloud).
Keys	A selection of keys from the specified table whose record values will be returned.
Output	The location where the results will go (variable or text or field element).

75.14 load table

Load table(s) into memory so they can be accessed by database actions.

Argument	Description
TableName	The table(s) to be loaded.

75.15 look up value

Retrieves the value associated with the specified key for a given record.

Argument	Description
Output	The location where the results will go (variable or text or field element).
TableName	The table where the specified record is located.
RecordID	The ID of the specified record (Either a variable or a value).
Key	The key whose value will be retrieved.
Target	The place to access the records (local or cloud).

75.16 merge record

Modifies a record's key value using comparisons like "is", "is not", "is in", and "is not in".

Argument	Description
Target	The place to create the record (local or cloud).
Data	An array containing key-value pairs.

Argument	Description
RecordID	The ID of the record to be modified.
Delimiter	The delimiter used to separate data in the record's key value.
TableName	The table where the new record will be created.
Operator	The operator used to compare the record's key value.
DelaySend	Optional argument to delay the action if the target is 'cloud'. Use 'Flush the Cache' to process delayed transactions.

75.17 ping

Pings the cloud server where the table is located and returns the response time in milliseconds. This action will be processed by the server, so a response indicates that the server is running and is processing requests.

Argument	Description
Table	The table name or table ID to ping. If a table is not specified, then a table will be chosen at random.
Output	The location where the result will go (variable or text element).

75.18 ping node

Only compatible with desktop platforms. Calling this action on a mobile platform will return the value 'N/A'. Pings the cloud server where the table is located and returns the response time in milliseconds. This function is NOT processed by the cloud server, so it only checks if the server is reachable from the client, not if the cloud server is actually running. Ping Node uses shell command 'ping'. As such, the response will be shorter than the 'ping' action response.

Argument	Description
Table	The table name or table ID to ping. If a table is not specified, then a table will be chosen at random.
Output	The location where the result will go (variable or text element).

75.19 query records

Searches the specified table and returns the records that match the provided query.

Argument	Description
ResultFormat	The format of the query's result, either a line-delimited list (recordList) or an array containing the records' data (recordData).
Key	The key in the table to query on.
Operator	The comparison operator for comparing values in the table against the input value.
Target	The place to access the records (local or cloud).
TableName	The table whose records will be queried.
Value	The value to compare the records against.
Output	The location where the results will go (variable).

75.20 read keys

Read a specific set of records and get only the specified keys for each returned record.

Argument	Description
Output	The location where the results will go (variable or element).
TableName	The table where the specified records are located.
Keys	A selection of keys (fields) used to make a record.
RecordIDs	A line delimited list where each line is the cdbRecordID of a record in the specified table. Can use "*" to specify all records.
Target	The place to access the records (local or cloud).

75.21 read record

Reads one or more records in a table.

Argument	Description
TableName	The table whose record(s) will be read.

Argument	Description
RecordIDs	A line delimited list where each line is the cdbRecordID of a record in the specified table. Can use “*” to specify all records.
Target	The place to access the records (local or cloud).
Output	The location where the results will go (variable or element).

75.22 sort records

Sorts a list of record IDs based on the specified key, direction, and sort type.

Argument	Description
Target	The place to access the records (local or cloud).
Key	The key to sort records by.
Direction	Which direction (ascending or descending) to sort the records by.
RecordIDs	A line delimited list where each line is the cdbRecordID of a record in the specified table.
TableName	The table whose records will be sorted.
SortType	The type of data stored in the key’s value to sort by.
Output	The location where the results will go (variable or text or field element).

75.23 sum column

Calculates the sum of all the values (default) or values of the specified records (optional) for the specified key of a given table.

Argument	Description
Records	Optional argument of a variable containing a line-delimited list of record IDs. Can use “*” to specify all records. If this argument isn’t populated, all records in the specified table will be used.
TableName	The table where the specified key is located.
Key	The key whose values will be summed.

Argument	Description
Target	The place to access the records (local or cloud).
Output	The location where the results will go (variable or text or field element).

75.24 sync records

Syncs a list of records in a specified table between local and cloud.

Argument	Description
DetectCollisions	Whether to compare record versions and avoid overwriting records if the source has a lower version (true) or to ignore record versions and have all source records overwrite target records (false).
Output	The location where the collision or sync results will go (variable).
RecordIDs	A line delimited list where each line is the cdbRecordID of a record in the specified table. Can use "*" to specify all records.
TableName	The table whose records will be synced.
Source	The direction of the sync (cloud or local).
AllowDeletes	Whether to remove records in the target if they don't exist in the source (true) or to ignore them (false).

75.25 update record

Updates an existing record in the specified table with the provided array.

Argument	Description
DelaySend	Optional argument to delay the action if the target is 'cloud'. Use 'Flush the Cache' to process delayed transactions.
TableName	The table of the record to be updated.
Data	An array containing key-value pairs.
RecordID	The ID of the records(s) to be updated.
Target	The place to update the record (local or cloud).

76 Actions: Control Structures

76.1 break

Break out of the current control structure and continue.

76.2 case

Create a case to use within a switch control structure.

Argument	Description
Value	The value to set variable to (input or another variable)

76.3 catch

Defines the block which runs when a try structure encounters an error.

Argument	Description
Variable	The variable to store the error, if one occurs

76.4 else

Adds a block for ‘all other cases’ of a control structure.

76.5 else if property

Additional condition to check the value of a property.

Argument	Description
Element	The element to have its property compared.
Property	The attribute that is to be compared.
Operator	The comparison operator to use
Value	The value to compare to

76.6 else if variable

Add an additional if condition to an existing control structure block, referencing a variable.

Argument	Description
ThisValue	The value to set variable to (input or another variable)

Argument	Description
ThisVariable	The variable to set the value of.
Operator	The comparison operator for comparing values in the table against the input value.

76.7 end if

Marks the end of a IF/ELSEIF/ELSE structure.

76.8 end repeat

Marks the end of a repeat structure.

76.9 end switch

Marks the end of a switch structure.

76.10 end try

Ends a try block.

76.11 exit repeat

Exits the repeat current repeat loop.

76.12 exit script

Exits the current script.

76.13 finally

Runs all lines below it, inside a try structure, whether there is an error or not.

76.14 if property

Creates the first condition referencing a property.

Argument	Description
Element	The element to have its property compared.
Property	The attribute that is to be compared.
Operator	The comparison operator to use
Value	The Value to compare to

76.15 if variable

Creates the first line of an if condition referencing a variable.

Argument	Description
ThisValue	The value to set variable to (input or another variable)
ThisVariable	The variable to set the value of.
Operator	The comparison operator for comparing values in the table against the input value.

76.16 next repeat

Skips the rest of the repeat control structure and returns to the top for the next repeat.

76.17 repeat

Repeat a code block n number of times.

Argument	Description
Number	Specify the number of times the repeat block will run

76.18 repeat over array

Repeat loop over the indexes or keys of an array.

Argument	Description
RepeatOver	Repeat loop over indexes or keys? Indexes are numeric, keys are strings like IDs. This action will repeat over every index in the array in order from lowest to highest. Or in an undetermined order of the keys. Note: If the number of indexes or keys changes during the loop, the new key or index will be included as part of the initialization of the next repeat.
ArrayVariable	The array variable whose indexes or keys you want to loop over.
ArrayPath	The path of the array key which contains the indexes or keys you want to repeat over. Expressed in '/key1/key2/' format.

Argument	Description
IndexKeyVariable	The variable to hold the current index or key. Note: When using repeats inside repeats,this variable must be different.

76.19 switch property

Begins a switch block where the value of a property is compared.

Argument	Description
Element	The element to have its property compared.
Property	The attribute that is to be compared.

76.20 switch variable

Begins a switch block where the value of a variable is compared.

Argument	Description
Variable	The variable to use in the SWITCH control structure

76.21 try

Runs the code within its block, and returns and runs the catch block if an execution error occurs.

77 Actions: Search

77.1 clear search

Resets the Search Element.

Argument	Description
Element	The search element to clear.

77.2 get search input

Retrieves the input of the Search Element.

Argument	Description
Element	The search element whose input will be returned.
Output	The location where the result will go (variable or field or text element).

77.3 perform search

Commands a search element to perform a search query.

Argument	Description
Element	The search element that will perform the search query.

78 Actions: Sockets

78.1 get my IP

Retrieves the IP address of the device.

Argument	Description
Output	The location where the IP address will go (variable or field or text element).

78.2 socket client connect to host

Opens a connection that can be used for socket communication.

Argument	Description
IP Address	Input, variable, text or field element containing the IP address of the host.
Port	Input, variable, text or field element containing the port to be used when opening the connnection.

78.3 socket client get message

Retrieves the message received by a socket client.

Argument	Description
Output	The location where the message will go (variable or field or text element).

78.4 socket client get sender IP

Retrieves the IP address of the sender of a socket message.

Argument	Description
Output	The location where the IP address will go (variable or field or text element).

78.5 socket client get sender port

Retrieves the port used by the sender of a socket message.

Argument	Description
Output	The location where the port will go (variable or field or text element).

78.6 socket client send message

Sends data to a socket.

Argument	Description
Port	Input, variable, text or field element containing the port of the receiver.
Data	Input, variable, text or field element containing the data to be sent.
IP Address	Input, variable, text or field element containing the IP address of the receiver.

78.7 socket client stop

Closes a socket connection.

78.8 socket host accept connections

Accepts an internet connection and creates a socket for that connection.

Argument	Description
Port	Input, variable, text or field element containing the port to be used to accept connections on.
AuthRequired	Whether new connections need to authenticate before sending other messages (True) or not (False).

78.9 socket host get last client IP

Retrieves the IP address of the last client to disconnect.

Argument	Description
Output	The location where the IP address will go (variable or field or text element).

78.10 socket host get last client port

Retrieves the port used by the last client to disconnect.

Argument	Description
Output	The location where the port will go (variable or field or text element).

78.11 socket host get message

Retrieves the message received by a socket host.

Argument	Description
Output	The location where the message will go (variable or field or text element).

78.12 socket host get sender IP

Retrieves the IP address of the sender of a socket message.

Argument	Description
Output	The location where the IP address will go (variable or field or text element).

78.13 socket host get sender port

Retrieves the port used by the sender of a socket message.

Argument	Description
Output	The location where the port will go (variable or field or text element).

78.14 socket host send message

Sends data to a socket.

Argument	Description
Port	Input, variable, text or field element containing the port of the receiver.
Data	Input, variable, text or field element containing the data to be sent.
IP Address	Input, variable, text or field element containing the IP address of the receiver.

78.15 socket host stop

Closes all open socket connections.

79 Actions: Camera

79.1 clear captured photos

Clears out the temporary photos captured or selected from a camera element.

79.2 create qr code

Creates a QR code in an image element with the given data.

Argument	Description
Image	The image element where the QR code will be displayed.
Data	Input, variable, text or field element containing the data to be in the QR code.

79.3 display captured photo

Displays a captured or selected mobile gallery photo from the camera element in the selected image element.

Argument	Description
Photo	The variable containing a single, temporary photo ID of a captured or selected mobile gallery photo from a camera element.
Image	The image element where the camera element's captured or selected mobile gallery photo will be displayed.

79.4 get captured photos

Outputs a line-delimited list of temporary IDs of the captured or selected photos from a camera element.

Argument	Description
Output	A variable to store the returned list of temporary IDs of the captured or selected photos from a camera element.

79.5 save photos to database

Saves the captured or selected mobile gallery photo(s) from the camera element to the database.

Argument	Description
Output	An optional variable to store the blob ID of the photo that was saved to the database.
Photos	The variable containing a temporary, photo ID or a line-delimited list of photo IDs of the captured or selected photo(s) from a camera element.
Target	The place to save the photo to. (local or cloud).

79.6 select photo

Allows users to select an existing photo from a library and use the image element's dimensions or modify the size.

Argument	Description
Image	The image element where the selected photo will be displayed.
Target	The place to save the image (local or cloud).
Width	The width for the selected image.
Height	The height for the selected image.

79.7 set camera output

Sets the output element for a camera element.

Argument	Description
AspectRatio	The aspect ratio for the image.
Camera	The camera element to have its output adjusted.
Image	The image element where the camera output will be displayed.

80 Actions: Player

80.1 go to player home

Exits the current app and returns to Appli Player's home screen. This action has no effect in Appli IDE.

80.2 quit player

Quits Appli Player and returns to the device's home screen. This action has no effect in Appli Builder.

80.3 refresh app

Refreshes the app that's currently loaded in Appli Player. This action has no effect in Appli IDE.

80.4 show player menu

Displays the in-app menu in Appli Player. The menu can be closed by clicking outside of it. This action has no effect in Appli IDE.

81 Actions: AI

81.1 GPT chat completion

This submits the prompt to the GPT Chat Completion endpoint. It includes also, the previous messages and responses in the conversation. These can be accessed independently using the ‘GPTConversations’ array functions. The documentation for this endpoint can be found here: https://platform.openai.com/docs/api-reference/chat/create.

Argument	Description
ResponseOutput	Where the resulting response text should be be placed.
ResponseSchema	JSON Schema for structured response. To learn more, and to use our helper bot, visit https://platform.openai.com/docs/guides/structured-outputs/
ConversationName	The name that will be used to track this conversation. This name gives you the ability to access the conversation’s responses directly using the ‘GPTConversations’ variable array.
Model	The model that should be used for the request. If you use ‘chatgpt-4o-latest’ your code will always use the latest 4o model. Visit https://platform.openai.com/docs/models for more info.
BasePrompt	The prompt which defines the desired behavior of the AI. This is where you tell the AI what role to play in this conversation. Describe it in as much detail as you can. For example: You are a world class designer, and you will help me design an app based on proper design rules and conventions.
RequestPrompt	Where the user’s next prompt should come from

81.2 GPT vision

This sends an image to the OpenAI Vision API, returning the GPTs response to your image based on the prompt. The call is synchronous.

Argument	Description
ResponseSchema	JSON Schema for structured response. To learn more, and to use our helper bot, visit https://platform.openai.com/docs/guides/structured-outputs/
ResponseOutput	Where the response text will be placed.
DetailMode	The input detail mode. Low = 512x512, High = 768x2048 or 2048x768. Input images are resized to those dimensions by OpenAI. It is best to aim to send the appropriate image dimensions. Limit per image is 12 MB.
ImageData	The binary data of the image. Your image will be resized based on the detail level you selected. When this variable is a numerical array, we use will use 'ImageData' as the key that contains the ImageData
ResponseModel	The GPT model to use to generate the text response.
TopicName	The name that will be used to track calls. This name gives you the ability to access the returned data directly using the 'GPTVision' array variable.
RequestPrompt	The prompt for the request. ie: What is in the image?
BasePrompt	The prompt which defines the desired behavior of the AI. This is where you tell the AI what role to play in this conversation. Describe it in as much detail as you can. For example: You are a world class designer, and you will help me design an app based on proper design rules and conventions.

81.3 clear GPT conversation

Clear the content of a GPT Conversation.

Argument	Description
ConversationName	Select the GPT conversation to be cleared.

81.4 convert text to speech

This converts text to speech using OPENAIs Text To Speech API. Documentation for this endpoint can be found here: https://platform.openai.com/docs/api-reference/audio/createSpeech.

Argument	Description
Response	Where the resulting audio data be be placed
Text	The text to be turned into audio.
Model	The model that should be used for the request.
Voice	The OPENAI TTS Voice to be used.
Speed	The speed of the speech.

81.5 generate image

Generates an image using OpenAI API and places the image in a variable or image. The documentation for this endpoint can be found here: https://platform.openai.com/docs/api-reference/images.

Argument	Description
ResponseFormat	Return Image BINARY Data or a URL.
Resolution	The resolution of the resulting image
RequestPrompt	The prompt for the image generation request. This is where you describe the image you wish to generate in as much detail as you can.
Style	Generation style to use.
ConversationName	The name that will be used to track this conversation. This name gives you the ability to access the conversation's array directly using the GPTConversations array.
Quality	The quality setting to use for the request. 'standard' produced lower quality results, and 'hd' produces the highest quality images the DALE-3 model is capable of.
ResponseOutput	Where the URL or image data should go.

82 Actions: Code

82.1 custom code

Allows code created in LiveCode to run in Appli.

Argument	Description
Select	Select the code you wish to run.

83 Actions: Report

83.1 clear report

Clears the data displayed in a report.

Argument	Description
Element	The report element to be cleared.

83.2 print report

Adds the specified report element to a print job. Either send to a printer (desktop only) or save as a PDF.

Argument	Description
Filename	The filename of the PDF to be saved if the ‘output’ argument is set to ‘PDF’. If this argument is empty, a default name with the app’s name and timestamp will be used.
Result	The action taken by the user when presented with the print dialog if the ‘output’ argument is set to ‘printer’. ‘True’ will be returned if the user proceeds with the print job. Otherwise ‘false’ will be returned.
RightMargin	Width of the right margin in inches.
Type	The type of print job, either sent to a printer (desktop only) or saved as a PDF.
Orientation	The orientation of the printed content.
TopMargin	Width of the top margin in inches.
Element	The report element to be printed.
LeftMargin	Width of the left margin in inches.
BottomMargin	Width of the bottom margin in inches.

83.3 refresh report

Refreshes the data displayed in a report.

Argument	Description
Element	The report element to be refreshed.

83.4 report current page

Returns the page number of the current page of a report.

Argument	Description
Element	The report element to have its current page number retrieved.
Output	The location where the page number will go (variable or text/field element)

83.5 report first page

Display the first page of a report.

Argument	Description
Element	The report element to display its first page.

83.6 report go to page

Display the specified page of a report.

Argument	Description
Element	The report element to display the specified page.
PageNumber	The page number corresponding to the page to display (input, variable, text/field element)

83.7 report last page

Display the last page of a report.

Argument	Description
Element	The report element to display its last page.

83.8 report next page

Display the next page of a report.

Argument	Description
Element	The report element to display its next page.

83.9 report number of pages

Returns the number of pages in a report.

Argument	Description
Element	The report element to have its number of pages retrieved.
Output	The location where the number of pages will go (variable or text/field element)

83.10 report previous page

Display the previous page of a report.

Argument	Description
Element	The report element to display its previous page.

83.11 reset report

Removes the database connections associated with the report and clears the data displayed.

Argument	Description
Element	The report element to be reset.

83.12 set report column names

Sets the display name for each column.

Argument	Description
Element	The report element to have its column name(s) set.

Argument	Description
ColumnNames	A comma-delimited list of the name corresponding to each column.

83.13 set report column sizing

Specify how column(s) in a report should be sized.

Argument	Description
Element	The report element to have its column sizing set.
ColumnSizing	A selection from the available sizing options.

83.14 set report group key

Specify which key to group the report by.

Argument	Description
Element	The report element to have its group key set.
Key	The key to group the report by.

83.15 set report keys

Sets the key(s) that a report is connected to.

Argument	Description
Element	The report element to have its key(s) set.
Keys	The key(s) to be connect to.

83.16 set report linked keys

Sets the key(s) that a report is linked to for relational data.

Argument	Description
Element	The report element to have its linked key(s) set.
LinkedKeys	The key(s) to be linked to.

83.17 set report linked table

Sets the table that a report is linked to for relational data.

Argument	Description
Element	The report element to have its linked table set.
LinkedTable	The table to be linked to.

83.18 set report linking key

Sets the key used for relational data.

Argument	Description
Element	The report element to have its linking key set.
LinkingKey	The key used to link data between tables.

83.19 set report recordIDs

Sets the recordID(s) that a report is connected to.

Argument	Description
Element	The report element to have its recordID(s) set.
RecordIDs	A line delimited list where each line is the cdbRecordID of a record to be displayed in the report. Can use "*" to specify all records.

83.20 set report sort direction

Specify which direction a report should be sorted.

Argument	Description
Element	The report element to have its sort direction set.
SortDirection	The direction to sort the report (ascending or descending).

83.21 set report sort key

Specify which key to sort the report by.

Argument	Description
Element	The report element to have its sort key set.
Key	The key to sort the report by.

83.22 set report sort type

Specify the data type used to sort the report.

Argument	Description
Element	The report element to have its sort type set.
SortType	The data type of the sort key (binary, dateTime, international, numeric, text).

83.23 set report table

Sets the table that a report is connected to.

Argument	Description
Element	The report element to have its table set.
TableName	The table to be connect to.

83.24 set report target

Specify where to access the data connected to the report.

Argument	Description
Element	The report element to have its target set.
Target	The place to access the data (local or cloud).

83.25 setup report

Complete setup for a report element.

Argument	Description
Target	The place to access the data (local or cloud).
LinkedTable	A table containing the LinkingKey.
LinkingKey	A key used to link data between tables.
GroupKey	The key to group the report by.
SortKey	The key to sort the report by.
SortType	The data type of the sort key (binary, dateTime, international, numeric, text).
SortDirection	The direction to sort the report (ascending or descending).
LinkedKeys	The key(s) to display from the LinkedTable.
TableName	The table to be connect to.
ColumnNames	A comma-delimited list of the name corresponding to each column.
Keys	The key(s) to be connect to.
ColumnSizing	A selection from the available sizing options.
RecordIDs	A line delimited list where each line is the cdbRecordID of a record to be displayed in the report. Can use "*" to specify all records.
Element	The report element to be set up.

84 Actions: Appli Connect

84.1 appli connect action

Call external functions and APIs from 3rd party providers using easy wizard like setup.

Argument	Description
Label	The label for this Appli Connect Action
Configure	Easily define calls to our ever growing library of 3rd party API endpoints, using a wizard style interface.
StatusFormat	How should your request's progress be returned?
StatusUpdate	The element or the variable where the progress update of your request is placed.

84.2 rest api call

Define a generic REST API call.

Argument	Description
ResposeCode	The array variable which contains the server's response code.
StatusFormat	How should your request's progress be returned?
ResponseBody	The array variable which should contain the result of the request. If JSON is returned, it is converted to an Appli Array. If it is not JSON, the returned value is placed there instead.
EndpointURL	The full endpoint to be used for the call
ParamsBodyFormat	Send body of the request sent as JSON or as URL Parameters
RequestBody	The array variable which contains the parameters / body of the request. This is array is converted to JSON and used as the payload/body of the request.
Method	The HTTP method to be used for the call
CustomHeaders	The array variable which contains the custom HTTP headers to use for the call. Most, not all APIs require this.
ResponseHeaders	The array variable which contains the headers of the server's response.
StatusUpdate	The element or the variable where the progress update of your request is placed.

85 Actions: Appli

85.1 get app names

Retrieves a line-delimited list of the current project's app names.

Argument	Description
Output	The location where the result will go (variable or text or field element).

85.2 get screen names

Retrieves a line-delimited list of the current app's screen names.

Argument	Description
Output	The location where the result will go (variable or text or field element).

86 Image Credits

- Header Photo credits: Fahmi Fakhrudin.
- Redemption Roasters photo from their website.

87 DOCUMENTATION CHANGELOG

Recent chapter changes. For Appli Builder changelog, check here.

- actions-ai.html
- actions-blobs.html
- actions-camera.html
- actions-disk-io.html
- actions-dropdown.html
- actions-layout.html
- actions-math.html
- actions-navigation.html
- actions-sockets.html
- actions-user-accounts.html
- actions-utility.html
- actions-variables.html
- appli_changelog.html
- changelog.html
- events.html
- low-code.html
- tutorial-code-reuse.html
- tutorial-relational-data.html

88 APPLI CHANGELOG

Recent Appli changes. For documentation changelog, check here.

88.1 VERSION 1.4.22 (2025-06-04):

- **New:** Report Element has been added to the list of "Container Elements" in the header.
- **New:** Report element. The report element represents a report with a header, body, and footer section. The header and footer sections are resizable, can collect elements as part of a group, and each section can be

enabled or disabled. The body section can be setup to display paginated records from a table. A linking key can be set to link records between tables. Records can be sorted and grouped. Column names are customizable and columns can be live resized. There is a line number column that can be enabled or disabled and the column name is customizable. The element has a nocode section and actions that can be used to setup the report. Report elements can be saved as a pdf or printed.

- **Improved:** Custom Code editor will now successfully compile your custom code when a handler, key name or a string in your script match the name of a variable.
- **Improved:** Syntax highlighting will now recognize most of the occurrences of a handler or a string matching a variable name. *Note: we are still working on fixing syntax highlighting for key names which match a variable. In such cases, your script will work, and you can safely ignore the highlighting.
- **Improved:** We added error detection for structured output JSON schemas in ChatGPT actions.
- **Improved:** New engine for Appli. This allows for further improvements and performance increase in the future.

88.2 VERSION 1.4.21 (2025-05-28):

- **New:** "Set text record ID" action in the "text" category to set the record ID that a text element is connected to if a table has been linked.
- **New:** "Refresh text" action in the "text" category to refresh the contents of a text element that is connected to a table and record.
- **Improved:** "Structured Output" in ChatGPT Chat Completion & Vision endpoints. Please see dictionary text for help and link.
- **Improved:** "JSONToArray" function is now more resilient to character encoding issues. In case of an error, conversion tries again after changing your string to UTF-8.
- **Improved:** Updated the "sync records" action to include "cdbBlobs" in the list of table names in the "TableName" argument.
- **Improved:** Added "sort" argument to "display records in layout" action to allow the option of sorting records based on the layout element's sort settings or maintaining the order of the specified record IDs.
- **Improved:** Arrays can now be edited in the data viewer. Double-click on an array value to edit the value.
- **Improved:** Variables can now be filtered by name in the Variable Viewer.
- **Fixed:** Fixed a bug in the text element's no-code setup where the templated text didn't populate the editor.

88.3 VERSION 1.4.20 (2025-05-14):

- **Improved:** Tables can now be unlinked in the no-code setup for text elements.

- **Fixed:** Fixed an error that could occur if an element was deleted while its low-code script was open.
- **Fixed:** Fixed a bug that prevented layout elements that don't have rows from scrolling under certain conditions.
- **Fixed:** Fixed a bug that may cause low-code events to dispatch multiple times.

88.4 VERSION 1.4.19 (2025-05-07):

- **Improved:** Improved parsing of custom SVG files to extract shape from differently formed SVG files.
- **Fixed:** Cancelling the custom event name dialog in the low-code editor will no longer display an error.
- **Fixed:** When setting the 'disabled' property of a new button, the boolean option was not showing. (Clarence)
- **Fixed:** Fixed a bug which could cause Appli to hang when saving to the cloud while offline.

88.5 VERSION 1.4.18 (2025-04-30):

- **Improved:** Button element can now use a custom SVG icon. Note: The SVG must be single unified path.
- **Improved:** The button element icon can now have a custom colour and does not need to be the same as the button's label.
- **Fixed:** Fixed an error that may occur in the low-code editor when adding an action after using undo.

88.6 VERSION 1.4.17 (2025-04-23):

- **Improved:** When holding Control and clicking TableName argument of query actions, the list will include cdb tables.
- **Fixed:** Fixed operator in FileType detection handler to avoid collisions.

88.7 VERSION 1.4.16 (2025-04-16):

- **New:** "Print text" action in the "printing" category. Use this action to print all text in a field or text element.
- **Improved:** We've added GPT 4.1 models to our Chat Completion and Vision actions. 4.5 preview has been removed, as it is leaving the API soon.
- **Improved:** Rectangle element now has a 'dashed border' property. A value of '10,5' will use dashes 10 pixels long with a 5 pixel space between them.
- **Fixed:** Fixed a bug where variables may be deleted if they were modified in the variable viewer while using filters.

88.8 VERSION 1.4.15 (2025-04-09):

- **Fixed:** Retain user variables when loading apps from disk.
- **Fixed:** Camera Element scale factor settings would sometimes cause issues in the player.

88.9 VERSION 1.4.14 (2025-04-02):

- **Fixed:** Zooming was not maintaining scroll position in certain circumstances.
- **Fixed:** Loading query element would sometimes have a reference error.

88.10 VERSION 1.4.13 (2025-03-26):

- **New:** “Remove whitespace” action in the “utility” category to remove leading and trailing whitespace from the specified input.
- **Improved:** Support “cdbDateCreated” and “cdbDateModified” keys in the layout element’s no-code setup when linking to an element or used in templated text.
- **Improved:** Support “cdbDateCreated” and “cdbDateModified” keys in the layout element’s no-code setup for sorting records.

88.11 VERSION 1.4.12 (2025-03-19):

- **New:** “Match Property Action” action lets you match 2 properties. 1 is the source, the other is the target. The offset argument gives you even more control.
- **New:** “Get Playground Info” action populates an array variable with useful information about the state of the Appli playground.
- **Improved:** “Allow variables to be used in the”Key” argument of the “look up value” action.
- **Improved:** “Allow variables to be used in the”TableName” argument of the “look up value” action.

88.12 VERSION 1.4.11 (2025-03-12):

- **Improved:** “Allow variables to be used in the”TableName” argument of the “create record” action.
- **Improved:** “Allow variables to be used in the”TableName” argument of the “update record” action.
- **Improved:** “Allow variables to be used in the”TableName” argument of the “sync records” action.
- **Improved:** “Allow variables to be used in the”TableName” argument of the “get table ID” action.
- **Improved:** Fixed Appli’s handling of a bug in the OpenAI API.
- **Improved:** Improved recognition when a file being sent to the Vision API is not an image.

- **Fixed:** Empty user input in a search element will now display all records when outputting to a layout element.

88.13 VERSION 1.4.10 (2025-03-05):

- **Improved:** Updated OpenAI model lists, including the addition of gpt-4.5-Preview. Caution: The price of inference of 4.5 preview is very high.
- **Improved:** GPT vision can now take in a set of images in a single prompt. Instead of giving the action the Image binary data, you. can now give it an array with indexes that contain a key "ImageData" for each each image you wish to upload. As before, the ImageData key in each index should contain the binary data of the image.

88.14 VERSION 1.4.9 (2025-02-27):

- **Improved:** Fixed a problem with dropdowns which have their options set using the two pipes delimiter '||'
- **Improved:** Save scroll location when expanding/collapsing project browser elements
- **Fixed:** Fixed an issue where elements within a Layout element would not properly delete and/or display records.
- **Fixed:** Fixed a bug where the "wait" action wouldn't work with variables.
- **Fixed:** Fixed a bug that could prevent nocode changes from saving for layout elements.

88.15 VERSION 1.4.8 (2025-02-19):

- **Improved:** Internal speed and stability improvements.
- **Fixed:** Fixed a bug which could prevent an app from loading.

88.16 VERSION 1.4.7 (2025-02-12):

- **New:** "count down timer ended" event. Use this event to perform an action when a count down reachers 0.
- **Improved:** Added "o3-mini" to the list of GPT Chat Completion models
- **Fixed:** Fixed a bug that could occur when using "count down timer" action.
- **Fixed:** Fixed a bug that would allow an event to be added without an event selected.

88.17 VERSION 1.4.6 (2025-02-05):

- **New:** "count down timer" and "stop count down timer" actions. Use these actions to start and stop a count down on a field or text element.

- **Improved:** When closing Appli, or closing an App by returning home, you will be prompted to save any unsaved lowcode tabs, and all tabs will be closed.

88.18 VERSION 1.4.5 (2025-01-29):

- **Improved:** Deleting elements from the playground will now show feedback to the user.
- **Improved:** When the Lowcode editor is the most recently focused pane tool, and ESC is pressed, you are prompted to save changes the same way you would if you had clicked X or Close all Tabs.
- **Improved:** Support for image element added to Form Element's No-Code binding.
- **Improved:** Improved performance when switching screens in play mode.
- **Fixed:** Fixed a bug where icons in the radio element don't resize when the zoom level is changed.

88.19 VERSION 1.4.4 (2025-01-22):

- **New:** "Scanner" keyboard type for field elements. Using this keyboard type along with actions, a field can be setup for hands-free processing when scanning multiple consecutive barcodes.
- **New:** "Line Number Column" and "Column Name" properties for the report element. Use these properties to enable and name a line number column.
- **Improved:**
- **Fixed:** Fixed a bug that prevented the buttons in the radio element from being selected if icons were used for the options.
- **Fixed:** When cutting lowcode actions, the editor now correctly updates the UNDO count.
- **Fixed:** Fixed a bug where the label in the radio element retained its text size if the zoom level was changed.
- **Fixed:** Fixed a bug where the low-code editor may remain visible after loading an app.
- **Fixed:** The low-code dispatcher ran element code in elements when it should only run openScreen code. This made opening a new screen appear slower than it should have been.
- **Fixed:** Fixed a bug that would cause the wrong records to be displayed in a report element.
- **Fixed:** Fixed a bug where opening a new tab in the low-code editor may undo changes to an action's argument in the current tab.

88.20 VERSION 1.4.3 (2025-01-15):

- **New:** Introducing Lowcode Editor tabs. Now you can have more than 1 lowocde script open and switch between them using tabs. Upon closing a tab or closing the lowcode editor, you will be prompted to save your unsaved changes.

- **New:** "Get list of Keys" action enables you to extract a return delimited list of the keys of an array variable.
- **New:** Text action to compose a message in SMS.
- **New:** "Get dropdown records list" action in the "dropdown" category to retrieve a line-delimited list of record IDs of the options in a dropdown element that has been connected to a database table.
- **New:** Array support for layout elements. Data context setup now has an "array" option.
- **New:** "ArrayToJSON" action turns an Array Variable into a UTF-8 encoded JSON String. "JSONToArray" action turn a valid JSON string into an Array Variable.
- **New:** 'selectPhoto.action' allows user to select an existing photo from a library to an image element.
- **Improved:** The options of a dropdown element can now be scrolled with a mouse wheel in play mode.
- **Improved:** Added HTMLText to the available properties of the Browser element. Now you can set or get the full content of the browser element's current webpage.
- **Improved:** Added an option to open an element's documentation from its contextual menu.
- **Fixed:** Fixed a bug that would occur if the variable used with the "go screen" action is a screen that does not exist.
- **Fixed:** Fixed a bug where moving an element within its parent element may cause it to relayer to the front.
- **Fixed:** Fixed a bug where changing the "password field" property of the field element may not correctly hide or show the password.
- **Fixed:** Fixed a bug with element selection in the query element no-code window.
- **Fixed:** Fixed bug where events like openscreen are needlessly sent to screens that are not the current screen.

88.21 VERSION 1.4.2 (2024-12-18):

- **Improved:** Made improvements to dragging elements to the bottom of the project browser.
- **Improved:** The answer file dialog importing exported Appli projects on disk will now allow you to select exported Appli apps.
- **Improved:** If a sorted line delimited list of recordIDs is provided in the 'Display Records In Layout' action, the order will be maintained in the layout element.
- **Fixed:** Fixed a bug which occured in large apps and large scripts where the lowcode script was not always successfully applied to its element.

88.22 VERSION 1.4.1 (2024-12-11):

- **New:** Custom lowcode events have been added. Double click or add "custom" event and give it a name. Then you can use "dispatch event" action to call it. (Must select "custom" and set CustomEvent argument)
- **Improved:** When you want to add actions to a specific event, just click its header before adding. Also, after adding a new event, it becomes the destination of all new actions until another action changes the context.
- **Improved:** GPT Vision and GPT Chat Completion actions now use the same GPTConversations array so that the two actions can part take in the same conversation. (Now your GPT Vision queries can be in context of a larger conversation.)
- **Fixed:** Fixed a bug where undo points were not made when performing some lowcode actions.
- **Fixed:** If you send just a 'number' to GPTChatCompletion or GPTVision, it no longer returns an error asking for a string.
- **Fixed:** Fixed a bug where a minimized window pane could still be interacted with.
- **Fixed:** Fixed a bug that could cause Appli to hang when switching modes after using the "print report" action.

88.23 VERSION 1.4.0 (2024-12-04):

- **New:**
- **Improved:** Fixed a reference bug with dispatching lowcode events.
- **Improved:** Code editor got another round of love. We added line numbers and further improved the compilation of custom code.
- **Fixed:** In certain circumstances references to appli elements in custom code would not work correctly.
- **Fixed:** Fixed an issue with indentation of custom code.
- **Fixed:** Custom code editor did not always recognize that a script has been changed again after it has been saved.
- **Fixed:** Fixed a visual bug that occurs when toggling full window view in the data viewer while in the array key view.

88.24 VERSION 1.3.39 (2024-11-13):

- **Improved:** Updated the "dispatch event" action to support screens.

88.25 VERSION 1.3.38 (2024-11-06):

- **Improved:** Expanded the list of selectable events in the "event" argument of the "dispatch event" action.
- **Improved:** Pressing the backspace key in the layout element's no-code template editor will now remove the entire key reference if applicable.
- **Fixed:** Fixed a bug in the "dispatch event" action where the "result" argument would not be populated after execution.

- **Fixed:** Fixed a bug where the inserted key in the layout element's no-code template editor may not match the specified key name.

88.26 VERSION 1.3.37 (2024-10-30):

- **New:** "Export as CSV" action to export the provided column headers and row data as a CSV file.
- **New:** "Reindex Array" action lets you reorder a numerical index that has had keys removed or added to it. "Timer Log" action lets you time how long things take and log the execution flow of your code.
- **Improved:** Added a "cancel" button to the template text editor in the layout's no-code
- **Improved:** All running repeat loops in Lowcode scripts will exit (as soon as possible) when returning to Edit mode. Requires re-applying the low code script.
- **Improved:** No-code option is now available in the element's right-click contextual menu if applicable.
- **Fixed:** Fixed a bug where the create and edit interface in the data viewer's users table may get cut off.
- **Fixed:** Editing template text in the layout's no-code window will no longer save immediately.
- **Fixed:** "repeat over array" action will now correctly indent its code block.

88.27 VERSION 1.3.36 (2024-10-23):

- **New:** "Update user record" action in the "user accounts" category to update a record of an existing user in the 'cdbUsers' table.
- **New:** "Resend verification email" action in the "user accounts" category to send a verification email to an existing user who has not been verified.
- **New:** "Delete account" action in the "user accounts" category to delete a user account.
- **Fixed:** The screen gallery is now automatically hidden when switching to play mode.

88.28 VERSION 1.3.35 (2024-10-16):

- **New:** "GPT Vision" action has been added. This action allows you to send an image to Chat GPT's Vision API.
- **New:** "Process Variable" action has been added. Available functions are: base64encode / decode, binary encode / decode, compress / decompress.
- **New:** "Batch create blob", "batch read blob", "batch update blob", and "batch delete blob" actions in the "blobs" category.
- **Improved:** Updated the icon that indicates an item can be dragged and moved.
- **Improved:** Improvements were made to table arrangement in the data modeler.

- **Fixed:** Fixed a bug in the “ask for file path” action which requested a location to save a file instead of an existing file.
- **Fixed:** Fixed a bug where cancelling the color dialog in the data modeler would change the color of the selected table.
- **Fixed:** Fixed a bug that could prevent records from being displayed in a layout element.
- **Fixed:** Correct sizing for a layout element’s scrollbar while in edit mode.
- **Fixed:** Fixed a bug where apps may not load with the last used orientation.
- **Fixed:** Fixed a bug which could cause elements in a layout element to draw out of place when an app was loaded.
- **Fixed:** Layout elements with “multiple rows” disabled can now scroll to the bottom.
- **Fixed:** “Learn more” text in the hover assist tooltips is now hidden if there’s no outgoing link.

88.29 VERSION 1.3.34 (2024-10-09):

- **New:** Added “clear current orientation” to the screen’s right-click contextual menu which will delete all elements on the current screen’s current orientation.
- **Improved:** Updated the list of GPT AI models available for the chat completion lowcode action.
- **Improved:** Updated the “ask for folder” action’s dictionary entry for clarity.
- **Fixed:** A bug in “save photos to database” lowcode action has been fixed.
- **Fixed:** Fixed a bug with the “list files” action where the output was incorrect if a variable was used in the “folder name” argument.
- **Fixed:** No longer default to all records if the “record IDs” argument in the “delete record” action is empty.
- **Fixed:** Fixed issues with “read blob” lowcode action.
- **Fixed:** Fixed issues with “create blob” lowcode action.
- **Fixed:** Fixed a bug which could cause the “create qr code” action to create the qr code in the wrong image element.
- **Fixed:** Fixed a bug where clicking in and out of the asset manager’s text size field would result in an error.

88.30 VERSION 1.3.33 (2024-10-02):

- **New:** “Set platform color” in the “utility” category to set the color of the platform’s background.
- **New:** “Get platform color” in the “utility” category to retrieve the color of the platform’s background.
- **Fixed:** Fixed a visual bug in the layout element that can occur if the zoom level is changed.

- **Fixed:** Fixed a bug where an empty "path" argument in the "count indexes in array" action would produce an incorrect result.
- **Fixed:** Fixed a bug which would cause layout elements with multiple rows enabled to display the same record in all rows.

88.31 VERSION 1.3.32 (2024-09-25):

- **New:** Lowcode action "Add 1 To Variable" is an easy way to increment a variable by 1
- **New:** Lowcode action "Get Key From Variable" reads a specific key from a variable which has array keys
- **New:** Lowcode action "Get Item From Variable" reads a specific item from a variable which contains one or more items separated by a comma or a specific character or string.
- **New:** Lowcode action "Save File To Path", saves a variable as a file to a specific path on the file system. "Load File From Path" loads a specific file from the file system into a variable. Optionally, both actions let you specify the mode of the operation: Text or Binary.
- **Improved:** Added input option to the "record IDs" argument in the "read records" action.
- **Fixed:** Fixed a bug in Select File action where an unfilled FilenameVariable argument created an Appli variable whose name is empty.
- **Fixed:** Components will now stay in sync when using "set value" action.

88.32 VERSION 1.3.31 (2024-09-18):

- **New:** New event and actions for sockets. Use event "socketHostClientDisconnected" along with actions "socket host get last client IP" and "socket host get last client port" to manage socket connects.
- **Improved:** "Path" argument for "count indexes in array" action. Can now provide a path for accessing nested arrays.
- **Improved:** New default property values for shadows and glows applied to elements.
- **Fixed:** Output for "count indexes in array" action could be incorrect.
- **Fixed:** "Element" was not an output option for "count indexes in array" action.
- **Fixed:** Dictionary entry for "count indexes in array" action.
- **Fixed:** Fixed a bug which could prevent all variables from being displayed when using the "delete array element" action.

88.33 VERSION 1.3.30 (2024-09-11):

- **Improved:** Modified desktop calendar to reflect updated appearance.
- **Improved:** Remove auto resizing of text in a button's label.
- **Improved:** Added compatibility for variables to be used in the search element's no-code input and output values.

- **Improved:** Unified some of the visual attributes of Appli containers to match on creation.
- **Improved:** Adjusted the default properties of layouts to use the most common settings that once needed to be set manually.
- **Fixed:** When clicking a locked field, a second mouseUp was being sent, which would cause the low code in that element to run twice.

88.34 VERSION 1.3.29 (2024-09-04):

- **New:**
- **Improved:** Improved responsive resize for button elements.
- **Improved:** The result provided from the sync action will now return true (good) or false (bad).
- **Improved:** Added dictionary entries for events to the low-code editor.
- **Improved:** Added mouse wheel scrolling to action and event panes in the low-code editor.
- **Improved:** Updated the rectangle element to be selectable via a double-click inside containers.
- **Fixed:** Fixed a bug with default row height of Layout elements. The default row height is now set to 50 pixels when "multiple rows" is activated.

88.35 VERSION 1.3.28 (2024-08-28):

- **New:** "Get character count" action in the "utility" category to retrieve the number of characters in the specified text.
- **New:** "Get text offset" action in the "utility' category to retrieve the character position of the first occurrence of the text to find in the text to search.
- **Improved:**
- **Fixed:** Elements that use visual effects in a layout may find their elements to not fully render the shadow or other effects.
- **Fixed:** Fixed a bug where changing the "text" property of a text element component with low-code actions wouldn't sync to all components.
- **Fixed:** Fixed a bug that can occur with saving custom code scripts.

88.36 VERSION 1.3.27 (2024-08-21):

- **New:** Create UUID action creates a v4 uuid made of 36 characters.
- **Improved:** Fixed Appli Connect bugs having to do with endpoint URL variables.
- **Fixed:** Fixed Appli Connect bug with keys that have an element input. The "full response variable" is emptied before the new result is written.

88.37 VERSION 1.3.26 (2024-08-14):

- **New:** Introducing Appli Connect, our newest low code action / input method. Appli Connect makes it easier than ever to work with external web apis and web services from any provider. Check out our demo app to see our mini Dropbox app & more.
- **New:** Helper action "Delete Empty Variables" which clears all empty Appli Variables. Useful in cleaning unused variables. Variables which your app populates will reappear.
- **Improved:** Display of long project and app names.
- **Improved:** Added a "pull to refresh" property to layout elements to control whether pulling down on the element on mobile devices will trigger a refresh.

88.38 VERSION 1.3.25 (2024-08-07):

- **New:** "Quit player" action in the "player" category to quit Appli Player and return to the device's home screen.
- **New:** "Get screen names" action in the "Appli" category to retrieve a line-delimited list of the current app's screen names.
- **New:** "Get app names" action in the "Appli" category to retrieve a line-delimited list of the current project's app names.
- **New:** "Create QR code" action. Creates a QR code in an image element with the given data. QR codes can be scanned with a camera element set up as a barcode scanner. The result will go into the assigned variable.
- **Improved:** Added convenient optional arguments to new file selection actions.
- **Improved:** The "disabled" property has been added. When set to true, the element is "greyed out" and all mouse / keyboard events are ignored.
- **Improved:** Added "duplicate" option to the screen contextual menu in the footer.
- **Fixed:** Fixed a bug which could cause the input field to draw over the hint label in the create account and login elements.

88.39 VERSION 1.3.24 (2024-07-31):

- **New:** "Select File" action lets you ask the user to select one or more files.
- **New:** "Ask For Folder" action lets you ask the user to select a folder.
- **New:** "Ask for File Path" action lets you ask user to select a filename and path.
- **New:** "Orientation changed" event that can be added to the low-code of screens.
- **New:** "Get orientation" action to retrieve the current orientation of the screen.
- **Improved:** Dialogue and interaction in the onboarding guide for new accounts.

- **Fixed:** Fixed a bug where commented actions that have an argument populated with an array would still run.
- **Fixed:** Fixed a bug where text elements with “auto height” sizing may change in width and x-coordinates when their contents are changed.
- **Fixed:** Fixed regression bug with repeat_over_array action. It will now work as expected.

88.40 VERSION 1.3.23 (2024-07-24):

- **New:**
- **Improved:**
- **Fixed:** Fixed a bug where “set property in layout row” and “set property in row from variable” actions would change the properties of the base elements in a layout element.
- **Fixed:** When typing in a cell of the data viewer, text past the width of the cell would be cut off and couldn’t be seen when moving the cursor with the arrow keys.
- **Fixed:** Changes to a field element’s font are now applied immediately.
- **Fixed:** Fixed a bug where the “layout refresh” event could be added to elements that weren’t in a layout element.
- **Fixed:** Fixed a bug where badge elements may not display the correct background, border, or text color properties.

88.41 VERSION 1.3.22 (2024-07-17):

- **New:**
- **Improved:** Adjusted the property inspector selection highlight color.
- **Improved:** UI and UX improvements. More control over the tables and data to copy when moving or copying an app.
- **Fixed:** Fixed a bug where an error may occur if a pinned badge element is moved to a layer below the element it’s pinned to.
- **Fixed:** Moving and copying apps to another project would not update certain references correctly.
- **Fixed:** Fixed a bug where saving a project to disk with an empty password would still require a password to import it.

88.42 VERSION 1.3.21 (2024-07-10):

- **Improved:** A local save now occurs when creating or deleting variables via the variable viewer.
- **Fixed:** Importing projects will now import all variables.
- **Fixed:** UseHTML option in the text element. When active, the text of the field will be interpreted and rendered as HTML in PLAY mode only. In edit mode the text remains as it was entered originally including the html tags.

- **Fixed:** Fixed a bug where text elements whose sizing property is set to auto height or auto width may display scrollbars.

88.43 VERSION 1.3.20 (2024-07-03):

- **Improved:** Unverified accounts can now log into Appli a limited number of times before account verification is required.
- **Fixed:** Fixed a bug in the dateDifference action where two compared dates might not compare dates or times in the right order, which would result in an empty value.
- **Fixed:** Manage “load app from disk” button when selecting apps.

88.44 VERSION 1.3.19 (2024-06-26):

- **New:** Socket support. New events and actions for creating connections and sending data to sockets. Actions can be found in the “Sockets” category.
- **New:** ‘dateClosed’ and ‘timeClosed’ events can now be dispatched when a date or time element is closed.
- **Improved:** The dateDifference action raw values are rounded to two decimal places.
- **Improved:** Badge elements can now be attached to field elements.
- **Fixed:** Fixed the wrong error message appearing when email has not been verified.
- **Fixed:** Fixed a bug with dropdown options not being refreshed when the options and values are set via lowcode, without a connection to a table.
- **Fixed:** The dateDifference action was out of sync with recent changes to the date/time pickers behavior.

88.45 VERSION 1.3.18 (2024-06-19):

- **New:** Added a new event called timeSelected for the low-code actions.
- **Fixed:** Clicking on ‘cancel’ button will now hide the time element on desktop.
- **Fixed:** Fixed a bug where “LF” stored in a variable or element for the “replacement text” argument of the “replace text” action didn’t translate to a line feed.

88.46 VERSION 1.3.17 (2024-06-12):

- **New:** bindTimePicker action added to define the field element to receive the selected time from the time picker.
- **Improved:** Added line-delimited list as an optional format for the returned keys of a database table.
- **Fixed:** Fixed the argument dropdown options for the “replace text” action.

- **Fixed:** Pressing “Cancel” when duplicating a screen will now correctly cancel.

88.47 VERSION 1.3.16 (2024-06-05):

- **New:** “cell selected” event. Use this event to perform an action when the cell of a table element is selected.
- **New:** “cell edited” event. Use this event to perform an action when the cell of a table element is edited.
- **Improved:** Updated argument names and descriptions of the “replace text” action.
- **Improved:** Onboarding for account creation.
- **Fixed:** Hide scrollbar when dropdown element options are empty.

88.48 VERSION 1.3.15 (2024-05-29):

- **New:** “Replace text” action in the “utility” category to replace text from the provided source with other text.
- **Improved:**
- **Fixed:** Fixed a bug where some events would not work if the element they belong to was in a layout element.
- **Fixed:** Fixed a bug which would cause component elements on other screens to lose their action scripts or not stay in sync when updating the current screen.
- **Fixed:** Fixed a bug where the “cdbRecordID” is not output if it’s selected with other keys in the “read keys” action.
- **Fixed:** Fixed a bug where nothing is output if “all items” is selected in the “keys” argument of the “read keys” action.

88.49 VERSION 1.3.14 (2024-05-22):

- **New:** “get all selected recordIDs” action for table elements.
- **New:** Repeat Over Array action has been added to enable you to iterate over a set of keys or indexes of an array. For example, the results of an API call. Allowing you to process the result and display it in your application’s front end.
- **New:** “Checkbox selected” event for the checkbox element that’s called when the element’s value is changed.
- **Improved:** GPT Chat Completion models list updated to include the newest model GPT-4o
- **Improved:** Made “get array value” action much faster for large arrays.
- **Fixed:** The “link to key” checkbox of the layout element’s no-code window will now populate correctly for non-text elements.
- **Fixed:** The low-code multi-selection argument input will now populate with selected items if applicable.
- **Fixed:** Corrected the dictionary entry of the “list records” action.

- **Fixed:** Fixed a bug with the “list records” action if “all items” is selected in the “keys” argument.
- **Fixed:** Made “get array value” action return only the exact path specified, rather than allowing multiple results if the same path was found deeper in array.
- **Fixed:** The action list dropdown of the blank action in the low-code editor will now move and display accordingly if the actions are scrolled or selected.

88.50 VERSION 1.3.13 (2024-05-15):

- **New:**
- **Improved:** “sort table” action now maintains highlight for the table.
- **Improved:** “update record” action can now take multiple recordIDs to update multiple records with the same value.
- **Improved:** Added compatibility for the checkbox, switch, and radio elements to the layout element’s no-code binding.
- **Fixed:** Fixed an error that may occur when deleting a screen from an app that contains a camera element linked to a deleted output element.
- **Fixed:** Fixed an error that may occur when removing a deleted element from a camera element’s output elements.
- **Fixed:** Fixed a bug where setting the value of a radio element to empty wouldn’t be retained in certain cases.
- **Fixed:** Fixed a bug that could prevent apps from being moved or copied to a different project.
- **Fixed:** Fixed a bug that can occur when ungrouping elements within a tab element.
- **Fixed:** The “mouse click” event will now work with non-native field elements.
- **Fixed:** Fixed a bug where arrow keys could be used to change the insertion point in a field element’s hint text.
- **Fixed:** Searching for data in arrays now properly indexes the array to search, providing accurate results.
- **Fixed:** Fixed the inability to set visible property of elements using custom code editor. Included some other potentially affected properties.

88.51 VERSION 1.3.12 (2024-05-08):

- **New:** “Get screen rect” action in the “utility” category to retrieve the dimensions of the screen’s rectangle of the current device.
- **Improved:** Modify date picker to match the date of the bound field.
- **Improved:** In the Custom Code Editor, active tabs are no longer colored. We removed the 3-dot menu in favour of right click to access the options for the tabs.
- **Improved:** Added HH:MM:SS formats to ‘get current time’ action.

- **Fixed:** Fixed an issue with resizing of charts when the app is first loaded. + Overall improvement of resizing of chart element.
- **Fixed:** Fixed a bug that caused an unstable state of the custom code editor.
- **Fixed:** Fixed a bug that could occur when using "cdbDateCreated" or "cdbDateModified" to sort a table element.
- **Fixed:** There was an intermittent error when using hotkey to minimize pane tools to the
- **Fixed:** Issue with disabled animations on chart elements which caused the charts to be un-selectable on the playground.
- **Fixed:** RestAPI Action now correctly passes endpointURL when specified as a variable.
- **Fixed:** Fixed a bug where the "reset element" action didn't clear the stored value of the radio element.
- **Fixed:** Fixed a bug that could cause projects to not load when saved to disk.
- **Fixed:** Fixed the delay between switching to play mode and the display of the mode-switching message.
- **Fixed:** Fixed a bug with the "combine multiple values" action where the "value4", "value5", and "value6" arguments didn't correctly populate their options.

88.52 VERSION 1.3.11 (2024-05-01):

- **New:** "REST API Call" LowCode action. This generic action allows you to define HTTP calls to any rest API on the web. GET, POST, PATCH, DELETE methods are available, You can define an array of custom headers, and an array for your payload. Status updates for your call can be sent to a variable or an element.
- **New:** "bind date" action to bind a date element with the provided field element.
- **New:** "dateSelected" event to use in low-code script for date element.
- **Improved:** "display records in table" action now maintains sort and highlight for the table.
- **Improved:** The custom code editor will now close when switching apps.
- **Improved:** Fixed the way Map API keys are handled, and improved help for users. Now, if a specific key is assigned to a map, the map will use this one. If none is assigned, the map will use the one specified in Appli Connect Settings.
- **Improved:** Chart element has animations turned off in Edit mode, to speed up the rendering of the chart when changing its properties.
- **Fixed:** When using setChartDataFromTable action, the chart no longer redraws 3 times, but only once.

88.53 VERSION 1.3.10 (2024-04-24):

- **New:** “set date” action to modify date element with provided date.
- **New:** “Get search input” action in the “search” category to retrieve the input from a search element.
- **Improved:** Added MaintainScroll and MaintainHighlight arguments to “refresh table” action.
- **Fixed:** Fixed a bug where the date element’s no-code setup didn’t save changes when resetting the connected element or variable output and clicking “done”.
- **Fixed:** Maintain consistent sizing for date element on ios platform.

88.54 VERSION 1.3.9 (2024-04-17):

- **New:**
- **Improved:** When inputting lowcode action arguments, the Keypad Enter key will now apply changes and close the input method, not close the lowcode editor entirely.
- **Improved:** You can now “Clear All Arguments” of a LowCode action using the right click menu.
- **Improved:** The createRecord action now supports storing multiple records. It also directly takes record data from a query action.
- **Fixed:** Fixed accidental flip of ENTER and RETURN functionality in the lowcode editor. Now, only they Keypad Enter button can “Apply” and Close the lowcode editor.
- **Fixed:** Fixed a regression where the project browser sometimes did not correctly show the status of the script icon of an element.
- **Fixed:** Minimized floating windows will now be behind the screens list popup.
- **Fixed:** Fixed a bug where date actions with a format of “seconds” would output an empty value.
- **Fixed:** Fixed a bug where dragging an action to the top of an event sometimes did not work.

88.55 VERSION 1.3.8 (2024-04-10):

- **New:** “Exit Field” event has been added and triggers when focus on a field element is lost WITHOUT changing the field’s content. “Close Field”, on the other hand, is triggered when the content HAS been changed.
- **Improved:** The order of the User Icon in the playground has been adjusted.
- **Fixed:** Fixed a bug that caused “open screen” and “close screen” events to run when switching screens in edit mode.
- **Fixed:** Addressed a bug that manifested when clicking on the background in the Projects screen.

88.56 VERSION 1.3.7 (2024-04-03):

- **Improved:** The custom code editor now has improved support for local variables inside your commands / functions handlers.
- **Improved:** Your app's lowcode action scripts will now be automatically compiled and stored in a more advanced state. Improving the performance of loading Appli apps in both the builder and the Player. This optimization happens when your app is loaded in the Appli builder, and occurs only once.
- **Fixed:** We've improved handling of properties that are both Livecode and Appli properties when using Custom Code functions.
- **Fixed:** Don't allow option copy for elements inside of components.
- **Fixed:** Actions with a parameter requiring an operator from a dropdown will eventually break due to a regression.

88.57 VERSION 1.3.6 (2024-03-27):

- **Fixed:** Fixed a visual bug with layout elements that have multiple columns and alternate row colors.
- **Fixed:** Fixed a bug where changing the layout element's "column" property would display alternate row colors despite the property being off.
- **Fixed:** Pasting into custom code editor will now work as expected.
- **Fixed:** Added checks to make sure the Lowcode editor does not open with its bottom below the footer when the user's resolution changes, or the appli window is made smaller.
- **Fixed:** Fixed a bug where the low-code editor settings may be cut off.
- **Fixed:** Fixed a bug with referencing elements in the custom code editor.
- **Fixed:** Fixed a bug where deleting a parent element containing a badge element could result in an error.

88.58 VERSION 1.3.5 (2024-03-20):

- **New:** Reset query element action clears the values of every field in the container.
- **Improved:** When the auto completion field displays above the cursor, pressing the up arrow focuses the auto-complete field to allow selection.
- **Improved:** "Edit Custom Code" action was added to the right click menu of a "custom code" action's argument.
- **Improved:** We've optimized the way action scripts are saved and applied, resulting in overall performance improvements. This improvement is only applied to scripts that are opened and resaved using the lowcode editor. NOTE: Once in use, this requires an updated version of the Appli Player.
- **Improved:** Improved usability of the operator and delimiter options in the search element's no-code window.
- **Fixed:** The playground settings window will now resize correctly.
- **Fixed:** Fixed a bug that caused an error if an element with a badge

element pinned to it is deleted.

- **Fixed:** The “return” button no longer closes the Lowcode Editor when typing in an action name. Only the “enter” button applies the LowCode changes and closes the lowcode editor.
- **Fixed:** Can now scroll layout elements to the bottom when they have multiple rows enabled.
- **Fixed:** Fixed a bug preventing the “get property from layout row” action from working.

88.59 VERSION 1.3.4 (2024-03-13):

- **Improved:** Appearance and usability of the search element’s no-code window.
- **Improved:** Make selection lines for components a different color.
- **Fixed:** Fixed a bug where changing the playground’s zoom level will cause the search element to lose its rounded corners.
- **Fixed:** “Query element” action will now work when using an element or variable for the “logicalOp” and “queryOp” arguments.
- **Fixed:** “Query element” action will now work when key names have spaces.

88.60 VERSION 1.3.3 (2024-03-06):

- **New:** A new LowCode action which lets you combine up to 6 values in one action.
- **Improved:** Typing speed in the custom code editor has been optimized. The displaying of the auto-complete is now delayed until the user stops typing.
- **Improved:** Our various pane tools now return to their previous known location if you happen to relaunch a tool that is already open but minimized.
- **Improved:** CTRL/COMMAND + W has been improved. All panes will have new state assigned when this hotkey is used. It will minimize all open panes if the last focused pane tool is not minimized. It will un-minimize all pane tools if the last focused pane is minimized.
- **Improved:** The “camera captured” event will now run if a barcode is successfully scanned.
- **Fixed:** Native Elements were not displaying properly on initial app launch.
- **Fixed:** Fixed undo functionality with element low-code.
- **Fixed:** Fixed sizing of badge element’s no-code window.
- **Fixed:** Mitigated 2 conditions which made custom code editor behave strangely.
- **Fixed:** Emails sent with low-code actions will now come from “@appli.io”.
- **Fixed:** “Get line from variable” action can now output to elements.

- **Fixed:** A LowCode editor that has been full-screened and closed will now correctly return to it proper position when opened again.
- **Fixed:** The cloud or local target was not stored in the query element no-code.
- **Fixed:** Fields that could not be found in any table would not show up in the query element no-code window.
- **Fixed:** The query element action is out of sync with recent internal improvements.
- **Fixed:** The camera element's barcode scanner can now output to a variable.

88.61 VERSION 1.3.2 (2024-02-28):

- **New:**
- **Improved:** Error handling of GPT actions has been improved further.
- **Improved:** Make gradient adjustment tools easier to grab.
- **Improved:** General appearance of answer dialogs.
- **Improved:** Added the ability to include an icon on answer dialogs through the "answer dialog" action and the "player" tab of the settings.
- **Improved:** Improved the auto-binding to handle more cases, like renaming a previously known element to a new name.
- **Improved:** Layouts can now have multiple columns.
- **Fixed:** Fixed a bug where custom answer dialog colors wouldn't display correctly.
- **Fixed:** Tab elements could not receive element children in their second tabs.
- **Fixed:** Creating elements would sometimes create elements in a parent that was not the front-most element.
- **Fixed:** Fixed a bug in creating a correct message history in a GPTChatCompletion.
- **Fixed:** Multi-selection input method would display wrong values after filtering.
- **Fixed:** Fixed a bug that prevented the first call in a GPT Chat or Image generation to work reliably.
- **Fixed:** Fixed a bug preventing the "count lines in variable" action from working.

88.62 VERSION 1.3.1 (2024-02-21):

- **New:** Added ClearGPTConversation action to enable you to clear GPT Conversations to start them from the beginning.
- **Improved:** Colors for the interface and its text in answer dialogs can now be customized in the "answer dialog" action and the "player" tab of the app's settings.
- **Improved:** GPT actions will display all errors returned from the API endpoint.

- **Improved:** Added an optional argument to the "answer dialog" action to allow custom titles.
- **Fixed:** Fixed a bug where an "input" argument which contained line breaks or other special characters would cause lowcode actions script not to start.
- **Fixed:** Addressed an issue where the auto-binding would not properly handle new fields added to an existing query element.
- **Fixed:** Larger argument inputs will be truncated to 100 characters when displaying in the lowcode argument box.

88.63 VERSION 1.3.0 (2024-02-14):

- **New:** Introducing Appli Connect, our 3rd party API gateway. Our first supported endpoints are OpenAI's Chat Completion, Image Generation and Text 2 Speech. You can find them in our new "AI" actions category.
- **New:** Lowcode action to count the number of indexes in an array
- **New:** Play Audioclip action.
- **New:** Added a input method for logical operators to be used in actions.
- **New:** Added CRUD actions for CanelaDB blobs. The blob's data filepath can be specified by a variable or the user can select the file from their device's file system.
- **Improved:** Increased height and added scrolling to answer dialogs.
- **Improved:** The operator's input method has been enhanced to support both the symbol and words. Eg: ~ contains
- **Improved:** The query element action has a new operator parameter that was initially hard-coded to '~ contains'
- **Fixed:** Fixed a bug that can cause blank apps to display within an app project folder.
- **Fixed:** ImageText property of images can now be correctly set to binary image data, which includes the transparency channel.
- **Fixed:** Query element no-code table and key selection is broken.

88.64 VERSION 1.2.44 (2024-02-07):

- **New:** New "Hello World" template app
- **Improved:** Updated new account onboarding process.
- **Improved:** Clicking on the grey area outside the device boundaries will deselect all elements instead of selecting the screen.
- **Fixed:** Fixed a bug that caused the fill color of a rectangle graphic element to not change if its fill gradient property was on.
- **Fixed:** Close open dropdown elements in play mode if a hotkey shortcut is used.
- **Fixed:** "set array value" action would not work when a variable was used for the path.
- **Fixed:** Fixed a bug where a dropdown element's options text may be misaligned.

88.65 VERSION 1.2.43 (2024-01-31):

- **New:**
- **Improved:** Clicking on the grey area outside the device boundaries will disengage edit internal elements mode and will select the screen.
- **Improved:** Updated the query element container to handle adding and removing elements.
- **Improved:** Updated the query Element to handle broken binding references. They will be ignored when the queryElement action processes.
- **Improved:** Minor improvements to the query element no-code UI.
- **Improved:** Interactions with the Asset Manager can now be undone.
- **Improved:** Further improved the management of our various pane tools. Now when you drag them to the bottom they collapse, into a compact rectangle, to just above the footer. CTRL W will collapse all open pane tools to the footer. The same hotkey will restore them.
- **Fixed:** Form elements would not fully scroll down to the bottom of their children.
- **Fixed:** Minimized Pane Tools now stick to the footer when the Appli window is made shorter or taller.
- **Fixed:** Assets were not saving for projects with multiple apps in them.
- **Fixed:** All pane tools now return to their previous size and position when returning from Fullscreen.
- **Fixed:** A bug in custom code editor where the compiler did not always correctly recognize Livecode properties.
- **Fixed:** Prevents a bug where auto arranging panes while having panes minimized does not always restore them correctly.
- **Fixed:** Keys dropdown in no-code for the query element will now correctly show available DB keys for the selected table.
- **Fixed:** Removed the cdbUsers table from the available tables in the no-code for the query element.
- **Fixed:** Fixed a bug where “close screen” and “close app” events didn’t run when switching from play mode to edit mode.

88.66 VERSION 1.2.42 (2024-01-24):

- **New:** Usage tab in the account screen.
- **New:** roundToNumber.action action to round a value to the nearest whole number.
- **New:** Add Save App to Disk – allows saving app to disk without having to save the entire app project.
- **New:** The query element is a special group used to define the fields to be used as sources of input for a query.
- **New:** The query element action runs the query as defined from the no-code window for the query element. The action runs the query using the logical operator parameter and output parameters to control the query and its result. The matching recordIDs are output on a successful query.

- **Improved:** Add more options when saving apps/projects to disk.
- **Fixed:** Button elements would freeze if autosizing when deleting all text.
- **Fixed:** Fixed a bug in the custom code editor where deleting a new function didn't close its tab.
- **Fixed:** Fixed incorrect positioning of Hover Assist.
- **Fixed:** Low-code date actions will now calculate variables as an input properly.
- **Fixed:** Sort rows were incorrectly showing when clearing connected table in Layout no-code.
- **Fixed:** Search elements were not maintaining their output element when copy pasted.
- **Fixed:** Setting the value of a dropdown element to empty will now show the hint text.
- **Fixed:** Hover assist for icons in the footer now consistently appears near the footer.

88.67 VERSION 1.2.41 (2024-01-17):

- **New:** Introducing the integrated Custom Code Editor, which enables you to add custom Livecode scripts to your LowCode event flows. It is used by adding the the "custom code" action from the "Code" category. You can access the editor itself via its icon in the footer. Documentation for this feature will arrive in the coming weeks.
- **Improved:** Can now expand a dropdown element by clicking anywhere on it.
- **Improved:** Pane tools can now be moved out of the way to below the footer and past the left/right edges. Going fullscreen and auto-arranging now also takes the playground's header and footer into consideration.
- **Fixed:** Changing the text size of a dropdown element will now manage the placement of its label.
- **Fixed:** Fixed errors that could occur when using the "set array" and "set variable" actions.
- **Fixed:** The screen selector will now appear overtop of any open pane tools.

88.68 VERSION 1.2.40 (2024-01-10):

- **New:** setDatePart action and getDatePart action to modify date components.
- **New:** findRelativeDate action output the first or last date of the month.
- **Improved:** Added a "Cancel" option to the popup for closing the LowCode editor.
- **Improved:** Updated the dictionary entries with more details for createRecord, setArray, and updateRecord actions.
- **Fixed:** Alignment tools would not work correctly if zoom level was not 100%

- **Fixed:** Sometimes reference would fail to launch custom scripts.
- **Fixed:** Fixed a bug that may cause an error if the dropdown element is open.
- **Fixed:** Fixed a bug preventing "Refresh Apps" from working in the Projects screen.
- **Fixed:** Fixed cases that would cause false positives for local changes needed to be saved to the cloud.
- **Fixed:** Fixed a bug that would disable date elements when they are within the rect of form or layout elements.

88.69 VERSION 1.2.39 (2023-12-20):

- **New:** SetFormTable and SetFormRecordID actions to set the table and recordID of a form.
- **Improved:** Allow filtering of variable viewer to show only variables that pertain to the specified screen or element.
- **Improved:** Add property "AutoUpdate" to forms – allows devs to control whether forms from auto-update whenever the form comes on screen.
- **Improved:** Add property "DontUpdateOnStart" to allow layouts to not update when app starts.
- **Improved:** Allow variables to be used as Target parameter for database actions.
- **Improved:** "queryRecords" action now accepts variables for the following parameters: TableName, Key, Target, ResultFormat.
- **Fixed:** Fixed a bug that would cause button elements to lose properties when making a copy or creating components.
- **Fixed:** You can now remove table reference in Layout element's no-code pane.
- **Fixed:** Fixed a bug preventing the "sort records" action from being used.

88.70 VERSION 1.2.38 (2023-12-13):

- **New:**
- **Improved:** Sign out from the projects screen and playground screen will force a re-authentication the next time the developer signs in.
- **Improved:** Create account will more actively get the tier information when switching regions.
- **Improved:** When an app is loaded, the orientation with elements is now displayed if one exists.
- **Fixed:** CloseField event now works with non-native fields.
- **Fixed:** Fixed a bug preventing native fields from being drawn.
- **Fixed:** Date picker on mobile will now close after a date selection.
- **Fixed:** Multi-selection input method visual fix.
- **Fixed:** Dropdown element was not handling scrolling or resizing of screen while in edit mode.

88.71 VERSION 1.2.37 (2023-12-06):

- **New:** Badge Element - this element can be used to decorate another element to show users a status, notification, or event.
- **Improved:** The “Use Native Field” property for field elements has been split into “Use Native on iOS” and “Use Native on Android”. Use these properties to control which platforms use native fields.
- **Improved:** Optimize cloud performance and experience.
- **Fixed:** Some property inspector fields were incorrectly showing as enabled.
- **Fixed:** Sometimes dialogs were unclickable due to being stuck behind loading screens.

88.72 VERSION 1.2.36 (2023-11-29):

- **New:** “Border”, “border size”, and “border color” properties for the checkbox element.
- **Improved:** Sorted records in the data viewer will now retain their sort when switching to cloud or local and back. Sorting for cloud and local data are independent of one another.
- **Fixed:** If an element is selected, right-clicking the screen background will now select it and display the contextual menu.
- **Fixed:** Fixed a bug that prevented the data viewer’s contextual menu from working.

88.73 VERSION 1.2.35 (2023-11-22):

- **Improved:** “Duplicate” option has been added to the low-code editor’s contextual menu to quickly copy and paste actions.
- **Fixed:** Fixed a bug where the orientation of the low-code’s element input method may be empty.

88.74 VERSION 1.2.34 (2023-11-15):

- **New:** “focusField” action. Use this action to focus on a field element.
- **New:** Hotkey to open the low-code editor if an element or the screen is selected (Shift+E).
- **Improved:** Made the border of window panes clearer.
- **Improved:** The age to birth date action does not handle spaces between the number and ‘y’ or ‘m’ value.
- **Improved:** Allow cdbDateModified and cdbDateCreated keys to be used in sortTable action.
- **Fixed:** Fixed an error that can occur when changing the size of “create account” elements whose “autosize” property is on.
- **Fixed:** Fixed bug where some commented actions did not end up being commented in final script assigned to the element.

88.75 VERSION 1.2.33 (2023-11-08):

- **Improved:** The low-code editor can now be accessed in the contextual menu when an element is selected.
- **Improved:** Label alignment can now be adjusted in the form element's no-code window.
- **Improved:** Make saving to disk occur from what is on your computer, instead of what is in the cloud. Also add option to save data tables from cloud or local.
- **Fixed:** Fixed a bug that prevented grouping multiple image elements.
- **Fixed:** Fixed a bug that may cause windows to open in Appli and become cut-off.
- **Fixed:** Fixed a bug where a window that was toggled to full size will continue to open to full size after being resized smaller.

88.76 VERSION 1.2.32 (2023-11-01):

- **New:** Added ability to quickly adjust child element properties in the form element's no-code setup.
- **New:** sortTable action will sort the chosen column of a table element with control of the sort direction (ascending/descending) by sortType (text, numeric, dateTime).
- **Improved:** Query input method in lowcode editor now has an "Ignore Empty Input" checkbox. This allows "batch query records" action to find matching records when "AND" is used and there is an empty input.
- **Fixed:** Fixed an error that occurred when selecting "use document color" from the color editor contextual menu if there were no colors added to the asset manager.
- **Fixed:** Fixed a bug where scrolling the playground with "show offscreen elements" disabled could cause the playground to move off-center.

88.77 VERSION 1.2.31 (2023-10-25):

- **Improved:** "ageToBirthDate" action - allow the last char of input to be a 'y' to represent year.
- **Fixed:** Query input method is sized to fit in the lowcode editor.
- **Fixed:** Fixed an error that would occur when using "set table columns" action and attempting to select keys before selecting a table element.
- **Fixed:** "clear search" action now resets the hint text.
- **Fixed:** Lowcode Settings tab opens correctly from within Lowcode Editor.
- **Fixed:** Map Element (API and Geolocation) errors were not correctly alerting the user.

88.78 VERSION 1.2.30 (2023-10-18):

- **New:** ageToBirthDate action will provide a rough approximation of birth date from the data provided.

- **Improved:** Don't allow form, table, and layout elements inside of form elements.
- **Improved:** Pane Tools now appear "focused" or not.
- **Improved:** Added an argument to deleteArrayElement.action that determines what gets deleted by variable.
- **Fixed:** Date inputs on mobile platform will now match with the picker.
- **Fixed:** Convert date actions now correctly outputs the result based on the format.

88.79 VERSION 1.2.29 (2023-10-11):

- **New:** resetTable and resetLayout actions.
- **New:** Plus icon in the Database Modeler provides quick table creation.
- **New:** Updated the form element's no-code window with an "appearance" tab to allow quick child element arranging and deleting.
- **Fixed:** Fixed a regression that prevented the ability to copy content from the internal action dictionary.
- **Fixed:** Fixed a bug where text elements with "vertical align" enabled may load with an incorrect "Y" value.

88.80 VERSION 1.2.28 (2023-10-04):

- **Improved:** When switching from edit to play mode in Appli Builder, Appli will not run low-code actions until after initializing the app.
- **Fixed:** Reset date input method after selection.
- **Fixed:** submitFormtoDB.action would create a new record for local/cloud when the expectation was to update an existing record.
- **Fixed:** Updated the dictionary entry and fixed a bug that caused the "set variable from row" action's "key" argument to remain unpopulated.
- **Fixed:** Updated the dictionary entry and added error handling to the "set property in row from variable" action.
- **Fixed:** Fixed issue with wrong number of panes being included in the resize operations.
- **Fixed:** Make locked fields remain locked even if their value is set from a different source.
- **Fixed:** Pane tools return to their previous position when exiting full screen.
- **Fixed:** Pane windows were sometimes drawing too large or partially hidden.
- **Fixed:** Fixed a bug that could cause apps to load with an incorrect platform size.
- **Fixed:** submitFormToDB.action will fail if the recordID in the form does not exist in a database table.

88.81 VERSION 1.2.27 (2023-09-27):

- **New:** Action “getDateFormat” added to retrieve a date format.
- **Improved:** Group labels and their elements together when creating a custom form with the form builder.
- **Improved:** In Appli Builder - when going into play mode, set the radio button to the last known value.
- **Fixed:** Other date formats added to options including seconds in findDayofWeek.action and convertDate.action.
- **Fixed:** Radio button graphic inner circle was not centered.
- **Fixed:** Corrected spelling in the title of the Data Modeler.

88.82 VERSION 1.2.26 (2023-09-20):

- **New:**
- **Improved:** When grouping elements, the default group name will now match the name of the first selected element.
- **Fixed:**

88.83 VERSION 1.2.25 (2023-09-14):

- **New:** “Start print process” action to start a print job to send to a printer (desktop only) or save as a PDF.
- **New:** “Add to print job” action to add the specified element’s rectangular boundary of a given screen to the print job as a new page.
- **New:** “End print process” action to end the print process and send the open print job to the printer or save it as a PDF, as specified by the ‘start print process’ action called prior to this action.
- **New:** “Set print orientation” action to specify the orientation of the page(s) when printed.
- **New:** “Set print margins” action to specify the margin widths of the page(s) when printed.
- **New:** “Get print setting” action to retrieve the value of the specified print setting.
- **New:** “Get current date” action to retrieve current date in chosen date format
- **Improved:** Added date picker sub-input to compareDates.action and checkDateRange.action.
- **Improved:** Allow Desktop applications to change size by dragging on the corners of the screen.
- **Improved:** Allow for editing of Desktop screen sizes via the property inspector.
- **Improved:** Moved the “print to pdf” and “print screen” actions from the “utility” category to the “printing” category.
- **Fixed:** Date picker sub input was not showing up properly in Low Code Editor.

- **Fixed:** Time/Date/Rectangle elements were not saving their “Stretch to fill notch” property.
- **Fixed:** Fixed a bug that caused the “exit script” action to run despite being commented out.

88.84 VERSION 1.2.24 (2023-09-07):

- **New:** “set table columns” action for table elements.
- **New:** “get last selected recordID” action for table elements.
- **New:** “rowSelected” event for table elements.
- **Improved:** Add button to copy appCode from App Project screen.
- **Fixed:** Make fields lose focus when clicking on the rest of the app.
- **Fixed:** Fixed a bug where the hover assist tooltip may remain on screen after the mouse has been moved.
- **Fixed:** Fixed a bug which would reset the lowcode of table elements after loading an app.
- **Fixed:** Fixed a bug where adding an action after using undo would insert a blank line into the event.
- **Fixed:** Fixed a bug where some arguments in date/time actions may display as blank.

88.85 VERSION 1.2.23 (2023-08-30):

- **New:** Default setting for date and time format in “General Settings”.
- **New:** Border properties for radio elements.
- **New:** Hybrid option added to local and cloud options to form element
- **New:** “Exit Script” Action will exit out of a script
- **Improved:** Auto-populate format parameters for date inputs and outputs based on default setting in low-code.
- **Improved:** Structured resizing of floating pane tools is now cancelled as soon as any pane is manually resized, moved, or closed.
- **Improved:** submitFormToDB action improved to support hybrid mode.
- **Improved:** Improvements when using a search element to filter a layout element.
- **Improved:** Allow double-click to edit screen names in screen list.
- **Improved:** Add right-click menu for screen list.
- **Improved:** “layoutRefresh” event is now available for all elements.
- **Fixed:** Fixed a bug where window panes may open up with sizes below the set minimum.
- **Fixed:** Fixed a bug where window panes with “full window” toggled on may be positioned incorrectly when reopened.
- **Fixed:** Fields wouldn’t correctly respond when tabbing in certain cases.
- **Fixed:** Maintain scroll location after reordering screens.
- **Fixed:** The low code editor dictionary can now be resized.
- **Fixed:** “Exit Repeat” action was not working

- **Fixed:** Fixed a bug which would prevent apps from being moved or copied to another project.
- **Fixed:** An open dropdown element's options will now retain its location when scrolling the playground in play mode.
- **Fixed:** Make screen thumbnails draw correctly and move correctly after duplicating a screen.

88.86 VERSION 1.2.22 (2023-08-24):

- **Improved:** Multiple table keys can now be added, removed, renamed, and rearranged with the "edit keys" option in the data modeler.
- **Fixed:** Fixed a bug where refreshing a table in the data viewer may display its keys in the incorrect order.
- **Fixed:** Creating a new app will correctly show the initial screen name as "Untitled".
- **Fixed:** Fixed a bug which would cause the cursor to jump to the end of the field element when typing.
- **Fixed:** The readKeys action does not correctly handle reading cdbRecordIDs as the key.

88.87 VERSION 1.2.21 (2023-08-23):

- **New:** We have added line/action numbers to the LowCode Editor. This new setting is on by default and can be turned off via new checkbox at the bottom of LowCode editor settings.
- **Improved:** Optionally, submitFormToDB action can take a target parameter that will overwrite the target as defined in the form's no-code setup. This is useful if one desires to store data in the cloud and locally using two submitFormToDB actions in a row.
- **Improved:** If calling submitFormToDB twice in a row and before using the resetForm action, they will share the same recordID.
- **Improved:** Resizing the Appli window with Pane tools that have been "arranged" now retains their relative size and position. The arrangement is deactivated if one of the pane tools is closed. With no arrangement active, the pane tools are unaffected by the resizing of the main Appli window.
- **Improved:** Adding new keys to tables in the data modeler will no longer move the table to the center of the window.
- **Improved:** The data viewer will now retain the order of table keys.
- **Improved:** Allow drag/drop to reorder screens visually.
- **Improved:** The process of creating new tables in the data modeler has been improved.
- **Fixed:** Setting the "text" property of a field element with an action will now work when the field is focused.
- **Fixed:** Time picker now closes properly with property toggled on.

88.88 VERSION 1.2.20 (2023-08-16):

- **New:**
- **Improved:** Pickers will close after setting date/time on desktop.
- **Improved:** LiveCode integrations should sync to get the latest version if possible.
- **Improved:** When you hover over the name of an action, the tooltip now shows its action / line number.
- **Fixed:** Make all other elements deselect when editing a text element.
- **Fixed:** Addressed an issue when saving a project to disk and then saving to cloud would result in a hang.
- **Fixed:** Removed 'hidePicker' property for consistency. Visibility on playmode will be set like other elements.
- **Fixed:** readKeys.action was not offering target of local or cloud.
- **Fixed:** Add back icon picker.
- **Fixed:** After HoverAssist is popped up, subsequent HoverAssists can be viewed without delay. The delay is made active again once after the user clicks on something on the Playground.

88.89 VERSION 1.2.19 (2023-08-10):

- **Improved:** Add checkmark besides named color if it matches current color when attempting to 'use document color'
- **Fixed:** Corrected and issue that may cause errors when writing data to the cloud.
- **Fixed:** Fixed a bug that caused the data modeler refresh button to go missing.
- **Fixed:** Fixed error messages for control structures in the low code editor's error inspector.
- **Fixed:** Fixed a bug which would prevent a new app from being created.
- **Fixed:** Fixed a bug which would prevent an app from being loaded.
- **Fixed:** Deleting an app through the right-click contextual menu will now manage the icons when viewing in list mode.

88.90 VERSION 1.2.18 (2023-08-09):

- **New:**
- **Improved:** Group icons in the header so it works better for small screen sizes.
- **Improved:** Add hotkeys to tooltips and hover assist panes.
- **Improved:** Add (Q) hotkey for Date elements and (Shift-Q) hotkey for Time elements.
- **Improved:** Date and time pickers will close upon clicking on the background when the 'closebg' property is toggled on.
- **Improved:** The data modeler now displays the "cdbUsers" and "cdbBlobs" table and allows user-created keys in these tables to be modified.

- **Improved:** Slider and checkbox elements can now be linked to table keys in the form's no-code setup.
- **Improved:** Text font dropdowns in the form builder's style screen can now take text input and will filter through the list of fonts.
- **Fixed:** Fixed an error that would occur when deleting a table from the data modeler while in the middle of adding keys to it.
- **Fixed:** Changing the "selected color" property in the checkbox element while the checkmark is visible will no longer hide the checkmark.
- **Fixed:** Fixed a bug where checkbox elements with the "sizing" property set to "fixed size" may load its UI incorrectly.
- **Fixed:** Fixed a bug where double-clicking to add an event in the low code editor would cause the undo count to increase by two.
- **Fixed:** Non-native field elements now correctly disable the keyboard when the 'lock text' property is enabled.
- **Fixed:** Properly convert from AM/PM to military time.
- **Fixed:** Date picker was clipped on mobile after initial creation.
- **Fixed:** Fixed an error that may occur when trying to create a user record in the cdbUsers table of the data viewer when the table has no records to display.
- **Fixed:** Fixed a bug where "try" control structures were being added incorrectly in the low code editor.
- **Fixed:** The low code inspector will now report errors involving control structures.

88.91 VERSION 1.2.16 (2023-08-03):

- **Improved:** Can now multi-select apps in the Projects screen when viewing in list mode. Multiple apps can be deleted, moved, and copied at the same time.
- **Improved:** An interface to facilitate the process of creating new users and modifying existing users in the "cdbUsers" table has been added to the Data Viewer.
- **Improved:** Boolean input method is back.Ê Set Property Action now resets the value argument when the property is changed, this prevents faulty inputs.
- **Improved:** Number values of LowCode action arguments now appear as numbers without quotation marks.
- **Improved:** Low Code "Comment" actions are greyed out.
- **Improved:** Low Code argument boxes resize better for smaller values.
- **Improved:** Add hotkey to open/close data modeler via cmd/ctrl-d.
- **Improved:** Updated icons in footer – added data viewer, variable viewer icons.
- **Improved:** Prebuilt Contacts form now connects to cloud table rather than local. Also resets form on submit.
- **Improved:** Make loading projects default to the filename.
- **Fixed:** Properly disable keyboard on field elements with 'lock text' prop-

erty toggled on.

- **Fixed:** The “value” argument dropdown of the set property action now correctly changes based on the property selected.
- **Fixed:** Fields were sometimes un-selectable after toggling the “Locked-Text” property.
- **Fixed:** Linking a time picker to a field was not correctly locking the field.
- **Fixed:** Hide/Show element actions now display elements properly.
- **Fixed:** Some fields were requiring use of “ctrl-return” on macOS instead of “cmd-return”
- **Fixed:** Invisible elements that are selected will no longer be draggable on the playground.

88.92 VERSION 1.2.15 (2023-07-26):

- **New:** Checkbox Element. This element shows a checkbox that can receive user input.
- **Improved:** Error handling for table creation has been improved in the Data Modeler tool.
- **Improved:** All pane tools now open in front of the others.
- **Improved:** Appli now remembers the last position of each of its pane tools. (On a per app basis). When exiting fullscreen the pane tools will now be restored to their position prior to pressing fullscreen.
- **Improved:** Data Viewer now has a new icon differentiating it from the Data Modeler.
- **Improved:** Keys can now be renamed or deleted in the data modeler through the right-click contextual menu.
- **Improved:** Can now view apps as a list in the Projects Screen.
- **Fixed:** the Lowcode editor’s top left collapse/expand button now has a tooltip when Lowcode first starts.
- **Fixed:** Refreshing the data viewer on cloud mode will now keep tables and their keys in sync with what’s in the cloud.
- **Fixed:** Fixed an error that may occur when renaming table keys in the data modeler.
- **Fixed:** Refreshing the data modeler will now keep tables and their keys in sync with what’s in the cloud.
- **Fixed:** Fixed a bug where deleting an element row of the form builder if it has only one row would produce an error.
- **Fixed:** Fixed a bug where the form builder would create an incorrect amount of options for the radio element.
- **Fixed:** Fixed a bug with the time and date elements where their dimensions may be inaccurate when created on zoom levels above and below 100%.
- **Fixed:** Maintain field element’s ‘lockedText’ property to match property inspector on mobile
- **Fixed:** Fixed a bug preventing apps from being loaded.
- **Fixed:** Fixed a bug which would occur when going to “Play” mode.

88.93 VERSION 1.2.14 (2023-07-19):

- **New:** A color swatch appears on any lowCode action argument that contains a color as a value.
- **Improved:** Improve speed of deletion of elements.
- **Improved:** Creating multiple elements via Form Builder and Copy/Paste was slow; improved speed.
- **Improved:** Table keys in the data modeler can now be renamed by double-clicking on them.
- **Fixed:** Fixed a bug that may display a blank app inside of a project folder.
- **Fixed:** Playground would sometimes shift while going into play mode.
- **Fixed:** Changing a field element's "password field" property will now switch its contents to the password font only if it's in play mode and doesn't contain hint text.
- **Fixed:** Fixes for errors that could occur in the player when there is a dropdown element inside of a form element.
- **Fixed:** Fixed a bug that could cause the low code editor to display an unnecessary, horizontal scrollbar.
- **Fixed:** Colorpicker will only be available as an inputMethod for Color related properties.
- **Fixed:** HoverAssist setting will not reset to true when Appli is restarted.
- **Fixed:** Saving and creating components would not update element low-code references in some cases.
- **Fixed:** Color picker is now displayed as an input method option for the set_property action when the selected property is a color.
- **Fixed:** Fixed a bug that caused the dropdown element's scrollbar to be positioned incorrectly and the scrolling to stop working.
- **Fixed:** Fixed an error that would occur when switching to play mode if a dropdown and layout element were on the same screen.
- **Fixed:** Fixed an error that can occur when loading an app.
- **Fixed:** Regular tooltips are removed while HoverAssist is active, and restored when the user turns it off.
- **Fixed:** Fixed missing HoverAssist for ellipse element.

88.94 VERSION 1.2.13 (2023-07-12):

- **New:** Move an app from one project to another through the right-click contextual menu in the Projects Screen.
- **New:** Copy an app from one project to another through the right-click contextual menu in the Projects Screen.
- **New:** "Load table" action.
- **New:** Introducing "Hover Assist", an advanced tooltip feature to help new users learn Appli faster. These advanced tooltips will grow in richness. It can be turned on and off in the General Settings.
- **Improved:** Option to create your own form in the form builder. This

allows you to specify keys and generate a form and connected table.

- **Improved:** AppliBot now exists on its own stack to avoid issues with native layers.
- **Improved:** Added the ability to quickly add text label increments in the slider element's no-code window.
- **Fixed:** Fixed a bug in the dropdown element that can cause an error when switching from edit mode to play mode.
- **Fixed:** Chart element would not draw.
- **Fixed:** Fixed an error that can occur when pressing option/alt key while the mouse is over the project browser.
- **Fixed:** Fixed cases where the horizontal scrollbar in the low code editor may not display correctly.
- **Fixed:** Browsers, maps, and charts will now be correctly hidden when a dropdown menu is opened and its options intersect with a browser, map or chart.
- **Fixed:** Native layers elements no longer show when switching back to Edit Mode.
- **Fixed:** Scrollbar no longer shows up in the middle of the screen when in play mode.
- **Fixed:** Removed drop shadow from create account screen.
- **Fixed:** Fixed display order of increments in the slider element's no-code window.
- **Fixed:** Missing tooltip for Bug Inspector, empty bug inspector field showing up, and highlight of Bug inspector when hovering.

88.95 VERSION 1.2.12 (2023-07-05):

- **New:** AppliBot support bot has been integrated into Appli Builder. Click on the Chatbot icon in the footer.
- **New:** "Refresh dropdown" action in the "dropdown" category to refresh the options of a dropdown element that's linked to a data table.
- **New:** Added a guide to walk through new users on starting a blank project or downloading a template when they first log in.
- **New:** Form builder with Prebuilt forms. You can use these prebuilt forms as a jumping off point for creating your own forms. Using these forms will create a form element along with the data table it is connected to.
- **Improved:** Add ability to save forms for future use.
- **Fixed:** Fixed a bug in the dropdown element that can cause an error when switching from edit mode to play mode.
- **Fixed:** Fixed a bug in the low code editor that can cause certain events to be excluded from the events list.

88.96 VERSION 1.2.11 (2023-06-28):

- **Improved:** Unify date and time toggler low-code actions into one.
- **Improved:** Smoother scrolling when moving tables in the data modeler.

- **Improved:** Added links to download Appli Player in the App Code tab of the settings and after account creation.
- **Fixed:** Properly dereference binded-field element from picker after deletion.
- **Fixed:** Enable mouseClick event for picker's binded-element (given lockText property is on and useNative property is off).
- **Fixed:** Undoing of adding screens would cause an error.
- **Fixed:** Fixed a bug where declining the prompt to delete a table in the data modeler caused the table to change in color and location.
- **Fixed:** Apps imported from disk now display their app code in the Projects screen.
- **Fixed:** Components now work with the undo/redo system.
- **Fixed:** Fixed a bug where tables in the data modeler could be moved off-screen.
- **Fixed:** Dragging actions into empty events now works consistently
- **Fixed:** Fixed a bug that could occur in the lowcode editor when the return key was pressed while using the field input method.
- **Fixed:** "Get array value" action would add a line feed to the value.

88.97 VERSION 1.2.10 (2023-06-21):

- **New:** Date Element - This element allow users to enter a date, using the native datepicker on mobile, or a datepicker widget on desktop. This element can be used by itself, or attached to a text/field element.
- **New:** Time Element - This element allow users to enter a date, using the native timepicker on mobile, or a timepicker widget on desktop. This element can be used by itself, or attached to a text/field element.
- **New:** "Get Date" action - used to get the date of a Date element in a specified format
- **New:** "Get Time" action - used to get the time of a Time element in a specified format
- **New:** "Show Picker" action - used to show a Date or Time element.
- **New:** "Hide Picker" action - used to hide a Date or Time element.
- **New:** "Toggle Picker" action - used to toggle visibility of a Date or Time element
- **New:** "Filter lines" action.
- **New:** "Get property from layout row" action.
- **New:** "Set property in layout row" action.
- **Improved:** Date difference action has been updated to support the new time and date picker elements.
- **Improved:** Updated Store tab in the Account screen for displaying current tier details, updating current tier, and switching tiers.
- **Fixed:** When a blank action is added, along with a new event, the blank action did not focus.
- **Fixed:** Prevents user from being randomly locked out of typing into a blank line field they just created.

- **Fixed:** Using an action to set the minValue or maxValue of a slider element would prevent subsequent actions from running in some cases.
- **Fixed:** Lowcode editor settings were displaying incorrectly.
- **Fixed:** When using Variable Editor to change the value of a variable, pressing enter no longer closes the LowCode Editor.
- **Fixed:** Reset input method variable when using "merge record" action.
- **Fixed:** The lockedText property of field elements on the mobile player will now be consistent with the IDE.

88.98 VERSION 1.2.9 (2023-06-14):

- **New:** "Merge record" action. This modifies a record's key value using comparisons like "is", "is not", "is in", and "is not in".
- **Improved:** All pane tools have been given a header with pane management features. You can toggle between full window panes, and you can also arrange them Horizontally and Vertically if more than one pane tool is open.
- **Improved:** Allow CSV and JSON files to update an existing database table if dragged onto an existing data table element. It is required that the schema matches.
- **Improved:** You can press Enter or Return in the LowCode editor to "Apply" the changes.
- **Improved:** When pressing ESC in the LowCode editor, you will be prompted with the option to save changes or not.
- **Improved:** Updated tier selection screen in account creation.
- **Fixed:** Each time a pane is clicked on, it now correctly takes focus above other panes.
- **Fixed:** LowCode Script Verification state is correctly updated if the LowCode editor is open when elements are deleted.
- **Fixed:** Fixed an execution error when going to the LowCode editor from one of the lines of the LowCode Verification Tool.
- **Fixed:** ESC will now exit LowCode editor without saving. Only the ENTER key saves the LowCode script.
- **Fixed:** "Populate form data" action will now populate the "lastSelectedText" property of dropdown elements within the form element.

88.99 VERSION 1.2.8 (2023-06-07):

- **New:** "Batch query records" action.
- **New:** InputMethodQuery for lowcode editor.
- **Improved:** Low-code verification inspector will no longer detect errors in commented out actions.
- **Improved:** The LowCode editor has been moved to a resizable pane similar to other tools in Appli. Using the top left button, the left most pane can be collapsed giving you a full screen LowCode experience. Click again to expand.

- **Fixed:** Fixed a bug with importing CSV files that caused the last key to go missing and ordered the keys incorrectly.
- **Fixed:** Fixed a bug in the low code editor where deleting and adding actions may add blank lines to an event.
- **Fixed:** Fixed dropdown element's option(s) to overlap other elements.

88.100 VERSION 1.2.7 (2023-05-31):

- **New:** "SliderValueChanged" low-code event for slider elements.
- **Improved:** Updated visuals for slider elements with a corner radius greater than zero.
- **Improved:** Updated the slider element's no-code setup to improve usability.
- **Fixed:** Creating lines while zoomed in would sometimes result in the angle of the line being different.
- **Fixed:** Creating lines is now undo-able.
- **Fixed:** Drawing a line on top of a group, tab, form, or layout will now create the line inside that element.
- **Fixed:** Thumbnails of apps, screens and components were showing as blank.
- **Fixed:** Prevent elements from being created in hidden elements when an element is first drawn.
- **Fixed:** Made bug fixes regarding the current value and visual improvements for the slider element.
- **Fixed:** Hide the screens list when the lowcode editor is opened.

88.101 VERSION 1.2.6 (2023-05-24):

- **New:** Added a Slider element to depict a numerical value. (Hotkey: Shift + S)
- **Improved:** Improved handling of various click actions in the data modeler tool. Added the ability to reset custom arrangement.
- **Improved:** Side and Bottom scrolling in the data modeler is now smoother, and the resize handle now accounts for the scrollbars.
- **Improved:** Dragging operations in the data modeler are now smoother.
- **Improved:** You can now rename a table via double click in the data modeler.
- **Improved:** "Get element value" and "set element value" actions have been updated to support the slider element.
- **Improved:** Dragging a single element around the screen was sometimes slow.
- **Fixed:** Data Modeler: Fixed an issue where tables were mispositioned on first switch to custom arrange mode, and when new tables were created.
- **Fixed:** Enter, Return, and Tab Key are now treated the same when using the data modeler.

- **Fixed:** Fixed a bug which would prevent using the no-code setup for a form element after deleting the table it was connected to.
- **Fixed:** You can now access the "low-code" button via the project browser even for deeply nested elements.
- **Fixed:** Fixed a bug which would prevent using the no-code setup for a layout element after deleting the table it was connected to.
- **Fixed:** Pressing "Return" key when typing in a button element will exit. Use "shift-return" to type a line-break.
- **Fixed:** The clipboard will no longer be cleared when Appli auto-saves.
- **Fixed:** Option-dragging to copy is now undoable.
- **Fixed:** Fixed an error that would sometimes occur when option-dragging to copy elements.

88.102 VERSION 1.2.5 (2023-05-17):

- **New:** Data Modeler Tool - Use this tool to visualize and model data in your Appli project. You can view and edit tables and their keys. You can auto-arrange or customize the arrangement of tables, as well as change their colors.
- **Improved:** Make corner radius adjustment dots larger and easier to click.
- **Improved:** Add variables, elements, and raw input as sources for the recordIDs argument.
- **Improved:** Add variables, elements, and raw input as sources for the recordIDs argument for deleteRecord.action.
- **Improved:** Lowcode Verification tool now lists only the present issues.
- **Fixed:** Stop corner radius adjustment dots from disappearing after autosave.
- **Fixed:** Tab elements would sometimes draw their child elements improperly if you had resized the tab.
- **Fixed:** Add error handling for downloading templates.

88.103 VERSION 1.2.4 (2023-05-10):

- **New:** Projects and apps that have been accessed online can now be accessed offline without internet connection.
- **Improved:** readKeys.action output is now a string.
- **Improved:** setBrowserURL.action updated to accept input for URL from a variable, element, or raw input.
- **Fixed:** ReadKeys output will produce a useful result with comma-delimited keys and LF-delimited rows.
- **Fixed:** readKeys will now properly support variables, elements, and raw input for the RecordID argument.
- **Fixed:** Issue when clicking on the icon of a pane that is already open.

88.104 VERSION 1.2.3 (2023-05-03):

- **New:** Added “show element” and “hide element” actions
- **Improved:** Clean action names by using lowercase characters and remove wordy articles.
- **Fixed:** Reset Create Account element after successful creation of new account
- **Fixed:** Fixed a case where an error may occur when deleting a camera element and a potential camera output element in quick succession.

88.105 VERSION 1.2.2 (2023-04-26):

- **New:** Added a “Dispatch Event” action to run actions from anywhere in an app.
- **New:** “Camera Captured” event that gets called after a photo is taken or selected in the mobile photo gallery with the camera element.
- **New:** “Display Captured Photo” action in the “Camera” category to display a camera element’s photo in an image element. The “Reset Element” action can be used to reset the displayed image of the image element.
- **New:** Lowcode Editor tab in Settings panel. Configure these settings to customize the look and feel of the lowcode editor.
- **Improved:** Add “Zoom To Selection” as option for right click
- **Improved:** Allow pasting text when renaming elements in the project browser.
- **Improved:** Events can be folded and unfolded from the context menu.
- **Improved:** Improved sizing and positioning of auto-complete actions list to prevent it from getting cut-off.
- **Improved:** Sign in performance improvement. Load template apps from local when possible.
- **Improved:** More actions have their verbose format descriptions
- **Improved:** Users can now specify how to sort action lines in Action List.
- **Improved:** Repeat control structure is now just “Repeat”
- **Improved:** Updated “Reset Element” action in the “Utility” category to support the image element.
- **Fixed:** Fixed another bug dealing with focusing new blank lines.
- **Fixed:** Fixed a bug where actions in an event may be incorrectly displayed or removed.
- **Fixed:** Prevent action selection after drag & drop
- **Fixed:** Low code bug with drag & drop.
- **Fixed:** Issue with focusing on blank actions.
- **Fixed:** Fixed a bug where clicking on an action in the body of the low code editor may cause an action to be added to the current event.
- **Fixed:** Data Viewer icon in Lowcode Editor will now display the data table when clicked.
- **Fixed:** Fixed text display of argument values that include “<” in the low code editor.

- **Fixed:** Fixed a bug that prevented “if variable”, “else if variable”, and “case” control structure actions from working when compared against a variable.
- **Fixed:** Update margins on text element so zooming will look more consistent

88.106 VERSION 1.2.1 (2023-04-19):

- **New:** “Get Captured Photos” action in the “Camera” category to output a line-delimited list of temporary IDs of the captured or selected photos from a camera element.
- **New:** “Clear Captured Photos” action in the “Camera” category to clear out the temporary photos captured or selected from a camera element.
- **New:** “Save Photo to Database” action in the “Camera” category to save the captured photo or selected mobile gallery photo from the camera element to the database.
- **Improved:** Optional Arguments no longer appear as gray “unfilled”.
- **Improved:** Action Blocks. Now when selecting actions from the control structures category, the “end” action is also added for you.
- **Improved:** Changed the desktop photo gallery back icon of the camera element.
- **Improved:** LowCode editor will now scroll to where new lines are added if they are off screen.
- **Improved:** Better reading of the user’s context when adding new lines. Lines will add one after the other when clicked consecutively. They will add to the bottom of an event after you select one. And when you select a line, the next action you add will be added below it.
- **Improved:** Display captured photos in the camera element’s desktop photo gallery with aspect ratio.
- **Improved:** Automatically resize the variable and data viewers if they are larger than the window size.
- **Improved:** Clear Argument right click menu option now works.
- **Improved:** setAvariable.action now accepts text and field elements as a source for ‘Value’.
- **Improved:** Multi-dimensional array support is more robust.
- **Improved:** Moved RecordIDs argument to the end as an optional entry. No entry is the same as providing an ’*’.
- **Improved:** Sign in performance improvements. Load from local if possible. Projects and Apps in the cloud created on different computers can be seen using “Refresh Apps” through the user icon in the Projects Screen.
- **Improved:** Added ability to output a camera element’s captured or selected photo ID to a variable. This photo ID represents a temporary image and will exist on the local device during the app’s session. The setting can be accessed in the camera element’s no-code setup.
- **Improved:** Support raw input as a source for dates.
- **Improved:** Accept fields as an element source.

- **Improved:** Updated and improved process for capturing photos and selecting photos from the mobile gallery in the camera element.
- **Improved:** setArrayValue.action will now accept variables and elements in addition to raw input for the path.
- **Fixed:** Fixed the ocassional bug with focusing newly added blank lines.
- **Fixed:** Commented Lines remain commented as user adds or deletes lines.
- **Fixed:** Adding an action to the bottom of an empty event would add the action to the wrong event.
- **Fixed:** Optional arguments now validate correctly
- **Fixed:** Commented actions also validate correctly
- **Fixed:** Fixed a bug in the tooltip for variable arguments in the low code editor.
- **Fixed:** New blank actions created using the + button will now correctly become focused.
- **Fixed:** Error when shift selecting from 2 different events. Shift-click is now ignored if clicked in a different event.
- **Fixed:** Fixed a bug where the orientation dropdown in the camera element's no-code setup was hidden.
- **Fixed:** When updating arguments of indented lines, the indent is no longer cancelled out. Similarly, unexpected indents also will no longer occur after updating arguments in an action line.
- **Fixed:** Make IfProperty and ElseIfProperty correctly display variable or input options.
- **Fixed:** Scrolling when the event flows have a vertical scroll bar is much improved.
- **Fixed:** Unfilled arguments will always render using the correct colour.
- **Fixed:** Lowcode display updates after adding blank event.
- **Fixed:** Make InputMethodColor work correctly.
- **Fixed:** Fixes left-clicking of events in the low code editor on Windows.
- **Fixed:** Fixes left-clicking of actions in the low code editor on Windows.
- **Fixed:** Can now shift-click to select a range of actions in the lowcode editor.

88.107 VERSION 1.2.0 (2023-04-13):

- **New:** Visual Low Code Editor: The Low Code Editor has been completely revamped. It now shows arguments visually, with an english-like syntax. You can type to search for actions and add them directly to events.
- **New:** You can now show App codes in the Apps screen.
- **Improved:** Explanation of "blank action" has been improved.
- **Improved:** Better Drag & Drop in Low Code editor
- **Improved:** Better recognition of events in a script. The dialog of whether to use MouseClick will only appear when the script editor has no prior context.
- **Improved:** Changed the order of arguments changed for better experience for setTheProperty.action.

- **Improved:** Merge notch properties into a single property
- **Improved:** Improved scrolling projects and template apps in the Projects Screen.
- **Improved:** Add image support for BMP and PICT files.
- **Improved:** Added properties “save to mobile device” and “show photo gallery” to the camera element.
- **Improved:** Added variables, elements, and input to RecordID argument for “get Array Value” action.
- **Improved:** getArrayValue.action now accepts variables, elements, and hard-coded input as a source for the path. Added record IDs delimited by LF or comma delimited and ’*’ for all record ids as an argument. The path can be in a multi-dimensional array and database records in Appli.
- **Fixed:** A bug that occured when you cut or delete actions from an event that contains commented actions.
- **Fixed:** Commented actions now copy, cut and paste correctly.
- **Fixed:** Allow setTheProperty action to take a variable or input.
- **Fixed:** Fixed a bug with the “repeat x number of times” control structure when the “number” argument is a variable.
- **Fixed:** Make wrap for text elements apply automatically when auto-height is set.
- **Fixed:** Text elements were moving incorrectly when zoom was changed while in a different orientation.
- **Fixed:** Fixed a bug which prevented scrolling template apps in the Projects screen with the scroll wheel.
- **Fixed:** Fixed a bug which would cause projects in the Projects Screen to cutoff when scrolling.
- **Fixed:** Fixed an error that could occur when editing elements of a parent element.
- **Fixed:** Autosave while zoomed and showing offscreen elements would sometimes cut off the screen
- **Fixed:** Fixes for importing csv file. Support for quotes in header. Improved error checking.

88.108 VERSION 1.1.0 (2023-04-05):

- **New:**
- **Improved:** Option-copy revisions and improvements.
- **Improved:** UI for the Projects Screen has been improved.
- **Improved:** Display an error message in the chart element if its data set contains invalid data types and cannot be displayed.
- **Improved:** Make text styles in assets manager manually add/delete.
- **Improved:** Make colors in asset manager manually add/delete.
- **Improved:** Support for drag and drop JSON file to create table.
- **Fixed:** Fixed a bug where the camera element may not be initialized after switching screens on desktop.
- **Fixed:** Fixed a bug where the browser, chart, and map elements may be

visible in play mode when they shouldn’t be on certain platforms.

- **Fixed:** Fixed an error that can occur when changing the text size or text style of a button element’s label.
- **Fixed:** Text styles in assets no longer track deprecated “dynamic text size”
- **Fixed:** Can now scroll templates in Templates Screen with the scroll wheel.
- **Fixed:** Handle errors that can occur when a chart element’s data set contains invalid data types.
- **Fixed:** Fixed a case where saving locally could cause the playground to have incorrect scroll positions.
- **Fixed:** Fixed a bug which would cause text elements to draw in the wrong location when an app is loaded.

88.109 VERSION 1.0.4 (2023-03-29):

- **New:** Added an action that generates random integers.
- **Improved:** Updated database actions dictionary entry where necessary. Introduced “ ”*for recordIDs argument. Certain database actions can accept ””* to specify all records in the table.
- **Improved:** Added “highlight elements” option to the contextual menu of components in the asset manager.
- **Improved:** Table element performance improvements when scrolling.
- **Improved:** Table element cache system improved.
- **Improved:** Improved scrolling apps in apps screen.
- **Improved:** Add option to save App projects from playground.
- **Improved:** Increase contrast for selection in project browser.
- **Fixed:** Closing an answer dialog will no longer cause native controls in elements to become visible in edit mode.
- **Fixed:** Fixed a bug where the preview image of the map element may display incorrectly in edit mode after loading an app.
- **Fixed:** Dropdown element no-code setup now shows the correct values in the key dropdowns when it is opened.
- **Fixed:** Fixed a bug that may prevent switching screens with the screen dropdown menu.
- **Fixed:** Table element would display empty rows when scrolling large data sets on desktop.
- **Fixed:** Table element visual fix on desktop.
- **Fixed:** Table element would highlight incorrect rows when pulling down to refresh on mobile.
- **Fixed:** Turning on multiple rows on layouts while zoomed would cause the rows to display with incorrect height.
- **Fixed:** Zooming in while showing offscreen elements would sometimes cut off the screen.
- **Fixed:** Fixed a bug where map and chart elements may not display live data when switching to a screen that contains these elements in play mode.

- **Fixed:** Fixed a bug where dragging elements within a tab element may cause it to be moved into elements on a different tab.

88.110 VERSION 1.0.3 (2023-03-22):

- **Improved:** Update template apps with descriptions and screenshot of apps
- **Fixed:** The location of newly created auto height text elements would jump up when resized.
- **Fixed:** Double-clicking on a button to change the label will now adjust the size of the label correctly to fit the new content.
- **Fixed:** Prevent elements from being renamed to blank name and remove any extra empty lines from the new name.
- **Fixed:** Input method variable will no longer use the hint text as the variable name.
- **Fixed:** Display correct sort order when loading datacontext for layouts.
- **Fixed:** Can now deselect one of the selected orientations in the player settings.
- **Fixed:** Clicking on elements inside a layout element row would not highlight the row.
- **Fixed:** Layout elements with multiple rows enabled that aren't connected to a table now highlight correctly.

88.111 VERSION 1.0.3 (2023-03-22):

- **New:**
- **Improved:** Update template apps with descriptions and screenshot of apps
- **Fixed:** The location of newly created auto height text elements would jump up when resized.
- **Fixed:** Double-clicking on a button to change the label will now adjust the size of the label correctly to fit the new content.
- **Fixed:** Prevent elements from being renamed to blank name and remove any extra empty lines from the new name.
- **Fixed:** Input method variable will no longer use the hint text as the variable name.
- **Fixed:** Display correct sort order when loading datacontext for layouts.
- **Fixed:** Can now deselect one of the selected orientations in the player settings.
- **Fixed:** Clicking on elements inside a layout element row would not highlight the row.
- **Fixed:** Layout elements with multiple rows enabled that aren't connected to a table now highlight correctly.

88.112 VERSION 1.0.2 (2023-03-15):

- **Improved:** Smarter responsive sizing for layout and tab elements.
- **Improved:** On account creation, trial tier is now permanently displayed and no longer requires entering a promo code.
- **Improved:** Shift-tab support for create account screen.
- **Improved:** Chart Element - Moved the Axis related properties to near the top of the property set, where they are closer to the dataset.
- **Improved:** Chart Element - Better calculations of SUGGESTED MIN/MAX and consistency of updates to graph.
- **Fixed:** Shift-click to highlight records in the data viewer would include duplicates of records.
- **Fixed:** Fixes for highlighting records in the table element. Shift-click to highlight a range of records now works. Records remain highlighted when scrolling.
- **Fixed:** Manage Lowcode Error Reporting Tool bug icon when a new app is created.
- **Fixed:** Prevent components from being added or pasted to a screen if the same component already exists on the screen.
- **Fixed:** Fixed a bug that would occur when pasting elements into a component.
- **Fixed:** Blank Chart Element Issue.

88.113 VERSION 1.0.1 (2023-03-08):

- **New:** Tooltips added to the property inspector's "text" section.
- **Improved:** Deleting elements is now much faster
- **Improved:** Added SuggestedMin/Max properties to enable better displaying of values that are at the min or max of the Y axis. Requires "begin at zero" property to be disabled.
- **Improved:** When creating components, name it with the element's name.
- **Improved:** dateDifference action by providing verbose and array as formats.
- **Improved:** Error checking for dateDifference action.
- **Improved:** Date calculation is more accurate and better supports a range of outputs.
- **Fixed:** Fixes for highlighting records in the data viewer. Shift-click to highlight a range of records now works. Records remain highlighted when scrolling.
- **Fixed:** Refresh_The_Form and Reset_Form actions will now correctly clear form data.
- **Fixed:** Fixed a bug where hint text in field elements may become highlighted when interacting with a field input dialog.

88.114 VERSION 1.0.0 (2023-03-01):

- **New:** Added Get Date Difference action to get the difference between two dates.
- **New:** Print Screen action (Desktop only)
- **Improved:** Appli's window will now retain its top-left location when switching to the login, projects, or playground screen.
- **Improved:** Reset scrollbars for groups and field elements when switching from play mode to edit mode.
- **Improved:** Added optional "records" argument to "sum the column" action in the "database" category to retrieve the sum of the values of the specified records.
- **Improved:** Combine The Values action now allows Value1 and Value2 to work with variables, input, and text/field elements.
- **Improved:** Combine The Values action now allows line feed (LF) to be an acceptable separator.
- **Improved:** Combine The Values action now allows the output to go to a text or field element.
- **Improved:** Added more details to the dictionary for launchURL action
- **Improved:** Allow users to copy content from the dictionary.
- **Improved:** Horizontal scrollbars will show up when needed.
- **Improved:** Added a 'dropDownSelection' event for the dropdown element.
- **Improved:** Make Undo/Redo when editing text element maintain cursor
- **Improved:** Make Undo/Redo when editing text element have more waypoints and act more consistently
- **Fixed:** Clicking "cancel" was still returning to app project screen
- **Fixed:** Pasting into an empty field element will now replace the hint text.
- **Fixed:** Fixed false positives with the Lowcode Error Reporting Tool related to actions being update with new optional arguments and events missing "end_if" actions.
- **Fixed:** Apps loaded from disk will now display the correct device type.
- **Fixed:** Can no longer uncheck both supported orientations for the player.
- **Fixed:** Fixed a bug where scrolling was affected when text was pasted into a text element.
- **Fixed:** Fixed a bug where the wrap property was affected when text was pasted into a text element.
- **Fixed:** Colors passed in name format would not convert to HEX for internal storage.
- **Fixed:** Fixed a bug where focus and blinking cursor may be missing when tabbing into a field element.
- **Fixed:** submitFormToDB action would not work with variable reference
- **Fixed:** Fixed a bug where element input selection doesn't occur and duplicates an action line in the low code editor.

88.115 VERSION 0.9.20 (2023-02-22):

- **Fixed:** Fixed a bug where new lines may sometimes be entered into a single-line field element.
- **Fixed:** No longer display fixed guidelines in play mode if zoom level is changed.
- **Fixed:** Fixed a bug that would prevent scrolling in an app with the scroll wheel.
- **Fixed:** Issue where error would be thrown when re-doing actions after undoing too many actions
- **Fixed:** Allow scrolling of the playground in play mode with mouse wheel and trackpad.
- **Fixed:** Allow both horizontal and vertical scrolling for groups, text, layout, and form elements with mouse wheel and trackpad.
- **Fixed:** Fixed a bug where ungrouping elements in a scrollable appligroup may cause the child elements to be positioned incorrectly.
- **Fixed:** Template apps will now display the correct device type when they are downloaded.
- **Fixed:** Handle errors with saving more gracefully.
- **Fixed:** Fixed a bug that would prevent populate_form_data action from working correctly on mobile.
- **Fixed:** Creating app would sometimes show error "The app you are trying to save no longer exists on the cloud".
- **Fixed:** Creating app would sometimes show error "There was a problem creating a guest user in your app: User already exists"
- **Fixed:** Contacts App would not show contact details.
- **Fixed:** Appli Invoice was updated on all regions.

88.116 VERSION 0.9.19 (2023-02-15):

- **New:** Added 'lock screen' and 'unlock screen' actions to help improve the visual performance of lowcode scripts which make multiple updates to the screen
- **New:** Compatibility settings in Player Settings tab. Enable this setting to prevent apps from running on older versions of Appli Player.
- **Improved:** Improved performance and error checking when adding a new screen.
- **Fixed:** Fixed an issue with crashing when using the Low Code Editor. Some coloration features were removed as part of this fix; they will be added back in the future.
- **Fixed:** Populate_form_data action now manages field element scrollbars.
- **Fixed:** On mobile, mouseClick actions will be skipped if the element's parent layout is being scrolled. (Fix adds code to the script)
- **Fixed:** Fixed a bug where the categories list in the low code editor wouldn't display all the categories until the refresh button was clicked.
- **Fixed:** Low code error inspector will no longer output errors for multi-

nested control structures that have matching open and end actions.

88.117 VERSION 0.9.18 (2023-02-08):

- **New:** “Show Player Menu” action in the “Player” category to display Appli Player’s in-app menu.
- **New:** “Refresh App” action in the “Player” category to refresh the currently loaded app in Appli Player.
- **New:** “Go to Player Home” action in the “Player” category to exit the current app and return to Appli Player’s home screen.
- **Improved:** Add selectedTabChanged event for tab elements
- **Improved:** When importing a project from disk that has a password, don’t attempt to create the project if the password is incorrect.
- **Improved:** Ask for new name when duplicating screens.
- **Improved:** Allow cdbRecordID to be the value of dropdown.
- **Improved:** appliGroup element will resize accordingly to fit newly added/removed elements from playground and project browser.
- **Improved:** The layout and row based layoutRefresh now work without error with search results. Also fixed stability and other less visible issues with using this combination of Appli features.
- **Improved:** Don’t allow changing of orientation for Desktop applications.
- **Improved:** Allow ESC key to save and exit low-code editor.
- **Fixed:** Fixed a bug that would cause apps to not load in the player.
- **Fixed:** Thumbnails would sometimes not update in UI.
- **Fixed:** Fixed a bug that prevented Cmd/Ctrl + Scroll to zoom in/out of the playground from working when the low code error inspector was open.
- **Fixed:** Fixed incorrect error reporting in the low code error inspector for control structures that have matching open and end actions.
- **Fixed:** Unnecessary undo steps were created with some dragging operations.
- **Fixed:** Fixed a bug where form data was not properly set internally, which would prevent access to the text value when referencing the affected element.
- **Fixed:** Fixed a bug where the low code editor error inspector may display an error when variables were used in action arguments.
- **Fixed:** Sidebar icons in the low code editor will now position correctly with horizontal scrolling.
- **Fixed:** App low code error inspector no longer outputs duplicate error reports for control structures.
- **Fixed:** Fixed a bug where the app low code error inspector may incorrectly output “missing screen reference” errors.
- **Fixed:** The scrollbar for non-native, multiline field elements will now display correctly when their height or width is changed.
- **Fixed:** Fixed an error that can occur when pressing the alt/option key while in edit mode.

88.118 VERSION 0.9.17 (2023-02-01):

- **New:** Added a setting to adjust how users can toggle the menu in Appli Player. Options include icon, long press, and triple tap. To change this setting, load an app, select the gear icon on the right-hand side of the header, and select the "player" tab.
- **New:** "searchPerformed" lowcode event for search element.
- **New:** UNDO feature for the LowCode Editor. Now you can undo recent actions you have taken. Simply click the Undo button, which shows you how many undo steps there are. The undo buffer clears each time the LowCode editor is closed.
- **Improved:** Lowcode's internal bug popup recognizes missing variable issues
- **Improved:** Added back button to device selection screen when creating new app.
- **Improved:** Allow non-"appliproject" files to be opened by Appli IDE (option must be selected in file picker).
- **Improved:** Lowcode Error tool - Added the ability to increase and decrease font-size by holding Control /Command and using wheel mouse, Verification tool now remembers the Vscroll when re-opening after low code editor has been closed.
- **Improved:** Adjust text size of the message in the loading dialog box.
- **Improved:** Autosize property added on login and createAccount element.
- **Improved:** Add option to save users when saving a project to disk.
- **Improved:** Updated Appli to the latest CanelaDB.
- **Fixed:** Display flicker issue when opening arguments.
- **Fixed:** Loading a project from disk would sometimes show incorrect snapshot and app name.
- **Fixed:** Deleting images/video from a disk-loaded project would sometimes cause the original to be affected.
- **Fixed:** Sorting a table element now displays local data correctly.
- **Improved:** Restored the message for the user telling them about saving fixes and changes to their scripts made by the error module. It is displayed only once per session.
- **Fixed:** Display issue with deprecated actions, shortened more error strings.
- **Fixed:** Appli projects with multiple apps can now be loaded from disk.
- **Fixed:** Stop dropdown element from making unnecessary db calls when zooming and resizing, which would sometimes cause Appli to freeze.
- **Fixed:** Re-index dropdown element table reference when project is loaded from disk.
- **Fixed:** Fixed a bug that would occur when applying lowcode changes to a project that was loaded from disk.
- **Fixed:** Fixed a bug that could occur with the create account and login element.
- **Fixed:** Handle scrollbar visibility for non-native field elements when text

property is changed and reset the scroll position if it isn't active.
- **Fixed:** Layout actions which specify row data now correctly display search results.
- **Fixed:** Layout actions which specify row data now work correctly with duplicated layouts.

88.119 VERSION 0.9.16 (2023-01-25):

- **New:** Lowcode Error Reporting Tool (Bug Icon in Footer). This tool will report: Missing Screens, Missing Variables, Missing Elements, Deprecated Actions, Changes in Argument Set, Missing Arguments, Errors in Conditional Structures. It will also check for Lowcode integrity issues. Use the pencil icon to open the script to correct the errors, and hit "apply" to save the corrected data.
- **New:** You can now save and load projects from disk.
- **Improved:** Image elements will now display the default image when their blob is deleted.
- **Improved:** UI update for Projects screen.
- **Fixed:** Fixed a bug that could occur in the lowcode editor when clicking on an event flow.
- **Fixed:** Fixed a bug with the button element that prevented the "fill" property from being changed in play mode if the "ripple effect" property was on.
- **Fixed:** Fixed an error that could occur when adding more than one event in the low code editor.
- **Fixed:** Don't scroll the playground when scrolling the variable viewer.
- **Fixed:** Screen will now stop jumping when scrolling to the beginning/end with a mousepad.

88.120 VERSION 0.9.15 (2023-01-18):

- **New:** New action set_camera_output. Sets the output element for a camera element.
- **New:** clearVariable.action added to set a variable's value to empty.
- **New:** "Ping" and "Ping Node" actions in the "database" category to ping the cloud server where a specified or random table is located.
- **New:** New action select_record_in_layout. Selects the specified record in a layout element.
- **New:** New action unhighlight_the_layout. Unhighlights the highlighted row of a layout element.
- **New:** LowCode Script Error Module
- **Improved:** Make text responsive resize based on smaller of width or height
- **Improved:** Add cancel button to icon picker. Make button icons default to previous chosen icon.

- **Improved:** Allow text elements to have mixed textSize (selected chunks of text can have different text size).
- **Fixed:** Height/Width property inspector editors were disabled every time Appli restarted.
- **Fixed:** Fixed an error that occur with the camera element when the device camera is being used by something else.
- **Fixed:** Clicking on icon in the dock (MacOS) will now un-minimize the program.
- **Fixed:** Label was not showing up properly for switch and radio element.
- **Fixed:** Dragging elements into tab elements via the project browser was causing errors.
- **Fixed:** display_records_in_layout would display all the records in the table when the recordIDs argument was empty.
- **Fixed:** Fixed a bug that occurred in the low code actions list when the low code editor was re-opened.
- **Fixed:** Icon Picker will now display the correct icon.

88.121 VERSION 0.9.14 (2023-01-11):

- **New:** New properties “Highlight Row on Click” and “Highlight Color” for layout elements with “Multiple Rows” enabled.
- **Improved:** Change Text Elements autosizing and responsive geometry options.
- **Fixed:** Fixed UI for icon dropdown menu on text/icon layout.
- **Fixed:** Fixed a bug where native controls in elements would be shown after making a selection in message dialog boxes in the low code editor.
- **Fixed:** Dropdown element backwards compatibility with actions.
- **Fixed:** Empty label on button element was causing app to crash after reopening from project screen.
- **Fixed:** Preview image was getting deleted after undo command following an image upload.

88.122 VERSION 0.9.13 (2023-01-04):

- **New:** New action display_records_in_layout. Displays the specified records in a layout element.
- **Improved:** Dialog prompt for user to specify output format and prevent empty search result to text & dropdown element.
- **Improved:** Dropdown element - lastSelectedText property has been corrected, and lastSelectedValue property has been added.
- **Fixed:** Save locally when pasting elements or creating line elements.
- **Fixed:** Prevent multiple clicks of the login button in the login element from firing.
- **Fixed:** clear_search action was only accounting for user input and not no-code queries.
- **Fixed:** Fixed an error that would occur when deleting copies of elements.

- **Fixed:** Creating a copy of an element with the option key would group the element and the new copy.
- **Fixed:** Lowcode/No-Code now open correctly right after creation of new field

88.123 VERSION 0.9.12 (2022-12-20):

- **New:** Includes new property editor to create label / value pairs for dropdown elements.
- **Improved:** "App Project" option to return to app project screen.
- **Improved:** The dropdown element has a new property editor for label / value pairs. Label / Value database keys can be specified for dropdowns accessing tables. (In the no-code editor)
- **Fixed:** Reset cursor after signing out.
- **Fixed:** Handle errors in actions where a referenced element no longer exists.
- **Fixed:** Graphic name changed to re-enable return/enter key when creating new app.
- **Fixed:** Fixed relayering bug in Appli Stack
- **Fixed:** There was a reference error in the relayering of controls which moved controls from one "setup" window to another.
- **Fixed:** Fixed a bug where deleting an app from a project would cause apps from other projects to disappear from the list.
- **Fixed:** Fixed issue with incorrect relayering, causing issues in selection of table names.
- **Fixed:** Filtered out date operators in FindValueInArray action.
- **Fixed:** Reset cursor when leaving playground.

88.124 VERSION 0.9.12 (2022-12-20):

- **New:** Includes new property editor to create label / value pairs for dropdown elements.
- **Improved:** "App Project" option to return to project screen.
- **Improved:** "App Project" option to return to app project screen.
- **Improved:** The dropdown element has a new property editor for label / value pairs. Label / Value database keys can be specified for dropdowns accessing tables. (In the no-code editor)
- **Fixed:** Reset cursor after signing out.
- **Fixed:** Handle errors in actions where a referenced element no longer exists.
- **Fixed:** Graphic name changed to re-enable return/enter key when creating new app.
- **Fixed:** Fixed relayering bug in Appli Stack
- **Fixed:** There was a reference error in the relayering of controls which moved controls from one "setup" window to another.

- **Fixed:** Fixed a bug where deleting an app from a project would cause apps from other projects to disappear from the list.
- **Fixed:** Fixed issue with incorrect relayering, causing issues in selection of table names.
- **Fixed:** Filtered out date operators in FindValueInArray action.
- **Fixed:** Reset cursor when leaving playground.

88.125 VERSION 0.9.11 (2022-12-14):

- **New:** Action to read a key value of an element in the current row of a layout into variable.
- **New:** Action to change a property of an element of the current row from variable.
- **New:** "layoutRefresh" event flow to be used in the low-code script of elements inside a layout.
- **Improved:** Changed the "Done" button to "Create" when creating a new app.
- **Improved:** Clear action panel search input when clicking on category.
- **Improved:** "Go To Screen" action now accepts variables as input.
- **Improved:** Auto-highlight search inputs on category and action panels.
- **Improved:** Display comment's input option directly with one click instead of from a dropdown.
- **Improved:** Variables created in the variable viewer can now be named and are shown in variable dropdown lists.
- **Improved:** Keys added to a variable in the variable viewer can now be named.
- **Fixed:** Handle errors in actions where a referenced element no longer exists.
- **Fixed:** Reset cursor after signing out.
- **Fixed:** Fixed a spelling mistake in a dialog message.
- **Fixed:** Apps would load when double-clicking on an app project.
- **Fixed:** "Open screen" and "close screen" events were missing from the events list when modifying low code for screens.
- **Fixed:** Moving action lines within the low code editor will no longer cause input method windows to occasionally appear on screen.
- **Fixed:** Allow "set element value" action to work with empty inputs in the "value" argument.
- **Fixed:** Lowcode search fields were accepting enter key and returning empty list.
- **Fixed:** Fixed an error that can occur while resizing columns in the data viewer.
- **Fixed:** Fixed an error that could occur when selecting text in a text element.
- **Fixed:** Values in nested arrays of variables can now be modified in the variable viewer and will remain in sync.
- **Fixed:** Fixed an error that would occur when moving an element inside

of the first row of a layout element with multiple rows enabled.

- **Fixed:** Increased minimum size for the variable viewer to fit the editor window.

88.126 VERSION 0.9.10 (2022-12-07):

- **New:** Added a “LastSelectedText” internal property to the dropdown element. This is needed when we want to read what the user had selected.
- **New:** Added lockedText property to lock Field elements.
- **New:** The FindValueInArray action is used to search through the keys of an array to match one of its keys to one of the actions parameters. When found, the key value or the whole record can be returned.
- **New:** resetForm action to clear form element.
- **New:** Create a layout from selected elements.
- **New:** Guidelines for layout row when layout element has Multiple Rows enabled.
- **Improved:** SubmitFormToDB action can now place the recordID into a variable.
- **Improved:** When pasting elements, if you have an element selected it will paste into that element (if it can have children) or into the element’s parent
- **Improved:** Enable cancel button to stop creation or grouping of elements from dialog box.
- **Improved:** Allow dragging of any direction when corner radius dots overlap, and make dots appear more consistently.
- **Improved:** Delimiter parameter added to appendToValue.action
- **Improved:** Variables are no longer deleted automatically if no elements or screens are using them. They can be deleted through the variable viewer (Shift + V).
- **Improved:** Improvements for copying elements with the option key. While moving an element around, press and hold the option key to create a copy of the element in the location it was in before it was moved. Release the option key while the element is still being moved and the copy is removed. Release the mouse while the option key is still down and the copy remains.
- **Improved:** Measure distances between elements via option key.
- **Improved:** Change hotkey for “show distance” to Alt/Option. This will also show distances when resizing from center.
- **Fixed:** Basic low-code mouse events are now available for dropdown, tab, table elements.
- **Fixed:** Renamed grouping contextual option on PB to match playground.
- **Fixed:** Allow dynamicHeight text elements to draw correctly when vertical align is set
- **Fixed:** Make corner radius dots display at the correct distance when zoomed.
- **Fixed:** When resizing window, sometimes the screens gallery would draw

in the wrong location.

- **Fixed:** Stop allowing resizing of elements less than 0 width/height beyond their bottom/right sides.
- **Fixed:** Fixed a bug where some keyboard shortcuts wouldn't work if the variable viewer or data viewer was open.
- **Fixed:** Fixed a case where local saves may cause the playground to scroll to the top and left.
- **Fixed:** When resizing from center, guidelines will properly snap on edges opposite of the dragged edge.

88.127 VERSION 0.9.9 (2022-11-30):

- **New:** appendToValue.action appends a value to a column of the selected recordID.
- **New:** Can now create a form element by right-clicking on selected elements and selecting "Create Form Element" in the contextual menu.
- **Improved:** Performance improvements for scrolling layout elements.
- **Improved:** Set tablename parameter before data parameter for better workflow.
- **Improved:** Can now view the cdbBlobs table in the data viewer.
- **Improved:** Resizing of the data manager to avoid getting covered by the scrollbar.
- **Improved:** Allow drawing from center via option key when creating new elements.
- **Improved:** Auto focus on search for low code editor inputs.
- **Improved:** Store new accounts with the email set to lowercase.
- **Improved:** Allow resizing from center by holding down the option key while resizing.
- **Improved:** Make buttons ripple by default.
- **Fixed:** Fixed an error that occurred when trying to delete a key from a table in the data manager.
- **Fixed:** Fixed an error that occurred when right-clicking a tab group in the project browser.
- **Fixed:** White, text components are now displayed with a gray background in the asset manager.
- **Fixed:** Guidelines no longer appear for elements that are not visible.
- **Fixed:** The asset manager search function will now filter components.
- **Fixed:** Data-linked images in a layout element now respect the images original aspect ratio.
- **Fixed:** TextStyle would sometimes cause error when selected.
- **Fixed:** Text pasted into fields of the property inspector will no longer retain original formatting.
- **Fixed:** Support older accounts where emails may have mixed casing.
- **Fixed:** The sign-in button hover state and adornments are broken.
- **Fixed:** Appli does not correctly handle signing into a region where a user does not have an account.

- **Fixed:** Option-drawing of line from center.
- **Fixed:** Newly created elements in the project browser will now match the left pane's width.
- **Fixed:** Fixed queryRecords action, re-ordered parameter.
- **Fixed:** sortRecords action had mismatched parameters.
- **Fixed:** Make nudging of elements work at all zoom levels
- **Fixed:** Make nudging of elements work when multiple elements are selected

88.128 VERSION 0.9.8 (2022-11-23):

- **New:** Validate date action.
- **New:** Guidelines appear near the edge of the screen when you are going to create an element. If an element is created when these guidelines are visible, the element will snap to those edges.
- **Improved:** Updated properties validator to make layout elements backwards compatible with the property name change.
- **Improved:** Checking for empty table names and rearranging TableName parameter for better workflow.
- **Improved:** Updated variable creation and selection in table element's no-code setup.
- **Improved:** newly created components are unlocked by default, keeping the master component in the asset manager in sync with its instances.
- **Improved:** Changed "don't wrap" property to "wrap" in text elements.
- **Fixed:** Prevent table names with multiples lines in the table setup.
- **Fixed:** Displaying the cdbUsers table in a table element would cause an error.
- **Fixed:** Comparing date was not outputting the correct result.
- **Fixed:** Fixed a bug where the "delete selected action(s)" option can cause blank lines in the low code editor.
- **Fixed:** Hide data viewer when an app is loaded.
- **Fixed:** Unify record ID property for layout and form element.
- **Fixed:** Fixed a bug that could occur when releasing the mouse over the screen background.
- **Fixed:** Reset scroll position in form elements when populating data into its linked elements.
- **Fixed:** Retain playground scroll position when an element has been deleted.
- **Fixed:** An error will no longer occur when renaming keys in the data manager.
- **Fixed:** Stop fixed guideline indicator from drawing strangely in some instances.
- **Fixed:** Error when going to play mode when there is a Form that has element inside it that have been moved.
- **Fixed:** Allow odd-lengthed dimensions to match centers for alignment.
- **Fixed:** Retain playground scroll position when capturing screen thumb-

nails.

88.129 VERSION 0.9.7 (2022-11-16):

- **New:** "findDayofWeek" action determines what day of the week it is from the given date.
- **New:** "checkDateRange" action checks if the date is 'below', 'above', or 'within' the specified range.
- **New:** "compareDates" action determine if the date occurs before or after the specified date.
- **New:** "addValueToDate" action adds value (day, month, or year) to the specified date.
- **New:** "subtractValueFromDate" action subtracts a value (in day, month, or year) from a date.
- **New:** Map Element can now geocode addresses and fill in lat & lng (uses Google GEOCODING API)
- **Improved:** Keyboard shortcut Cmd + Q (macOS) or Alt + F4 (Windows) can be used to quit Appli.
- **Improved:** Update guidelines to work more smoothly while zoomed.
- **Improved:** Display of maps on mobile devices. Ability to select maps with click. Map does not update when resizing height. Maps can not be edited without an API Key.
- **Improved:** arrow key support for numerical input fields in the property inspector. Use the up and down arrow keys to adjust numerical values
- **Improved:** Updated keyboard shortcuts for element layering. Bring to front is Shift + Cmd +] or Ctrl + Shift +]. Bring forward is Cmd +] or Ctrl +]. Send backward is Cmd + [or Ctrl + [. Send to back is Shift + Cmd + [or Ctrl + Shift + [.
- **Improved:** All native controls will hide correctly when displaying answerDialog, in play or edit mode.
- **Improved:** Map Element Title & Subtitle now both display when clicking on a map marker
- **Improved:** Remove cut/copy screen from contextual menus in favor of duplicate screen.
- **Improved:** Display the current orientation for all screens in the screen thumbnail view.
- **Improved:** Layout element performance improvements.
- **Improved:** Adjust the default text alignment when clicking to create text elements to Left.
- **Improved:** Turn off text wrapping as a default.
- **Improved:** Make left pane minimum width larger to accomodate for scrollbars in asset manager
- **Improved:** Make elements move to front of layouts/forms/groups when they are dragged in
- **Fixed:** Fixes for errors that could occur when moving elements into, out of, relayering/deleting/creating child elements, and ungrouping appli-

group components.

- **Fixed:** PI was losing focus on map element's marker property field input.
- **Fixed:** PI regression bug that didn't disabled copy+paste shortcuts

88.130 VERSION 0.9.6 (2022-11-09):

- **New:** Updated Map Widget. This is the first iteration of the new map widget based on Google Maps API. You will need a google API key, which you can get here: https://console.cloud.google.com/project/_/google/maps-apis/credentials
- **New:** "closeField" low-code event to detect when field input has changed.
- **New:** "deleteArrayElement" action.
- **New:** "setArrayValue" action provides a method to modify a value in an array.
- **New:** "parseArray" action to allow developers to get values from an array.
- **New:** "get_user_first_name" action. Gets the first name of the logged in user.
- **New:** "Populate Form Data" action in the "Form" category to fill a form's linked elements with data.
- **New:** Low-code editor: Add support for updated value to a dictionary entry.
- **Improved:** Converted Linegraph element to be named "Chart"
- **Improved:** Ability to scroll the form element by toggling the "show scrollbar" property.
- **Improved:** Add layering commands to right-click and hotkeys. [,],{, and } will send backward or forward or to back or to front
- **Improved:** Data manager UI improvements – will default to first table, and will resize when resizing the left pane.
- **Improved:** Make dynamicWidth for text elements anchor based on text alignment.
- **Improved:** Add elements as an optional value in setArrayValue.action.
- **Improved:** Improve form performance when looking at cdbUser table.
- **Improved:** Show cdbUsers table in data viewer.
- **Improved:** Provide indication refresh categories have been completed with a beep.
- **Improved:** Adjust readUserRecord.action to not have userEmail as the primary key in the array output.
- **Fixed:** App projects will sort correctly when returning from the playground to the Apps Home screen.
- **Fixed:** Display dictionary entries for commented-out actions in the low code editor.
- **Fixed:** Problem where screens would sometimes partially draw.
- **Fixed:** Tables created from a CSV file were defaulting to "cloud" instead of user selection.
- **Fixed:** Make formatting of hex values in color of asset manager more consistent.

- **Fixed:** Make fixed guideline indicator hide when changing zoom levels.
- **Fixed:** Issue where clicking on a property would give an error that the element does not exist.

88.131 VERSION 0.9.5 (2022-11-03):

- **New:** "Filter Duplicate Lines" action in the "Utility" category to remove duplicate lines from an input source.
- **New:** "Sort Lines" action in the "Utility" category to sort lines from an input source.
- **New:** "Convert Month" action in the "Date/Time" category to convert to a month's name or number.
- **New:** "Perform Search" action in the "Search" category to command a search element to perform a search query.
- **New:** Array support for data viewer. Arrays are displayed as "Array Value" in the table. Clicking on a cell with "Array Value" in it will display the array in a tree where you can view its contents
- **Improved:** Added LineGraph setting for using Font Size Ratios on Apple or Android
- **Improved:** Added LineGraph sliders for the font size of hover tooltips (Title and Body)
- **Improved:** The changelog now uses New,Improved,Fixed for categories.
- **Improved:** Updated launchURL action - date to the expected format.
- **Improved:** Added more URL examples to the dictionary
- **Improved:** Ability to set HTMLText using the low-code editor and render the field as HTML by checking new "Render as HTML" checkbox. If htmlText is not set then it is ignored and the "text" property is rendered.
- **Improved:** Resize items in the project browser when the left pane is resized.
- **Improved:** Right-click menu of action lines now lets you Comment, Un-Comment one or more actions.
- **Improved:** Moved the "Clear Search" action from the "Text" category to the "Search" category.
- **Improved:** Allow field elements to be used as parameters for launchURL action.
- **Improved:** Allow field elements to be used as parameters for setElement-Value action.
- **Improved:** Add multiline support for table element.
- **Improved:** Made deleting records in a table element more reliable.
- **Improved:** Add support for cdb_Users table in table setup.
- **Improved:** Layouts of the buttons on creating an account screen. Added dialog after create account UI creation for user feedback.
- **Fixed:** Handle an error that can occur when selecting the "property" argument in property related actions if the selected element doesn't exist.
- **Fixed:** Possible crashing bug when selecting arguments in Low Code editor.

- **Fixed:** Pie chart now displays correct min / max values of Y axis.
- **Fixed:** Copying and pasting multiple lines at once would cause the pasted lines to be out of order
- **Fixed:** You can now view downloaded template apps on the player without having to re-save the app.
- **Fixed:** Fields contents are not cleared as expected.
- **Fixed:** Fix a bug that could occur when deleting a newly created record from a table element.
- **Fixed:** Stop a bug that could occur when updating a record in a table element
- **Fixed:** Fixed a bug that would prevent a table element error dialog from displaying.
- **Fixed:** Horizontal scrolling in the low code editor body was causing issues.
- **Fixed:** Possible execution error when generating app code.
- **Fixed:** Issue where downloading a template app would delete blob data associated with the template.
- **Fixed:** No-code editor wasn't working on layout with images.
- **Fixed:** Fixed a bug where some cases of text elements may not apply "show scrollbar" or "truncate text" properties.

88.132 VERSION 0.9.4 (2022-10-26):

- **New:** Data viewer. Data manager now has a "View Table" button. This view allows you to create, read, update, and delete data.
- **New:** Low code - "Delete Selected Action(s)" command
- **New:** "Get Element Value" action in the "Utility" category to retrieve the value of the specified element.
- **New:** "Set Element Value" action in the "Utility" category to set the value of the specified element.
- **Improved:** Low code - Confined selection to one event flow at a time
- **Improved:** New properties to set a mobile ratio for the axis label font size and font size for LineGraph Element
- **Improved:** LineGraph Element interactivity as numerical values are all sliders and update as you slide
- **Improved:** Make project browser's contextual menu to match playground
- **Improved:** Added cdbUsers table to layout no-code.
- **Improved:** Corner radius property on login element and creating account element.
- **Fixed:** Low code - Fixed selection highlight being removed by red/green indication
- **Fixed:** Low code - Shift Selection sometimes created inaccurate visual state
- **Fixed:** Low code - Other issues fixed by improvement below
- **Fixed:** Fixed a bug where an error may occur when selecting a radio element's options or changing its "orientation" property.
- **Fixed:** Add better error messaging when downloading template apps

- **Fixed:** Right clicking on locked element was causing execution error
- **Fixed:** Allow users to edit forms after child element was deleted
- **Fixed:** Issue with autosized buttons not retaining the correct fontsize when opening app.
- **Fixed:** Clearing recordIDs when selecting tables in layout no-code UI.
- **Fixed:** Don't show dropdown element scrollbar when resizing the element in edit mode.
- **Fixed:** Fixed a bug that prevented scrolling the dropdown element's options with the scroll wheel.
- **Fixed:** Fixed a bug where child elements of the initial, saved component element weren't in sync.
- **Fixed:** No-code editor was not able to work on layout element containing images
- **Fixed:** 'Done' button wasn't closing dimension setting modal view
- **Fixed:** Issue where Appli would stay on a blank screen after deletion
- **Fixed:** Allow clicking to change cursor position when modifying names in the asset manager.
- **Fixed:** Changes in properties of nested child elements in components will now stay in sync.

88.133 VERSION 0.9.3 (2022-10-19):

- **New:** "Verified" argument for create_account action. Users that are verified will not receive a verification email.
- **New:** count_Lines_In_Variable action. Gets the number of lines in a variable.
- **New:** read_user_record action. Reads the user record from the cdbUsers table of the user with the email provided.
- **New:** The action script allows other default applications to open the URL scheme (sms, telephone, websites) within their respective devices
- **Improved:** Appligroup elements can now use the "loginSuccess", "loginFailed" and "accountCreated" events
- **Improved:** Updated argument names to be consistent for the following actions: "convert date", "convert date time" and "convert time".
- **Improved:** Removed unneeded argument "TimeFormat" from "convert date time" action.
- **Improved:** Improved error checking for LiveCode integration action.
- **Improved:** Low code - Ability to copy entire eventflow handlers
- **Improved:** Low code - Ability to right click event flow header and select paste to paste to the end of handler
- **Improved:** Low code - You can now double-click on an event to add it to a flow
- **Improved:** Updated low code error verification to handle additional cases of arguments with deleted or renamed elements.
- **Improved:** "value" argument for query_Records action can now take a variable

- **Improved:** Ask user to define layout/form names on creation
- **Improved:** Stop icon is now shown in the header when currently in play mode.
- **Improved:** Restrict certain hotkeys from triggering in play mode.
- **Improved:** Change hotkey for toggling between play and edit mode from "P" to "Escape".
- **Improved:** Change hotkey for clearing copied elements and screens from "Escape" to "CMD + Escape/CTRL + Escape.
- **Fixed:** Low code - Fixed bug where the incorrect action argument is updated
- **Fixed:** Make multiline fields wrap text on desktop platform
- **Fixed:** Make dynamic text sizing work with wrapped text
- **Fixed:** Zooming with offscreen elements shown was sometimes causing issues
- **Fixed:** Toggling visibility of fixed guidelines now apply to all screens.
- **Fixed:** create_account action would trigger accountCreated event when account failed to be created
- **Fixed:** Send_email action was using the ToEmail argument instead of the EmailBodyTEXT argument for the body of the email
- **Fixed:** Removed a trailing LF in the output after using readKeys action.
- **Fixed:** Made dictionary entry corrections to the following actions: "convert date", "convert date time" and "convert time".
- **Fixed:** Add a check to "submit form to DB" action to prevent an error if the specified form element no longer exists.
- **Fixed:** allow renaming of screens in Project Browser. Improved speed when renaming screens
- **Fixed:** Populate correct keys in the text element's no-code dropdown for templated text.
- **Fixed:** Layouts will now replace templated text with empty when the key was never set in the database

88.134 VERSION 0.9.2 (2022-10-12):

- **New:** "Reset element" action in the "utility" category to reset certain types of elements to their default state.
- **Improved:** Make touchpoints for dropdowns in PI bigger
- **Improved:** Make button elements autosize by default
- **Improved:** Manage singular vs plural language for script error dialog.
- **Fixed:** Fixed "copy" and "cut" options when right-clicking a screen in the project browser.
- **Fixed:** Low code - Prevent line selection when arguments are unfolded
- **Fixed:** Low code - Prevent popup of arguments when dragging lines and letting go
- **Fixed:** Low code - Fixed hiding of arguments checkbox when low-code editor closed
- **Fixed:** Low code - Fixed drag/drop broken operation

- **Fixed:** Low code - Fixed Argument Triggering
- **Fixed:** Low code - Invisible Script Error Number
- **Fixed:** Low code - Fix for dragging actions to another line
- **Fixed:** Low code - Copy not reliable when selected lines are not contiguous.
- **Fixed:** Low code - Fixed a bug where input methods would not appear when clicking on action arguments
- **Fixed:** Low code - Check actions for errors when commenting or uncommenting events.
- **Fixed:** Low code - If an element has already been selected in Element Input Method, maintain that selection, including orientation
- **Fixed:** Low code - Fixed a bug where selecting the value argument in the "else if property" control structure in the low code editor always displayed a dropdown.
- **Fixed:** Low code - Fixed a bug where the incorrect input method may be shown in the low code editor when selecting the value argument in "set the property" action, "if property" control structure, or "else if property" control structure.
- **Fixed:** Low code - Adding an action after selecting a line in an event other than the first event would cause the action to display incorrectly
- **Fixed:** Low code - Fixed a bug where the low code right-click menu for a selected action line may not display "cut" or "copy" options.
- **Fixed:** Minimum size for low code editor events section
- **Fixed:** fixed a bug that would occur when deleting a screen with the right-click menu in the project browser
- **Fixed:** Make screen responsively resize when changing to custom platform size
- **Fixed:** Make refresh the form action work when there is no data
- **Fixed:** Make copy paste of forms and layouts retain internal links to their children
- **Fixed:** Make copy paste of low code actions that specify specific element types maintain links
- **Fixed:** Copying and pasting dropdown element was retaining original values event after changing properties.
- **Fixed:** Add overlay to template editor for datacontext/field

88.135 VERSION 0.9.1 (2022-10-05):

- **New:** New "get the version" action in the "utility" category which retrieves the current version of Appli IDE or Appli Player the device is running.
- **New:** Allow Dropdown elements to be collapse/expanded via context menu while in edit mode
- **Improved:** Low Code - Commented Lines are now ignored in script verification.
- **Improved:** Low Code - New actions can be added to the bottom of event

or below the last of the selected lines. To control where new actions are added, select a line and the next actions will be added below it.
- **Fixed:** Issue when Radio Element had multiple options with a number as their first word.
- **Fixed:** Allow setting of text from asset manager when text size is dynamic
- **Fixed:** Update script inspector's error description if it's displayed when low code changes are made.
- **Fixed:** PI was losing focus and deleting lines after each option input.
- **Fixed:** References to LowCode events are deleted from message dispatch when elements are deleted.
- **Fixed:** Low Code - Disabled the possibility of accidentally selecting the hidden "end" line of an event flow
- **Fixed:** Low Code - fixed a bug where adding an action to the default event would cause subsequent actions added to the event to be disconnected
- **Fixed:** Fixed a bug where setting the line height of a text element would have no effect
- **Fixed:** Maintain accuracy when moving rows in the table element.
- **Fixed:** low code script error inspector now displays errors for deleted screen and element arguments
- **Fixed:** Handle setting text font/alignment/style for selected chunks better
- **Fixed:** Fix an issue where selecting the font dropdown would cause text to be unselected
- **Fixed:** Fixed a case where some deleted or renamed elements in arguments of the low code actions weren't being handled
- **Fixed:** Fixed a bug that would cause a color component in the asset manager to disappear if its color was changed and the same color was selected
- **Fixed:** The text element was storing the name "text" even if user entered a different name.

88.136 VERSION 0.9.0 (2022-09-28):

- **New:** Low Code - Script Error Inspector
- **New:** Version number to sign-in screen
- **New:** Copied/cut actions can now be pasted into another element.
- **New:** 'Count Record' action for layout element
- **Improved:** Low Code - Better recognition of missing END statements in control structures.
- **Improved:** Low Code - Dragging action lines between events
- **Improved:** Low Code - UI of code line options
- **Improved:** Make editing button label resizing more real-time.
- **Improved:** Ensure Appli developer is authed when switching from play mode to edit mode.
- **Improved:** Add marketing opt-in checkbox to create account page
- **Fixed:** Low Code - Script Errors Field appearance

- **Fixed:** Hide photo viewing interface when re-initializing camera element.
- **Fixed:** Fix execution error when opening Appli for the first time
- **Fixed:** Elements created from a component will no longer stay in sync when their component is deleted from the asset manager
- **Fixed:** search element field was disabled after changing zoom levels
- **Fixed:** Don't change text font for password input fields of create account element when text font property is changed.
- **Fixed:** Fix issue where text element text couldn't be highlighted and changed.
- **Fixed:** Setting the password property of a field from a "set the property" action to true/false did not work on desktops.
- **Fixed:** Make all options work for button layout
- **Fixed:** Update seach output field with correct element types
- **Fixed:** Search element incorrectly repopulating data
- **Fixed:** Layout element was getting the incorrect endpoint range for the number of rows and chopped off data after resetting the search element.
- **Fixed:** Prevent an error from occurring when adjusting text color property while editing a button element's label text.
- **Fixed:** Double-clicking to edit a button element's label didn't allow the insertion point to move to the clicked location.

88.137 VERSION 0.5.1 (2022-09-21):

- **New:** Action script to clear search element
- **New:** Low Code - Comment Action: displays as Greyed Out (after argument is set). To change comments, user must unfold action
- **New:** getLineFromVariable action. Get the specified line from a variable
- **New:** Answer dialog action. Display a message with 1-3 choices as buttons
- **Improved:** Low Code - Argument Unfolding
- **Improved:** Low Code - Code Line Options
- **Improved:** Low Code - Collapsing / Uncollapsing
- **Improved:** Low Code - Rearranging within Event Flow
- **Improved:** Low Code - When adding a conditional structure, a whole block is added to help give clarity to user
- **Improved:** Low code - update visual UI when lines are cut
- **Improved:** Low code - Improved repositioning of Event Flows
- **Improved:** Scrolling for layout element on desktop
- **Improved:** The minimum IDE width support 1080 for Mac and 1350 for Windows.
- **Improved:** Allowed sizing the IDE to a width of 1080 to help when developing on smaller monitors.
- **Improved:** Search element - set Element types with drop down options.
- **Improved:** Automatically handle "missing data from array" error.
- **Improved:** Make scrolling speed faster in Low Code
- **Improved:** Setting the file of an image element through the property inspector will now reset to the natural aspect ratio of the image

- **Improved:** Label Placement Options and Text alignment for switch element.
- **Fixed:** Low Code - Fixed Display error when removing entire events
- **Fixed:** Low Code - Fixed indenting of IF conditions
- **Fixed:** Low Code - Fixed error triggered when Appli IDE already has existing flows
- **Fixed:** Fixed a bug that would cause an error when refreshing a layout element on desktop
- **Fixed:** Don't move the cursor to the end while typing in non-native field elements.
- **Fixed:** Search element - Clearing output with trash can icon.
- **Fixed:** Fixed a bug that would prevent actions deleteKeys and readKeys from displaying the table keys correctly
- **Fixed:** Lock Text toggle now works with autoHilite set to false. Text Lock Icon is fixed to match with dynamic text size toggle.
- **Fixed:** Make spacings with tab key always disappear correctly
- **Fixed:** Make CSV import work again
- **Fixed:** Fixed a bug that would prevent listRecords action from displaying the table keys correctly
- **Fixed:** Prevent redundant pop-up in duplicated text element
- **Fixed:** Fixed a bug where some properties weren't hiding/showing in the property inspector after another property was toggled.

88.138 VERSION 0.5.0 (2022-09-14):

- **New:** lock for components in asset manager. Locked components are unaffected when elements created from that component are updated. Unlocked components are updated when elements created from that component are updated.
- **New:** New "toggle password" action in the "utility" category.
- **New:** "Disconnect from Component" option in right-click menu for component elements. This option will disconnect the selected element from other component elements making it so it no longer stays in sync with them
- **Improved:** Lowcode Editor - Visual changes for events and inputs
- **Improved:** Retain scroll position when a layout element is refreshed.
- **Improved:** Set minimum element height and width to 1
- **Improved:** improved drawing rect visibility when drawing elements on dark backgrounds
- **Improved:** Added properties to the login element to turn dialogs on or off.
- **Improved:** Added properties to the login element to allow custom dialog text.
- **Improved:** Error prevention for empty fields. Enabling login events for custom low-codes.
- **Improved:** components on different orientations will now stay in sync

- **Fixed:** Use contains instead of begins with for searching fonts
- **Fixed:** fixed a bug that would occur when returning to edit mode after using the get_auth action in play mode
- **Fixed:** fixed a bug that would occur in play mode after an unsuccessful login
- **Fixed:** fixed error reporting for get_auth_status action
- **Fixed:** allow multiple image/media files to be dragged into project at once
- **Fixed:** Fixed an issue where Windows OS would not allow a mouseUp to take place during a 'repeat while mouse is down'.
- **Fixed:** Fixed occasional bug when adding empty event
- **Fixed:** fixed a bug that would occur when toggling the "border" property of a text element
- **Fixed:** Add prompts for missing password/email on login.
- **Fixed:** Fixed automatic focus on argument inputs
- **Fixed:** components from the app project will now load for newly created apps
- **Fixed:** Prevent populating empty, field element inputs with hint text if user is currently typing.
- **Fixed:** Make cut/copied elements that refer to themselves in low code still refer to themselves after pasting
- **Fixed:** Fixed password's font to update after toggling pw checkbox. Remask pw field after mode change.
- **Fixed:** fixed a bug that would cause an error when binding data to a text element
- **Fixed:** Disabling false login on dev credentials on fields requiring user credentials. Updating proper global auth output.
- **Fixed:** fixed a bug that prevented player settings from being updated
- **Fixed:** fixed a bug that would cause the login element to show a successful login message when login failed

88.139 VERSION 0.4.4 (2022-08-31):

- **New:** "Send Verification Email" property for create account element
- **Improved:** Validation for email and password matching on createAccount action script.
- **Improved:** Update internal database and backend
- **Fixed:** Enable createAccount event for field and button elements.
- **Fixed:** Fixed an error that occurred when an image element inside of a layout element is linked to a key.
- **Fixed:** Fixed a bug where changes to the "text size" and "text style" properties of a switch element wouldn't occur.
- **Fixed:** Changed scope of the field and button element to allow visibility on login's events for low-code customizations.
- **Fixed:** Make password matching confirmation work for CreateAccount element on mobile

- **Fixed:** Clear password input fields when switching screens.

88.140 VERSION 0.4.3 (2022-08-26):

- **Improved:** Use native fields for CreateAccount element
- **Improved:** Use native fields for Login element
- **Improved:** Add scrolling support for appligroups
- **Fixed:** Fixed a bug where renaming an app caused issues displaying apps from other projects.
- **Fixed:** Fixed a bug that prevented non-native field elements from using the password font when the "password" property is on.
- **Fixed:** Addressed not updating the a common variable used to check the auth status making it difficult to cover common usage of the feature.
- **Fixed:** Fixed a bug that caused the login and create account elements to lose their low code.

88.141 VERSION 0.4.2 (2022-08-24):

- **New:** Low code action: create_account - alternative Create Account Element to allow a user to create an account with their name, email, and password.
- **Improved:** Smoother auto-scrolling when reordering columns in a table element
- **Improved:** Low code action get_auth takes in end-user's username and password– get_auth_status will then return the status of the authentication based on the targeted location (cloud/local).
- **Fixed:** Allow field to be blank when hintText is removed
- **Fixed:** Allow deletion of last screen in app
- **Fixed:** Make scrollbars in field element more consistent when changing zoom
- **Fixed:** Fixed bugs that could occur after setting all the key values of a record to empty in a table element
- **Fixed:** Fixed a bug that could occur when reordering rows and columns in a table element
- **Fixed:** Table element destination indicator accurately displays where a row is being moved to
- **Fixed:** Stop scrolling when there is no horizontal/vertical scrollbars
- **Fixed:** Make script icon appear correctly when adding code to new screens

88.142 VERSION 0.4.1 (2022-08-17):

- **New:** Low Code action - list_the_files
- **Improved:** Update element and screen names in an action's arguments if they have been renamed.
- **Improved:** Mark errors in actions if their arguments contain an element or screen that has been deleted.

- **Fixed:** Fix issue with cleaning variables when deleting screens
- **Fixed:** Low Code Editor - second mouseClick handler would be added if an empty one had previously been saved.
- **Fixed:** Actions related to lineGraph do not work
- **Fixed:** Saving Appli Low Code actions would cause issues
- **Fixed:** Error when dragging in project browser
- **Fixed:** Make Login element functional
- **Fixed:** Make graph element show up correctly as a preview in edit mode

88.143 VERSION 0.4.0 (2022-08-11):

- **New:** New Charting Element - replaces LineGraph element. This element allows line graphs, bar charts, and pie charts, utilizing the Chart.js library.
- **New:** File I/O Actions for lowcode editor - Create_Folder, SetFolder, Save_Text_File, Save_Binary_File, Load_Text_File, Load_Binary_File
- **Improved:** Option to autosize or specify width and height of search element's icon.
- **Improved:** Restrict "openScreen" and "closeScreen" low code events to screens. These events are still backwards-compatible, but it's recommended to move all actions from these events into screens.
- **Improved:** Restrict "loginSuccess" and "loginFailed" low code events to the "login" element. These events are still backwards-compatible, but it's recommended to move all actions from these events into the "login" element.
- **Improved:** Restrict "accountCreated" low code event to the "create account" element. This event is still backwards-compatible, but it's recommended to move all actions from this event into the "create account" element.
- **Improved:** Updated platform setup to match on-boarding setup.
- **Improved:** You can now edit the label of a button element by double-clicking on it
- **Improved:** Lowcode editor now remembers mouseClick as default behavior once it asks you whether to use it
- **Improved:** Make saving action scripts undo/redoable.
- **Improved:** Use current device orientation in "print to PDF" action.
- **Fixed:** Custom sizes can now be set for phone and tablet when updating the current device.
- **Fixed:** Change mouse cursor appropriately.
- **Fixed:** Prevent naming elements with multiline text.
- **Fixed:** Actions assigned to an appligroup will now trigger if the target child element of the appligroup does not have an action assigned to it
- **Fixed:** Fixed a bug where "print to PDF" action wasn't working on Android.
- **Fixed:** Adjust paper size when using "print to PDF" action to prevent cutting off the rect of the specified element.

88.144 VERSION 0.3.5 (2022-08-03):

- **New:** Elements created through the components section of the asset manager will now stay in sync when they are updated
- **Improved:** Show screen in Project Browser. Allow screens to apply Low Code.
- **Improved:** Default to the element type when asking for element name
- **Fixed:** Prevent actions in mouse events from running in parent elements when interacting with their child element.
- **Fixed:** Hide elements from display on top of the "show loading" action dialog.
- **Fixed:** Elements with the same properties in different sections were incorrectly being shown/hid
- **Fixed:** Allow "move out" of parent from inside project browser by dragging to the left
- **Fixed:** Prevent unrelated hotkeys from triggering when viewing the low code editor.

88.145 VERSION 0.3.4 (2022-07-27):

- **New:** Components section for the asset manager. Right-click on an element to save it as a component. Components are global to all apps in an app project. Components can be added to a screen of an app by clicking on the component in the asset manager
- **New:** Low-code - New "get the platform" action.
- **New:** Low-code - New Date and TIme actions allows users to get the current time and date.
- **New:** Low-code - New Date and TIme conversion actions
- **Improved:** Added ripple effect option to buttons
- **Improved:** UI for Zooming, especially on Windows platform
- **Improved:** Added hotkey commands and additional zoom options to zoom level dropdown.
- **Improved:** Zoom to fit content and zoom to selection now zooms to the nearest whole percentage.
- **Improved:** Upon creating a text element, the user is prompted to name their field after entering the initial text.
- **Improved:** Upon creating a button element, update the text of the button to match the name.
- **Improved:** Make Form setup save as a single command, add cancel button to Form and Camera Setup
- **Fixed:** Avoid an execution error when working with lineGraph Element
- **Fixed:** Fixed issue with incorrect parameters in low-code

88.146 VERSION 0.3.3 (2022-07-20):

- **New:** Low code - "Switch" control structure

- **New:** Low code - “Try” and “Catch” control structure
- **New:** Low code - “If/ElseIf Property” control structure
- **New:** Low code - “If/ElseIf Variable” control structure
- **New:** Low code - “Else/End if” control structure
- **New:** Low code - “Repeat” control structure
- **New:** Low code - “Wait” action
- **New:** Low code - “Set Item Separator” action
- **Improved:** Verification of low code editor
- **Improved:** Display valid input(s) as ghosted text in field elements that have a validation type and no hint text.
- **Improved:** Actions in elements will auto-update to be compatible with changes from Appli ver. 0.3.1 upon loading the app.
- **Improved:** Search element now allows customization of icon
- **Improved:** No-code setup for dropdown element. Can now assign a table and key to display options
- **Improved:** Indentation and Colorization in the low code editor
- **Improved:** Increased size of event flow headers
- **Fixed:** Colorization issue in low code editor
- **Fixed:** General bug fixes for displaying hint text in field elements.
- **Fixed:** Icons with the same name in different families wouldn’t be chosen
- **Fixed:** Can now scroll the asset manager with the mouse wheel when the cursor is over the left pane background
- **Fixed:** Can now scroll query builder value list with mouse wheel
- **Fixed:** Fixed a bug that would cause Appli to throw an error when changing textStyle in the asset manager
- **Fixed:** Fixed text style hotkeys and typing into text elements.

88.147 VERSION 0.3.2 (2022-07-13):

- **New:** Hotkeys to apply text styles while editing text elements.
- **New:** “Validation type” property to field elements.
- **New:** getTheDifference low-code action
- **New:** getTheDivision low-code action
- **New:** getThePower low-code action
- **New:** getTheRoot low-code action
- **New:** getTableKeys low-code action
- **Improved:** Add optional error handling parameters for low-code math Actions
- **Improved:** Tooltips for settings, sync to cloud, account, no-code, and low-code buttons
- **Improved:** Low Code Element input method now shows parent/child relation for elements
- **Improved:** Allow text elements to use templates when setting their datacontext
- **Improved:** Added shortcut keys for zoom levels, ability to zoom to selected elements, and ability to zoom to fit all.

- **Improved:** Clicking on a line that contains an action in the low-code editor will highlight the line
- **Improved:** Allow multiple keys to be used for sorting layouts
- **Improved:** Actions will be added when clicking the "+" button, rather than when the action is selected.
- **Fixed:** Fixed a bug with the element input method which prevented filtering screens
- **Fixed:** Fixed a bug where drag-and-drop in the low code editor would lose accuracy after scrolling.
- **Fixed:** Fixed a bug that would occur when pasting into a "mouseClick" event.
- **Fixed:** Size Low Code dropdown list input method to fit content
- **Fixed:** Allow events and actions to execute when optional parameter are not provided
- **Fixed:** Error where some dialogs in the low-code area were not showing

88.148 VERSION 0.3.1 (2022-07-01):

- **New:** Event Handling – you can now handle different events, such as "openScreen" and "mouseEnter" in the low code editor. Full tutorials on will be released in the future
- **New:** Low code validation – your low code events will not fire and will notify you if arguments are not correctly filled.
- **New:** Low code Folding – you can now click to minimize event handlers
- **Improved:** When creating fields, buttons, or grouping elements, display a dialog to name the element. The default name will be used if no name has been entered.
- **Improved:** Added "secondary color" property to graphic elements with gradients.
- **Improved:** You can now sort app projects by date created and date modified.
- **Improved:** Updated header title on Projects screen
- **Improved:** Fields default to their name for hint text unless manually set
- **Improved:** Low code: Clicking on an event header will fold and unfold code
- **Improved:** Low code: You can now drag/drop line(s) of code, both within and between event handlers
- **Improved:** Copy/paste of low code: Copy whole handlers by right clicking the handler line
- **Improved:** Copy/paste of low code: Paste into the top of another handler by right clicking handler name > paste
- **Improved:** Copy/paste of low code: Paste below another line by right click line > paste
- **Improved:** Text elements can now be used in data binding for form elements.
- **Improved:** Form element data binding is now more visual. All form

elements will need to have keys and elements linked in the no-code window in order for form element actions to function.

- **Improved:** Low code element input method will default to the current orientation
- **Improved:** Allow variable viewer to resize smaller
- **Improved:** Allow apps to be sorted by device type
- **Improved:** UI and UX improvements for viewing and creating app projects and apps
- **Improved:** Update Notch handling - Top Level elements that touch the edge of the screen are now given the option to stretch into the notch (X or Y)
- **Fixed:** Using the platform setup will now modify the existing platform instead of creating a new platform
- **Fixed:** fixed a bug which caused the sort icon to move behind the sort label when sort is set to "ascending" for app projects
- **Fixed:** Add cancel button to Layout's datacontext setup
- **Fixed:** Allow dragdrop to work for CSV/Images
- **Fixed:** Clear the project browser search and close its dropdown when switching screens or apps.
- **Fixed:** An app with multiple platforms will be split into multiple apps, one for each platform, when it is loaded. This will only happen if you approve it; however, apps with multiple platforms are no longer supported
- **Fixed:** Added "paste" option, if applicable, to right-click contextual menu when elements are selected.
- **Fixed:** Stop deletion of non-existing rows in layout
- **Fixed:** Fix issue in SetTheProperty action where clicking on the property/value would always error out saying that the Element had not been selected yet
- **Fixed:** Add Cut/Copy Screen when right-clicking
- **Fixed:** App sort label will now display correctly on startup
- **Fixed:** Fixed an error that will occur when opening a project after cloning a duplicate sample app.

88.149 VERSION 0.3.0 (2022-06-15):

- **New:** Appli now has app projects. App projects can contain multiple apps. Apps within an app project can share data
- **Improved:** Smoother local save process when the playground has scrollbars.
- **Improved:** Unified the look and feel for a few more contextual menus.
- **Fixed:** Fixed a bug where the order of elements in the project browser may be displayed incorrectly.
- **Fixed:** Position switch element's label correctly when an app is loaded.
- **Fixed:** Hide native elements beneath a dropdown element's options when the dropdown is expanded.
- **Fixed:** Hide fixed guideline indicator when resizing an element on the

edge of the playground.

- **Fixed:** Adjust a layout element's scrollbars when zoom levels are changed.

88.150 VERSION 0.2.42 (2022-06-08):

- **Improved:** Add property/support for ios notch
- **Improved:** Added spacers between logical groupings in contextual menus
- **Improved:** Elements can be renamed from the right-click, contextual menu
- **Improved:** Grouping elements will now put the new group into the lowest common parent of the elements
- **Improved:** Panning will only occur while spacebar is held down – no longer a toggled mode
- **Fixed:** Display an error message for "get the sum" action if the parameters passed are not numbers
- **Fixed:** Fixed a bug where fixed guidelines can be dragged outside of the playground area if there are off-screen elements
- **Fixed:** Don't select a child element whose parent element is not within the selection area
- **Fixed:** Error when changing style/color of text chunk
- **Fixed:** Handle dynamic text size in asset manager
- **Fixed:** Keep text in asset manager from losing title if property is changed to the same value
- **Fixed:** Fixed a case where elements that were highlighted by the asset manager don't stay highlighted
- **Fixed:** Make Cmd/ctrl-scroll work with when over an element
- **Fixed:** Undoing grouping was throwing errors in some circumstances

88.151 VERSION 0.2.41 (2022-06-01):

- **New:** Pan tool. You can click the Pan tool (or press spacebar) to pan around
- **Improved:** Added ability to unlink table and record ID in form no-code setup window.
- **Fixed:** Asset manager options can get cut off when collapsing a section and the scrollbar is hidden.
- **Fixed:** Dropdown element's border property doesn't remain off between loads.
- **Fixed:** Make first line of CSV show up as keys when dragging onto table element
- **Fixed:** Allow side-scrolling in table element using left/right scroll and shift scroll
- **Fixed:** Make tabstops work with zooming
- **Fixed:** Make textElement scrolling turn off vertical alignment
- **Fixed:** Offscreen elements can show in play mode.

- **Fixed:** First screen may not be selected when loading an app that has multiple platforms and screens.
- **Fixed:** Fixed an error that can occur when loading an app with multiple platforms.
- **Fixed:** Handle zoom switching while in play mode
- **Fixed:** Add backwards compatibility for lineElement zooming
- **Fixed:** Allow undoing of lineElement deletion to work at different zoom levels
- **Fixed:** Form no-code setup window wasn't displaying linked record ID.
- **Fixed:** Make sure fixed guidelines have correct behavior when loaded

88.152 VERSION 0.2.40 (2022-05-25):

- **New:** Zoom capabilities. You can now zoom in and out of the playground by using the dropdown in the top right of the screen, or by using cmd/ctrl+scroll
- **Improved:** Fixed guidelines will now save
- **Improved:** Hide guidelines and indicators when switching to play mode
- **Fixed:** Update Ask/Answer icon
- **Fixed:** Handle elements on both orientations correctly when undoing delete screen
- **Fixed:** Make Form Element draw correctly when there are multiple
- **Fixed:** Make Tab Element draw correctly on responsive resize
- **Fixed:** Keep playground scrollbar visibility consistent when "go to screen" actions are performed in play mode
- **Fixed:** The position of screens could get out of sync after entering play mode or when a save occurs
- **Fixed:** "Download Sample App" button was not functional.
- **Fixed:** Allow hotkeys to be used immediately after opening project, without having clicked on anything
- **Fixed:** Make fixed guidelines draw in correct location when platform is resized, or when elements are partially offscreen
- **Fixed:** fixed various bugs that could occur with a radio element

88.153 VERSION 0.2.39 (2022-05-18):

- **Improved:** Adjust apps scrollbar after an app is deleted or a new app is created
- **Improved:** Visual improvements when resizing a radio element
- **Improved:** Hiding and showing the label of the radio button positions the options correctly
- **Improved:** UI improvements for apps screen
- **Improved:** Added errorHandling parameter to get_the_product action. This specifies the action to take if an error occurs. By default, it will now display an error message and stop all execution. The errorHandling options include: display error message, stop this execution, stop all execu-

tion, display error message and stop this execution, display error message and stop all execution.

- **Fixed:** Handle bug where making table column too small would cause Appli to hang
- **Fixed:** Allow deletion of an app that has just been loaded and returned to the home screen without selecting another app first
- **Fixed:** Fixed guideline indicators now retain their position and size when elements are outside of the playground.
- **Fixed:** Fixed a bug where the guideline indicator does not become visible.
- **Fixed:** IconList editor was incorrectly showing the template icon
- **Fixed:** Make tabElement default properties match what is actually drawn (coloration)
- **Fixed:** Fix an issue where deleting elements would sometimes cause selection error
- **Fixed:** Update switch element sizing to be more consistent at smaller sizes
- **Fixed:** Radio Element label no longer cuts off
- **Fixed:** Radio Element orientation no longer changes when its width or height are changed
- **Fixed:** Fixed bug where typing doesn't occur in email and password fields of login screen when Appli is initially opened on Windows.
- **Fixed:** Fixed a bug that could occur when updating the thumbnail of a screen
- **Fixed:** The menu for the user icon now shows the same menu for both left and right click
- **Fixed:** Selection UI now retain their position when a local save occurs.
- **Fixed:** Fix issue where line element selection would draw off by 1 pixel
- **Fixed:** Allow selection of nested table elements with double-click.
- **Fixed:** Fixed selection UI when a table element is resized.
- **Fixed:** Fixed bug where some elements don't return to their original appearance when resizing is initiated but doesn't occur.
- **Fixed:** Fix issue where roundrectdots were still showing on images when not usable
- **Fixed:** On desktop, deleting all text in a field element would deselect the field
- **Fixed:** Make CreateAccount element save its width/height when set in Property Inspector

88.154 VERSION 0.2.38 (2022-05-11):

- **Improved:** Accounts on LiveCloud plans can now switch to Appli tiers.
- **Improved:** Variable viewer can now be resized
- **Improved:** Updated thumbnails to show 'Highlight elements' in the asset manager (WIP).
- **Improved:** Removed clipboard usage when updating the thumbNails of screens.

- **Improved:** All thumbnails are kept current in real-time.
- **Fixed:** Autosave will no longer reset scrolling on screen.
- **Fixed:** Handle switching to a delayed, promo tier correctly.
- **Fixed:** Fixed an error that can occur when loading an app with more than one platform.
- **Fixed:** Fixed a bug where elements can be dragged into nested elements whose parent elements aren't visible.
- **Fixed:** added filename parameter to print_To_PDF action. Will no longer default to "invoice"

88.155 VERSION 0.2.37 (2022-05-06):

- **New:** Variable Viewer. Can be accessed in the low code editor by clicking the variable viewer icon in the bottom left of the code section. Can be accessed in the playground with shift + V
- **Improved:** clearTemporaryBlobs.action - Action name changed for consistency with other actions
- **Improved:** LowCode editor minimum dictionary height changed to 310
- **Improved:** LowCode editor panels remember their sizes for the entire session
- **Improved:** Update thumbnails view for all screens in real-time.
- **Improved:** Allow hotkeys when the low code editor is open
- **Improved:** Added error checking and dialogs to "send email" action.
- **Fixed:** Apps renamed in Appli will load correctly in the player
- **Fixed:** Fixed a bug where clicking on an action's name in the low-code editor doesn't display its dictionary entry on Windows.
- **Fixed:** You can now create fixed guidelines when showing offscreen elements
- **Fixed:** Don't show the fixed guideline indicator in play mode
- **Fixed:** Reset Category pane minimum sizes
- **Fixed:** Fixed bug where "EmailBodyHTML" parameter is always used in "send email" action.

88.156 VERSION 0.2.36 (2022-05-03):

- **New:** Low-Code Events pane controls added
- **New:** New Actions - checkInternetStatus, getLoggedInUser
- **Improved:** Key the appCodes to a particular region.
- **Improved:** UI tweak to events Pane
- **Fixed:** Don't show scrollbar for layout element in edit mode when it is not needed
- **Fixed:** Fixed a bug that could cause the text size property to be disabled when it should be enabled
- **Fixed:** Fixed a bug that could cause app thumbnails to draw smaller and have a black background

88.157 VERSION 0.2.35 (2022-04-26):

- **New:** More low code actions: clearTemporaryBlobs, getAuth, getTheResult, Logout
- **New:** Actions for setting linegraph data - SetLinegraphAxis, SetLinegraphData, SetLinegraphDataFromTable
- **Improved:** fixed guidelines no longer appear in app thumbnails
- **Improved:** hold down the "g" key to remove all guidelines from the current screen or from the entire project
- **Improved:** Allow payment method removal if not on a paid plan.
- **Improved:** Nudging elements with arrow keys: if shift is down, large nudge (default 10) will be used. Small and Large nudge are adjustable in settings
- **Improved:** read_keys action - If receiving only one record of data, structure the output with only the value.
- **Fixed:** Fixed a UI bug that occurs after adding a payment method when there are no existing payment methods.
- **Fixed:** If file can't be found when dragdropping onto playground, ask user for file
- **Fixed:** Nudging will no longer move the parent's selection box when in "Edit Element" mode.
- **Fixed:** Nudging is now undo/redo-able
- **Fixed:** Sometimes the Password field in login will not allow one to type.
- **Fixed:** Fixed a bug that could occur when using the round rect dots to adjust the corner radius of a graphic element
- **Fixed:** Handle dragging of multiple elements better when considering moving into/out of containers
- **Fixed:** getAuthStatus.action - fixed Element Output to list text fields
- **Fixed:** Make "Edit Elements" mode respect all ancestors/children
- **Fixed:** Fix issue where grey rectangle would show on top of elements that were moved out of their parent
- **Fixed:** Form elements that are not opaque will now be highlighted correctly when dragging elements into them
- **Fixed:** Handle dragging into/out of appliGroups
- **Fixed:** Fix ungrouping inside of forms
- **Fixed:** native elements and widgets no longer draw over dialog messages
- **Fixed:** Forms will now reset even if they are not on the first screen
- **Fixed:** Forms opaqueness and background color are now saved correctly

88.158 VERSION 0.2.34 (2022-04-20):

- **New:** New action "SendEmail"
- **New:** New action "GetTableID"
- **New:** New action "SetAVariable"
- **New:** New action "GetAuthStatus"
- **New:** Linegraph element. This will allow you to draw a line chart.

- **New:** Prompt user before resetting password
- **Improved:** Updated project browser text font and text size to match the property inspector
- **Improved:** Project Screen Visuals
- **Improved:** Updated the tooltips with capitalization and other visual improvements. Tooltips are better in sync with previous visuals.
- **Improved:** Added a higher quality appli logo for projects screen
- **Improved:** a group in edit mode will remain in edit mode when an element is deleted from it
- **Improved:** Low Code section - disabled the showFocusBorder on the search fields on the category and actions panel
- **Improved:** Low Code section - Decreased the size of the search field to fit within border graphic.
- **Improved:** Toggle play/pointer mode with p and shift-P engages pointer mode.
- **Improved:** Adjusted new text elements to have their textAlign set to center.
- **Improved:** Make Asset name changes persistent (Save when you save your project)
- **Fixed:** Handle hiding of screen list at bottom of screen so it won't cut off playground
- **Fixed:** Reset Asset Manager correctly after exiting project and opening different project
- **Fixed:** Save default white background for platforms on creation
- **Fixed:** SaveCloud toast will no longer disappear immediately have save is requested.
- **Fixed:** Make sure screen background will always be unhighlighted after dragging files onto the playground
- **Fixed:** Fixed an issue in when saving a project that prevented the thumbnails from updating.
- **Fixed:** dontWrap was not correctly set when creating a new text element. Expected setting is true.

88.159 VERSION 0.2.33 (2022-04-13):

- **Improved:** Offline improvements for loading apps
- **Improved:** Updated DB settings to improve performance
- **Improved:** Check if user is online for actions that require internet
- **Improved:** Add search field to asset manager
- **Improved:** "Edit Elements" option will now show border more consistently and handle drag-select
- **Improved:** Asset Pane now tracks Text - font, size, style
- **Improved:** Store/Account updated UI and plans
- **Fixed:** Delete local data when forgetting password so the system doesn't throw errors
- **Fixed:** Fix execution error which sometimes occurs when loading app

with 2 platforms.
- **Fixed:** Visible selection border will always be removed in play mode
- **Fixed:** Handle scrolling in Asset Pane
- **Fixed:** Fixed a bug that could occur if an app doesn't have a thumbnail in the apps screens
- **Fixed:** Fixed an error that would occur when deleting the only screen in an app
- **Fixed:** Appli groups now honor styling properties - border properties and background properties

88.160 VERSION 0.2.32 (2022-04-06):

- **New:** HotKey for the Project browser (CMD + Y)
- **New:** HotKey for the Database manager (CMD + D)
- **New:** HotKey for the Asset manager (Shift + CMD + Y)
- **New:** Tooltips for changing orientation (shift - O)
- **New:** Tooltip for hiding/showing thumbnail view (V)
- **New:** Method to toggle the thumbnail UI on/off
- **New:** elements only snap to fixed guidelines when the "~" key is down
- **Improved:** Added additional details to error messages when loading an app.
- **Improved:** Update right-click options for colors on asset pane
- **Improved:** Lock icon next to text size. Text size will no longer be invisible when dynamicTextSize is on.
- **Improved:** Add/modify hotkeys
- **Improved:** Add option to highlight elements from asset pane
- **Improved:** Move deselect elements to (CMD + Shift + A)
- **Fixed:** Fix hash issue with CDB on Windows
- **Fixed:** Handle deletion of elements correctly in asset pane
- **Fixed:** Be able to drag the out of the bounds element back into the appli group element by selecting the element from the project browser
- **Fixed:** Fixed a bug where certain actions displayed blank dictionary entries when their names were clicked on in the workflow of the low-code editor.
- **Fixed:** fixed a bug where adding an item to a list in the property inspector would throw an error
- **Fixed:** Fixed issue where pressing "Enter" would throw an error in the asset manager

88.161 VERSION 0.2.31 (2022-03-30):

- **New:** Fixed Guidelines – You can click and drag from the edge of each screen on the playground to create your own fixed guidelines. Hide and show these guidelines with the "g" key. Drag a fixed guideline outside of the screen area to delete it.
- **New:** Edit Mode for container elements. All container elements (group,

layout, tab, form) can now be set into “Edit mode”, where you can click on child elements. This will also show a thicker line around the parent element. You can access this mode via context menu with a right click, or by double-clicking a child element inside a container element.

- **New:** Asset Pane with colors. This will track all colors in the document and will allow you to edit them. You can also click to set the main color of a selected element
- **Improved:** Update screen thumbnails in real-time. Visible in thumbnail view and project screen.
- **Improved:** Add button in Property Inspector to reset image aspect ratio
- **Improved:** Clicking on the action name in a line of code will now display the dictionary entry for the action
- **Improved:** You can now right-click on colors in the Property Inspector in order to use existing document colors.
- **Fixed:** Drag selection of child elements when in edit mode for tab and form now works corrrectly.
- **Fixed:** Option+dragging child elements inside tab elements will now work correctly.
- **Fixed:** Fixed a bug where highlighting rows didn’t populate the correct record IDs into the selected variable.
- **Fixed:** Make corner radius dots appear only when usable.
- **Fixed:** Ensure that internal variables are set correctly before deleting projects; this previously caused Appli to hang.

88.162 VERSION 0.2.30 (2022-03-23):

- **New:** expand/collapse sections in property inspector
- **Improved:** Make Images default to the correct aspect ratio and lock aspect ratio
- **Improved:** Load app improvements
- **Improved:** performance improvements when going to the “My Apps” screen
- **Improved:** Rename “mobile” platform to “phone”
- **Improved:** Add tooltips to Property Inspector with actual property names (as used in actions)
- **Improved:** Clicking a switch element’s label will also toggle it.
- **Fixed:** Right-click > Reset Image Aspect Ratio now correctly resets to the image’s natural aspect ratio
- **Fixed:** Fix issue where saving locally would fail when clicking save to cloud button (Clarence M.)
- **Fixed:** Fixed error that can occur when moving multiple elements that contain a parent element.
- **Fixed:** Actions now work when toggling a switch element.

88.163 VERSION 0.2.29 (2022-03-16):

- **New:** Orientation support for the player. This can be found in the "Player" tab in Settings.
- **New:** "o" as a hotkey to toggle on/off show offscreen elements
- **Improved:** Improved promo code selection and tiering in store
- **Improved:** Move the play button to the left of the screen
- **Improved:** Going to play mode will hide everything in the header expect the play button
- **Improved:** Add more human-readable display names to property inspector
- **Fixed:** Truncation of element names when editing the name using the Project Browser will no longer cause the ellipses to become part of the element name
- **Fixed:** Stop option-dragging when no elements are selected
- **Fixed:** Stop Text Elements from being selectable in play mode
- **Fixed:** Make downloading included apps update tableIDs for form elements
- **Fixed:** Update Form element to show native controls on player
- **Fixed:** Fixed errors that can occur when loading an app that has multiple platforms.
- **Fixed:** Fix issue where elements removed/deleted from Forms would cause issues

88.164 VERSION 0.2.28 (2022-03-09):

- **New:** Ask which platform to load when loading an app if there are more than 1 platform set up
- **New:** Keyboard shortcut to toggle between play and edit mode with shift + P
- **Improved:** Property Inspector visual rework – make font sizes and spacing smaller
- **Improved:** Hide footer in play mode
- **Improved:** The play button now toggles between play and edit mode
- **Improved:** Reset table name and record ID selections when unchecking "link to key" in a text element's no-code screen.
- **Fixed:** Handle Vertical Alignment and Ellipses clipping of text elements better
- **Fixed:** Creating a new app will clear previous apps in the playground
- **Fixed:** Handle spacing on Project Browser when expanding/collapsing groups
- **Fixed:** Toggling the vis of the left pane would cause the selection rect to draw incorrectly in some cases

88.165 VERSION 0.2.27 (2022-03-02):

- **Improved:** visual improvements after selecting an app to load and transitioning to the playground screen
- **Fixed:** Consistently resize low-code editor to the correct size.
- **Fixed:** Prevent a local save from occurring after a project has just been loaded.
- **Fixed:** Use the correct color for the first row in layout elements after a project is loaded.
- **Fixed:** setting the text of a field element to empty with an action will now manage the hint text on mobile devices
- **Fixed:** applying an action to an element will no longer interfere with its built in behavior
- **Fixed:** Toggling the switch element will now set its “value” property correctly.
- **Fixed:** Make Undo/Redo with images/media work correctly.
- **Fixed:** fixed an error that would occur when signing in to Appli a second time
- **Fixed:** can type in search element again

88.166 VERSION 0.2.26 (2022-02-23):

- **New:** Ability to export invoices in the account page.
- **Improved:** Renamed “Video” element to “Media” element, as it can be used for both video and audio
- **Improved:** Increased width for lowcode editor
- **Improved:** Prevent the lowcode editor inputMethods from drawing outside the lowcode editor
- **Improved:** Cutting/Pasting lines in the lowcode editor will now manage the scrollbar
- **Improved:** Delete line icons in the lowcode editor now move with scroll
- **Improved:** Visual improvements after signing in and transitioning to the apps screen
- **Improved:** Make form set its children’s value when recordID is set in play mode
- **Fixed:** Action scripts now work for search elements
- **Fixed:** Included Projects will now download images/media when cloned
- **Fixed:** Make Undo/Redo of image/video deletion/resetting of images work
- **Fixed:** Delete line icons no longer draw outside lowcode editor
- **Fixed:** InputMethods in the low code editor now draw above other ui elements when they are displayed
- **Fixed:** Getting the text of a field element with an action will now return the correct value on mobile

88.167 VERSION 0.2.25 (2022-02-16):

- **New:** New Invoices tab on the accounts page. This will allow you to see you billed invoices.
- **Improved:** Internal handling of indexes for Layouts
- **Improved:** Internal handling of types for appliGroups
- **Fixed:** Fixed an issue where action scripts could not be assigned to image elements.
- **Fixed:** Allow older apps with no app-code to be deleted
- **Fixed:** Fixed an error that may occur when outputting a search element to a layout element.
- **Fixed:** The layout element no longer displays extra rows after receiving output from a search element.
- **Fixed:** All platform controls will now be correctly returned instead of just those on the selected screen.

88.168 VERSION 0.2.24 (2022-02-10):

- **New:** Account screen located in the user icon dropdown of the projects screen.
- **New:** "Payment methods" screen in the account menu.
- **New:** Form Element. This element can contain other elements that take user input and submitted to the DB
- **New:** new "print_to_PDF" action
- **New:** new "email_the_PDF" action
- **New:** new "hide_Loading" action
- **New:** new "show_Loading" action
- **New:** new RefreshTheForm action: will reset form to empty or to the linked table/record's current value
- **New:** new Submit Form to DB action. This will add or update a record from a form
- **Improved:** preserve the low code editor vScroll when moving lines of code
- **Improved:** Added Subtypes to appliGroups – these will be improved with addtional functionality in coming releases.
- **Improved:** Add empty button to inputMethodField
- **Improved:** App Code Management. You will now need to check your app code under "settings" in order to add it to your player.
- **Improved:** Newly created text elements now have "text" as their default text
- **Improved:** Appli no longer automatically deletes empty text elements
- **Fixed:** border now draws for native multiline field elements on mobile devices
- **Fixed:** Applying the multi-selection input method after filtering will now populate the argument with all selected items.
- **Fixed:** Appli no longer saves local changes when switching between edit

mode and play mode

- **Fixed:** Appli no longer saves local changes when switching between edit mode and play mode
- **Fixed:** Layout elements with multiple rows display a row color in play mode despite alternate row colors being off.
- **Fixed:** Keep position of selection UI in sync with the selected element(s) or screen.
- **Fixed:** Don't populate empty rows in layout elements with templated text.
- **Fixed:** Don't allow scrolling with mouse wheel in layout elements whose "show scrollbar" property is off.
- **Fixed:** Allow Layout template binding to be backwards compatible with text elements that previously didn't have their "text" property explicitly set
- **Fixed:** Allow auto-ellipses for text elements work with data binding
- **Fixed:** Update cases where cdbApps table was not available locally
- **Fixed:** Fix issue with downloading sample project when a project of the same name already exists in your account

88.169 VERSION 0.2.23 (2022-01-26):

- **New:** New "Format the Text" action
- **New:** New "Combine the Values" action
- **New:** App code feature - when creating a new app, a code is generated. It can be copied in the settings window.
- **New:** Load app code when it is not known. Show it from cache if available.
- **New:** The ability to store the record IDs of selected rows in a table element into a variable. This variable can be assigned in the table element's no-code setup window.
- **New:** You can now download sample apps to edit/view in Appli
- **Improved:** Update actions with record ID arguments to accept variables as input.
- **Improved:** Added "cdbRecordID" to the keys argument of "List Records" action.
- **Fixed:** Fixed dictionary entry for "Delete Keys" action.

88.170 VERSION 0.2.22 (2022-01-19):

- **New:** New "Get the Sum" action
- **New:** New "Set Browser URL" action.
- **New:** New "Get the Date" action
- **New:** New "Get the Tax" action
- **New:** New "Sum the Column" action
- **New:** Ability to copy, cut, and paste actions in the low-code editor. Hold CMD/CTRL/SHIFT and left-click to select actions. Right-click to see the options menu.

- **Improved:** Highlight rows in table elements with a single click. Click a cell in a highlighted row to edit it, if applicable.
- **Fixed:** Actions that take an element for a parameter will now correctly perform the action on elements inside of a layout.
- **Fixed:** refresh_the_layout fix for layout elements that are not nested
- **Fixed:** get_the_property action will now return the correct "text" value for field elements

88.171 VERSION 0.2.21 (2022-01-12):

- **New:** New Get_The_Product action. This action takes 2 numbers and stores the product in a variable.
- **New:** new "Refresh the Layout" action
- **Improved:** Dropdown element - options no longer show empty space
- **Improved:** Dropdown element has new property to control number of options to display
- **Improved:** Setting a field to empty using an action will show its hint text
- **Improved:** Variables will now show up in the variable picker as soon as they are first created
- **Improved:** LookupValue action can now choose recordIDs from variables
- **Improved:** allow DeleteTheRecord action to take either a variable or a value
- **Improved:** allow UpdateTheRecord action to take either a variable or a value for the recordID
- **Improved:** Added property "hintTextColor" to fields
- **Fixed:** Dropdown elements will display above it if there isn't room below the label
- **Fixed:** Set text color for fields correctly when changing text property programatically
- **Fixed:** Fix issue where projects wouldn't load if you had deleted a screen using a different computer
- **Fixed:** Changing the size of a dropdown element if its options has a scrollbar now resizes correctly.
- **Fixed:** Nested layout elements will now draw correctly in play mode.
- **Fixed:** Fixed issue where scrollbars were appearing on fields unnecessarily
- **Fixed:** Moving an element with the arrow keys will now update the position of any elements within it correctly

88.172 VERSION 0.2.20 (2021-12-22):

- **New:** New action deleteKeys allows a developer to delete the content from specific keys in a record.
- **New:** New "Get the Property" action.
- **New:** New "Count Displayed Records" action.

- **New:** New Count Records action will count the number of records in a given table.
- **New:** can now put the text of a field element into a variable in the low code editor.
- **New:** can now set the text of a text element from a variable in the low code editor.
- **Improved:** Elements and variables can now be used as values in the array input method.
- **Improved:** The Actions column is now sorted ascending in the low-code editor.
- **Improved:** Allow template strings for text elements inside of layouts.
- **Improved:** readTheRecord can optionally output to a text element. Data is converted to comma and lf delimited data.
- **Improved:** Added better support for placing the result into a text element.
- **Improved:** Using inputMethodMultiSelection instead of dropDownList.
- **Improved:** allow search element to use Layout as output.
- **Improved:** When using the drag to create method for a text element, dontWrap, and verticalAlign are now automatically set to true. These settings should better align with how this element is most commonly used.
- **Fixed:** Fixed empty orientation text in element input method when an action's element argument is empty.
- **Fixed:** LC integration action did not support all forms of commands correctly.

88.173 VERSION 0.2.19 (2021-12-15):

- **New:** New "Calc Update the Record" action.
- **New:** New "Sync Records" action.
- **New:** New "Sort Records" action.
- **New:** New "Read Keys" action.
- **New:** New "Query Records" action.
- **New:** New "Set Variable From Property" action.
- **New:** New "Set Property from Variable" action.
- **New:** New "Set Variable from Context" action.
- **Improved:** Add Local/Cloud options to search element.
- **Fixed:** Gracefully handle deleted elements when selecting an element in lowCode.
- **Fixed:** Make delete icon for lowcode align correctly on Windows.
- **Fixed:** Make dragging of code lines align correctly on Windows.
- **Fixed:** Handle case where dragging lines of code was causing errors.
- **Fixed:** setVariableFromProperty fixed api command name
- **Fixed:** When hiding a group with a native control, the native control will now hide correctly.
- **Fixed:** The dynamicTextSize checkbox in the property inspector is no longer disabled for newly created text elements.

88.174 VERSION 0.2.18 (2021-12-08):

- **New:** inputMethodElement for low code editor.
- **New:** inputMethodMultiSelection for low code editor.
- **New:** inputMethodArray for low code editor.
- **New:** New “Refresh the Table” action.
- **New:** New “Update the Record” action.
- **New:** New “Create the Record” action.
- **Improved:** update setTheProperty action to use inputMethodElement. This will allow setting of properties for elements on different screen/orientations.
- **Improved:** Set dynamicTextSize to true by default to better support responsive design expectations.
- **Improved:** Add sort option to order layouts.
- **Improved:** Allow local/cloud setting for layout data linking
- **Improved:** Allow changing of layout data table, data recordID and data target using property setting
- **Improved:** Dictionary entry for listRecords.action updated.
- **Improved:** listRecords.action functionality improved
- **Fixed:** Locked elements could be moved with the arrow keys.
- **Fixed:** Error no longer occurs when toggling an element’s lock icon and moving it with the arrow keys.
- **Fixed:** field elements now observe their text alignment when they are loaded
- **Fixed:** appliGroups inside of a locked appliGroups will no longer be selected with drag-select

88.175 VERSION 0.2.17 (2021-12-01):

- **New:** New “Delete the Record” and “Flush the Cache” actions.
- **Improved:** Updated CanelaDB libraries to newest
- **Improved:** right-click contextual menu in project browser now allows copy/cut when multiple elements are selected
- **Fixed:** Allow elements with action scripts to be copy/pasted
- **Fixed:** Delete screen dialogue text will display the correct name after the screen has been renamed.
- **Fixed:** Fixed bug where multiple selected elements could be dragged into a non-visible tab, group, or layout element on the playground.
- **Fixed:** Correctly display screens in thumbnail view after screen order was changed.
- **Fixed:** Playground no longer loses its background color when toggling “overridePlatformColor” in the property inspector.
- **Fixed:** Allow action scripts to work inside of groups, layouts, etc.

88.176 VERSION 0.2.16 (2021-11-24):

- **New:** New action, look_up_value, to look up the value of a specified key for a given record.
- **New:** New action, readTheRecord, to read one or more records in a table.
- **New:** Notion of variables to the low-code editor.
- **New:** New APIs to retrieve variable names and manage variable metadata.
- **New:** New "showBorder", "lineSize", and "lineColor" properties added to the field element to allow customization of its border.
- **New:** New action, cdbList, to retrieve one or more keys from a table.
- **New:** New APIs to retrieve a list of a table's keys or record IDs.
- **Improved:** Clear out text before displaying the field input method.
- **Improved:** Allow right-click deletion of screens.
- **Improved:** Improved field element to work with look_up_value action.
- **Improved:** Fixed resizing of inputMethodVariable when there are no existing variables.
- **Improved:** Resizing code for inputMethodVariable UI.
- **Improved:** The dictionary field no longer wraps text to support more data.
- **Improved:** Add dates to the changelog.
- **Improved:** The management of errors returned from selecting an action's argument.
- **Improved:** Switched dictionary's header from "options" to "arguments".
- **Improved:** Manage scrollbars in the workflow area when an action's argument receives an input.
- **Fixed:** Allow fields to be un-deleted correctly.
- **Fixed:** Forced conversion to RGB when setting a native android field.
- **Fixed:** Make sure property change references work correctly.
- **Fixed:** Groups with layouts in them will now load/be copied correctly.
- **Fixed:** Reset projects completely when the refresh button or logout is clicked. Otherwise, if you reload the same project, it will not attempt to download updates from the cloud.
- **Fixed:** Make dragging to move actions set correct variables.
- **Fixed:** Prompt users to save their local changes to the cloud before closing Appli's window.
- **Fixed:** Fixed management of spaces in front of the search term.

88.177 VERSION 0.2.15 (2021-11-17):

- **New:** Error checking for loading actions. If an error happens, the action is removed from the element, and the developer is notified. This issue is generally due to the beta state of actions and is not expected to be a problem once the code is finalized.
- **New:** liveCode_integration action allows developers to utilize code written in LiveCode and run it inside of Appli projects. The action currently

supports functions only. The code written in LiveCode should do computation or database calls. We do not support references to UI. This action is experimental in nature and will probably change.

- **New:** Refresh button in the low-code editor will reload actions and sync the LiveCodeIntegration table to make new code available to Appli.
- **New:** Now copy and cut elements with the right-click contextual menu in the playground and project browser. Paste elements by right-clicking on a selected screen.
- **New:** New API to return a list of sorted, non-Appli-internal tableNames to publicAPI.lib
- **New:** New input method for the low-code editor that works with date values.
- **New:** Verified support for actions for every element in Appli. Note: Not all elements are ready for actions yet.
- **Improved:** Login element improvements. Element has new properties in the Property Inspector to control the size of the fields and buttons.
- **Improved:** Create account element improvements. Element has new properties in the Property Inspector to control the size of the fields and button.
- **Improved:** Display a dropdown for the “value” argument in the “set the property” action if the selected property takes certain values.
- **Fixed:** The Text Element was immediately deleted if created with a single click (dynamic width/height).
- **Fixed:** The Text Element now updates its size immediately when a dynamic option is chosen.
- **Fixed:** The input method filter does not show all the results with partial match.
- **Fixed:** This action stopped working with the recent changes to low-code support. All uses of this action will need to be applied again.
- **Fixed:** Hide icon when creating a button element.
- **Fixed:** The dropdownlist input method didn’t get cached on the first search attempt.
- **Fixed:** Better handling of arguments with commas in their values (eg: RGB values).
- **Fixed:** Dropdown search in low-code editor now resets correctly on launch.
- **Fixed:** Fixed a bug that prevented the categories and actions search bar in the low-code editor from repopulating their respective lists.

88.178 VERSION 0.2.14 (2021-11-10):

- **New:** Add InputMethodColor for action scripts. The method uses logic for handling color names, RGB, and Hex values. Includes an eye dropper, color picker, and color sample.
- **New:** Added inputMethod for whole numbers.
- **Fixed:** Make the camera element use no-code action for output instead

of low-code.

- **Improved:** Make camera output undo-able.
- **Improved:** General improvements to input methods when clicking on an argument.
- **Improved:** Make DataContext changes to layout undoable.
- **Improved:** Added search capability to the dropdown list input method.
- **Improved:** Made creating dictionary entries a lot easier by separating the design from the content.
- **Improved:** Updated slider properties in property inspector to return whole numbers if the input contains decimals and slider's "numberformat" is empty.
- **Improved:** "Set the Property" action can now handle numerical inputs in the value argument.
- **Improved:** Allow dragging to move lines of low-code action flows.
- **Improved:** low-code editor input field UI and functionality
- **Fixed:** Make Camera element use no-code action for output, instead of low-code.
- **Fixed:** Elements with a "set the property" action now load correctly when loading an app.
- **Fixed:** Fixed a function related to goToScreen.action.
- **Fixed:** Allow actions to parse inputs with commas.
- **Fixed:** Can now set the text alignment of a button element.
- **Fixed:** Fixed issues when deleting lines in a flow of actions.
- **Fixed:** Fixed issues where the Project Browser may balk at property changes to an element made after switching to a different screen.
- **Fixed:** Addressed issues where action flows with multiple lines would not draw correctly.
- **Fixed:** Addressed a bug where clicking the 'Cancel' button in the low-code editor would try to update the element's code anyway.
- **Fixed:** Moving the mouse out of a selected screen will no longer throw an error.

88.179 VERSION 0.2.13 (2021-11-04):

- **New:** set_the_property.action can process certain valid requests. The only supported properties are those that have boolean values. Other property types are coming soon.
- **New:** New input methods (UI when changing arguments in the low-code editor: Boolean properties.
- **New:** Action scripts now support all Appli elements.
- **New:** Search in Categories and Actions for low-code editor working now.
- **New:** New graphics property for gradients.
- **New:** UI to adjust gradient coordinates for graphic elements.
- **New:** Show/Hide Gradient Coordinates option in the right-click menu which will toggle whether to keep gradient coordinate UI on-screen when the mouse is not over the selected graphic element.

- **New:** color settings for “dropShadow”, “innerShadow”, “outerGlow”, and “innerGlow” properties
- **Improved:** Project browser dragging improved for dragging into tab element and groups.
- **Improved:** Positioning of low-code editor window.
- **Improved:** Error checking for broken references in “Go to Screen” action.
- **Improved:** Can now resize the dictionary in the low code editor.
- **Improved:** Make cut its own undo-able action.
- **Improved:** Update undo/redo to be separate for each platform. Jump to correct screen/orientation on undo/redo.
- **Fixed:** Fixed many bugs related to adjusting the arguments of an action.
- **Fixed:** Element selection in inputMethodDropdownList now displays all elements. Previously one element was being left out.
- **Fixed:** Clicking on an empty line in the element selection dropdown will now hide the dropdown instead of putting an empty argument. This will prevent us from being in a state where we can no longer select an element argument.
- **Fixed:** Graphic elements with gradients will apply correctly in a repeat grid.
- **Fixed:** Make screen deletion work with undo/redo.
- **Fixed:** Make screen copy/paste work with multiple elements on screen.
- **Fixed:** Gradient coordinates of a graphic element within a layout element are no longer offset after loading an app.
- **Fixed:** Graphic elements no longer have a border after loading an app when the showBorder property is toggled on and off.
- **Fixed:** API goToScreen.action was not correctly referencing its inputMethod.
- **Fixed:** Empty actions code will no longer try to store partial scripts.

88.180 VERSION 0.2.12 (2021-10-27):

- **New:** [WIP] Initial release of setTheProperty action. Only the first two arguments work as expected. The third argument is still under development.
- **New:** [WIP] New input methods for certain arguments in the low-code editor.
- **New:** new “dropShadow”, “innerShadow”, “outerGlow”, and “innerGlow” properties in the property inspector for graphic and button elements. Clicking on the checkbox will engage the property and reveal more settings for the effect.
- **New:** New properties have been added to apply gradients to rectangle and ellipse graphic elements.
- **New:** Error handling for broken element references.
- **New:** There is now a script icon in the project browser entry for elements. This icon is ghosted when the element does not have a script and unghosted when an element has a script. Clicking on the icon, whether it is ghosted

or unghosted, will open the low-code editor for the element.

- **Improved:** Sizing of Input Methods more robust.
- **Improved:** Actions with multiple arguments are more robust.
- **Improved:** Placement of inputMethods for low-code editor is more robust.
- **Improved:** No longer auto select first Category and Action if the element already has code applied.
- **Fixed:** We no longer get duplicate values when clicking on the property argument in setTheProperty action.
- **Fixed:** Error/Notification dialog is hidden behind the low-code editor.
- **Fixed:** Text in the Settings page is no longer cutoff on Windows.
- **Fixed:** Clicking on the General tab in the Settings page no longer throws an error on Windows.
- **Fixed:** mouseUp action scripts no longer interfere with the right-click contextual menu in Appli.

88.181 VERSION 0.2.11 (2021-10-20):

- **New:** New API appli_getPropertyListCurrentElement in publicAPI.lib
- **New:** New API appli_getElementListCurrentScreen in publicAPI.lib
- **New:** Dictionary support in .action files.
- **New:** Support for displaying dictionary entries in the editor.
- **New:** Settings page. The sprocket icon now takes you to the settings page.
- **New:** New table added to project to support LiveCode a future integration feature.
- **New:** Ability to reset account password.
- **Improved:** Updated the dictionary entry for goToScreen.action
- **Improved:** Allow vertical scrolling of the dictionary if needed in low-code editor.
- **Improved:** Refactored robustness of low-code editor argument management.
- **Improved:** Set argument names in actions to one word if possible.
- **Improved:** UI for drop-down list input method. The method places itself near the selected argument. Removed unnecessary UI from the method.
- **Fixed:** When resizing Appli with the editor open, the dictionary resets to the default value.
- **Fixed:** Arguments do not follow specified order for placement in the editor.
- **Fixed:** Multi-argument support does not properly set link values when opening existing code for an element.
- **Fixed:** appli_buildLowCodeDetailsArray is not building its internal array correctly when the argument count is greater than one.
- **Fixed:** Adjust the layering of the dictionary field so it does not draw over the buttons.
- **Fixed:** Key presses now work on Windows after creating a browser el-

ement, changing a browser element's URL, and switching between play mode and edit mode when a browser element is on-screen.

- **Fixed:** Project browser search field no longer gets focus after loading an app.
- **Fixed:** Region dropdown now closes when you select it again.

88.182 VERSION 0.2.10 (2021-10-13):

- **New:** Argument support uses a reusable input method. Currently, we have a drop-down list method with the ability to support other forms of input in the future.
- **New:** Support for managing one of many new input methods for arguments.
- **New:** UI elements for drop-down list input method
- **Improved:** Appli low-code editor received a pass on plane placement and color palette.
- **Improved:** Added more comments for developers in goScreen.action.
- **Improved:** Simplified the code dramatically in goScreen.action.
- **Improved:** Removed deprecated code in publicAPI.lib.
- **Improved:** LowCode Editor and Action model can now support multiple arguments per action API.
- **Improved:** Relaying between child elements in a parent element is more robust in the project browser.
- **Improved:** new "autoSize" property for appliGroup elements. When "autoSize" is true, resizing an appliGroup will scale elements within the group. When "autoSize" is false, resizing an appliGroup will only resize the group, and elements in the group will be unaffected.
- **Improved:** Can now change the data table being displayed in a table element by using the no code UI.
- **Fixed:** Opening the code of an element that already has actions does not properly set up the arguments to be modified.
- **Fixed:** Can not change the value of an argument that has already been assigned within the same session.
- **Fixed:** Clicking on a category does not update the actions pane.
- **Fixed:** Use the file path of the current OS for map and browser element preview images.
- **Fixed:** Remain on the projects screen if an app fails to load when errors in syncing tables happen.
- **Fixed:** New platforms and switching platforms will now show the correct screen name.
- **Fixed:** Prevent a button's icon from drifting to an edge when the "maintainIconRatio" property is toggled.
- **Fixed:** Existing buttons will draw correctly again.
- **Fixed:** Table creation from CSV after typing an existing name will now work.
- **Fixed:** Tables will now work with copy/paste without throwing errors.

88.183 VERSION 0.2.9 (2021-10-06):

- **New:** Introduced an ‘Apply’ button to save the code to the selected element and close the low-code editor.
- **New:** Introduced a ‘Cancel’ button that ignores changes to the selected element and will close the low-code editor.
- **Improved:** Allow copy/paste to work for screens.
- **Improved:** Undo/redo system is more robust.
- **Improved:** Removed the close widget from the low-code editor.
- **Improved:** Simplified the go_screen action and added comments to help others make their actions.
- **Improved:** Improved error reporting when data usage has consumed plan limit.
- **Improved:** Added lock icon to size section of the property inspector to maintain the aspect ratio of an element when it is resized.
- **Improved:** Action script parser will consider the platform and make mouse/touch messages native to the intended device.
- **Improved:** Changed the global hiliteColor of the low-code editor to blue.
- **Fixed:** Fix issue where creating a table after deleting a table would cause Appli to get into an unrecoverable state.
- **Fixed:** Error message for saving to the cloud now remains on screen.
- **Fixed:** Clicking on a category does not update the actions pane.
- **Fixed:** UI was initially drawing all categories with all actions for each category on startup of the editor. It was changed to draw each category and only the actions for the selected category.
- **Fixed:** Empty searches in a search element will now display the entire contents of a table element when a query contains more than one key.

88.184 VERSION 0.2.8 (2021-09-29):

- **New:** Add functionality to interact with answer dialog using keyboard.
- **New:** Add frontscript to handle enter/return key, along with l/r arrows
- **Improved:** Set minimum dimensions for Appli to support laptop values most frequently used.
- **Improved:** Center the lowCode Options dialog.
- **Improved:** Reorganized some of the elements in the lowCode editor.
- **Improved:** Namespace prefix added to lowCode behaviors.
- **Improved:** Automatically vertically align hint text, unless we are multiline.
- **Improved:** Check for non ascii keys before showing the csv import ui.
- **Improved:** All new lines of code should automatically have a delete delete line widget.
- **Improved:** Convert colors in defaultProperties.lib to hex
- **Improved:** Switched the default url from liveCloud.io to appli.io in the browser element.
- **Improved:** Make “stretch” the default responsive option. Most of the

time we will want to use stretch to make sure everything stays on screen.
- **Fixed:** Results fields and body are drawing too small.
- **Fixed:** Delete widgets are missing when opening the editor from and element that has code.
- **Fixed:** Unify hiliteColor for results and body fields.
- **Fixed:** When resizing Appli, code needs to align delete widgets correctly.
- **Fixed:** When a line of code is removed, the delete widgets need to realign.
- **Fixed:** Fixed an error that occurs when dragging a tab element within itself in the project browser.
- **Fixed:** Set textSize correctly when creating element.
- **Fixed:** App icon manager text is cut off on Windows.
- **Fixed:** Updated to use cdb_importCSV. CSV import will now handle commas correctly.
- **Fixed:** Resize allows group "center" to grow out of control.
- **Fixed:** Delete line widgets are too large.

88.185 VERSION 0.2.7 (2021-09-27):

- **New:**
- **Improved:** Make sure mobile keyboard doesn't cover fields when they are focused.
- **Improved:** Trap touch messages on mobile
- **Improved:** New "gap" slider in PI. This slider controls the gap size between the fields and button.
- **Improved:** Don't flag an element with "update" if pValue is the same as the initial value. This will prevent save dialogs after loading an app then going home without making any changes.
- **Improved:** Removed test browser buttons.
- **Fixed:** Fixed crashing bug when attempting to create a browser element.
- **Fixed:** Fixed an isuse with being able to use the map element.

88.186 VERSION 0.2.6 (2021-09-24):

- **New:** Temporary test buttons in header for checking Appli crashing with browser element
- **New:** Support for setting the background color of a project.
- **New:** Ability to reset your icon to the default app icon in the app icon manager
- **Improved:** Use the appli icon as the default app icon.
- **Improved:** Setting default orientation on new projects based on device type. Mobile and tablet are portrait. Deskop defaults to landscape.
- **Fixed:** Added backwards compatibility to platform and screen properties
- **Fixed:** When saving to the cloud, check if screens in local changes exist before trying to update their thumbnail.
- **Fixed:** Unsubscribe controls from topics if the screen they are on is deleted.

- **Fixed:** Remove deleted screens from local changes and save locally.
- **Fixed:** Tabs may be removed from the tab element before an option is selected in the dialog.
- **Fixed:** SelectedTab value in the property inspector isn't updated after removing tabs.
- **Fixed:** Index for the tab element may get out of sync if tabs are removed.

88.187 VERSION 0.2.5 (2021-09-22):

- **New:** "adjustColumnsOnPlayer" property. This property will adjust the column widths of a table on the player when the table is resized with responsive resize. It will increase or decrease each column by the same ratio used on the width of the table.
- **New:** Low-code to the following elements: images, rectangles, ovals, lines, display fields, search, radio buttons, switches, groups, input fields
- **Improved:** Resize column support for mobile player
- **Improved:** Removed special code to handle low-code support when adjusting properties of an element. Now using a method specific to low-code support.
- **Improved:** No longer saving script prematurely when adjusting parameters of action. This will allow us to introduce a cancel button in the low-code editor.
- **Improved:** Adjusted namespace using appli_ as a prefix for public facing APIs.
- **Improved:** Command applyActionScript will work with elements of liveCodeTypes other than groups.
- **Fixed:** Setting the lineColor will now change the borderColor for the table, not just the lines of the header and body field.
- **Fixed:** "selected element" in the low-code editor would not properly identify the name of the element selected.
- **Fixed:** Found an appli API that did not reference the updated naming scheme of appli_APIname
- **Fixed:** Missed a touchpoint for renaming the "placeholder" graphic in tab elements.
- **Fixed:** Renamed "placeholder" graphic in tab elements to fix a bug with resizing the tab element when a browser or table is inside it.
- **Fixed:** Removed duplicate "visible" property from the customProperties of button, browser, and repeat grid.
- **Fixed:** While in Play mode, allow mouse wheel to scroll horizontal scrollbar.

88.188 VERSION 0.2.4 (2021-09-20):

- **Improved:** Updated Appli executable to use LiveCode 9.6.3 stable.
- **Improved:** Updated icon to match expected standards for macOS and Windows.

- **Improved:** Updated internal pathing routines to make Appli's shared libraries with the player more compatible.
- **Improved:** Changed out disk icon for a cloud icon to be more representative of what the feature does.
- **Fixed:** Removed some debugging code from auto-update code.
- **Fixed:** Action scripts would not work after reloading a project.
- **Fixed:** Added a missing function that does ascii range checks.
- **Fixed:** Fixed a bug in pathing related to patch notes.

88.189 VERSION 0.2.3 (2021-09-17):

- **Fixed:** Search Element fixed to handle non-matching searches and
- **Fixed:** Handle app Icon pathing correctly in player
- **Fixed:** Fix Data Manager key adding/deleting

88.190 VERSION 0.2.2 (2021-09-15):

- **New:** App Icon Manager for setting the icon for apps.
- **Improved:** Made drag-drop of images/video/csv work even if over other elements. Dragging images onto graphics will only create a mask if the cmd/ctrl key is down.
- **Improved:** UI of LowCode Editor
- **Improved:** resize quality of images is now sharper
- **Improved:** Color picker now allows for pasting Hex,RGB, or Color names
- **Fixed:** Fixed issue where list editor in Property Inspector couldn't take v key

88.191 VERSION 0.2.0 (2021-09-08):

- **New:** Visual Changelog. You can now see the changelog when Appli is updated.
- **New:** Actions. You can click on the "Low-code" button underneath the property inspector to bring up the actions editor. Here, you can choose things to do. For now, there is "goScreen".
- **New:** Video Element. You can drag mp4 videos onto the playground to create video players, or draw from the toolbar.
- **New:** Icons for Apps. You can now create an icon for your app to show within the player.
- **New:** Undo/Redo System. You can now undo and redo.
- **New:** Element validation system. This allows updates to automatically fix old elements that don't have correct properties.
- **Improved:** There are now "No-code" and "Low-code" buttons at the bottom of the Property Inspector. These allow you to set more complex properties/actions for some elements.
- **Improved:** Siginificant improvements to scrolling for native objects

- **Improved:** Added a lot of properties to Tables to allow for more customizability. Includes rowHeight, rowVerticalAlign, TabStops
- **Improved:** Improved click areas and tab stop handling for Tables
- **Improved:** Table/Layout refresh UI
- **Improved:** Updated the Project Browser to have easier drag/drop and nesting
- **Improved:** Changed some keyboard shortcuts, added ctrl/cmd-z, ctrl/cmd-d
- **Improved:** Blob backend handling for Images/Video. This makes things more performant.
- **Improved:** Updated UI of camera element
- **Improved:** Add properties for letterboxing and maintaining aspect ratio for images
- **Improved:** Add drag-handle to left pane for resizing
- **Improved:** Allow custom regions in LiveCloud so we can connect to custom regions
- **Fixed:** Fixed an issue with opaqueness of layouts
- **Fixed:** Fixed an issue where layouts wouldn't work with multiple rows without data-binding
- **Fixed:** You can now drag multiple elements into a tab/layout at once
- **Fixed:** Tab children are now placed correctly on load
- **Fixed:** Fixed a bug where copying and image and then deleting the copy would affect the original image.
- **Fixed:** Stack shouldn't be resizable on login/project screens
- **Fixed:** Stop slider editor from changing properties as soon as it is drawn
- **Fixed:** Don't select locked elements with Ctrl-A
- **Fixed:** Prevent non-ascii characters when dragging in CSVs
- **Fixed:** Fix some dynamic text sizing issues

88.192 VERSION 0.1.23 (2021-06-18):

- **Improved:** Refactor updateProp to be common among elements (internal change)
- **Fixed:** Dropdown, Login, Create Account elements can now be nudged correctly (arrow keys).

88.193 VERSION 0.1.22 (2021-06-15):

- **Improved:** Error messages from creating a new app now give more information.
- **New:** [WIP] Ability to scale the playground canvas. Use the scroll wheel to adjust the scale of your project. The feature is not complete and does not properly scale every element. The potential is there and it will very useful when we are feature-complete.

88.194 VERSION 0.1.21 (2021-06-04):

- **Fixed:** Made a correction for loading landscape elements into the index.

88.195 VERSION 0.1.20 (2021-06-03):

- **Fixed:** Project browser search no longer shows elements on both orientations.
- **Fixed:** Project browser search field no longer auto-fills with "all items" if the field is empty and user is still typing.
- **Improved:** Hide project browser dropdown if clicking into the search field.

88.196 VERSION 0.1.19:

- **Fixed:** Project thumbnails will update after saving to the cloud and refreshing the projects screen.
- **Fixed:** Project UI is no longer duplicated when refreshing, sorting, or deleting a project.
- **Fixed:** Update projects in user settings if a project is deleted or renamed.

88.197 VERSION 0.1.18:

- **Fixed:** Screen thumbnails no longer show an incorrect screen or have a large gray space.
- **Improved:** All changed, screen thumbnails will be updated when saving.

88.198 VERSION 0.1.17:

- **Improved:** Removed double-clicking to change image source.
- **Improved:** Added support for resetting the natural aspect ratio of single and multi-selected images to a contextual menu.
- **Improved:** Removed shift-double-clicking method to reset the natural aspect ratio of an image.

88.199 VERSION 0.1.16:

- **Improved:** Allow scrolling a layout element with the scroll wheel while in edit mode when multipleRows is false.

88.200 VERSION 0.1.15:

- **Fixed:** Thumbnails in projects screen no longer have large gray space.
- **Improved:** Adjusted default fontSize of tables to 20
- **Improved:** Log is using updated cdb_autoSetupAuthKey() to support init code being in a behavior.
- **Improved:** Using latest CanelaDB libraries

88.201 VERSION 0.1.14:

- **Fixed:** Make background for transparent elements show up when selected

88.202 VERSION 0.1.13:

- **Fixed:** Option click drag to copy works again.

88.203 VERSION 0.1.12:

- **Fixed:** Make right click work consistently on Windows

88.204 VERSION 0.1.11:

- **Fixed:** Make Option-Drag to duplicate groups that have groups inside of them work.

88.205 VERSION 0.1.10:

- **Fixed:** Allow copy/paste of text inside text elements and other fields

88.206 VERSION 0.1.9:

- **Fixed:** Add checks for thumbnail in case of corruption
- **Fixed:** Make line deselectable with cmd/ctrl/shift click

88.207 VERSION 0.1.8:

- **Improved:** Text support for chunks - you can now set the font, size, and color of selections within a text element
- **Fixed:** Fixed Ellipsis truncation in text element so it won't unstyle chunks or delete text
- **Fixed:** Fix deletion of empty fields so they will delete more cleanly

88.208 VERSION 0.1.7:

- **Improved:** Alignment of multiple objects and Selection is now faster
- **Fixed:** Alignment of multiple objects now works in standalone

88.209 VERSION 0.1.6:

- **Improved:** Show selected lines when they are multi-selected
- **Improved:** Make all footer fields have consistent capitalization
- **Fixed:** Make new platforms layer correctly under selection/guideline layers
- **Fixed:** Add lock screen when switching orientations

- **Fixed:** Make sure “show offscreen elements” continues to stay toggled when creating new screens/platforms/etc.

88.210 VERSION 0.1.5:

- **Improved:** Double click on tab name while in edit mode will switch the active tab of tab element.
- **Fixed:** Tabs will no longer incorrectly show as highlighted/selected on project load.

88.211 VERSION 0.1.4:

- **Fixed:** Cursor issue with Gap Rects in RepeatGrid (Thanks Clarence Martin)

88.212 VERSION 0.1.3:

- **Fixed:** Don’t allow selection of both parents and children at the same time.

88.213 VERSION 0.1.2 (2021-05-17):

- **Improved:** You can now drag/drop multiple items in the Project Browser at once

88.214 VERSION 0.1.1 (2021-05-07):

- **New:** First closed beta release