

# Oracle® Cloud

## Creating Low-Code Apps with Oracle Mobile Cloud, Enterprise



18.1.1  
E92743-01  
January 2018



Copyright © 2017, 2018, Oracle and/or its affiliates. All rights reserved.

Primary Author: John Bassett

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

**U.S. GOVERNMENT END USERS:** Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

Audience	vi
Documentation Accessibility	vi
Related Resources	vi
Conventions	vi

## 1 What Is Mobile Application Accelerator?

---

MAX at a Glance	1-1
What Kind of App Can I Build?	1-1
Video: An Introduction to Mobile Application Accelerator (MAX)	1-2

## 2 Getting to Know MAX

---

How Do I Build a Mobile App Using MAX?	2-1
How Do I Get Started?	2-1
What Do I Need to Access MAX?	2-2
Creating the First Screen of Your App	2-2
What's Next?	2-7

## 3 Developing Your App

---

Getting to Know the Designer	3-1
The Navbar	3-2
The Preview	3-5
The Preview Settings	3-6
The Properties Inspector	3-7
Testing and Publishing Tools	3-10
The Components Palette	3-10
Configuring Graph, Chart, and Meter Components	3-14
Legend and Titles	3-14
Screens	3-22
Screen Actions	3-23

Screen Summary	3-25
Dashboards	3-28
Working with Tiles	3-29
Cards	3-37
Lists	3-41
List Layouts	3-41
Adding a Search Field	3-44
Adding Swipe Gestures	3-47
Forms	3-50
Data Mapping for Forms and Form Fields	3-53
Field Types for Lists and Forms	3-55
Creating Select Lists	3-60
Images	3-61
Maps	3-65
Use Case: Adding a Map Component	3-67
Enabling Your App to Display Content	3-74
Exploring Services	3-76
Getting to Know Business Objects	3-76
Service Upgrades	3-81
Taking a Look at the Data Mapper	3-82
Mapping Field Values to Forms	3-85
Use Case: Mapping Data to a Line Chart	3-86
Adding Actions	3-88
Application Properties	3-92
How Do I Set Application Properties?	3-92
Using QuickStarts	3-95
Use Case: Creating a List Using QuickStarts	3-97
How Do I Add Another Create or Edit Screen?	3-108
Using the Image Gallery	3-109
Managing Your Screens	3-111
Adding Landing Screens to Your App	3-112
Rearranging Screens	3-116
Video: Mobile Application Accelerator (MAX) Demo	3-116

## 4 Building and Testing Your Apps

---

Previewing Your App	4-1
Testing Your App	4-2
Installing the MAX App	4-8
Installing the MAX App on iPhones	4-9
Installing the MAX App on Android Devices	4-10

MAX App Quick Reference	4-10
Sharing Your App	4-12

## 5 Managing Your Apps

---

Importing and Exporting Your Apps	5-2
-----------------------------------	-----

# Preface

Welcome to *Creating Low-Code Apps with Oracle Mobile Cloud, Enterprise*.

## Audience

This guide is intended for anyone who wants to develop cross-platform mobile apps with little to no coding in Oracle Mobile Cloud, Enterprise.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Resources

- *Getting Started with Oracle Cloud*
- *Developing Applications with Oracle Mobile Cloud, Enterprise*
- *Administering Oracle Mobile Cloud Enterprise in a Customer-Managed Environment*
- *Analyzing Customer Data with Oracle Mobile Cloud, Enterprise*
- *Managing Oracle Mobile Cloud, Enterprise*
- *Known Issues for Oracle Mobile Cloud, Enterprise*
- *Licensing Information User Manual for Oracle Mobile Cloud, Enterprise*
- *REST API Reference for Oracle Mobile Cloud Enterprise - Platform APIs*
- *REST API Reference for Oracle Mobile Cloud Enterprise - Tooling APIs*

## Conventions

The following text conventions are used in this document:

---

Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

---

# What Is Mobile Application Accelerator?

Mobile Application Accelerator (MAX) is a feature of Oracle Mobile Cloud, Enterprise (OMCe) that allows anyone to quickly create, test, and publish sophisticated cross-platform mobile apps. No coding required.

## MAX at a Glance

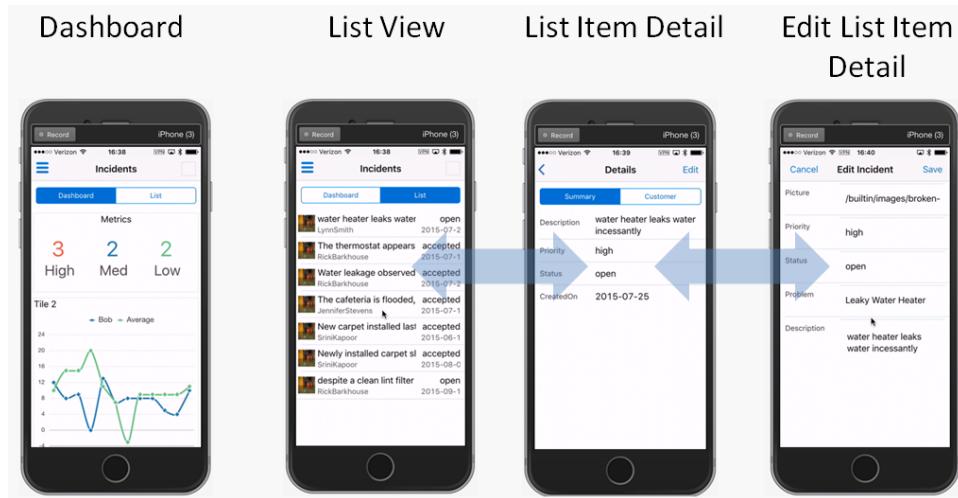
MAX provides you with everything that you need to build, test, and publish a mobile app that run on both iOS and Android devices.

To build an app, you don't have to download platform-specific development tools like Android Studio or Xcode. And you don't have to worry about keeping them up to date or configuring device simulators, either. With MAX, you're not even tied to a development machine. Because it's a web-based tool, you can log in from any computer or tablet and immediately begin building your app.

## What Kind of App Can I Build?

Using MAX, you can create apps that support your mobile workforce. For example, you can build an app for sales organization that gives users an at-a-glance view of the sales pipeline by populating a dashboard screen with counters and graphs for active accounts, pending deals, deal sizes, and so on.

To get you familiar with the MAX UI, we're going to use a mobile app for appliance repair technicians who work in the field as an example. This app has a dashboard that's divided into upper and lower regions. The dashboard's upper region has a set of counters that shows the technician the number of open incident reports (IRs) by priority (high, medium, and low). Its lower region contains a line graph that plots IRs assigned to the technician on a monthly basis. The app also supports the technician's workflow with a screen that displays a list view of the open IRs. From this screen, a field technician drills down to a detail page for any list item to learn more about the IR. She can return to the list, or she can edit the IR or create a new one.



## Video: An Introduction to Mobile Application Accelerator (MAX)

To see how you can build, test, and publish mobile apps using MAX, take a look at this video:

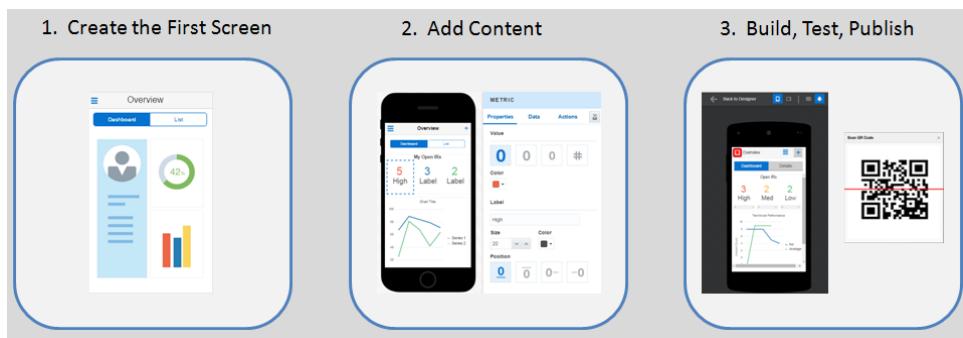


## Getting to Know MAX

### How Do I Build a Mobile App Using MAX?

It's a three-step process to get your app to its users.

1. Create the foundation of your app by choosing its first screen. You can decide how you want to arrange the content by selecting a template for a commonly used screen type, like a dashboard or a list. If none of these templates reflect your use case, you can choose a blank page and then populate it with UI components later on.
2. Add content to your app. You can drag and drop UI components to quickly build your app's screens. You enable these screens to display data using a service published by your development team. To help you integrate the right data from the service into your UI, your development team defines a set of *business objects*, which use concrete terms to describe the data and how to use it.
3. Build, test, and publish your app. You can test your app in a hosted simulator to see how the actual data renders in your UI. You can see how your app interacts with live data on an actual device by viewing it through the MAX App, which you can download from the App Store or Google Play. After you've built your app, MAX generates a QR code that enables you to share the app with users. This QR code remains the same even when you've updated the app.



### How Do I Get Started?

Before you build a mobile app, you should define a use case, how the user experience can be expressed by the screen flow of your app, and how your app (and its users) will interact with the data.

Even as you gather requirements and construct your use case scenarios, you can login to and get oriented by watching the videos. From there, you can peruse MAX to see which of its page and UI components fit your needs. The central piece of your mobile app will be the data that it accesses. While MAX enables you to give your mobile app access to backend data, you need to work with your dev team to learn about the available resources before you actually start building your app.

## What Do I Need to Access MAX?

To access MAX and run your mobile apps, you need accounts for MAX itself, as well as a test user account, and mobile user account.

Check with your administrator to make sure that you have the following:

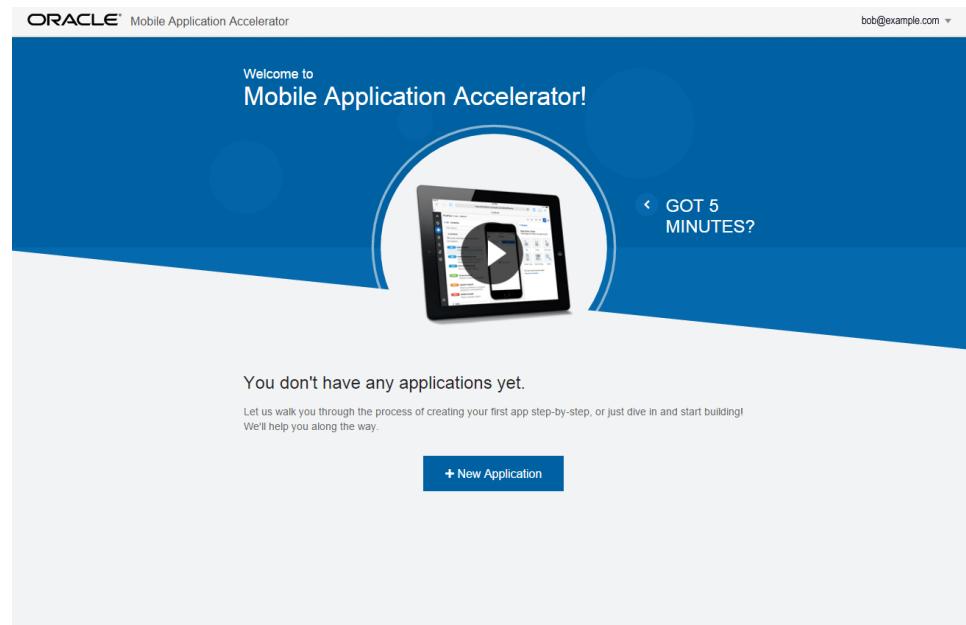
Account Type	Description
A MAX account	<p>This is the user name and password that enables you to login to MAX. If your account has been set up properly, then you can build and publish mobile apps. You might have problems if your account configuration is incorrect or incomplete, so keep the following in mind:</p> <ul style="list-style-type: none"><li>• Contact your administrator if you don't see the MAX UI after logging in to Oracle Cloud.</li><li>• Contact your administrator if you see an error message similar to the following when you're trying to publish an app: <i>A Mobile Cloud Production environment has not been configured for your user ID.</i></li></ul>
A test user account	<p>You need a test user account to preview your app using live data. Ask your administrator to create this account, which you also need to generate the QR code for testing your app on a device. You can also use the test user account to test an app on a device.</p>
A mobile user account	<p>Anyone (including you) who runs mobile apps needs a mobile user account. If you're already using mobile apps created by your company, then you have this account. Otherwise, make sure that you have it so that you can run your apps on a device.</p>

## Creating the First Screen of Your App

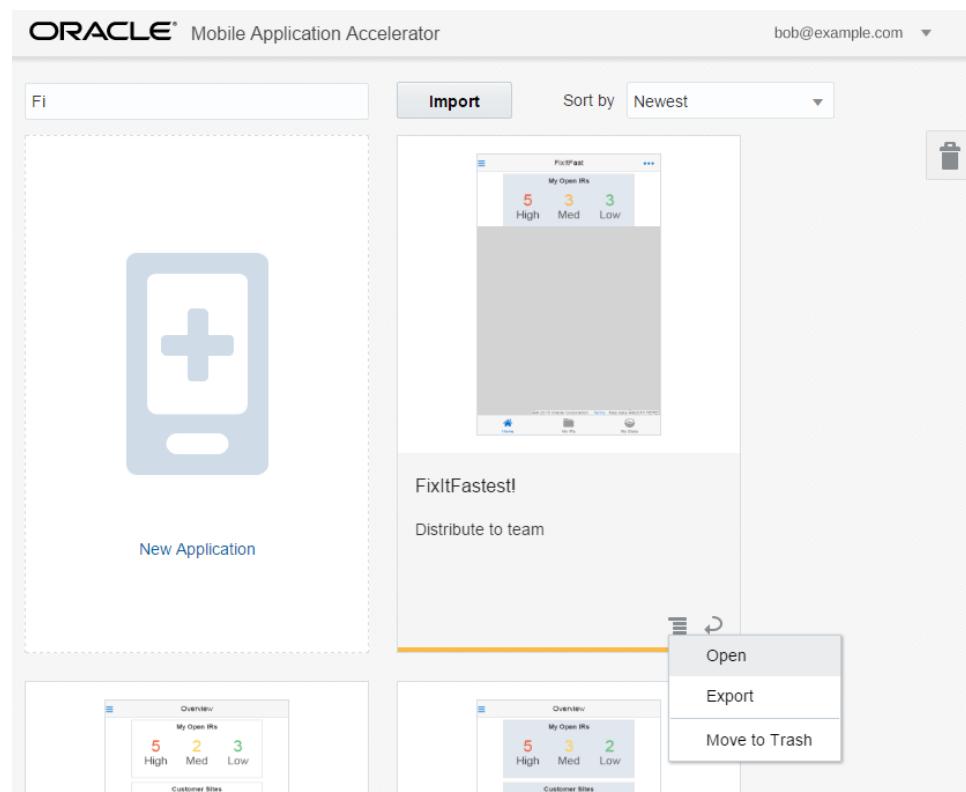
You can create the foundation of your app in just few clicks. Just give it a name and pick out a style for its first screen.

**1. After you login, click **New Application**.**

If this is your first visit to MAX (or if you haven't created an app yet), you'll see a welcome page that looks like this:

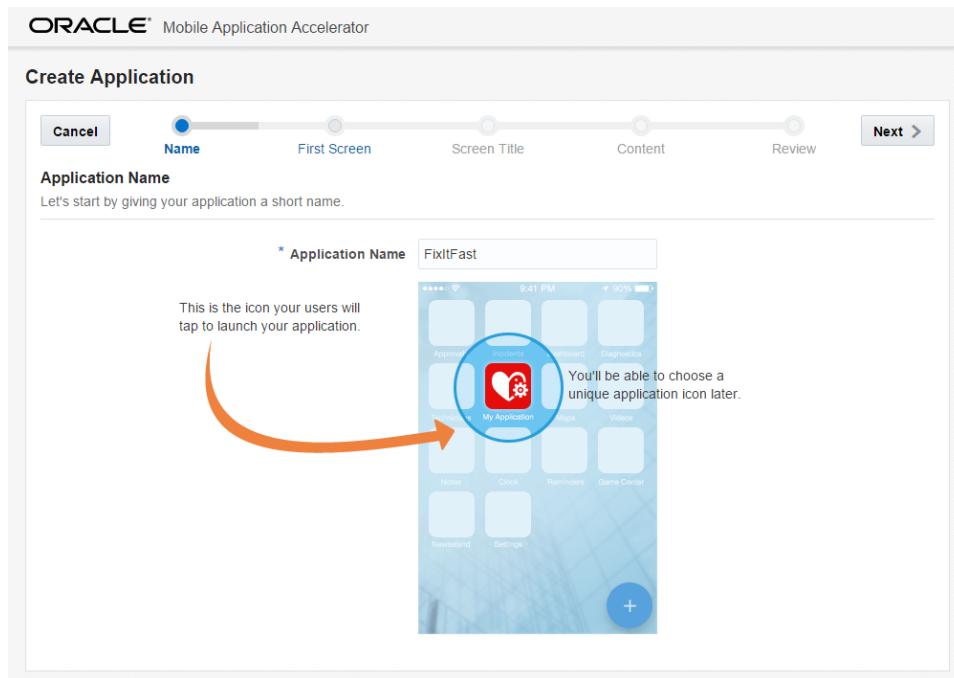


If you've created an app already, then welcome back! In this case, you'll see the Applications page after you login.



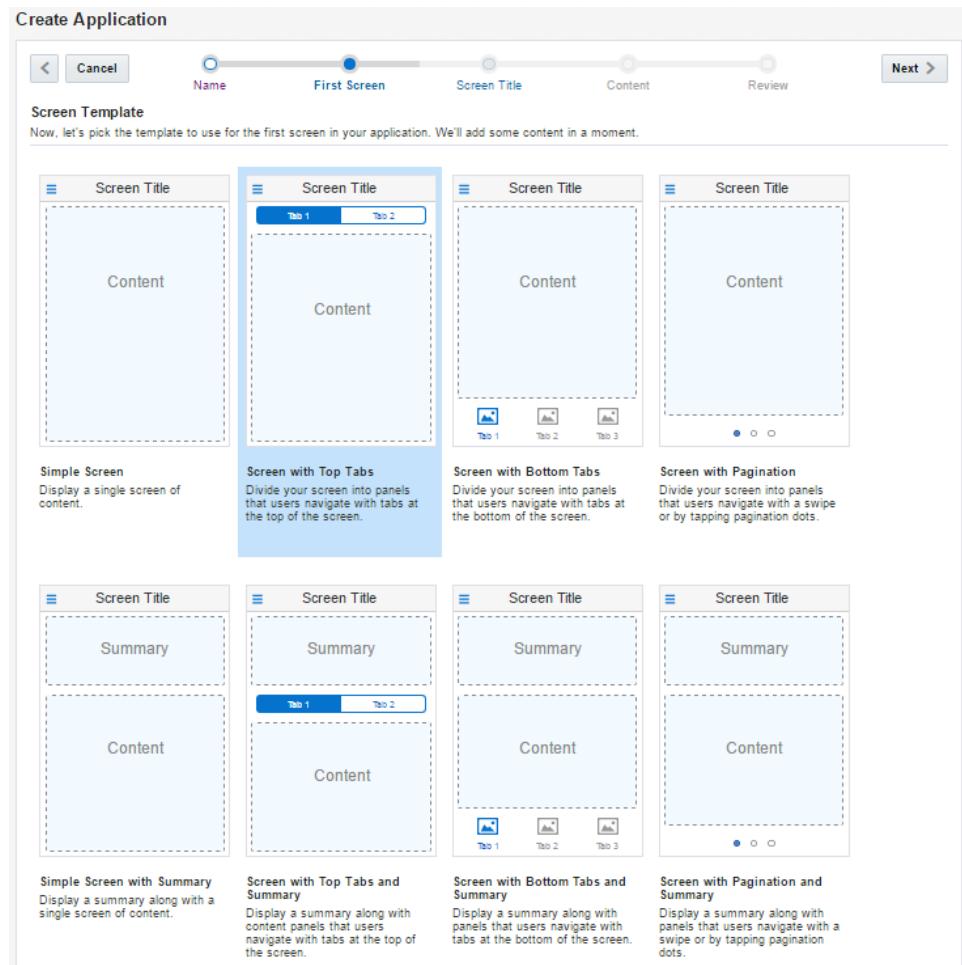
You can also create a new app from here as well as delete, update, or republish your existing apps. For now, however, just click **New Application**.

2. Next, give your app a display name and then click **Next**.



You can change the default icon (and other default images). To find out more, see [Using the Image Gallery](#).

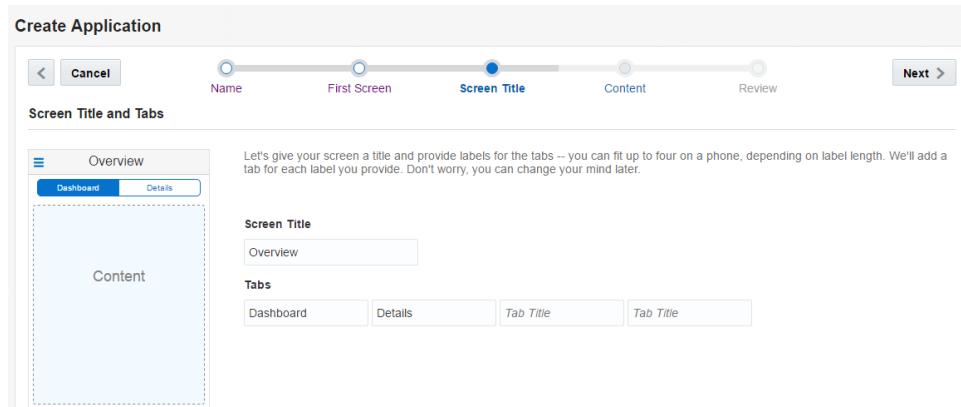
3. Now that you've named your app, you're ready to create the first screen that users see after logging in. First, choose from among these commonly used formats. You can opt to create a landing page with non-scrolling, banner-like summary pane at the top. [Screen Summary](#) describes these components, including how to add or remove them.



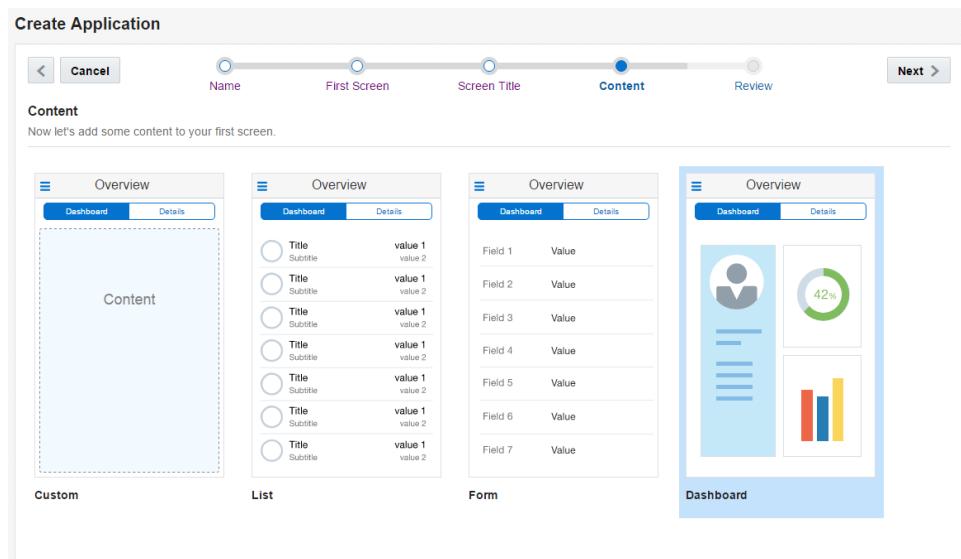
These templates all have a main menu icon (≡). After you finish your first screen, the main menu becomes active – you can open and close it. See [What's Next?](#) to find out more. There are two versions of each screen type: one that has a summary field where you can display things like KPIs or images, and one that doesn't.

4. Next, enter a name in the Screen Title field.

If your first screen includes tabs, you can create them by replacing the *Tab 1*, *Tab 2*, and *Tab Title* text in the Tabs fields with your own names. You can add up to four tabs top tabs or five bottom tabs.



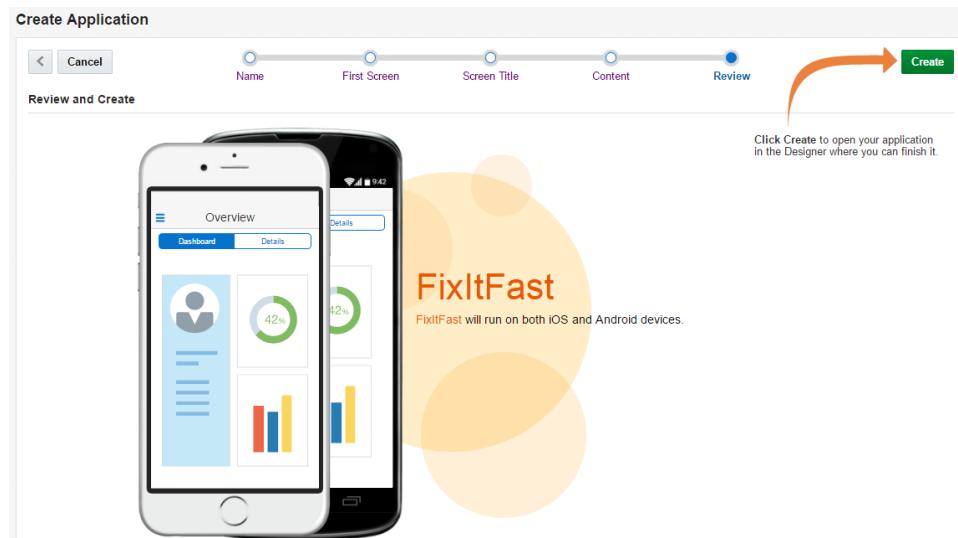
5. Select the layout option for your content (that is, for the UI components you want to add and ultimately, for the data that they display). Click **Next**.



 **Tip:**

If you're not sure how you'll lay out your content, then choose the **Custom** option for now. You can explore other options in the Designer later. To find out more, see [Getting to Know the Designer](#).

6. If everything looks OK (so far), click **Create**. If you want to pick a different format, you can go back through the wizard using either by clicking **Previous** () or by clicking on the progress trail at the top of the page.

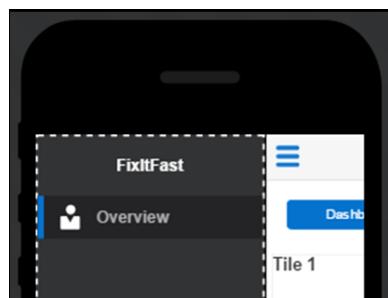


 **Tip:**

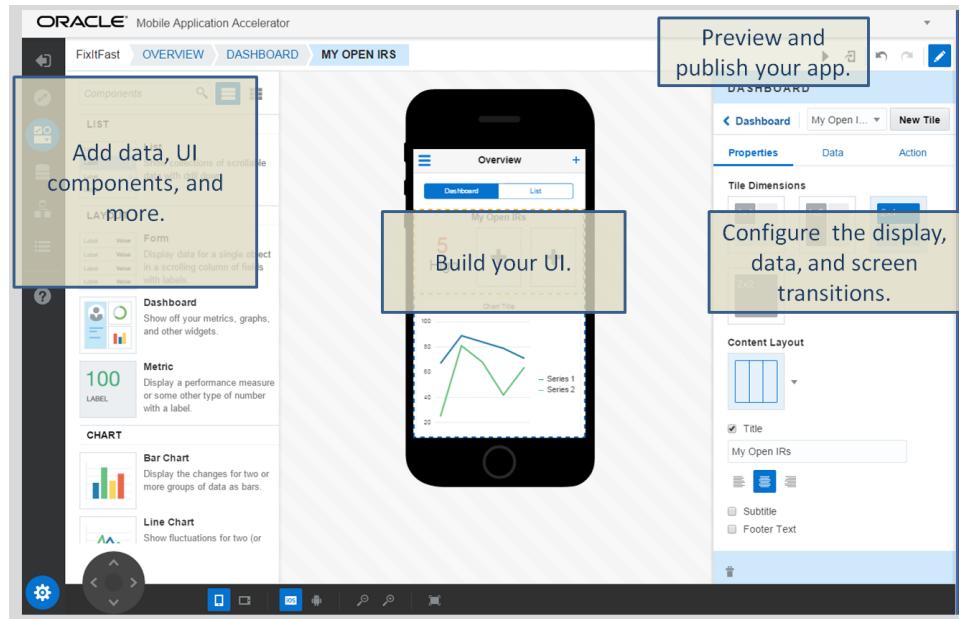
Later on, you can add another screen and make it the first screen that your users see when they open your app. See [Rearranging Screens](#) to find out more.

## What's Next?

MAX has already added some functionality to the first screen that you've just created, like navigating to your app's main menu when you click the menu icon (≡). You can also traverse through your app using tab or pagination functions.



With the basic structure of your first screen now complete, you can use the Designer to dress it up with UI components, add other screens to it, and sequence them into the flow that supports your scenario. The Designer lets you integrate your UI with the service that provides your app with its data. It also lets you build, test, and publish your app. To find out more, see [Getting to Know the Designer](#). If you want to jump in, check out the  [tutorial](#).



# 3

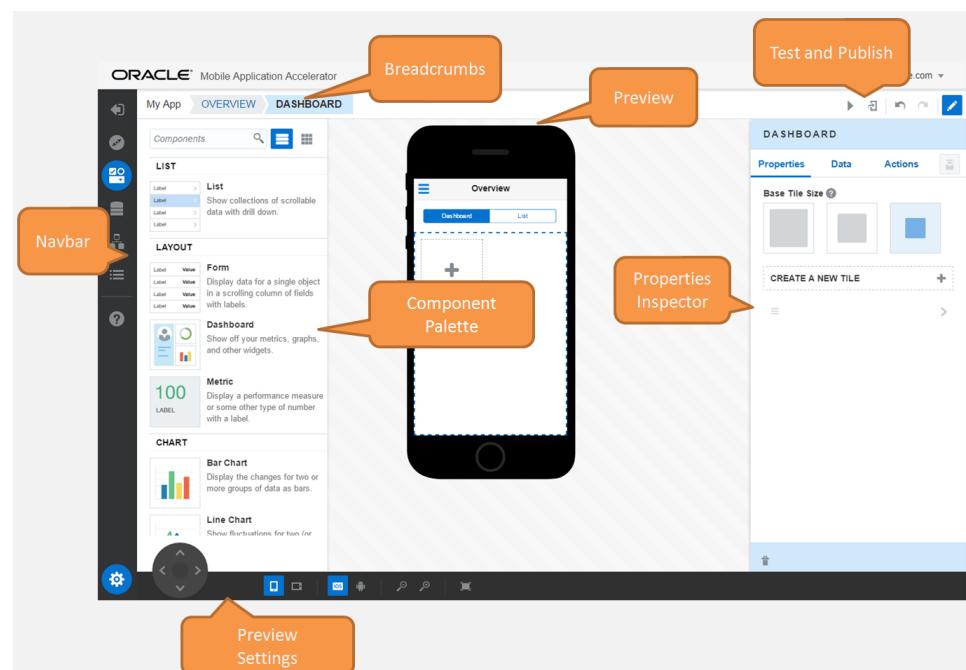
## Developing Your App

### Getting to Know the Designer

The Mobile Designer lets you add UI components like graphs and lists to your app and wire it to data. None of this requires any coding, just simple configuration.

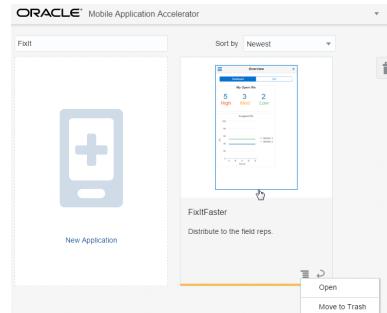
The Mobile Designer is your one-stop shop for designing, testing, and distributing your app. Its tools let you compose screens and pick out the kind of data that they display. While you're building your app, you can preview it to see how it handles live data. When you're ready to see how your app behaves on an actual device, the Designer lets you distribute it to your test group. The Designer also lets you publish the finished version of your app. With a couple of clicks, it generates the QR code that you use to distribute your app to your users. To get you oriented, here's a Cook's tour of the Designer:

- [The Navbar](#)
- [The Preview](#)
- [The Properties Inspector](#)
- [The Components Palette](#)
- [Testing and PublishingTools](#)

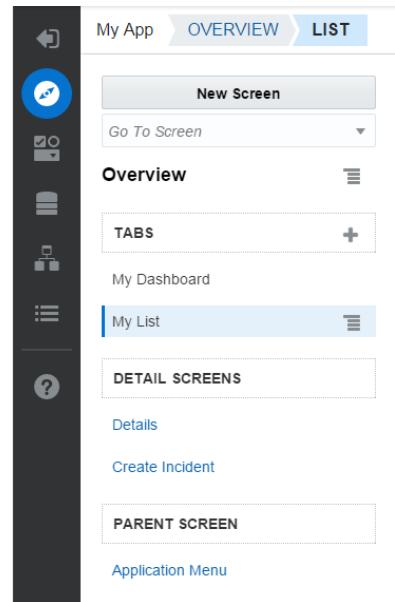


## The Navbar

Icon	Tool	Uses
	Back to My Apps	Takes you to the Application page, which is your landing page after your first visit to MAX. After you've created the first page of your app, you can open it from this page to continue working on it with the Designer. You can access all of your apps from this page. It's also where you can add new ones, delete the ones that are no longer in use, and republish your apps. You can access import and export functions from this page, which can help you troubleshoot and debug apps. To find out more, see <a href="#">Managing Your Apps</a> .

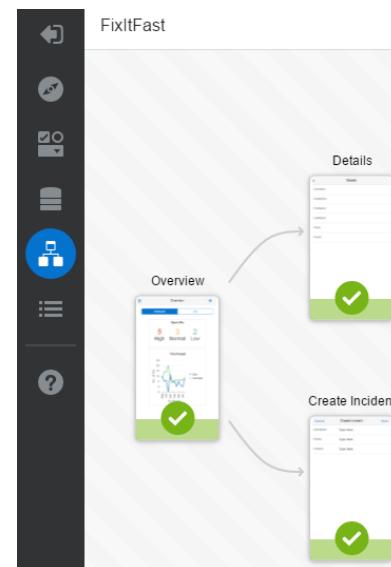


Icon	Tool	Uses
	Application Screens	<p>Explore, create, edit, and delete your app's screens. You can add a screen to your app by clicking the <b>New Screen</b> button. To find out more, see <a href="#">Managing Your Screens</a>.</p> <p>To access any screen in your app, first click the <b>Application Screens</b> icon. Next, choose a screen from the Go To Screen menu to open it in the Preview.</p>



The Application Screen's page provides configuration options that are specific to the screen that you select from the menu. It also describes the screen's position within the hierarchy of the screen flow by listing the parent screen and detail screens. These related screens are represented as links. Clicking them opens the parent or detail page in the Preview and Properties Inspector.

Icon	Tool	Uses
	Components	Opens and closes the Components Palette. To find out more, see <a href="#">Configuring Graph, Chart, and Meter Components</a> .
	Data	Opens the Services Catalog, a repository for the services published by your development team. To find out more, see <a href="#">Enabling Your App to Display Content</a> .
	Screen Flow	Lets you see how all of the screens in your app are related to each other in terms of the application flow.



If your app has any errors that might prevent you from publishing it, you can quickly spot them here. Double-clicking a screen opens it in the Preview and the Properties Inspector.

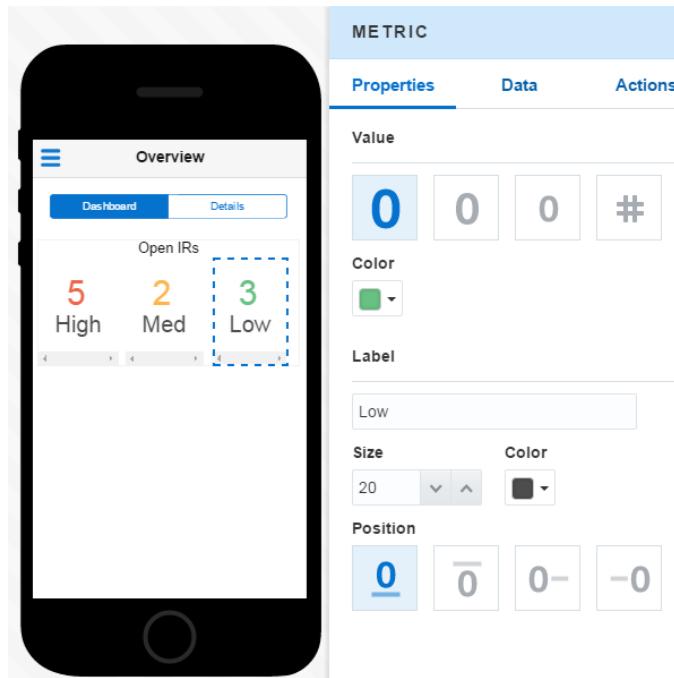
Icon	Tool	Uses
	Application Settings	Change the basic information for your app. When you created your first screen, MAX added a default icon. If you want to change the image for this icon, you can do that here. You can also change your app's name and add a description. For more information, see <a href="#">Managing Your Apps</a> . To find out about changing the launch icon, see <a href="#">Using the Image Gallery</a> .
	Help	Opens the help center. You can refer to the user guide and revisit the MAX videos.
	Preview Settings	Opens the Preview Settings tools, which let you change the appearance of the Preview. To find out more, see <a href="#">The Preview Settings</a> .

## The Preview

The Preview allows you to approximate how your app looks on a device.

You can build your UI by dragging UI components from the Components Palette into the Preview. Any change that you make to your app, such as adding or removing a UI component, or selecting an application that allows a UI component to display data, are immediately reflected by the Preview. As described in [The Properties Inspector](#), when

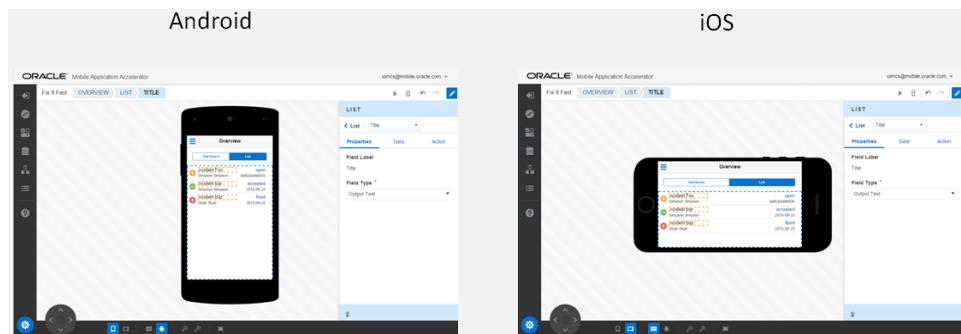
you select a component in the Preview, you can then configure it using the Properties Inspector. To find out how to adjust the appearance of the Preview to reflect different platforms, sizes, or orientations, see [The Preview Settings](#).



## The Preview Settings

You can change how the Preview displays your app while you're working on it using the Preview Settings. Use these settings to get an approximation of how your screens look on different platforms, in different sizes, and in different orientations. You can use various combinations of the iPhone and Android options along with the size and orientation options to transform the Preview as you see fit. You can also move the

Preview around the designer using the wheel (  ) or your mouse.

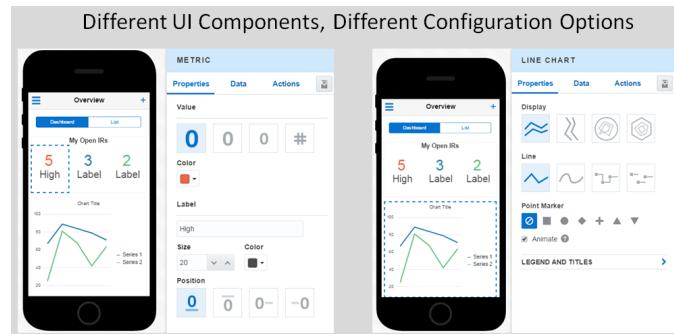


To see how your app looks in different orientations and as different platforms on an emulator (and how it works with live data), see [Building and Testing Your Apps](#).

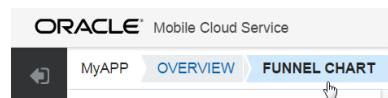
## The Properties Inspector

Use the Properties Inspector to configure your UI.

In addition to adjusting the look of any selected UI component, you can also use the Properties Inspector to add data to your component, or to configure your component to trigger an action, like a tap gesture to open a screen or send an e-mail. Depending on the component you're currently working on, the Properties Inspector lets you change the display, data, or triggered actions.

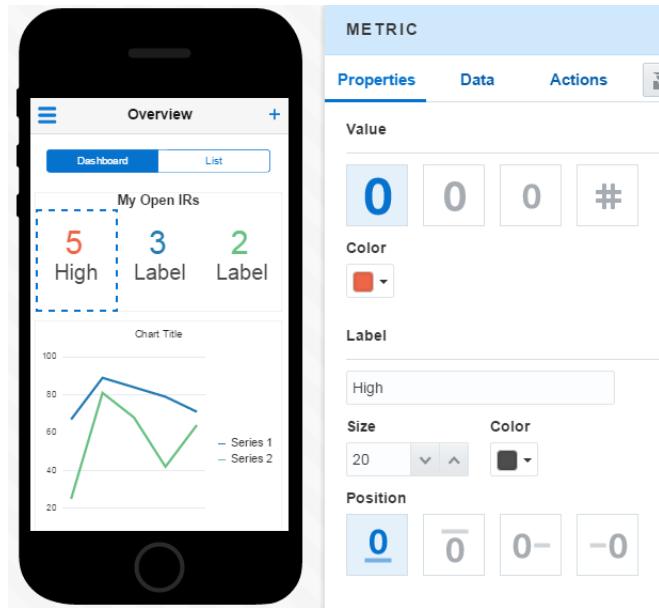


In other words, the configuration options available in the Properties Inspector depend on the current component that you've selected (that is, the one highlighted in the breadcrumbs at the top of the page or selected in the Preview). For example, if you're working on a line chart component, the Properties page displays chart-specific options that let you pick the point marker and define the chart's legend and title. If you've selected a different type of UI component, like a list, the Properties Inspector display a different set of options.



The Properties Inspector has the following tabs:

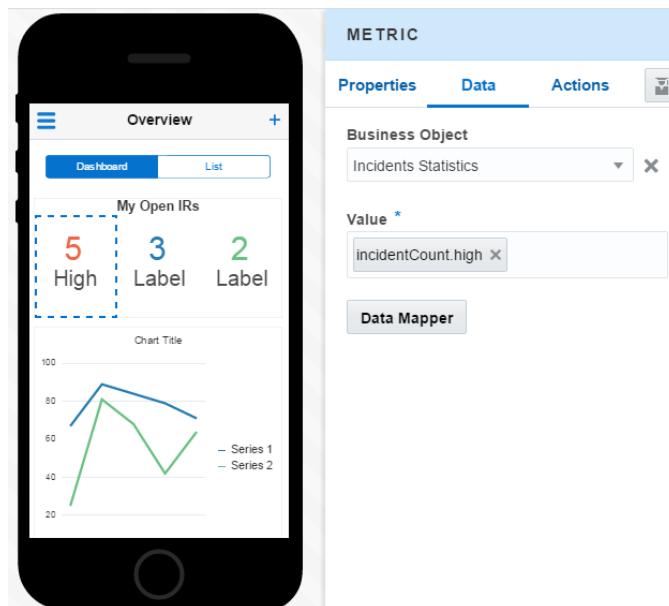
- **Properties**—Use configuration options in this page to change the appearance of a component.



- **Data**—This tab lets you bind your UI to data. Like the display and layout options, the options that display depend on the UI component that you've selected. To help you identify the data that you can display in a UI component, your development team creates models of data called business objects. After you've selected a business object for a UI component (that is, map the UI to the data that you want it to use), the Preview displays mock data. This isn't the actual data that your app will use. Instead, it's a representation, or placeholder, to help your data mapping. To find out more, see [Taking a Look at the Data Mapper](#).

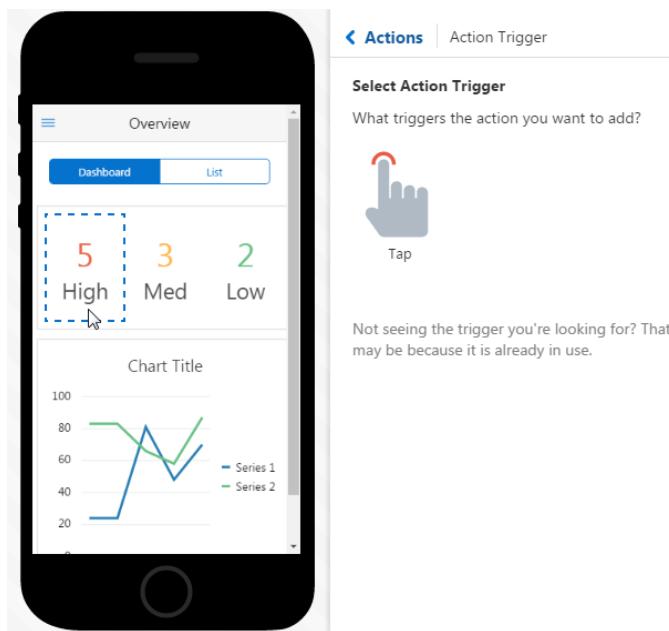
 **Note:**

Be sure that you have a business object that supports the data requirement of the component.

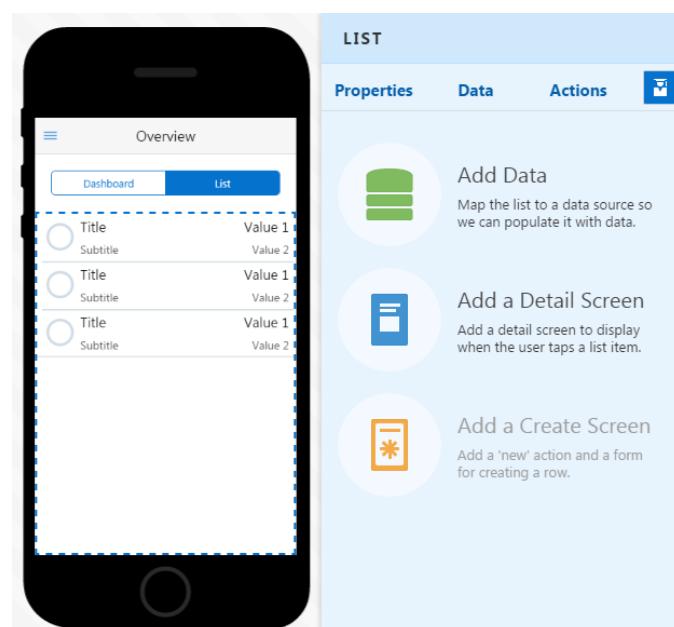


- **Actions**—You can add a user gesture, like a tap, on a selected component. You can define the action that's triggered by that gesture, which can be:
  - Navigating to (or back to) another screen in your app.
  - Actions defined by the business object used to return data to the app. These can include actions to refresh data, and actions that correlate to CRUD (create, read, update, and delete) operations. For example, you can configure a tap gesture to refresh the data for a counter.

Not every UI component can have an action associated with it (like output fields, for instance).

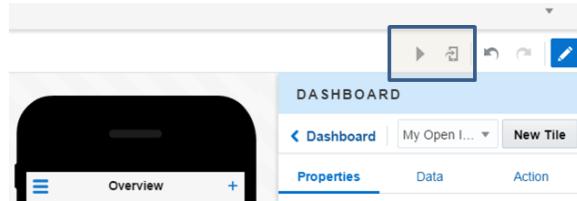


- **QuickStarts**—A set of wizards that speed up your app development. To find out more, see [Using QuickStarts](#).



## Testing and Publishing Tools

The Designer has a set of tools that you can use to preview your app using live data, test it on a device, and publish it. To find out more, see [Testing Your App](#) and [Sharing Your App](#).



## The Components Palette

The Mobile Designer provides you with a set of UI components that are optimized for mobile apps. They include layouts for forms and itemized lists as well as data visualization tools like charts and graphs. You drag these components from the Components Palette into the Preview. You can then configure how they display and use data using the Properties Inspector.

### List and Layout Components

You can arrange your data as the following:

Component	How Do I Use It?
List	Use it to organize content into a list. For information on configuring how your list displays, see <a href="#">Lists</a> . To find out how you can enable users to drill down from a list to detail pages and update a list using create or update screens, see <a href="#">Using QuickStarts</a> .
Form	Use it to organize input and output fields. You can use forms as child pages of list. To find out more, see <a href="#">Forms</a> .
Dashboard	Use it to assemble high-level information, like KPIs, charts and graphs. You'd typically use a dashboard page as the first screen of your app. Before you add UI components to a dashboard, you need to lay down tiles for them first. To find out more, see <a href="#">Dashboards</a> .
Card	Use it as a small container for displaying summary information. To find out more, see <a href="#">Cards</a> and <a href="#">Screen Summary</a> .

Component	How Do I Use It?
Metric 	Use it to display a metric: a performance measure, or some other type of number with a label.

## Chart and Graph Components

You can visualize your data as various charts and graphs, like pie charts, scatter charts and bubble charts. See [Configuring Graph, Chart, and Meter Components](#) to find out more.

Chart Type	How Do I Use It?
Line 	<p>Use a line graph to compare changes for two groups over time. For example, you can use this chart to compare the number of mystery novels sold by a bookstore from March to June and the number of romance novels sold during the same period.</p> <p>Line charts plot data points for the things that you want to measure on the vertical y-axis (like the number of books sold) against the horizontal x-axis, which categorize this data. See <a href="#">How Do I Create a Line Chart?</a> to find out more.</p> <div style="border: 1px solid #4CAF50; padding: 10px; background-color: #e0f2e0; margin-top: 10px;"> <p> <b>Tip:</b></p> <p>Use a bar chart instead of a line chart for data that shows greater changes (higher peaks and steeper valleys) between intervals, or that covers a long period.</p> </div>
Bar 	<p>You can use a bar chart to see changes over time for different groups, or to compare different items in a related category.</p> <div style="border: 1px solid #4CAF50; padding: 10px; background-color: #e0f2e0; margin-top: 10px;"> <p> <b>Tip:</b></p> <p>Use a bar chart instead of a line chart if the data along the x axis is not continuous (that is, it represents things like categories or geographical regions).</p> </div>
Pie 	<p>Bar charts have a vertical y axis and a horizontal x axis. To show the number of books sold by category, you'd populate the y axis with amounts (50, 100, 200, etc.) and the x axis with categories like mystery, non-fiction, science fiction, self-help, etc. See <a href="#">How Do I Create a Bar Chart?</a> to find out more.</p> <p>In a pie chart, different data points (represented as slices) are presented as parts of a whole.</p> <p>For a bookstore app, these slices might show the percentages of different categories of books (like mystery, romance, etc.) in terms of the bookstore's entire bookstore's inventory. See <a href="#">How Do I Create a Pie Chart?</a> to find out more.</p>

Chart Type	How Do I Use It?
Area	<p>You can use an area to track changes over time for two or more related groups. It's similar to a line chart, but the areas beneath the lines are filled in to give you a picture of changes over time. Like a line chart, the area chart uses the vertical y axis to measure something (like sales amounts) and the horizontal x axis to display time periods.</p> 
Scatter	<p>A scatter chart helps you to see the correlation between two different types of data. It helps you ascertain if these data sets change together. You plot this data on the x-y axis of the graph. For example, to show the relationship between site visits to book sales, you can plot the sales amounts along the vertical y axis (\$100, \$200, \$300, and so on) and the number of site visits (1000, 2000, 3000, etc.) along the horizontal x-axis. See <a href="#">How Do I Create a Scatter Chart?</a> to find out more.</p> 
Funnel	<p>Funnel charts compare how many users start a workflow against how many complete it. Funnels segment the workflow into a sequence of steps that guide users to some goal. Typically, users drop off at each step.</p> <p>If your bookstore has an online shopping app, you can use a funnel to see how many users finish a checkout process compared to how many started it. You can see where the app is losing users. See <a href="#">How Do I Create a Funnel Chart?</a> to find out more.</p> 
Bubble	<p>A bubble chart has numeric values for both the x and y axes. Along these axes, you can plot related data for the bookstore, like sales data on the vertical y axis (25, 75, 100, etc.) and book prices on the horizontal x axis (\$5, \$10, \$15 and so on). These points are depicted as bubbles, whose size conveys a third dimension. For the book store, the size might be gauged by the proportion of total sales. See <a href="#">How Do I Create a Scatter Chart?</a> to find out more.</p> 
Donut	<p>Like a pie chart, a donut chart show data sets in proportion to a total. You can compare data sets as concentric rings.</p> 

### Gauge Components

Gauges are measuring instruments that indicate a quantity such as sales, stock levels, temperature, or speed. Gauges identify thresholds using colors. For example, a gauge value axis might show ranges colored red, yellow, and green to represent low, medium, and high states. Typically, gauges show a single data value. If your data support status meters, then you can use them as fields in the Form, List, and Card components. [Field Types for Lists and Forms](#) describes how to include these components within lists and forms.

#### Tip:

Use gauges instead of charts to display a single data value.

Gauge Type	How Do I Use It?
Linear Status Meter	To compare a current measure against another value. An inner rectangle shows the current level of a measurement against the ranges marked on an outer rectangle. The business object used to provide data for this component must provide minimum and maximum threshold values. The data also needs an update increment. See <a href="#">Getting to Know Business Objects</a> to find out more. You can find out how you can add an image in a form field or list row in <a href="#">Field Types for Lists and Forms</a>
Circular Status Meter	Show a task's progress (like a download) as a numeric value, or percent complete, within a completing circle.
LED	Show a measurement, such as a key performance indicator (KPI).
Rating	Show ratings for products or services, such as the star rating for a movie.

## Media Components

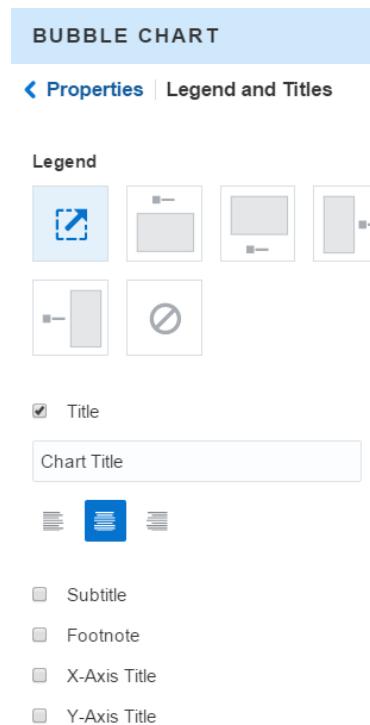
Media Type	Description
Image	You can spruce up your UI by adding an image from your data collection or from a local directory. You can images (including ones that users can replace) in various places, such as the fields of Form, Card, and List components. <a href="#">Images</a> tells you about choosing a data source for an Image component, and how to size it. You can find out how you can add an image in a form field or list row in <a href="#">Field Types for Lists and Forms</a> .
Map	Equip your mobile workforce with maps that display points of interest (POIs), such as customer locations or repair sites. To find out how to plot maps from your collection data, see <a href="#">Maps</a> .

## Configuring Graph, Chart, and Meter Components

Like the other UI components, you add graphs, charts, and meters to your screens by dragging them from the Component Palette into the Preview. The Properties Inspector displays the configuration options that are specific to the UI component at hand, such as segment labels for a funnel chart or point markers for a line chart. In addition to these component-specific properties, The Properties Inspector also displays configuration options related to a category of data visualization components. When you add a chart component, for example, the Properties Inspector displays the chart-specific options, like legends, titles, and the ability to animate the chart when users first open a screen or whenever the data that backs the chart changes.

### Legend and Titles

The Properties tab lets you add a descriptive text to your chart or gauge and also position the legends (or keys).



#### Legend

The business object that you used to add data to your graph has attributes that describe the different data sets used by the data visualization component. These attributes display as legends (or keys), which tell you what each bar, line, point, bubble, or slice represents. Using the Properties tab, you can position the legend for your graph.

## Titles

You can add a title to your UI component and also further describe it with a subtitle and a footnote. Depending on the chart style, you can also title a chart's axes or data slices.

## How Do I Create an Area Chart?

Use area charts to show trends over time, such as sales for the last 12 months.

1. In the Preview, double click **Add Component** (+ Component) to open the Component Palette.



2. Drag the area chart component ( ) from the Component Palette into the Preview.
3. Use Property tab of the Properties Inspector, to adjust the display.
4. Add data by either using the Data Quickstart or by opening the Application Catalog from the Data tab of the Properties Inspector. From the Application Catalog, choose the application that has the business object that supports an area by providing at least two sets of data.

## How Do I Create a Bar Chart?

1. In the Preview, double click **Add Component** (+ Component) to open the Component Palette.



2. Scroll to **Charts** and then drag the bar chart component ( ) into the Preview.
3. Use the Properties Inspector to configure the chart's display properties and the data it consumes.
  - a. Configure the chart's display properties: You can display bar charts in both horizontal and vertical orientations, or you can display them as a circle. For example, you can use a horizontal bar chart to show trends or compare values. You can display your data as a stacked bar chart, where the bars for each set of data are appended to previous sets of data. The size of the stack represents a cumulative data total.

**BAR CHART**

**Properties** **Data** **Actions**

**Display**

**Value Label Location**

**Animate**

**LEGEND AND TITLES**

- b. Pick a location for the label.
- c. Click **Legends and Titles** and then add a title for the chart and pick the location for the legend.

**BAR CHART**

**Properties** **Legend and Titles**

**Legend**

**Best Fit**

**Title**

**Closed Incidents**

**Subtitle**  
 **Footnote**  
 **X-Axis Title**  
 **Y-Axis Title**

4. Add data to your chart either by using the Data Quickstart or by opening the Application Catalog from the Data tab.

Keep in mind that for a bar chart, you need two different sets of data: one that displays on the y axis and one that displays on the x axis. Your development team may have created a service and business objects that are tailored specifically for your mobile app, or for general use. Check with them if you're not sure which service or business object to use.

## How Do I Create a Bubble Chart?

Use the Property Inspector to add data and configure the appearance of your bubble chart.

1. In the Preview, double click **Add Component** (+ Component) to open the Component Palette.
2. Drag the bubble chart component (  ) into the Preview.
3. Use the Properties Inspector to configure the chart's display properties and the data it consumes. See [The Properties Inspector](#) to find out more.
  - a. Configure the orientation of the chart. You can plot the bubbles along x and y axes, or as a radar chart.
  - b. Pick a location for the label.
  - c. Add a title for the chart and pick a location for the legend.
4. Add data to your chart using either the Add Data QuickStart or by clicking the Data tab and then selecting an application with the appropriate business objects from the Application Catalog.
  - a. Using the descriptions created by your development team, choose the business object that can provide your bubble chart with three data sets that it needs. Ask your development team if you're not sure which business object to use. Because a bubble chart requires three data sets, be sure to pick two fields for the y axis.
  - b. Map the response by choosing the fields for the x and y axes. Since the bubble chart requires three data sets, be sure to pick two fields for the y axis.

## How Do I Create a Donut Chart?

A donut chart, like a pie chart, uses slices (or segments) to show proportions of a whole. In a donut chart, the slices of a circle represent a contribution to an item that is named in the center of the donut.

1. In the Preview, double click **Add Data** (+ Component) to open the Component Palette.
2. Drag the donut component (  ) from the Component Palette into the Preview.
3. Use the Properties Inspector to configure the chart's display properties and the data it consumes. Typically, chart components don't trigger subsequent user interactions, so there are no options to configure in the Actions tab. See also [The Properties Inspector](#).
  - a. In the Properties tab, enter a name for the center of the donut. If you don't use the **Best Fit** option (which is selected by default), you can pick the location for the labels for these slices (inside the chart, outside of it, or no labels at all).
  - b. Choose the format for the labels: label the slices as numbers, percentages, or using names provided by the business object.
  - c. Add a title for the chart and pick a location for the legend.

4. Add data to your chart using either the Data QuickStart or by opening the Application Catalog from the Data tab.
5. Using the descriptions created by your development team, choose the business object that can provide your pie chart with three data sets that it needs. Ask your development team if you're not sure which business object to use.

## How Do I Create a Funnel Chart?

A funnel chart represents data related to steps in a process.

Within the funnel, these steps appear as vertical slices across a horizontal cone-shaped section. As the actual value for a given step or slice approaches the quota for that slice, the slice fills. Typically, a funnel graph requires actual values and target values against a stage value, which might be time. For example, use the funnel graph to watch a process where the different sections of the funnel represent different stages in the sales cycle. There are no variations of the funnel graph. The Funnel chart is used to visualize the progressive reduction of data as it passes from one phase to another. Data in each of these phases is represented as different portions of 100% (the whole).

1. In the Preview, double click **Add Component** (+ Component) to open the Component Palette.



2. Drag the Funnel Component ( ) into the Preview.
3. Use the Properties Inspector to configure the chart's display properties and the data it consumes. To find out more, see [The Properties Inspector](#).
  - a. Configure the orientation of funnel chart.



- b. Choose how each slice (or step) of the funnel will be labeled. The **Name** option renders each slice with a value from the business object. Picking **Number** labels each slice with a numeric value of the overall target. Picking **None** leaves each slice blank.
- c. Open **Legends and Titles** to pick a location for the legend.
4. Add data to your chart using either the Data QuickStart or by opening the Application Catalog in the Data tab using the Data Mapper.

Data in each of these slices (or phases) is represented as parts of the whole (which is 100%). Your business object will dictate the number and sizes of the slices in your funnel, so pick one whose properties can provide your bar chart with

the comparative data it needs (such as sales figures for the y axis and department names for the x axis).

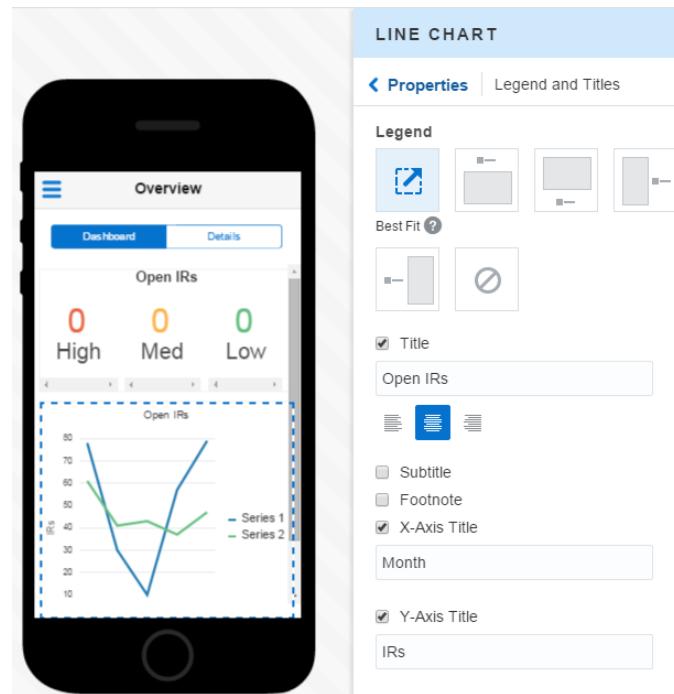
## How Do I Create a Line Chart?

1. In the Preview, double click **Add Component** (+ Component) to open the Component Palette.



2. Drag the line chart component ( ) into the Preview.
3. Use the Properties Inspector to adjust the display.

You can display a horizontal line graph, or circular line graph known as a radar (or web) graph. Use radar graphs to show patterns that occur in cycles, such as monthly sales for the last three years.



4. Add data using either the Data QuickStart or by opening the Application Catalog from the Data tab using the Data Mapper. See [Use Case: Creating a List Using QuickStarts](#) for more information.

Keep in mind that business objects that support line graphs need to provide two data sets (one for the y axis and one for the x axis).

## How Do I Create a Scatter Chart?

Scatter charts (also known as X-Y plots) depict the relationships between a number of items. Use scatter charts to show correlations between two different kinds of data values, such as sales and costs.

1. In the Preview, double click **Add Component** (+ Component) to open the Component Palette.



2. Drag the scatter component ( ) from the Component Palette into the Preview.
3. Use the Properties Inspector to configure the chart's display properties and the data it consumes.
  - a. Configure the orientation of the chart. You can plot the points along the x and y axes, or as a radar chart.
  - b. Pick a location for the label.
  - c. Add a title for the chart and pick a location for the legend.
4. Add data to your chart using either the Data QuickStart or by opening the Application Catalog in the Data tab.

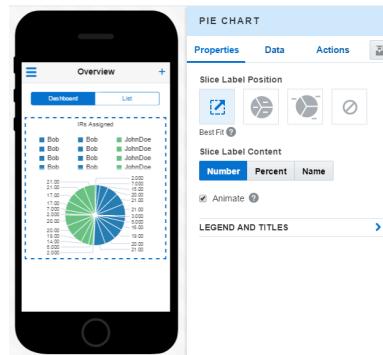
## How Do I Create a Pie Chart?

Use pie charts to show the relationship of parts to a whole, such as how much revenue comes from each product line.

1. In the Preview, double click **Add Component** (+ Component) to open the Component Palette.



2. Drag the pie component ( ) into the Preview.
3. Use the Properties Inspector to configure the pie chart's display properties and the data it consumes.
  - a. In a pie chart, the data items are displayed as sections of a circle causing the circle to look like a sliced pie. In the Properties tab, if you don't use the **Best Fit** option (which is selected by default), you can pick the location for the labels for these slices (inside the chart, outside of it, or no labels at all).
  - b. Choose the format for the labels: label the slices as numbers, percentages, or using names provided by the business object.



- c. Add a title for the chart and pick a location for the legend.
4. Add data to your chart using either the Data QuickStart or by opening the Application Catalog in the Data tab.

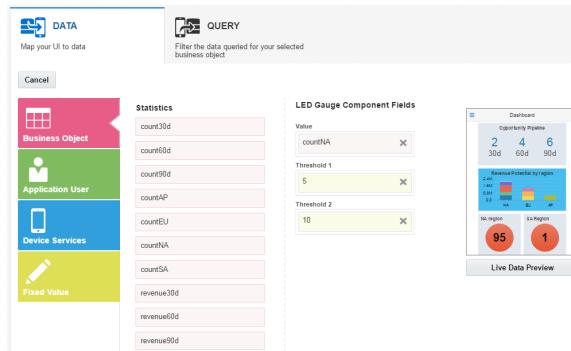
## How Do I Configure Meters and Gauges?

Because you use gauges and meters to measure a quantity or express a state, you need to define thresholds for them. Adding thresholds allows you to use colors to show various states, like high, medium, and low.

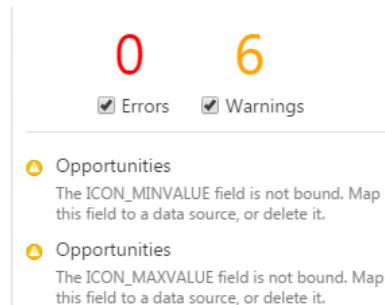
Using an LED Gauge component as an example, you can add thresholds as follows:



1. Start off by dragging an LED Gauge component (  ) into the Preview.
2. Using the Data Mapper or the Add Data QuickStart, wire the component to a data source that returns a numeric value.
3. Map a field from one of the data options (like **Data Source**, **Business Object**, or **Fixed Value**, for example) to the Value field. For the LED Gauge component, this field value is considered the maximum value. The Data page for other components, like the Circular Status Meter, include minimum and maximum value fields along with those for setting the thresholds.



You'll get a warning if you don't define all of the values for a meter. While you can ignore these warnings, keep in mind that values and colors make your gauges easier to read and more informative.



4. Use the Data page of the Properties Inspector to configure the thresholds. You'll add them from the lowest to the highest.
  - a. Click **Add** (  ).
  - b. Use the color picker to assign colors appropriate to each threshold, like red (minimum), yellow (medium), and green (maximum).

c. Add values for the minimum and medium thresholds. Because the value mapped for the gauge is considered the maximum value, you don't add a maximum threshold value here. You only need to assign a color.

## Screens

Use the Properties Inspector to change the appearance of the screen component.

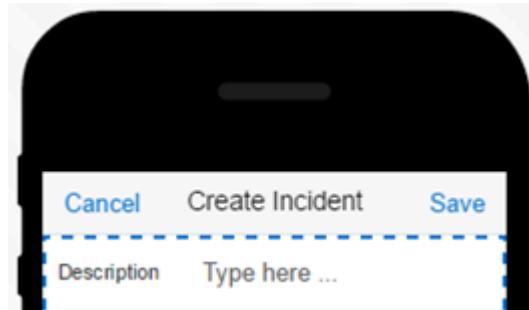
The Screen component configuration properties let you:

- Change the screen title.
- Choose another screen template.
- Add or hide a summary field.

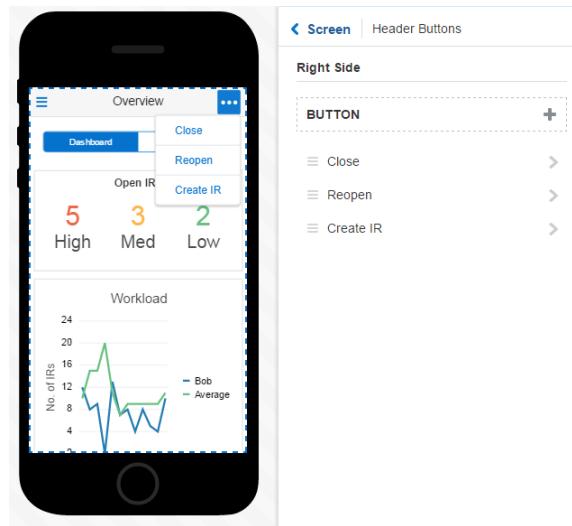
## Screen Actions

You can expand your application flow by creating your own screen actions.

The Mobile Designer's screen templates already have functioning actions within the screen header, like the application menu (≡), the **Back** action (⟲), and the **Cancel**, **Save**, **Create**, and **Edit** actions.



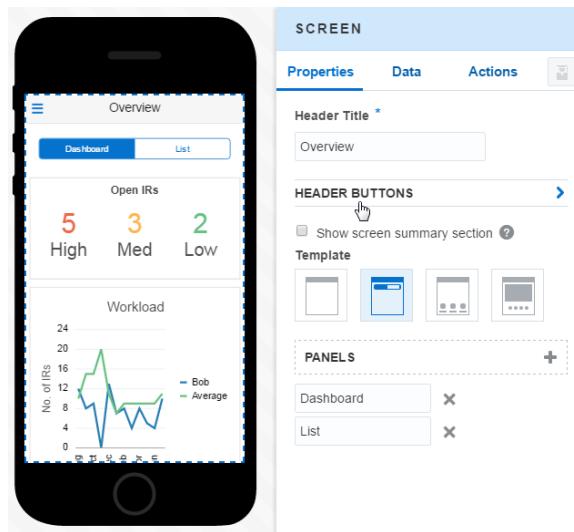
You can't change these standard actions; their appearance, names, and behavior are baked in. If your app requires its users to perform some specific task, then you can add your own action using the Properties Inspector for the Screen component.



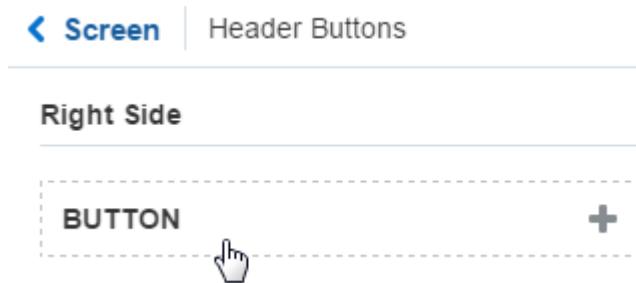
## Adding Screen Actions

In the Properties Inspector for the Screen component:

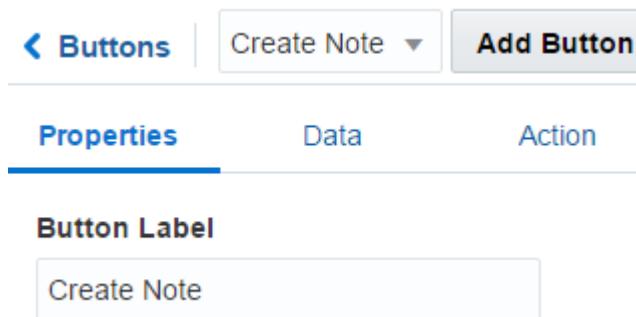
1. Click **Header Buttons**.



2. Click **Add Button**.



3. Enter a name in the Button Label field.



4. Click the Actions tab in the Properties Inspector.

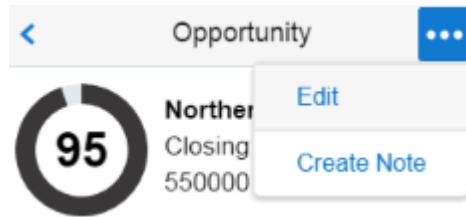
Because you can't map data to a button, you'll use only the Property and Actions tabs when creating and configuring a button.

5. Click **Add Action (Tap Button)**.

6. For the button's trigger action, choose **Tap** (  ). See [Adding Actions](#) to find out more.

7. Choose from among the navigation or business actions in the Configure Action panel. Depending on what you want this action to do (open a screen or send some value back to the service), you can use the Navigation Data Mapper or the Business Action Mapper. To find out more about these tools, see [Adding Actions](#).

All of the actions that you create display on the right side of the screen header. If this is the first action that you've added to a screen that already has an action there (like an Edit action), then the Mobile Designer accommodates them both by creating a menu.



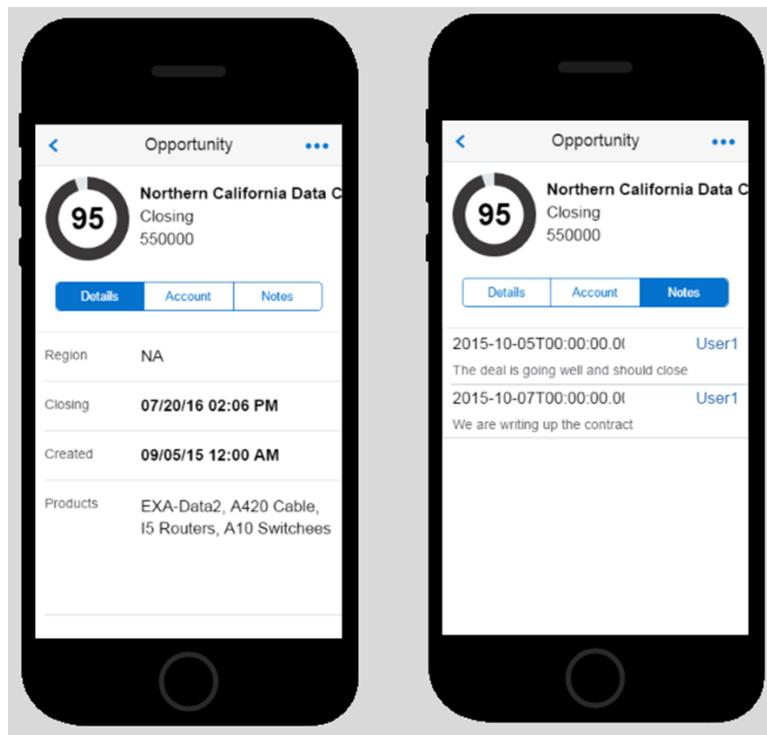
 **Tip:**

**Add Button** () appears in the Properties Inspector after you add your first button. You can add as many buttons as you need. Use  to reorder them. Click **Delete** () to remove buttons that you don't need.

## Screen Summary

You can enhance your screens by adding a summary panel above the main content area.

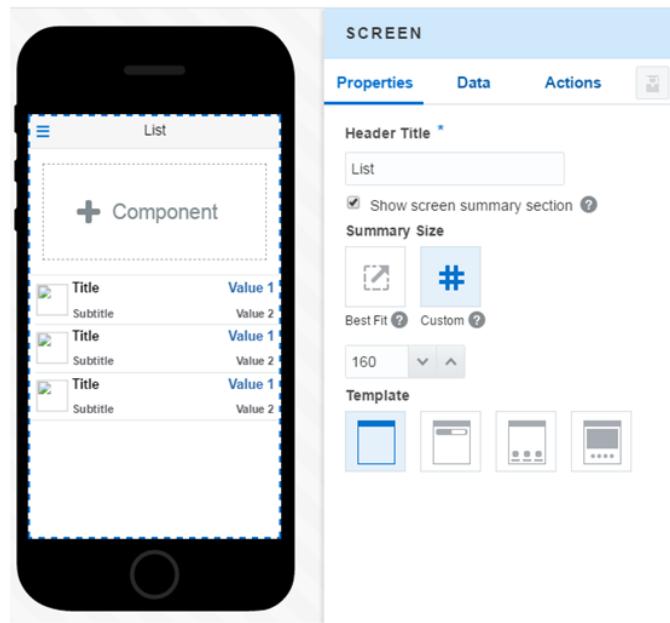
In terms of display, the summary panel is a constant feature in the screen, one that doesn't scroll. By using the summary panel to present salient information, for example, you can create a point of reference for users while they view more detailed information in the screen's tabs or pages.



You can add a summary to the top of any screen, including the landing screen of your app. If you didn't choose a summary panel when you created a screen, you can add one by selecting the **Show screen summary section** option in the Properties Inspector. After that, you can populate the panel by dragging and dropping a component into the Preview. If needed, you can then bind the component to data using the Add Data QuickStart or the Data Mapper. To find out more, see [Enabling Your App to Display Content](#).

 **Tip:**

Whether you added summary panels at the outset, or added them later on, you can hide them by clearing the **Show screen summary section** option.



### What Type of Components Can I Add to a Summary Panel?

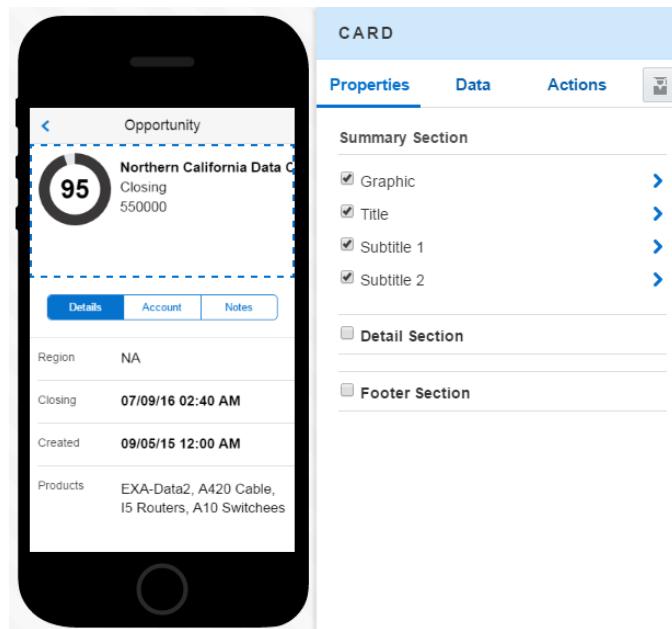
You can add any type of component that you like to a summary panel, but you can add only one component. If a single component won't suffice, you can add more by adding



a layout component, like the card ( ) to the summary panel.

#### Tip:

Because the card component re-sizes itself to fit the summary panel, you can fill the panel by populating the card component's Summary and Detail fields.



### Sizing a Summary Panel

Some components have a fixed height, others don't. By selecting the **Best Fit** option



(), the Mobile Designer resizes the height of the summary panel to accommodate the component that you've added to it. For this option, the Mobile Designer sizes the component using a 4:3 aspect ratio, which is based on screen width. While you can let the Mobile Designer resize the summary panel for you, you can do this yourself by first



choosing the **Custom Fit** option () and then entering the height, in pixels.

#### Note:

Keep in mind that this option keeps the summary panel at the height you've specified, which means that some components might not fit. And because the summary panel doesn't scroll, they won't display properly.

## Dashboards

You can give your users a bird's-eye view of the information they need by creating assemblings metrics, KPIs and data visualization tools into a single screen: a dashboard.

Typically, you'd position a dashboard as the first screen in your mobile app to give users a quick glimpse of their stats. The Mobile Designer gets you started with a dashboard when you pick the dashboard template, or when you add a dashboard component to a blank page.

Your dashboard might be geared to sales, health care, or field service requests. The Mobile Designer provides you with components for any use case. For example, for a sales organization, you might dress up a dashboard with a group of line and bar

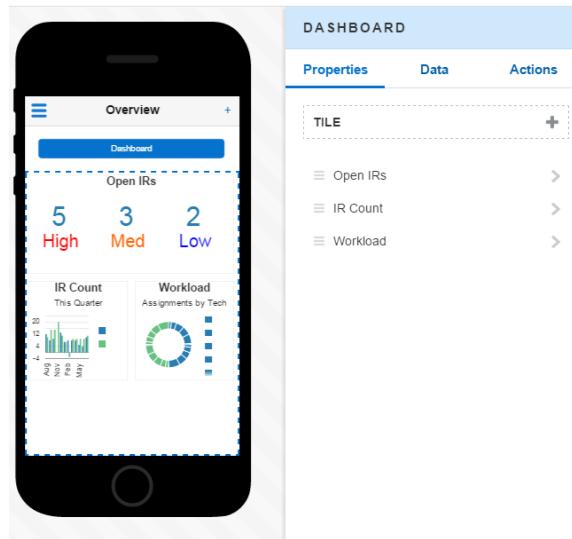
charts. A repair technician in the field might not need to see a bunch of charts. Instead, she might need a set of metrics for the open, closed, and pending IRs assigned to her. She might also want to see a line chart that compares her workload to that of other field repair techs. You can fashion a dashboard from the Mobile Designer component palette to suit your mobile use case as long as you have the business objects to back it up.

## Working with Tiles

Think of tiles as the foundation of your dashboard. You can't add any UI components without them.

### Important:

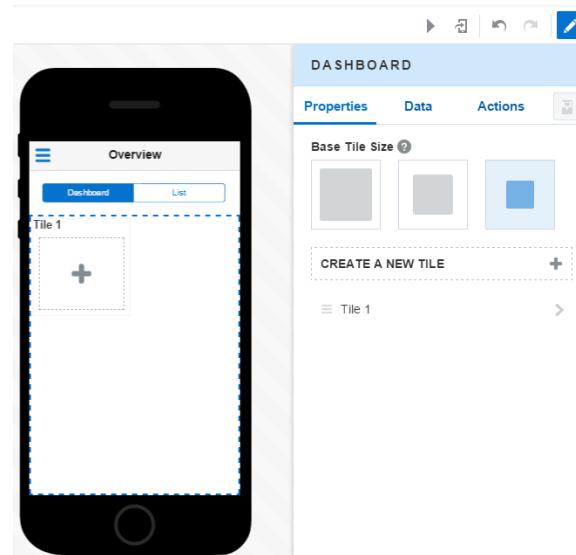
Before you can add metrics or other UI components to a dashboard, you first need to landscape it with tiles.



Once you've added tiles to a dashboard, you then add UI components to the tiles themselves. Tiles are the containers for the UI components: they anchor the UI components in place, enabling them to maintain their positions in both landscape and portrait orientations.

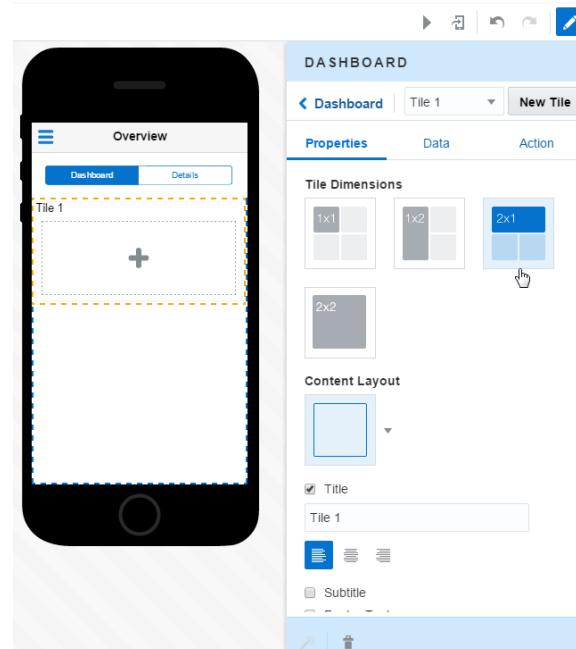
### Getting Started with Tiles

After you create a dashboard (either by selecting it as your first page or by dragging the dashboard component into the Preview), the Mobile Designer has a tile ready for you. But before you add a UI component to it, you might want to think about how you'll create the different regions of the dashboard using tiles of different shapes and sizes. And you might also want to think about how the UI components will take up the screen real estate.

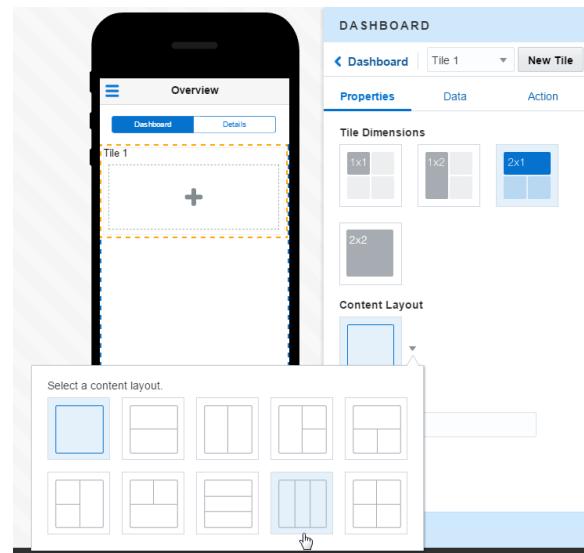


To begin, set the size for all of the tiles that comprise your dashboard. You can choose small, medium, or large (small is selected by default) from the **Base Tile Size** options. Because you're setting this for the dashboard component (and not for an individual tile), the size that you select here applies to every tile. As a consequence, large tiles might sprawl and can't be accommodated by the layout that you later select for an individual tile.

With the base size set, you choose tile shape and position that best suits your UI component by selecting from among the **Tile Dimension** options.

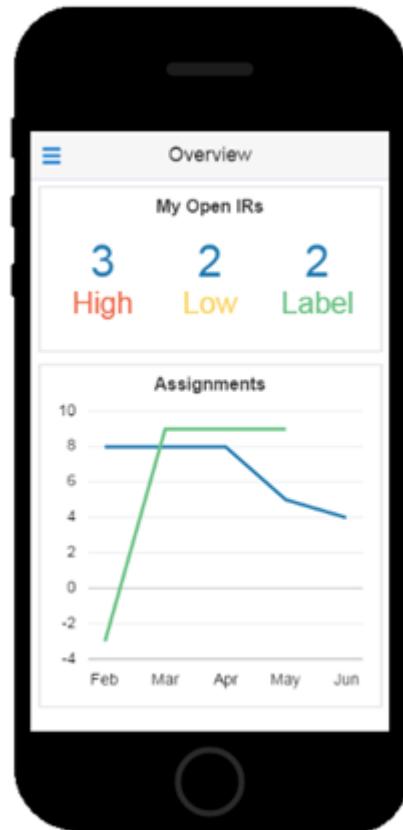


Finally, you'll choose the layout within the tile itself from among the **Content Layout** options. You can also opt to label your tiles with titles, subtitles, and footers, or pick a background color to enhance the UI component.

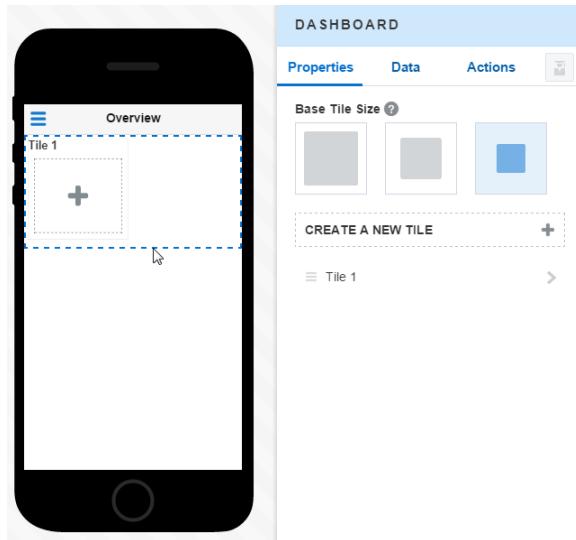


## Use Case: Creating a Dashboard

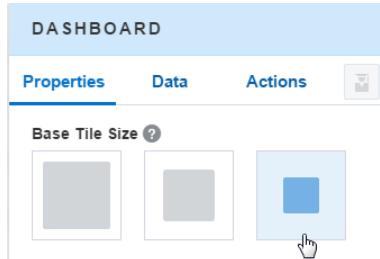
This use case shows you how to create and layout the tiles for a dashboard that has a set of three counters (or metrics) positioned above a line graph.



1. To get started with setting the basic size and position of the tiles, click the dashboard component in the Preview.

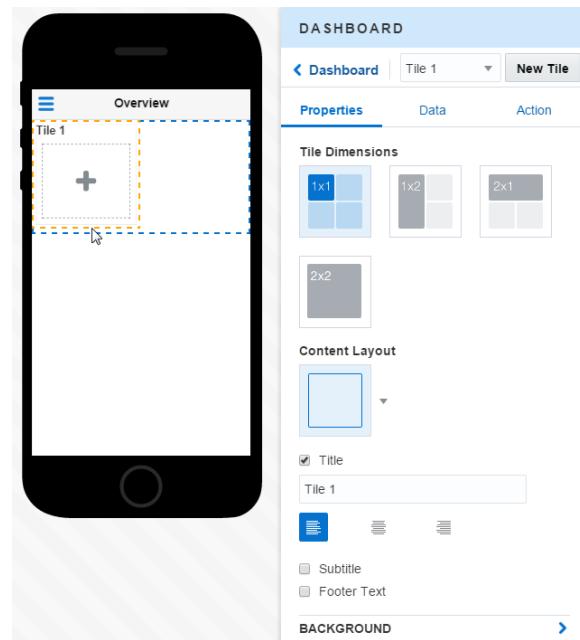


2. In the **Base Tile** section of the Properties Inspector, choose **small** as the size for all of the tiles used in this dashboard.

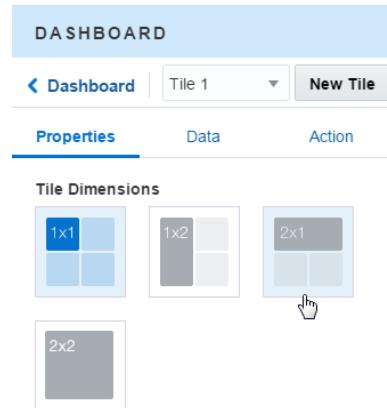


To get you started, the Dashboard component has a default tile called *Tile 1*. With the base tile size set, you're now ready to use the Properties Inspector to rename this tile, position it in the dashboard, and choose its layout.

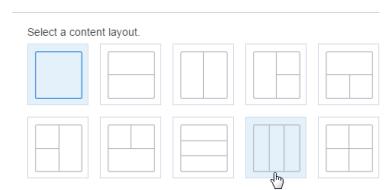
3. To open the Property Inspector, click the Tile 1 component in the Preview, or open Tile 1 in the Properties Inspector.



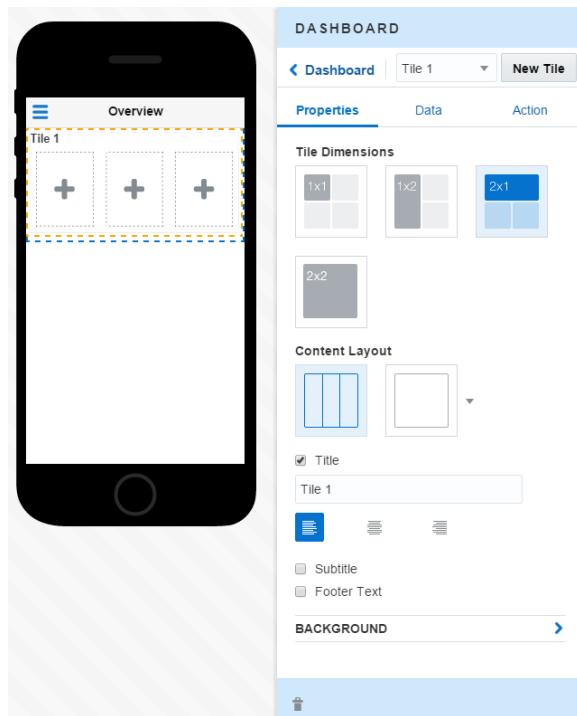
4. Because the metric counters occupy the top half of the screen, choose **2 x 1**.



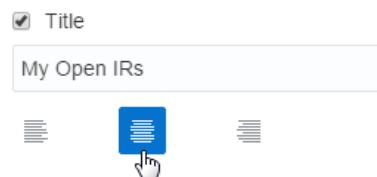
5. Because you'll include three counters (high, medium, low) that display from left to right, pick the layout option that segments the tile into three areas, **Layout 9**.



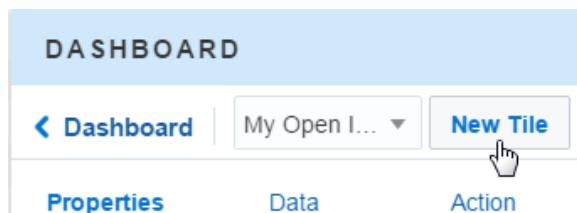
Choosing this option multiplies the tiles accordingly – you now have a tile for each counter.



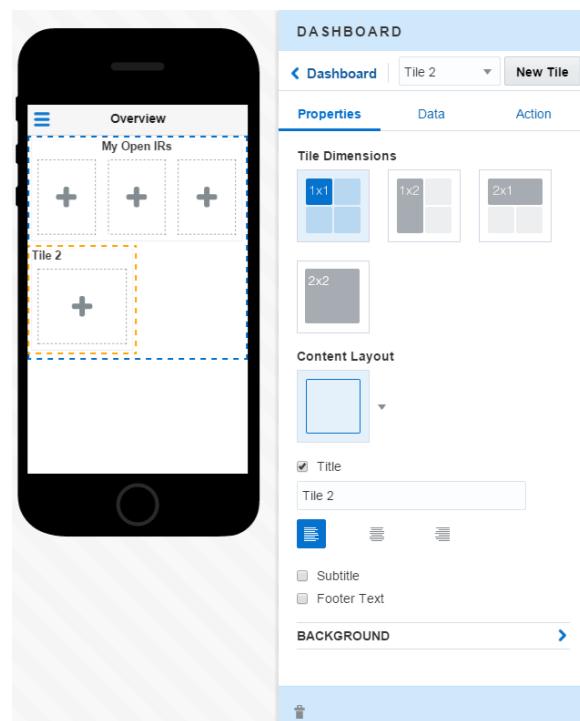
6. Change the name of Tile 1 by entering a new one (like *My Open IRs* ) in the Title field. Then center it over the tiles that will house the Metric components.



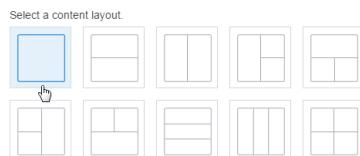
7. Next, create a tile to house the line graph by clicking **New Tile**.



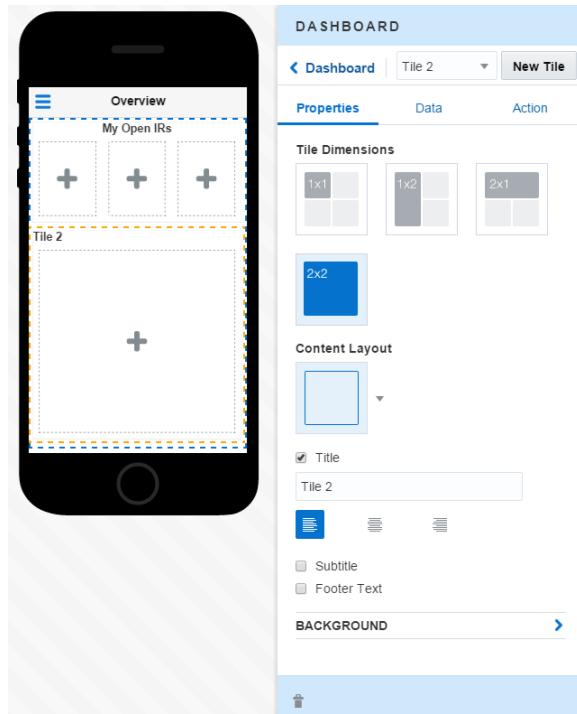
The Mobile Designer creates Tile 2.



8. Because the line graph will occupy the remainder of the dashboard, choose **2 x 2** for its dimensions.
9. Tile 2 will hold only the line graph component, so choose the plain layout provided by **Layout 1**.

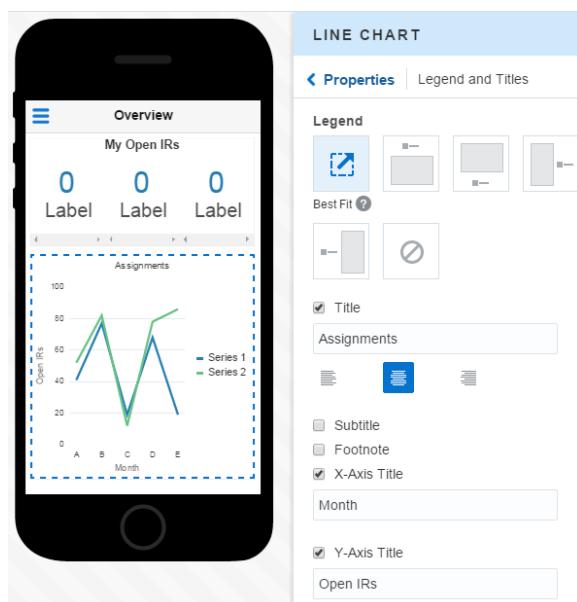


The dashboard might look something like this:

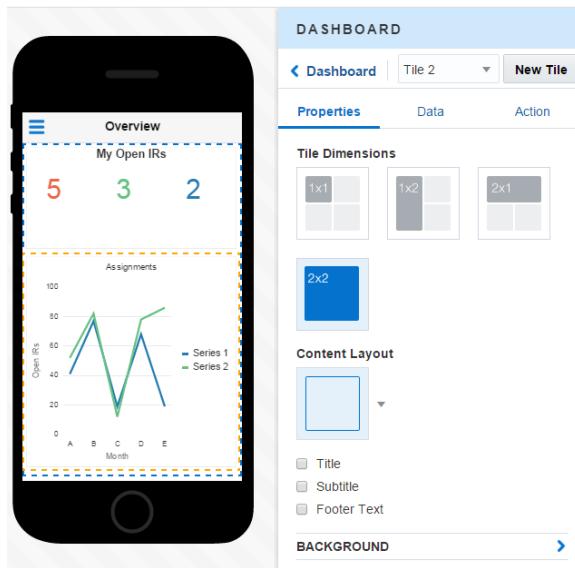


10. The title is an optional component. The business object that backs a UI component might already provide enough information to make a title redundant. That's the case here, so clear the **Title** option ( **Title**).
11. With the tiles configured, the UI components now have a place to go:
  - a. Drag a Metric component into each of the tiles at the top of the dashboard.
  - b. Drag the Line Chart component into its tile at the bottom of the dashboard.

After you've added all of the UI components, the dashboard might look something like this:



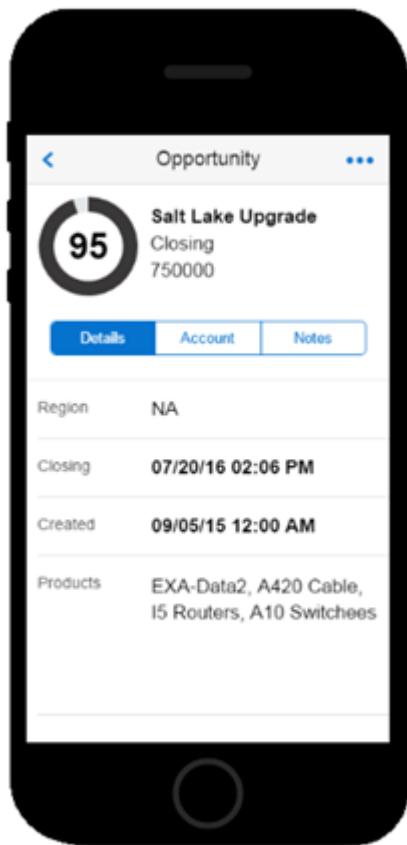
12. After you've mapped the UI components and configured their display properties, the dashboard should look something like this in the Mobile Designer.



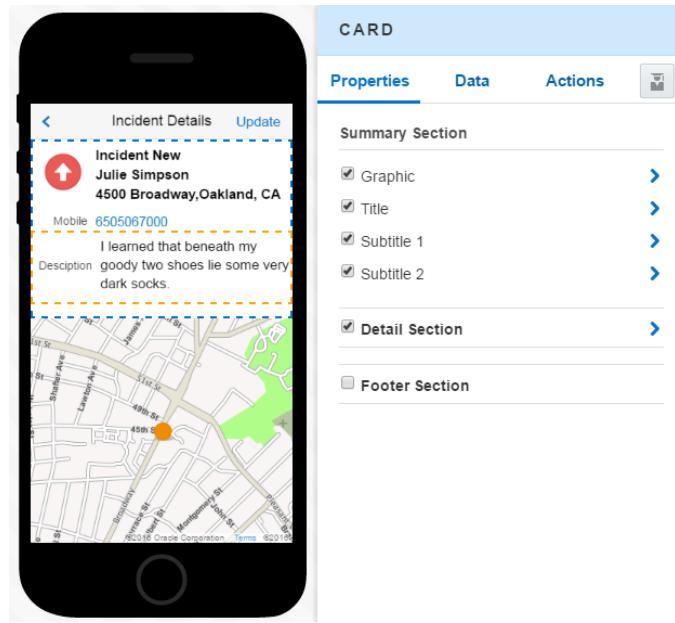
## Cards

Use the card component if you want to display summary information within a small pane.

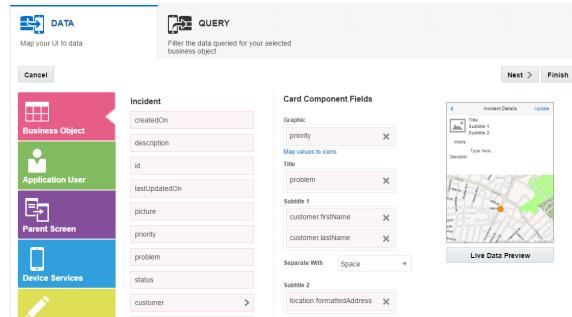
The card component is a small container for displaying various combinations of data visualization tools, images, and text. You might use this component to create a business card, but you can also use it to summarize an item from a larger data set, or provide cursory information about a physical object.



You can drag the card component into different types of containers, such as a tile or a screen's summary panel. Because it has no size of its own, the card component adjusts to the size of the container that holds it. Even though this component is flexible, keep in mind that the Mobile Designer truncates the card's content if you drag it into a container that's too small. To get an idea of how the component resizes itself, take a look at [Screen Summary](#). The card is divided into three content areas: the summary section, the detail section, and the footer section. Each of these sections is comprised of fields, like a form component, which you can add or delete. With a few exceptions, these fields are, by default, output text. Like the fields in a form component, you can change their type. To find out more, see [Field Types for Lists and Forms](#).

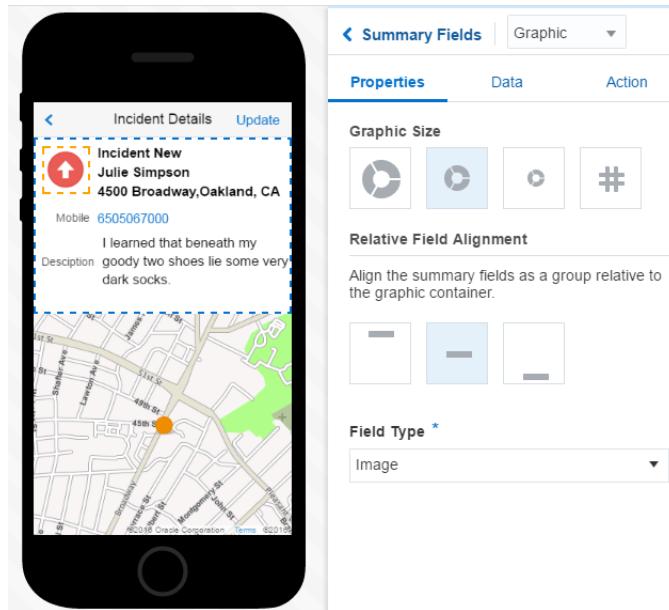


You can hide or display fields by toggling the various field and section options in the Properties Inspector. Only the fields that currently display are available in the Data Mapper.



### The Summary Section

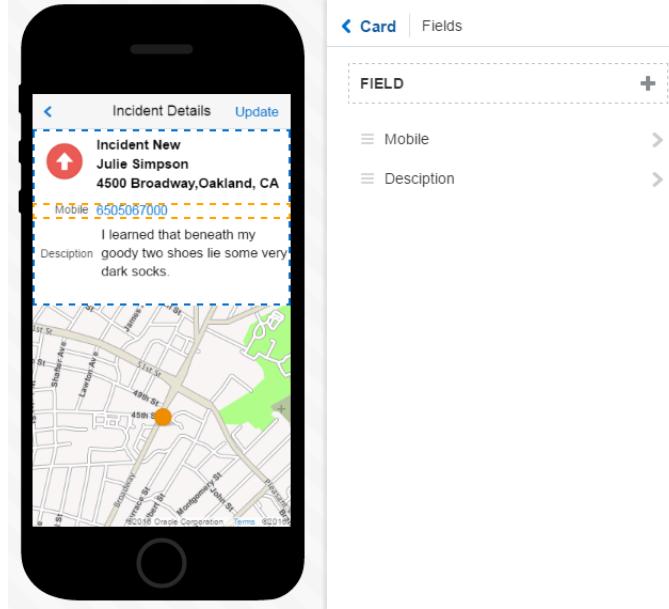
The Summary section includes the graphic container, an optional field for displaying an image or data visualization component, like status meters, ratings, or LED gauges. You can use the Properties tab to change the size of the graphic container and adjust the alignment of the adjacent fields.



The Summary section also includes the boldface Title field, which is output text-only (you can't change its field type). You can select any combination of the fields in the Summary section, but you can't add any additional fields, nor can you rename these fields. Instead, you add field names by data mapping.

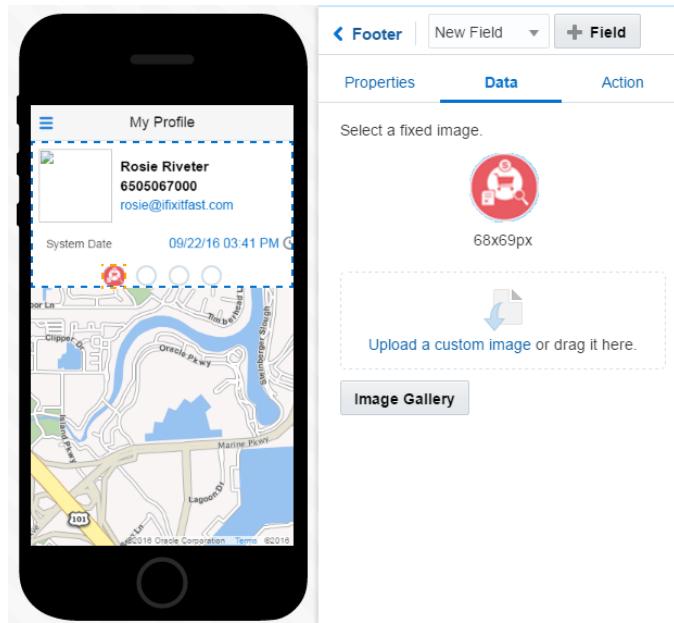
### The Details Section

Using the detail screen, you can augment the card by adding fields. You can name these fields yourself (or rename them, depending on the mapping) and also change the field type.



### The Footer Section

The footer section enables you to add up to six controls appear below the card content. By default, they are centered, but you align them to the left or right as well. These controls are exclusively image fields (meaning that you can't change them to another type) and the labels that you enter for them display as alt text at runtime. You can't use data mapping to add an image to the control container. Instead, you upload custom images using the Data tab. You can size the image to the container, or set it to its original size.



## Lists

Using the Properties Inspector for the List component, you can choose the types of fields that best suit your data.

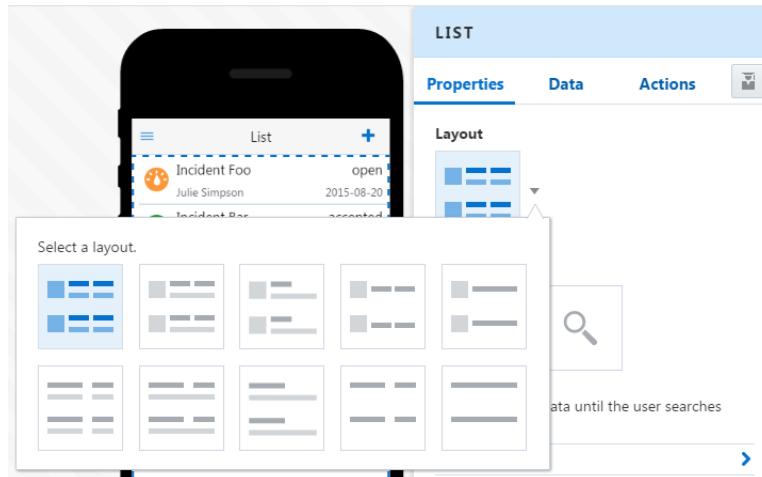
### List Layouts

Each row of the default list template has a group of fields called *Graphic*, *Title*, *Subtitle*, *Value 1*, and *Value 2*.

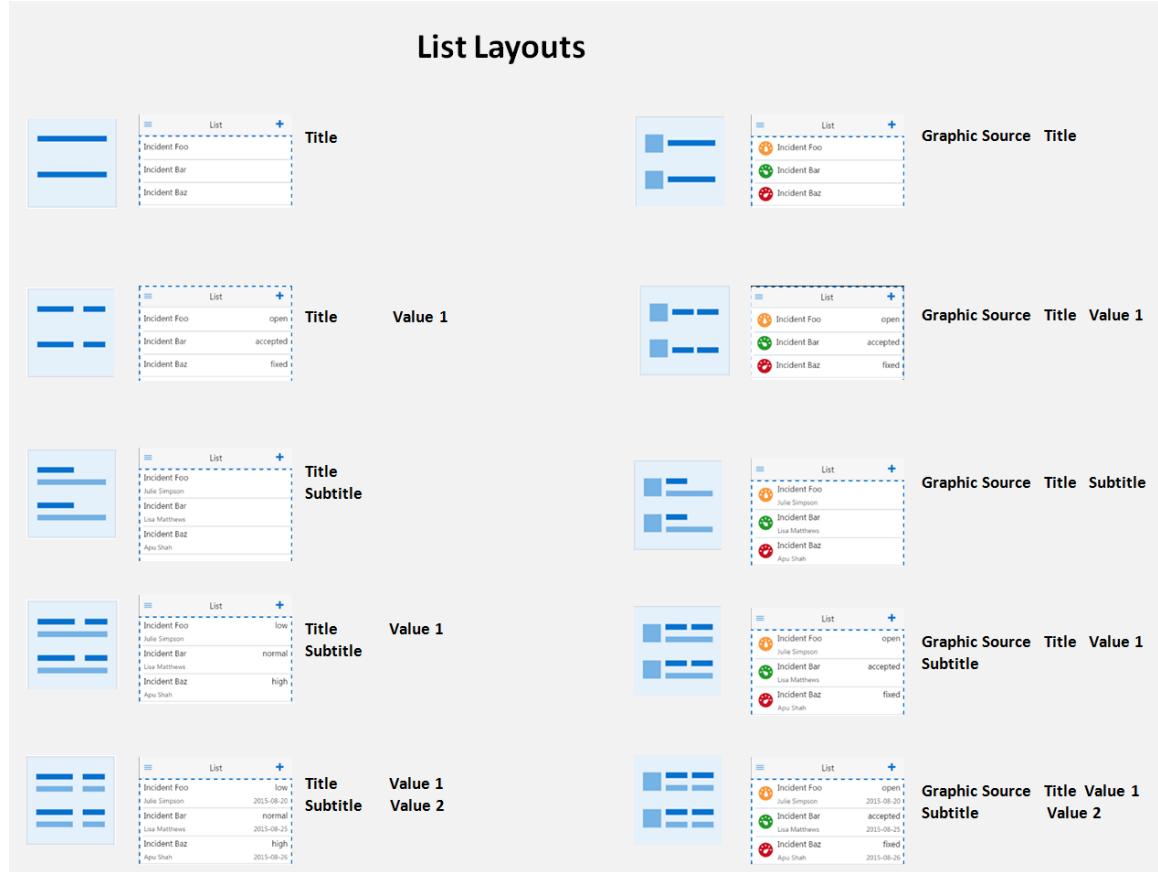
List		
	Title Subtitle	value 1 value 2

If your business object can populate each of these fields with data, then you can stick with this arrangement. But if, say, your business object doesn't provide images for the graphic field, or you just want to pare down the number of fields, then you can pick a different layout that better suits for your needs.

You can pick a layout using Add Data QuickStart wizard or the Properties Inspector. To find out more, see [Use Case: Creating a List Using QuickStarts](#).

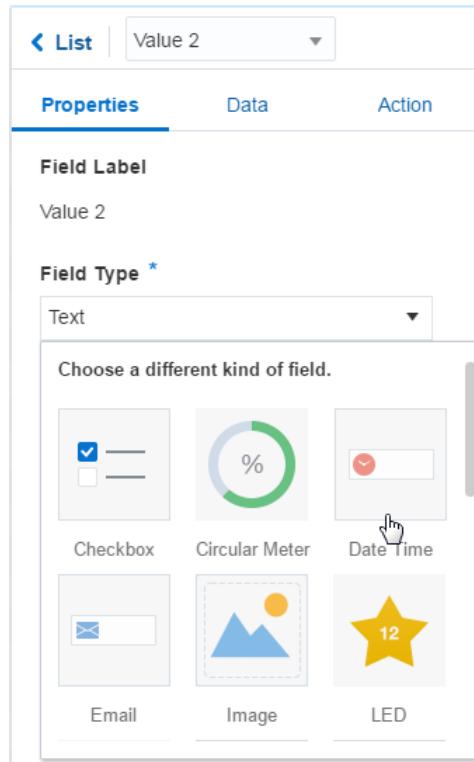


Each layout option is a variant of the default list template. The Title field is the only required field for a list. All of the fields are output text fields, which means that they are read only. If you'd like to change the format for these fields, see [List Field Types](#).



## List Field Types

The default format for the List component organizes each row into a set of output fields that are read-only for app users. As described in [List Layouts](#), each row of the list is a combination of the Graphic field (on the left) and the Title, Subtitle, Value 1 and Value 2 text output fields. If you're setting up a task list with text fields, then the default field type settings cover your needs. If not, you can change them by selecting from among the component options in the **Field Type** list in the Properties Inspector. Data might not display correctly (or in some cases, at all) when the field type clashes with the data mapping. When this happens (say, when you map a field that returns an integer value to a field that expects an image), you can quickly remedy the situation by switching to an appropriate field type. You can find out more about the **Field Type** menu and its component options in [Field Types for Lists and Forms](#).



## Adding a Search Field

If your application supports searching, then you can add a search field for a List component.

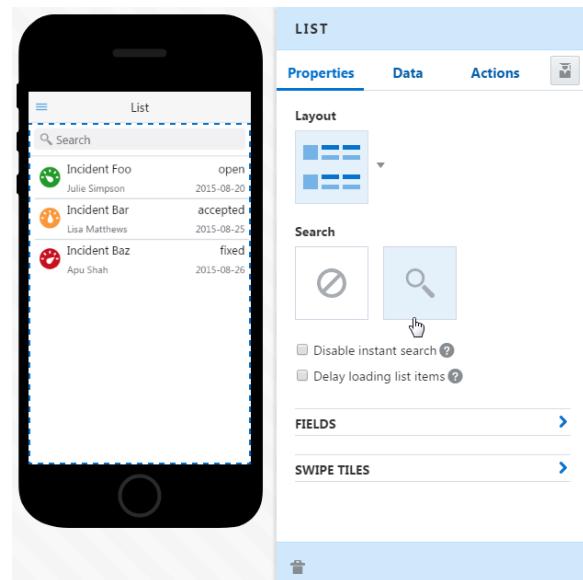
To add a search field:

1. Select the List component.
2. In the Properties Inspector, choose **Always Visible** (🔍) to add the search field.

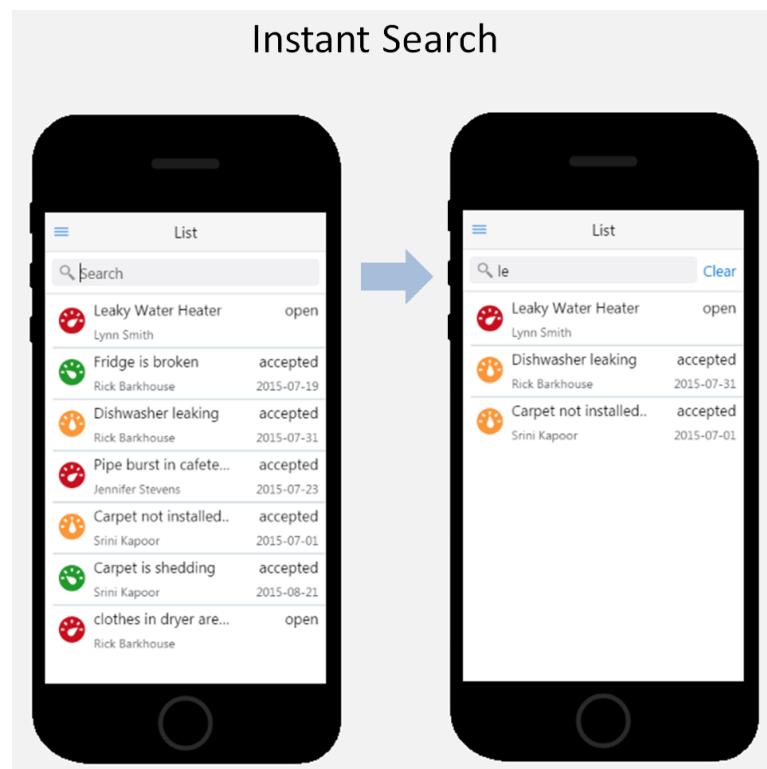
The alert (Data 1) appears because you need to map a query parameter to the search field.

### Tip:

To hide the search field, choose the default option, **None** (∅).

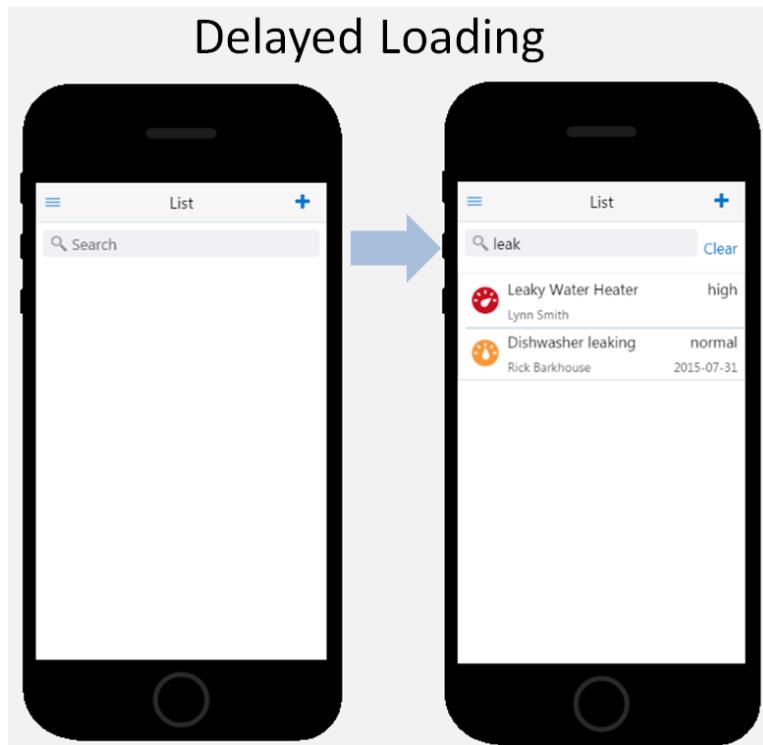


3. By default, your app uses the Instant Search mode, where the search field appears above a list that contains all of the records returned by your service. Users don't need to tap a search button to filter this list. Instead, the app executes the search as they type their entries into the search field. Instant Search is best-suited for small data sets whose limited size won't compromise the app's performance or require users to scroll continuously as they browse the list.

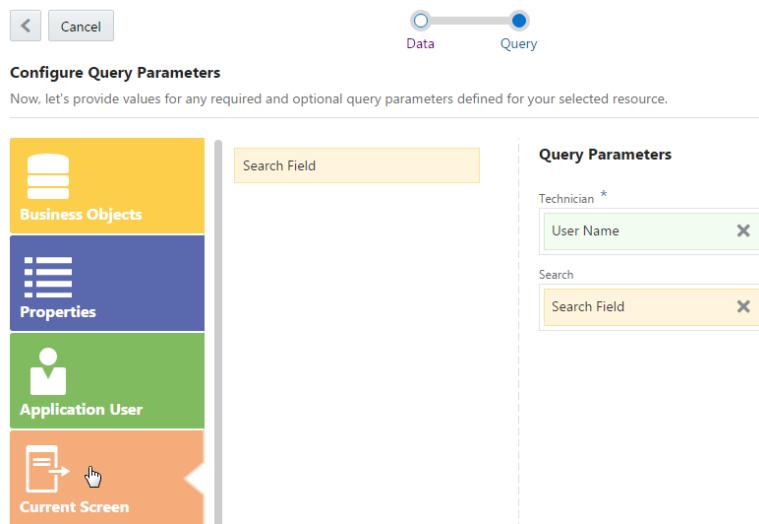


To preserve the user experience for large, or even moderately-sized data sets, consider choosing **Disable instant search** and **Delay loading list items**. For a mid-sized data set, use the **Disable instant search** option to suppress the default behavior and instead allow your app to load the list items before users can start

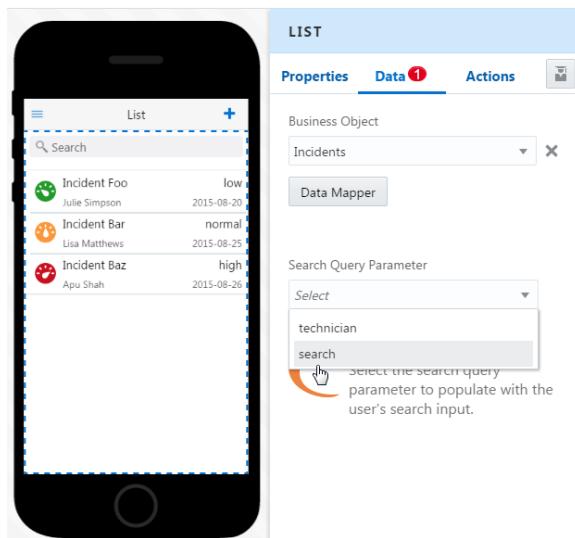
searching. When you choose this option, users can only narrow down the list by tapping Enter or the search button. If your service returns a lot of data, choose **Delay loading list items** so that your app doesn't have to load all of the data at the outset. This option enables your app to display only the list items that match the search criteria. By default, the app displays the search results after users stop typing in the search field, so they don't have to tap Enter. To allow users to see their search results only after they tap Enter, combine the **Disable instant search** and the **Delay loading list items** options



4. Bind the list to data. To find out more about the Data Mapper, see [Taking a Look at the Data Mapper](#) and [Use Case: Creating a List Using QuickStarts](#). In the Query page of the Data Mapper:
  - a. Click **Current Screen**.  
The Search Field value appears.
  - b. Next, choose the query parameter for the search field. To do this, drag the Search Field value into a field in the Query Parameters section of the page.  
The query parameters, which are created by your development team, are specific to the business object that you used for the List component. In the Fix-It-Fast example, the development team created a query parameter called *Search*. By adding the Search Field value to this parameter, you allow the app user to search the list of open incident reports by technician name.
  - c. Click **Finish**.



Clicking the Data tab in the Properties Inspector shows the search in the Search Query Parameter field.

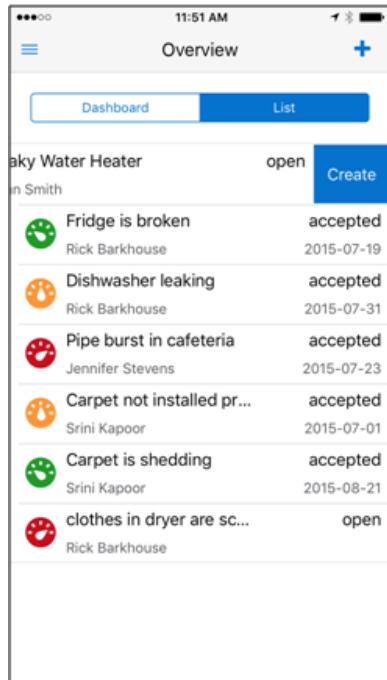


If you delete this value and then map this field to another query parameter, be sure to verify your mapping by checking the Query page of the Data Mapper. The Mobile Designer populates the Search Query Parameters field with the search value if you add a search field after you've already mapped the list to data.

## Adding Swipe Gestures

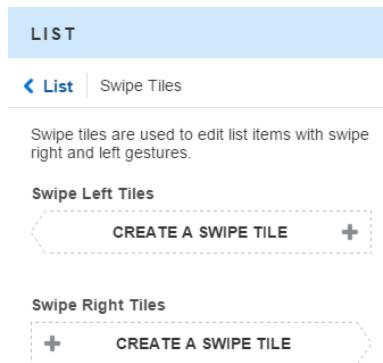
You can enable app users to update lists using left- and right-swipes using the Properties Inspector for the List component.

A swipe gesture can be made from any row in a list and can trigger different actions, such as navigation to another screen or access to a device service like the phone. A swipe can also trigger a custom action created for the mapped business object.

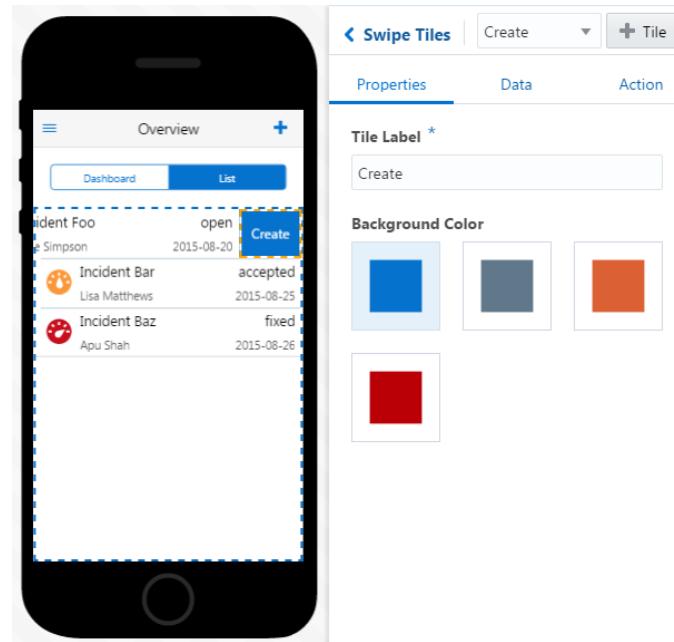


To add a left or right swipe gesture:

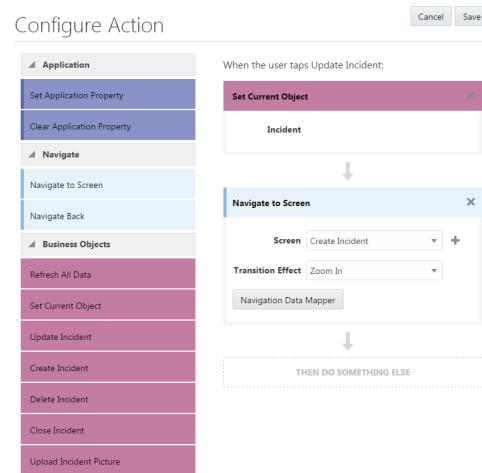
1. Select the List component.
2. In Properties tab, click **Swipe Tiles**.
3. Click **Create a Swipe Tile**.



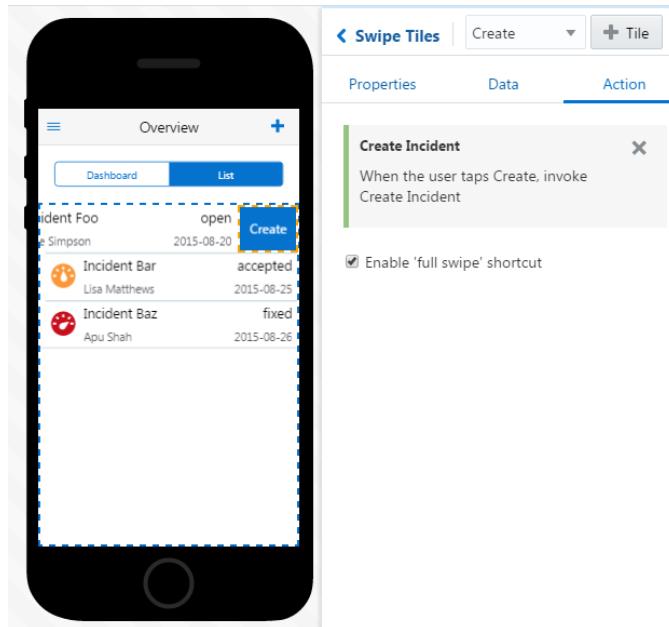
4. Add a label and pick a color.



5. Open the Action tab of the Properties Inspector, click **Add Action (Tap Tile)**, then add the action appropriate to the tile.



You can pick from the business object-specific actions (like refresh, create, update, delete, etc.), and if needed, pick a navigation option (navigate back, or navigate to screen). The Navigation Mapper lets you choose the destination screen and a transition effect.



If your swipe tile action doesn't navigate to another page, but instead updates a value, you can use the Navigation Data Mapper to pass a parameter from one screen of your app to another.

You can add up to three swipe tiles for each side of the list rows (three right-swipe, three left-swipe).

**Swipe Left Tiles**

No more tiles will fit on this side.

Update	>
Refresh	>
Close	>

**Tip:**

Use the **Move** icon (≡) to rearrange tiles.

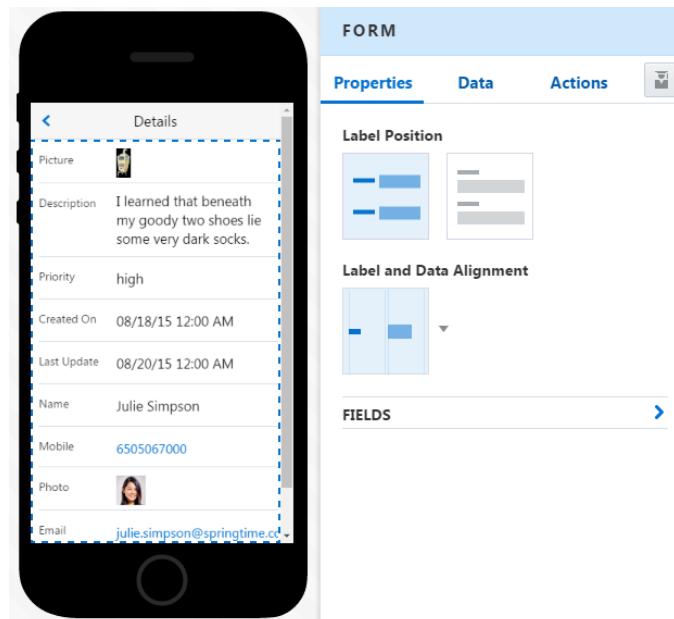
## Forms

Forms enable you to create screens with fields that can accept user input, or can be read-only.

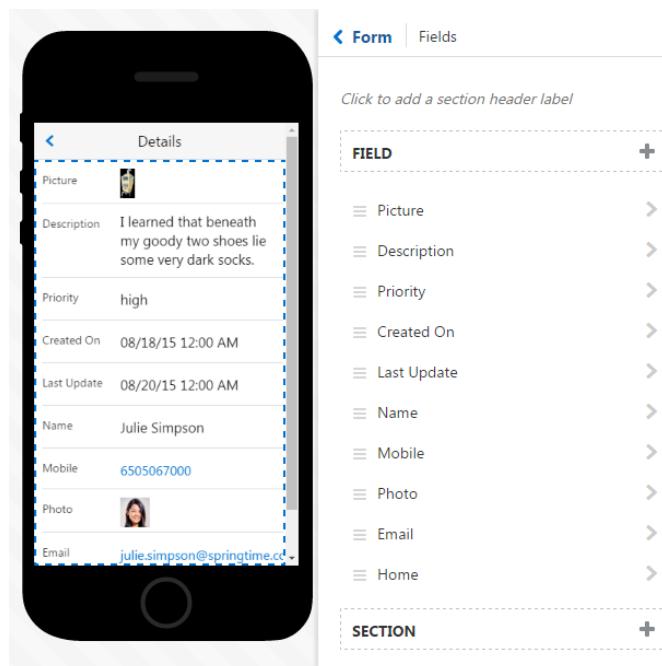
You can use the Form component in different ways. As you'll see in [Use Case: Creating a List Using QuickStarts](#), the Form component is flexible: it's the framework for both the record detail screens that can only display data and the edit and create screens that have combinations of read-only and editable fields. Unlike a List

component, the Form component is not a fixed pattern for fields and images. You can add as many fields (and field types) as you need. Your form's content depends on your service. See [Mapping Field Values to Forms](#).

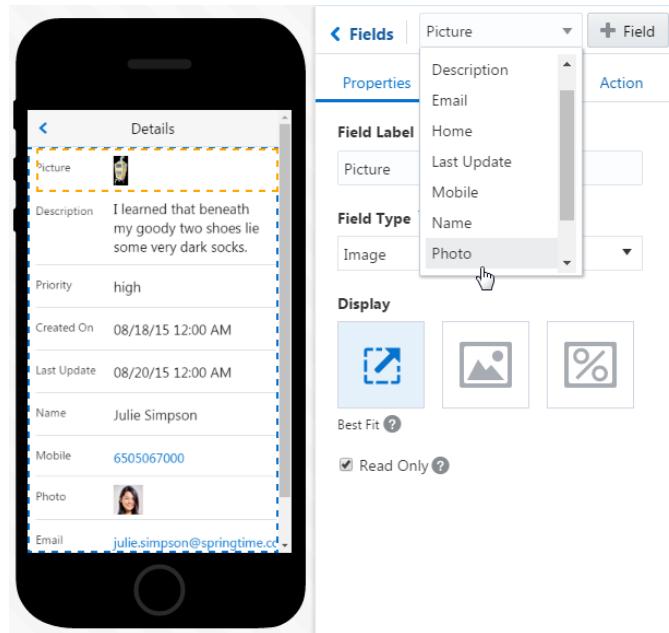
From the Properties page, you can choose a layout option for your form.



The **Fields** option in the Form component's Properties Inspector is your top-level view of the fields that you've added to your form.

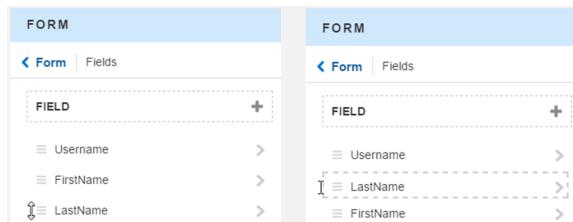


You can manage your fields from here by reordering them, or drilling down to configure the display, behavior, data, and actions for a particular field.

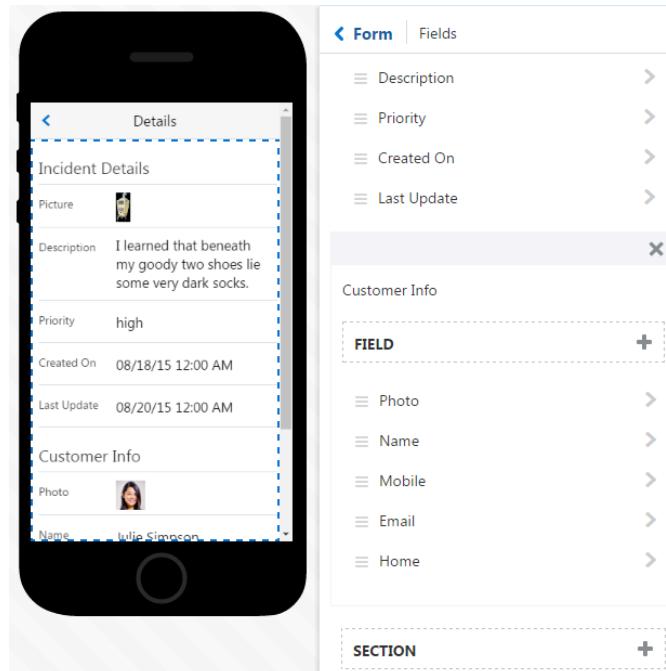


In the Fields page you can:

- Reorder the form's fields—Use the **Move** icon (≡) to reorder fields.



- Add a field—You can populate form fields by dragging and dropping business object fields in the Data Mapper, or by clicking **Add Field** (+) in the Fields page, or the **Add Field** button (+ Field) in the Properties Inspector for an individual field. After you add a field, the Mobile Designer displays a badge over the Data tab (Data 1), a signal that you need to map a value. To find out more, see [Mapping Field Values to Forms](#).
- Organize your fields into sections—Click **Add Section** (+) in the Fields page, enter a section name, and then organize the fields within the section using **Move** (≡). You can move fields between sections.

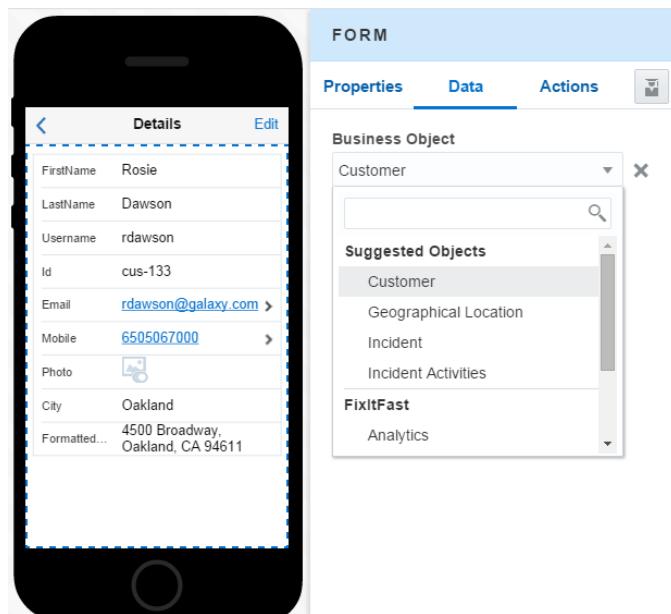


- Delete a field—Select a field and click **Delete** (  ). You can also delete the entire form (and all of its fields) by clicking **Delete** in the Properties Inspector for the Form component.

## Data Mapping for Forms and Form Fields

The Data tab lets you map a business object to the Form component itself and also change the values for the forms fields.

The Data tab shows the business object that is currently mapped to a Form component. Using the Business Object list, you can explore the business objects that belong to the application that backs your app, or choose another one. To help you map data for a detail screen, the Mobile Designer organizes the business objects chosen for the parent screen along with their child and reference objects, into the **Suggested Objects** section of the list. To find out more about child and reference business objects, see [Getting to Know Business Objects](#).

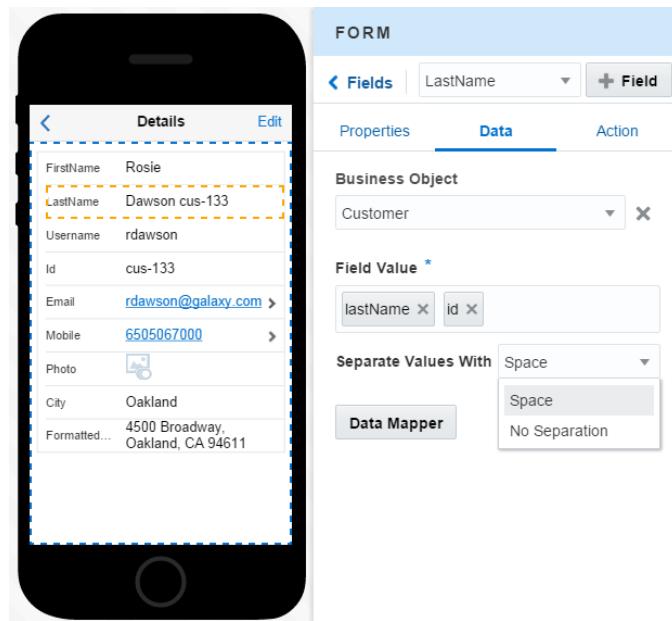


If you haven't mapped any data, then click the **Data Mapper** button (  ). To find out more, see [Taking a Look at the Data Mapper](#).

When you select a field, the Data tab displays its current mapping. It might display a field value that belongs to the business object of the application selected for your app, or it could be a different type of value, like a fixed value. Clicking the Field Value field opens a list of values. You can add or change the field mapping by clicking on an item in this list (if you're selecting from among business object field values), or by entering the name of a value into the field itself. If you're mapping a value from a business object, this list lets you pick from all of the field values that belong to the business object that is used to populate the form.

 **Note:**

If you enter a value into the field that doesn't match any of the values that belong to the business object, then the Mobile Designer considers this a fixed value.



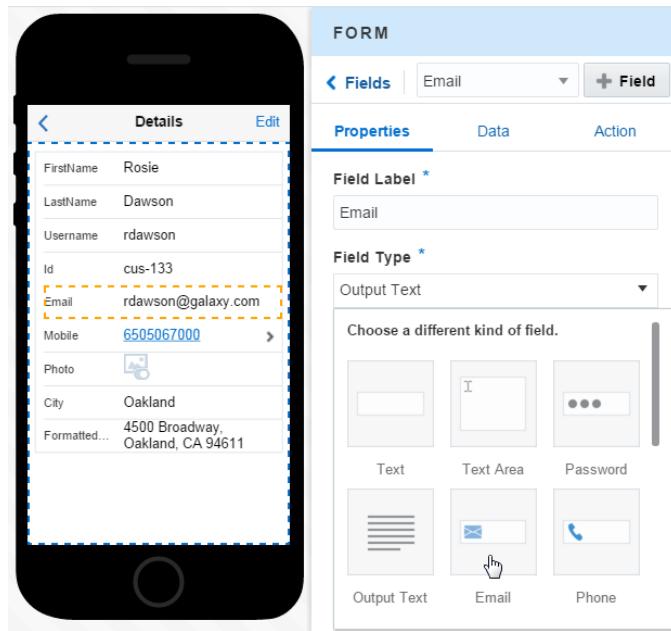
 **Tip:**

You can map more than one value to a field. The values display in the order that you added them. You can concatenate these values, or use a space to separate them.

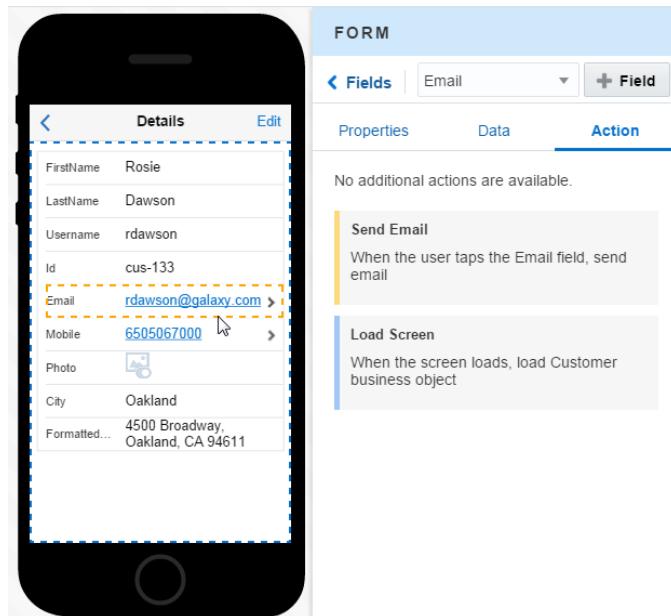
## Field Types for Lists and Forms

Your business object's fields can return different types of data, like text strings, integers, or images. To allow your list and form fields to reflect your data mapping, you need to change their field types.

Using the **Field Type** menu in the Properties Inspector, you can choose the component for field format that best suits the field data.

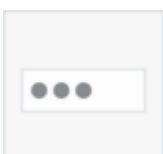
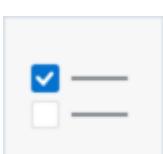


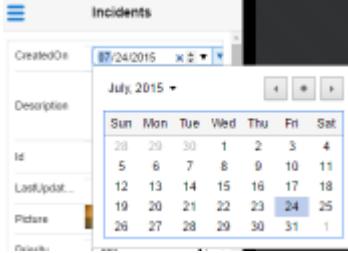
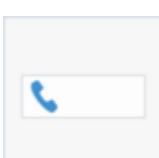
For example, if a field value displays an e-mail address as text string, you can make this field more meaningful (and useful) by transforming it from plain text into an active link to the e-mail account. To do this, you'd first select the field in the Preview and then pick **Email** from the **Field Type** menu in the Properties Inspector. With the field type changed, the Preview displays the value as a link. Clicking the **Actions** tab shows that the Mobile Designer has added a tap gesture to the field, which triggers e-mail. You can find out more about data mapping in [Enabling Your App to Display Content](#).

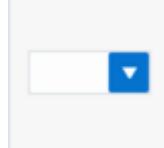


By default, fields don't accept user input because their **Read Only** option is switched on (). If your application supports user input, you can switch this option

off (). When you create an edit screen using the QuickStarts, you'll notice that this option is turned off by default ().

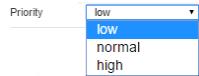
Field Type	Uses	Can I Make This Field Read Only?
Text	A short text field	Yes
		
Text Area	A long text field. You can adjust it to accommodate up to ten lines of text.	Yes
		
Password	A password entry field	Yes
		
Checkbox	Adds an options list.	Yes
		

Field Type	Uses	Can I Make This Field Read Only?
Date Time	Displays a date picker.	Yes
		
	You can display this field in the following formats: <ul style="list-style-type: none"> <li>• date—mm/dd/yyyy</li> <li>• date and time—mm/dd/yyyy 0:00</li> <li>• time—0:00</li> </ul>	
Switch	Displays a toggle switch.	Yes
		
Link	Displays a link that can open other pages in your app, or access external services.	N/A—The link field is always read only.
		
Email	Displays the text as an e-mail link. When you add this field type, the Mobile Designer configures an action to open the device's e-mail client. The <i>To:</i> field is populated with the address from the field.	N/A—The e-mail field is always read only.
		
Phone	Displays the text as a phone link. When you add this field type, the Mobile Designer configures an action to open the device's phone dialer.	N/A—The phone field is always read only.
		

Field Type	Uses	Can I Make This Field Read Only?
Select	Creates a select list. See <a href="#">Creating Select Lists</a> to find out more. 	Yes
Image	Displays the Image component. The field value that's mapped to this component is an absolute URL to the image file. 	Yes. <a href="#">Images</a> describes how to add images that users can replace.
Rating	Displays a Rating component. Yes To find out more about this component and its data requirements, see <a href="#">The Components Palette</a> .	
Linear Meter	Displays a linear status meter component. To find out more about this component and its data requirements, see <a href="#">The Components Palette</a> .	Yes
Circular Meter	Displays a circular status meter, whose value can display as a number or as a percent complete. See <a href="#">How Do I Configure Meters and Gauges?</a> to find out more about the data requirements for meters and configuring their threshold values.	Yes
Slider	Displays a slider component.	Yes. You can disable the slider by setting it to read only. 

## Creating Select Lists

Picking the **Select** (  ) field type enables you to create a list of options, or select list.



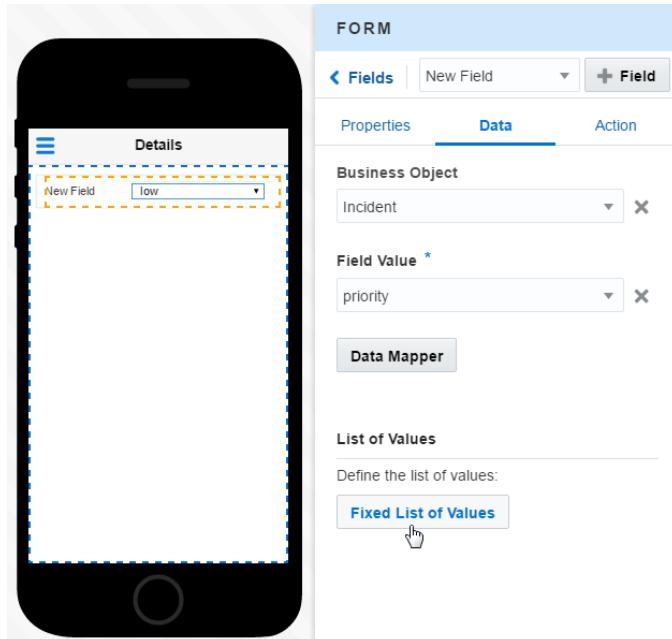
You can create a select list by binding the field to a business object, or by entering a fixed value. To do this:

1. Select the field in the Properties Inspector. In the Properties page:
  - a. Enter a field name.
  - b. Choose **Select** from the Field.

 **Note:**

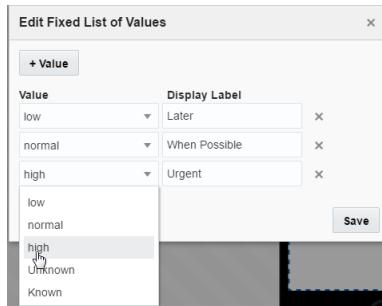
When you choose the **Select** field type, the **Read Only** toggle is switched off ().

2. In the Data page, select business object and field value, or add a fixed value. To find out more, see [Data Mapping for Forms and Form Fields](#).



You can use a field that returns three values.

3. If needed, rename the list values or add new ones. To do this, click **Fixed List of Values** to open the Edit Fixed List of Values dialog.



You can pick the value returned by the mapped field value, or click **Add Value** (+ Value) to create a new one. Click **Save** when you're done.

## Images

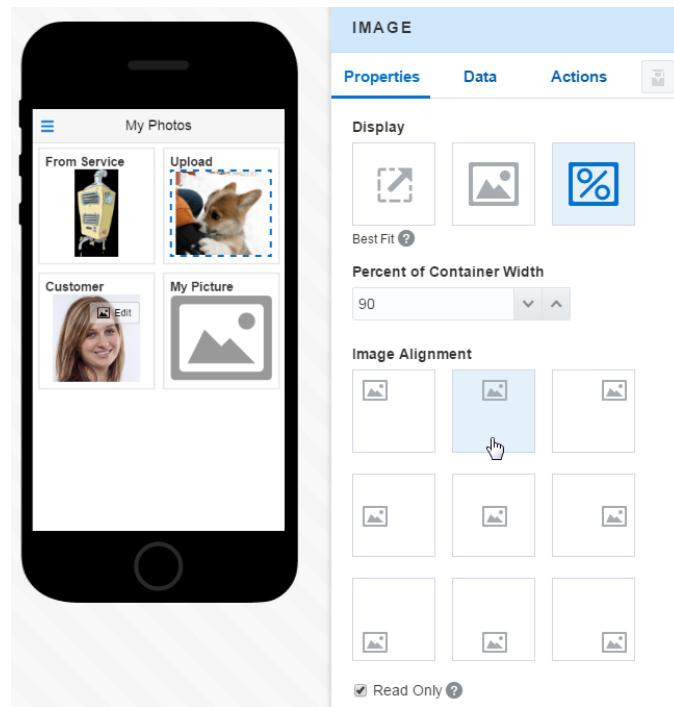
You can add an Image component to a dashboard tile or to a simple screen. To populate this component, you can upload an image or map a value from a business object. Users can't update the images that you upload, but they can update the ones that are mapped from a business object.

The Properties Inspector for an Image component enables you to set the size for an image, select an image itself, and configure a tap action on the image.

### Sizing the Image

The Properties tab enables you to set the size of the Image component. You can let the Mobile Designer size images for you by clicking one of the following:

Option	Description
<b>Best Fit</b> 	The Mobile Designer sets the size for you by calculating the longest side of the image against the corresponding axis.
<b>Percentage of Container Width</b> 	Set the size yourself by entering a value. You can adjust the size further using the spinner.
<b>Actual Size</b> 	Choose this option if your image already fits the container, or doesn't need adjusting.



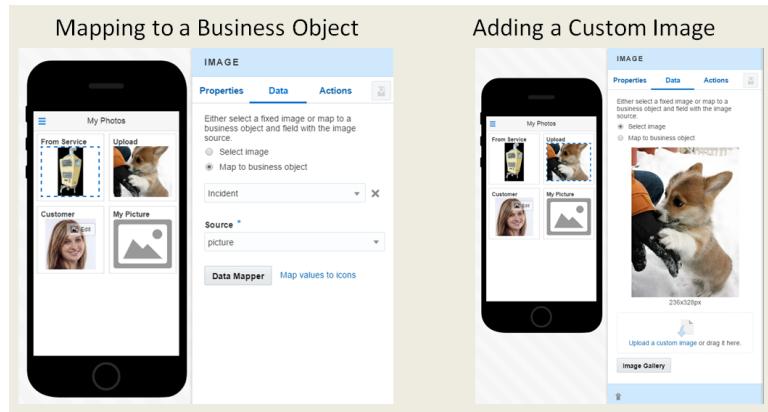
### Selecting the Image

The data tab lets you choose how your image is delivered when you select one of these options:

- **Select an image**—These are static images that users can't update.
- **Map to business object**—Users can replace these images, or they can remain as read-only. Business objects with upload actions allow access to photo galleries and cameras from within the mobile app.

#### Note:

The value returned from the application can be an absolute URL, a relative URL, or a base64-encoded image. For the latter, the value must start with `data:image/....` Also, check with your administrator to make sure that the business object supports binary streams.



To add a custom image:

1. Drag an Image component (  ) from the Component Palette into your screen.

 **Note:**

You can't drag an Image component into a List or Form component. The Image field type is read only.

2. Click **Select an image**.
3. Click **Image Gallery** (  ), choose an image from the gallery, and then click **Select**. Alternatively, you can upload an image to the Properties Inspector, or drag it in. (This also adds the image to the Custom category of the Image Gallery). See also [Using the Image Gallery](#).

 **Upload a custom image or drag it here.**

The Properties Inspector displays a thumbnail of the image that you've added, along with its dimensions (in pixels). The images that you upload are read-only and can't be changed by app users.

To add an image using field value mapping:

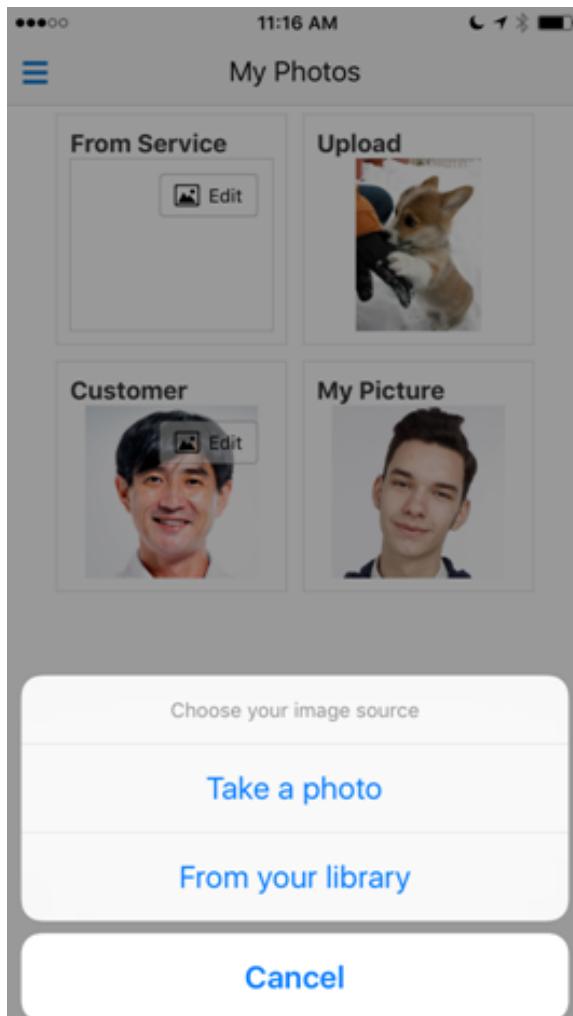
1. Drag an Image component (  ) from the Component Palette into your screen.
2. In the Properties Inspector for the Image component, click the Data tab and then select **Map to business object**.
3. Select from among the data options, like Data Source or Business Objects.
4. In the Data tab of the Data mapper, drag the value for the image into the Source field (located in the Image Components Fields section). You can only map a single

value to this field. This value represents an absolute URL to the image file that's returned from the application.

**Map Data Fields to UI**  
Now let's map the fields for your selected resource to the components in your list.



5. In the Query tab of the Data Mapper, choose your filter criteria and then click **Finish**.  
See [Taking a Look at the Data Mapper](#) to find out more about selecting business objects and mapping field values.
6. If needed, select an image upload action.
7. To enable users to replace the image, clear the **Read Only** option in the Properties tab. The Mobile Designer superimposes an overlay over the image that indicates that users can update the image. By tapping this overlay, users replace the image using the device camera or photo library.



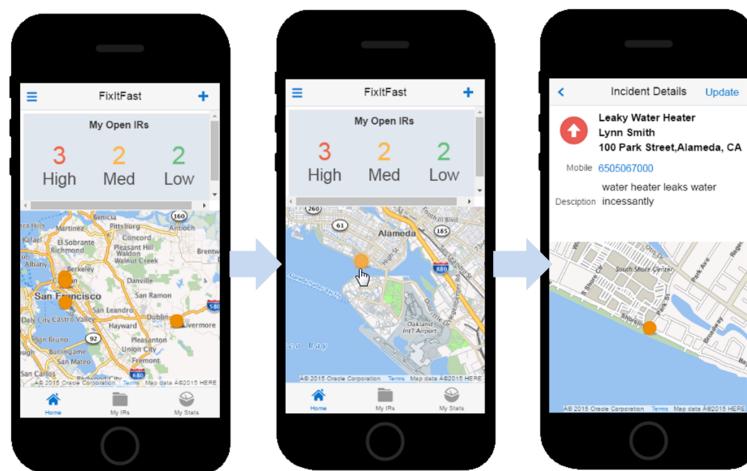
## Tap Gestures

You can combine the tap gestures configured for the Image component (☞) with actions that are specific to a business object, or with standard navigation actions like *Navigate to Screen* or *Navigate Back*.

## Maps

The Map component lets you create an interactive map, one that plots markers from addresses or geographic coordinates. Your users can explore these maps using standard gestures like pans or pinches to zoom in and out.

Your maps can drop markers for a set of addresses, or show just a single point of interest (POI). Because the Map component supports tap gestures, you can combine these two views to allow users to zoom in to a single location from a high-altitude view that contains a set of markers. [Use Case: Adding a Map Component](#) describes how the QuickStarts help you do this.



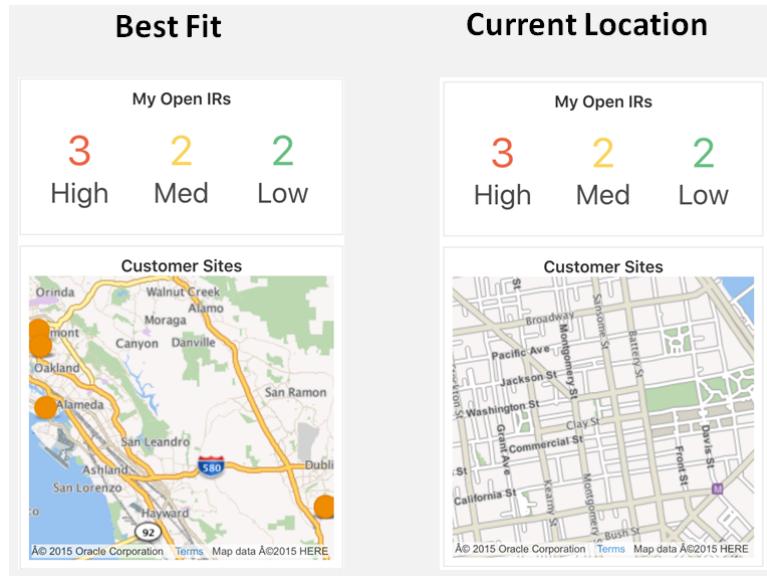
### How Do I Set the Zoom Level?

Depending on the number of data points returned from your application, and your intent, you can choose between the following two options in the Properties tab:

- **Best fit for points**—The Mobile Designer adjusts the zoom level so that all of the markers display in the center of the map view.
- **Center on current location**—The Mobile Designer displays the current location of the mobile app, which may not be near any markers. If you choose this option, users may not see any markers when they first land on the screen, so they'll need to pan and adjust the zoom to find them.

#### Note:

The Mobile Designer displays the app user's current location by default when there's no data mapping.

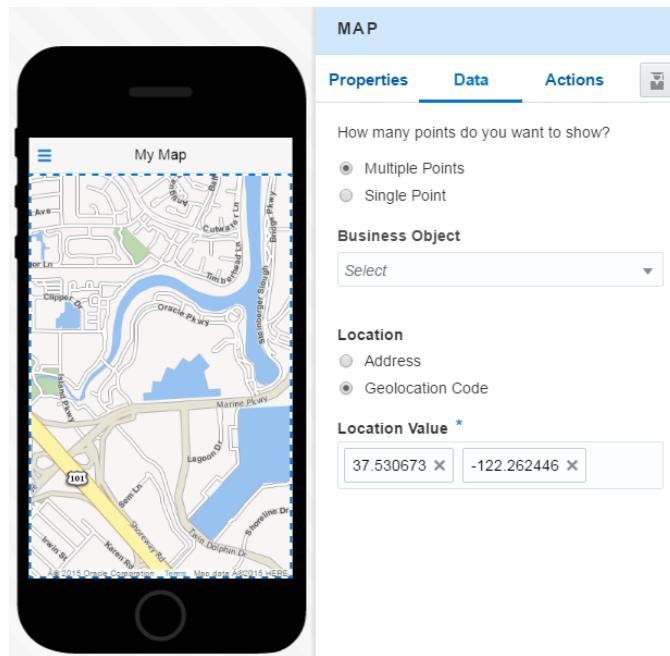


### What Kind of Location Data Should I Use?

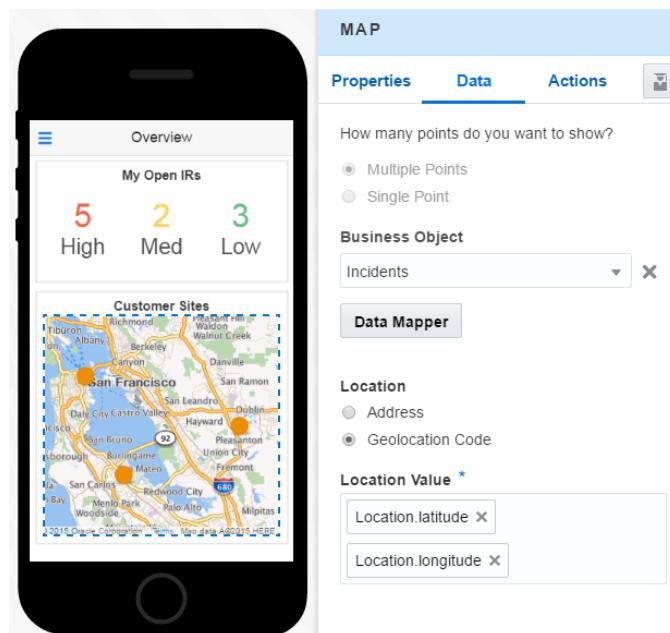
The Mobile Designer plots markers using geocoded locations, which are longitude and latitude coordinates that pinpoint a specific place. Your service might return these coordinates, or it might return human-readable addresses instead. The Mobile Designer geocodes these to transform them into longitude and latitude coordinates. For example, if your application returns *500 Oracle Parkway, Redwood City, CA 94065, USA*, the Mobile Designer converts this to *37.530673, -122.262446*. Using the Data tab of the Properties Inspector, you can configure the Map component to display one or more points from data provided from an application, or from the addresses or geocoded locations that you enter yourself.

 **Note:**

You need to enter these values separately, not as a single, comma-separated entry. In other words, parse your entries by clicking *Return* after you add each value to the Location Value field: it's *37.530673*, then *-122.262446*, not *37.530673, -122.262446*.



While you can configure the location data for your map manually using the Data tab, you can also use the Map component's QuickStart wizard to create single- and multi-point maps and bind them to business objects. After you've completed this wizard, you can revisit the Properties Inspector to not only update the data mapping, but also change other aspects, like the zoom level.



## Use Case: Adding a Map Component

For this use case, we're returning to our mobile technician app scenario. This time, we're going to equip the team with a map that plots all of their active customer sites.

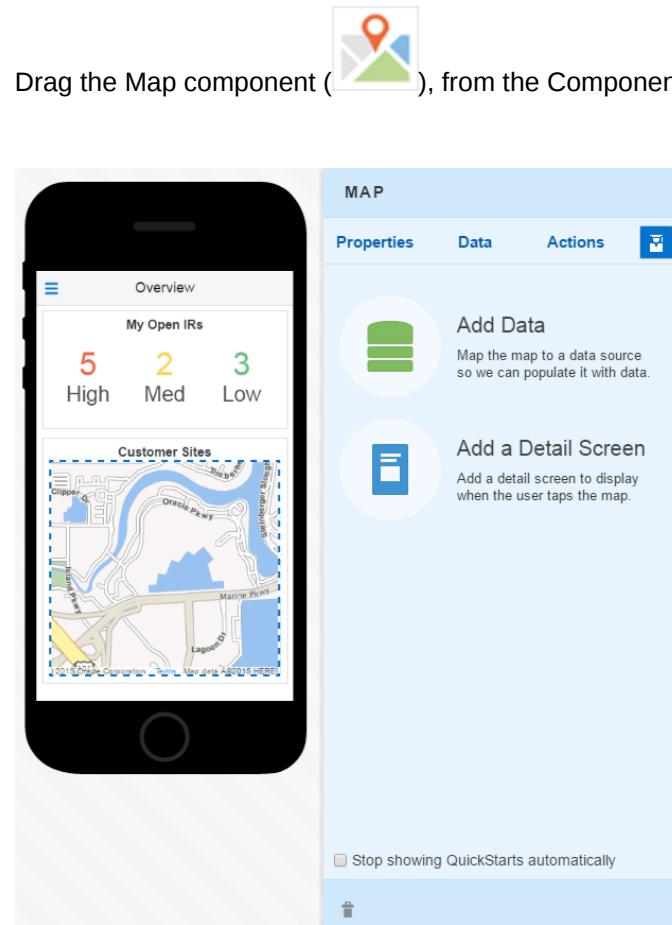
They'll drill down from a high-level view to get a closer look at a selected locale. The Map component's QuickStarts will help us create maps that not only display data returned by business objects, but support tap gestures that open detail views.

The Mobile Designer sizes the Mobile Designer component to different types of containers, be they entire screens, screens with summary panels, or dashboard tiles. For this use case, we'll add a Map component to a dashboard that is very like the one in [Use Case: Creating a Dashboard](#). In this case, we're substituting a map for the chart that occupies the lower part of the screen.

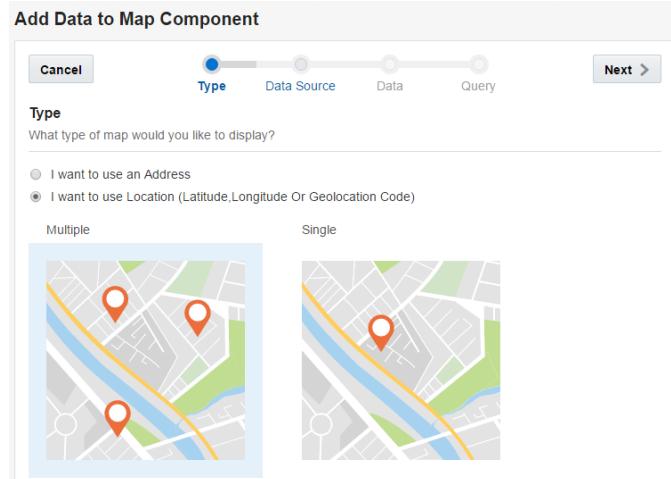
 **Note:**

This use case starts off after the dashboard has been created and its high, medium, and low priority Metric components have been mapped.

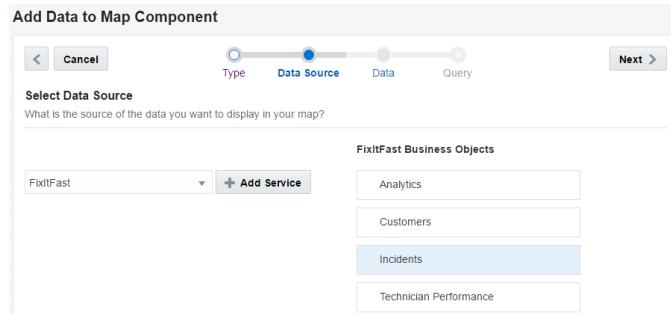
To add the map:



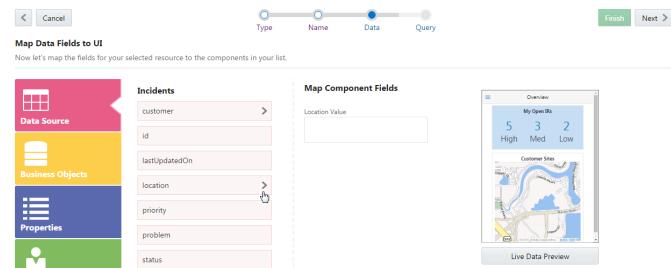
1. Drag the Map component () from the Component Palette into the Preview.
2. Click the **Add Data** QuickStart.
3. Because the FixItFast application includes a business object that returns a collection of customer sites as geocoded location coordinates, you'd choose **I want to use Location (Latitude, Longitude, or Geolocation Code)**. To display the entire collection within the map, you'd complete the Type page by selecting **Multiple**. You're ready to choose the aforementioned business object, so you can move on to the Data Source page by clicking **Next**.



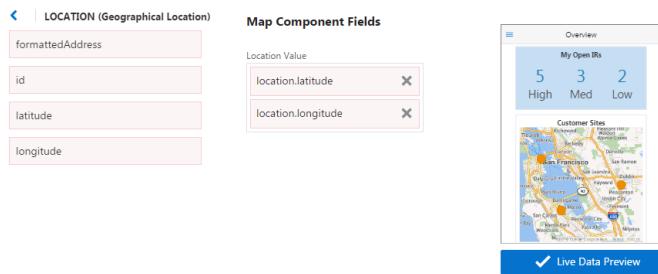
4. In the Data Source page, you'd choose the Incidents business object because it references another business object called Location that provides the map's data.



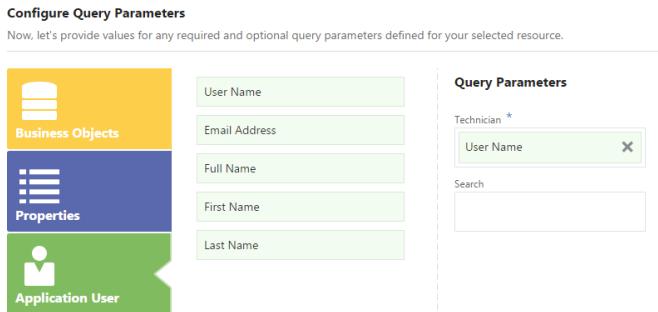
The Data page lists the fields defined for the Incidents business object along with the two peer objects that it references, Customer and Location.



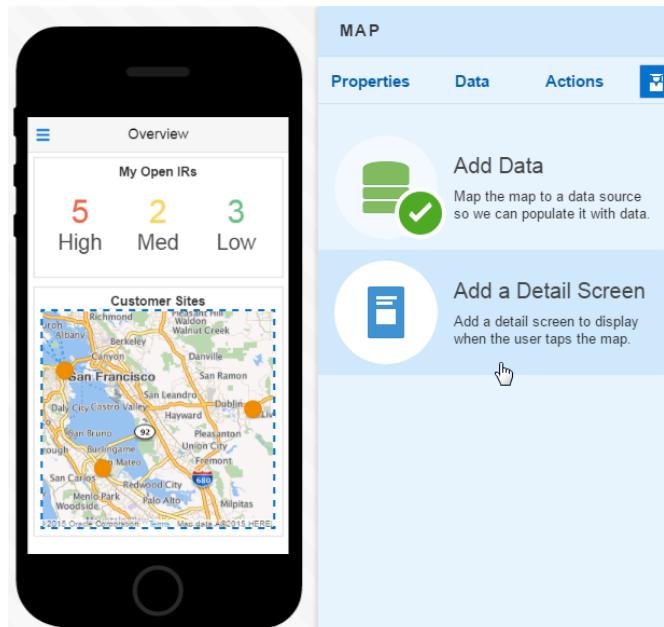
5. By expanding the Location business object and then dragging the **latitude** and **longitude** fields into the Location value field, you've wired the map to the data you need. Clicking **Use Live Preview** bears this out, as the mock data generates a set of markers.



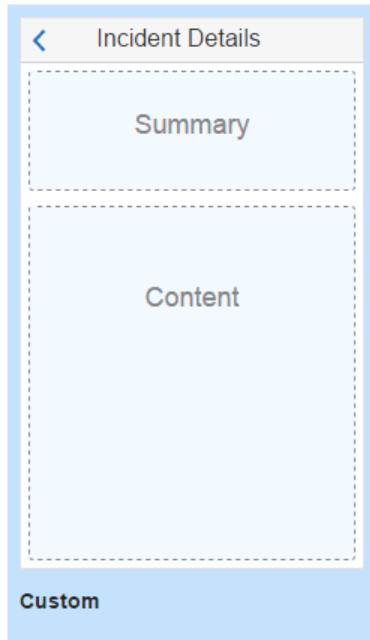
6. Because you want to display only the addresses of customers who are assigned to the technician, you'd filter the search results by the technician's user name.



You're done with the top-level screen. Now you're ready to create the detail view. The QuickStarts can help you out.



7. Clicking **Add a Detail Screen** opens the Screen Type screen. Because you want to add some context to the map location, you'd choose **Simple Screen with Summary** so that you can display the customer information that corresponds to the selected marker.



8. With the template complete, you're now ready to add content to it . Click **Go to Detail Screen**.



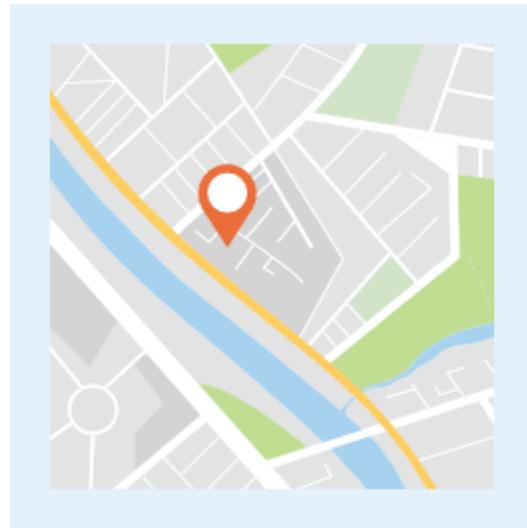
9. To add the customer context, drag a Card component ( ) into the summary



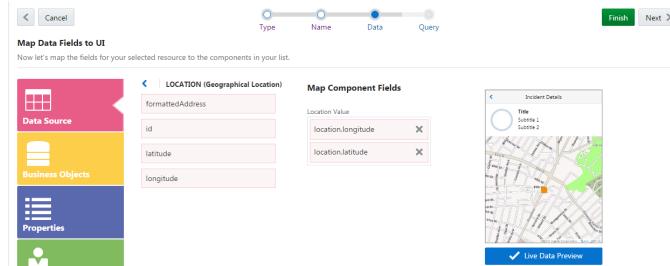
panel. Then drag a map component ( ) into the lower panel. Using the Add Data QuickStart, you can populate the card with customer and incident information. To find out more, see [Cards](#).

10. Clicking the Add Data QuickStart for the map component returns you to the Type page. Because this detail page describes the details and location of a selected customer site, you'd pick **Single**. For the data source, choose **Location (Latitude, Longitude, or Geolocation Code)**.

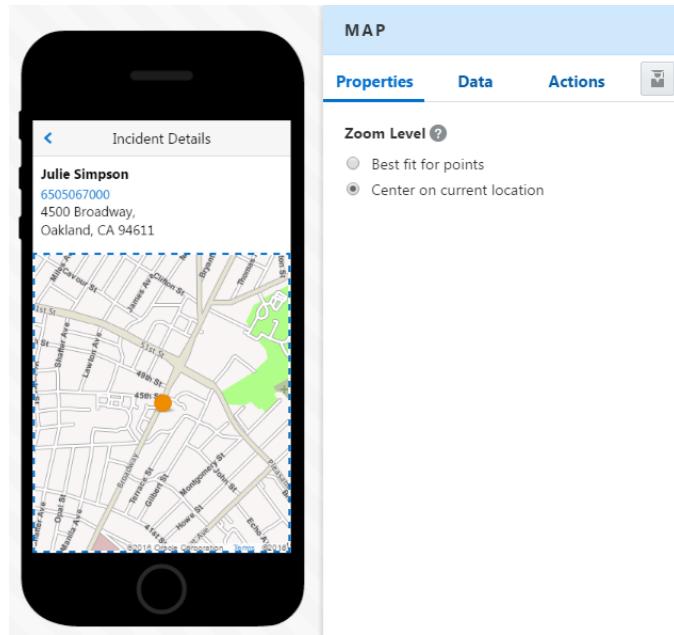
Single



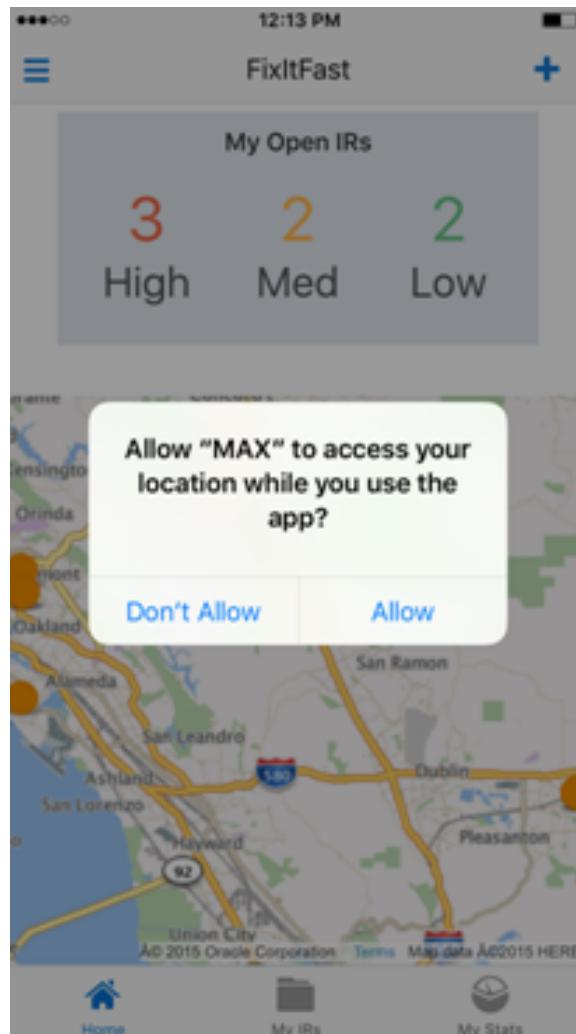
11. Again, you'll map the point using the longitude and latitude fields.



12. When you've completed the Add Data QuickStart, you'll adjust the zoom level for this single point by selecting **Center on current location**. By selecting this option, the maker will be in focus when users open the screen.

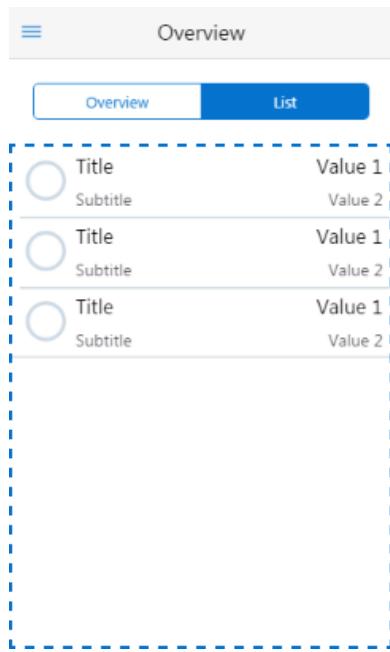


13. When you run the app on a mobile phone, you (and your users) will click **Allow** when prompted to enable MAX to access your current location.

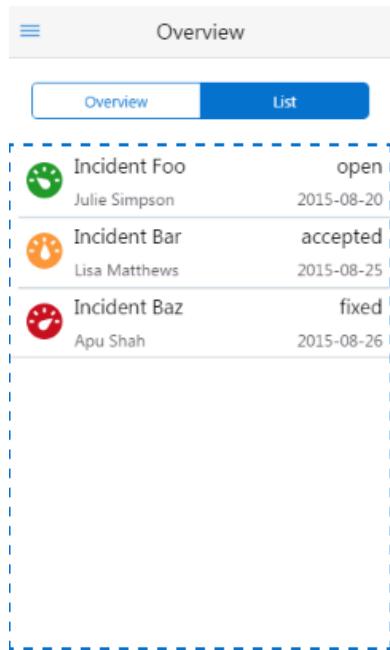


## Enabling Your App to Display Content

Although creating an easy-to-use and compelling UI is important (and a fun task when you use MAX), you should generally think of your mobile app as a way for users to get at data. Without services, your app can't display any data at all, including the mock data that helps you build your screens. Your app might look something like this:



When you've hooked it up with data, it might look like this:



Your app might need to access servers that host your company's data. It might also need information from the third-party providers, such as geolocation services. In planning your app, you might have discussed your app's data requirements with your development team. In turn, they developed services that provide this data. They may have tailored a service for your app, or created services that address more general needs. Services group together business objects (which you'll learn more about in [Getting to Know Business Objects](#)). Using services lets you enliven your app by allowing it to display data and react to user interactions. The specific task that enables

your app to use data is data mapping. Like building your UI from the MAX screen templates and UI components, data mapping can be done quickly by drag and drop operations.

## Exploring Services

The Service Catalog contains all of the services created by your development team.

The screenshot shows the Service Catalog interface. At the top, there is a search bar labeled 'Filter Services' and a checkbox 'Show latest versions only'. Below the search bar, the services are listed in a grid. The first service is 'FixItFast' with the sub-label 'MAX FIF demo/test API' and version 'v1.0'. The second service is 'HelloWorld' with the sub-label 'Hello World test API' and version 'v1.0'. At the bottom of the grid, there is a 'Select' button. Below the grid, there is a pagination bar showing 'Page 1 of 1 (1-2 of 2 items)' and a 'Select' button.

You can access the Service Catalog by using the Add Data QuickStart when you select a component in the Preview. To find out more about using the Add Data QuickStart (which is the quickest way to wire your app to your data), see [Use Case: Creating a List Using QuickStarts](#).

The screenshot shows the 'Add Data' QuickStart interface. It features a green icon of three stacked bars. The text 'Add Data' is displayed, followed by the instruction 'Map the list to a data source so we can populate it with data.'

You can also open the Service Catalog by clicking **Data** (≡) in the navbar.

## Getting to Know Business Objects

Business objects let you map your UI to data.

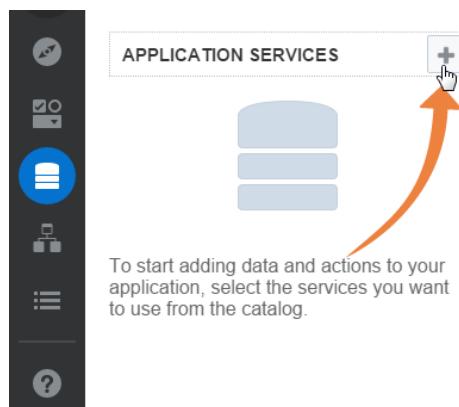
Business objects represent data as real-world entities like people, places, events, or business processes. They describe various aspects of the data and how (or where) it can be used. The business objects represent items in the data collection that support your mobile app. For example, say you're creating a mobile app that enables field service technicians to both respond to and track customer incident reports (IRs). You'll need the app to display the number of IRs assigned to a particular user. You'll also want your app to show each user a list of assigned IRs and enable users to open forms to create new IRs.

To allow your users to access data, your development team has created a service that includes a business object called *Incident*. To enable various aspects of IRs to display as a list in your UI, your development team has defined a set of fields for this business object which have names like *Incident ID*, *Incident Description*, and *Priority*. To allow

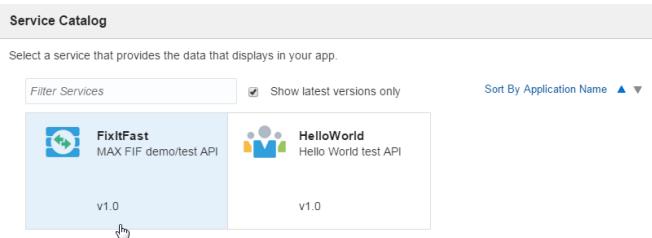
users to update, create, and delete a business object, your development team has also created a set of actions. These actions enable you to add create, edit, and update pages. They also allow user gestures, like tap and tap-and-hold to trigger an action.

You can pick an application and explore its business objects using the Data Palette. To open the Data Palette:

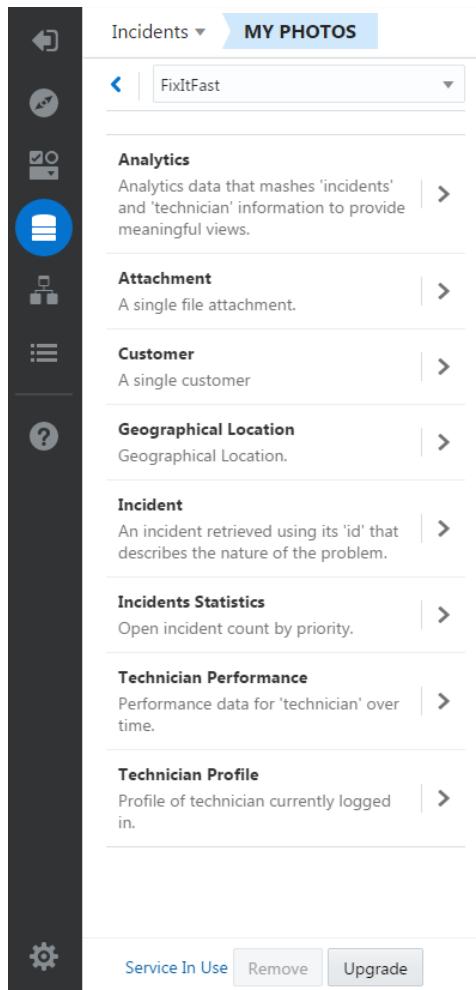
1. Click **Data** (  ) in the Designer's navbar.
2. Click **Add** (  ) to open the Application Catalog.



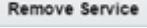
3. Choose the application that you want to make available to your app (along with the version) and then click **Select**.



4. After you choose an application, the Data Palette opens. It lets you explore its business objects, which allow you to bind your UI to data.



 **Tip:**

Click **Remove Service** () if you've selected a service that's not used by your app.

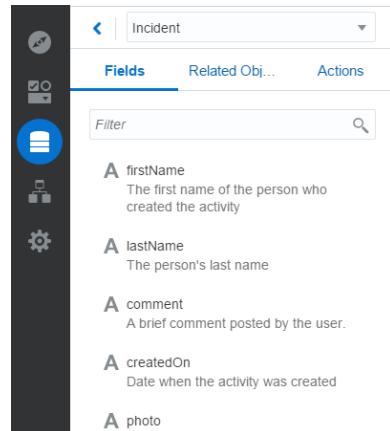
When your development team models the data into a business object, they also provide descriptions and hints about you can use that display in the Fields, Related Objects, and Actions tabs of the Data Palette. If you want to display a list, for example, your development team will document the business object accordingly by describing not only its purpose (like *Incidents' history and assignments by technician*), but also the fields that you can use in your app (like *Customer First Name, Last Name, Address*, and so on). The Data Palette also tells you about the actions that your users can perform on the data represented by the business object.

**Tip:**

The Add Data QuickStart is the fastest way to map business objects to your UI components. Using the QuickStart wizards, the Mobile Designer can infer how you might want to build your UI according to the business object that you choose. If you choose a business object that returns a list of items, for example, the QuickStarts will prompt you through populating the list view page. For create and edit pages, the Mobile Designer selects the fields from the business object that can be used for data input. See [Use Case: Creating a List Using QuickStarts](#) to find out more.

### Field Values

Each business object has a set of fields, which represent how data can be delivered to the UI. Using the Fields tab, you can browse through your development team's descriptions, such as the type of data it returns (like alphanumeric strings or numbers) and how this data is formatted (e-mail, URL, date, etc.) The developer's descriptions might also tell you how you can use the field. For example, they might include such details as whether app users can update the field, or whether it can support searches. The descriptions might also tell you how the field supports the app by stating that it's mandatory a field when users update or create records.



The screenshot shows the 'Fields' tab of the Oracle Mobile Designer interface for the 'Incident' business object. The tab bar also includes 'Related Obj...' and 'Actions'. A 'Filter' input field is present. The list of fields includes:

- A firstName**  
The first name of the person who created the activity
- A lastName**  
The person's last name
- A comment**  
A brief comment posted by the user.
- A createdOn**  
Date when the activity was created
- A photo**

### Related Objects

Business objects can have either direct or indirect relationships. They can be in peer relationships, or parent-child relationships. Child objects belong to their parent objects and can't exist on their own. For example, a child object called *Incident Activities*, which provides details about the incidents described by its parent object, *Incident*, has no meaning outside of the context of the parent object. A business object can also be referenced by another object in a less direct relationship. For example, a *Customer* business object that includes fields describing customer contact information would be related to the *Incident* business object, as it provides complementary information. Unlike a child object, however, a referenced business object exists on its own.

The screenshot shows the Oracle Mobile Designer interface for an 'Incident' object. The 'Related Objects' tab is selected. Under 'Child Objects', there is a section for 'Incident Activities' with a description: 'Collection of incident-activities.' Under 'Reference Objects', there are sections for 'Customer' (described as 'Entity representing a customer') and 'Geographical Location' (described as 'Geographical Location').

## Actions

Business objects may also have a set of actions that manipulate the data returned by the field, or can be used to support user interactions, like swipe gestures. The Mobile Designer provides a set of standard operations that enable fields to display data and allow user interactions:

Operation	Description
read	Returns an individual record.
collection	Returns a set of records.
create	Adds a new record.
delete	Removes a record.
edit	Updates a record.

In addition to these basic operations, your development team might also include operations that are unique to the business object.

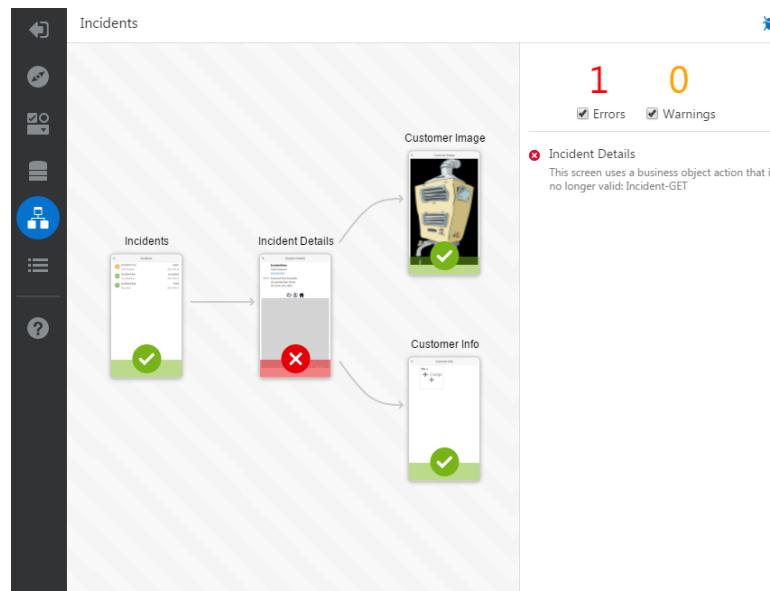
The screenshot shows the Oracle Mobile Designer interface for an 'Incident' object. The 'Actions' tab is selected. A message at the top says: 'The following actions can be performed on this business object.' Below this, a list of actions is provided with descriptions:

- read**: Get a specific incident using 'id'.
- collection**: Get a list of incidents.
- put**: Replace a specific incident.
- create**: Create/add a new incident.
- delete**: Remove/close the specific incident.
- edit**: Replace a specific incident.
- parent**: Get a list of incidents.
- instances**: Get a list of incidents.
- closeincident**: Closes the incident.

## Service Upgrades

If your development team has updated a service with new business objects, new, fields, or new query parameters that you'd like use, click **Upgrade** (  ). But before you do that, check with the service developer to find out about the changes to the service and if they'll prevent your app from making a graceful transition to the newer version. A newer version that addresses a bug fix, for example, could be backward compatible and not affect how your app consumes data. But in other cases, a newer version might include profound changes that require you to rework the data binding and other configurations: some fields currently in use might now expect different types of data, or may no longer exist. Query parameters can change, and entire business objects can be removed. When you upgrade a service, MAX checks your current bindings against the new service. MAX's error reporting (  ), lets you

spot the broken data mappings within the screen flow (accessed by clicking  in the left navbar). From there, you can begin making your repairs by clicking a snapshot to open an affected screen.



### Note:

Before upgrading, save a copy of your app by exporting to your system. See [Importing and Exporting Your Apps](#). Clicking **Upgrade** also lets you revert to the prior version of the service after you've upgraded.

### Re-Publishing Your Apps After Upgrades

If you're upgrading the service for an app that you've already published, then you'll need to publish that app again after you've tested its compatibility with the new version. This means that you'll have to rebuild the app before you scan its test QR code. When you've completed your testing, you'll need to re-publish the app and then

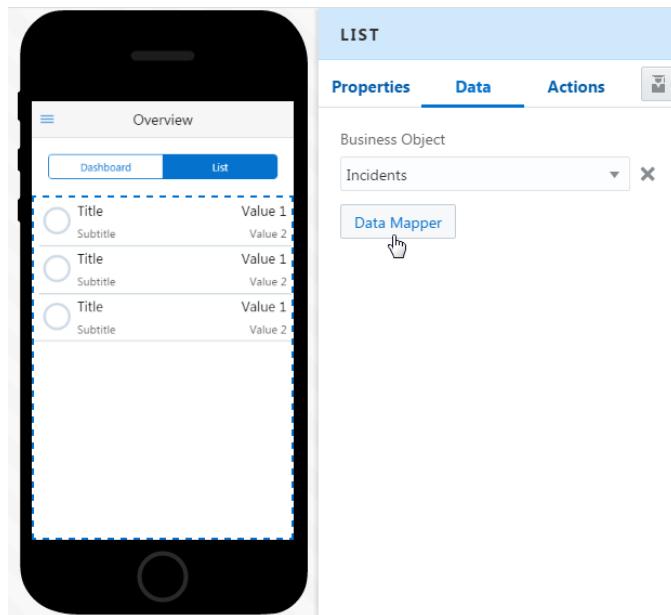
inform users that they need to re-install the app by scanning the production QR code. If you don't care about retaining the test and production QR codes, then you can perform the service upgrade and subsequent testing on a duplicate of the app that you've created using the **Import** and **Export** functions. In this case, your users can delete the version that runs the previous version of the service and install the new app, which has a different QR code. See [Building and Testing Your Apps](#) and [Importing and Exporting Your Apps](#).

## Taking a Look at the Data Mapper

After you've chosen the application, you can then specify how its data becomes available to your users through the Data Mapper.

### Tip:

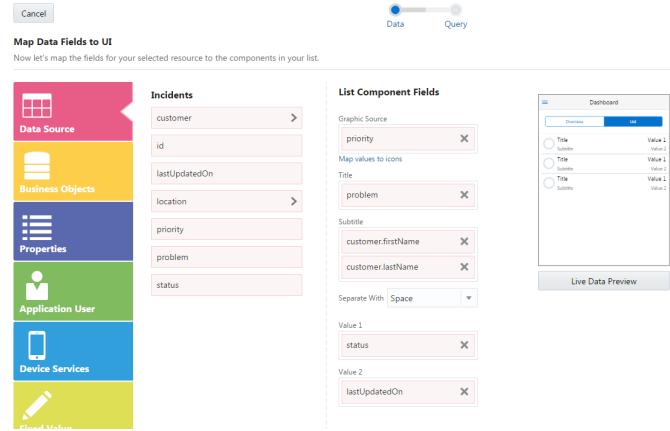
You can open the Data Mapper from the Data Page of the Properties Inspector, or through the Add Data Quickstart. This wizard provides the fastest route to adding data to a UI component.



The Data Mapper tool lets you set the values from the following sources:

- **Data Source**—The fields that belong to the business object currently in use. To find out more, see [Getting to Know Business Objects](#).
- **Business Objects**—Lets you select from among the business objects that belong to an application.
- **Properties**—Lets you define an application-wide value for a query parameter. See [Application Properties](#).
- **Application User**—Information about the authenticated application user, including the user name, e-mail, and so on.

- **Device Services**—The system date and the app user's current geographic location.
- **Fixed Value**—A specific value, or a value that enables the business object to return specific data.
- **Current Screen**—Use this category to map values that are passed between pages and components, including search fields. See [Adding a Search Field](#).



The Data Mapper color codes each of these categories to help you identify the data source. For example, fields belonging to the Data Source category (like those shown in the figure above) are pink. Fields belonging to the Application User category are green and those belonging to the Device Services category are blue, and so on. After you pick a source, you then drag and drop the fields to populate components and build forms.

#### **Tip:**

Choose **Live Data Preview** to help you map your values. The mock data (which is provided by the developer), guides you through your mapping. To find out more, see [List Layouts](#) and [Use Case: Creating a List Using QuickStarts](#). To see how the images, field entries, or data visualization tools look using like data, see [Previewing Your App](#).

**List Component Fields**

Graphic Source

Map values to icons

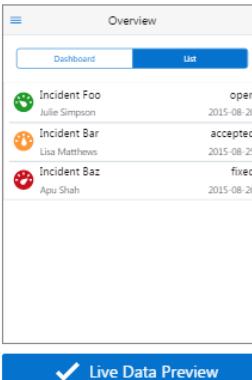
Title

Subtitle

Separate With  ▾

Value 1

Value 2



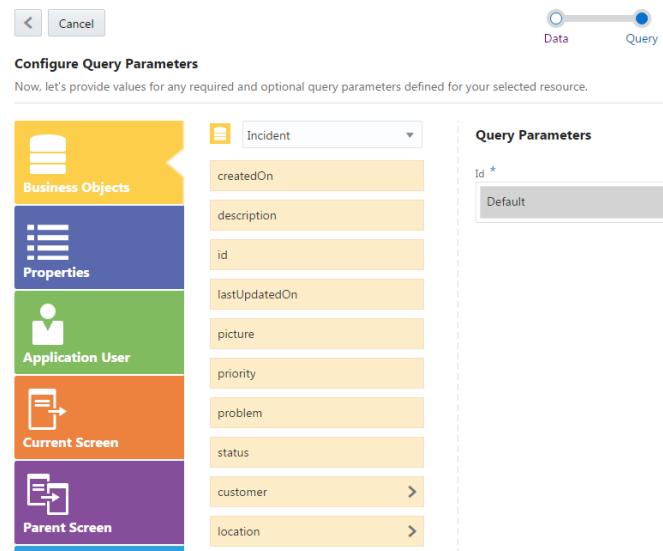
Live Data Preview

When you've finished the field mapping, click **Next** (or click the **Query** tab) to filter the data that the application returns to the app. You don't always need to add a filter, but when you do, you can pick from the **Query** page's filtering categories, which mirror those in the **Data** page. In addition to the categories that appear in the **Data** page, the **Query** page includes one more category that you use when you're mapping data for child screens, the **Parent Screen** category. Sometimes you add the filtering criteria yourself, but other times (as is the case when you're creating child screens), the Mobile Designer will volunteer the default value that's set by the application. For drill-downs from list to detail screens, this value is typically an ID. MAX notes these implicit mappings with a *Default* field value.

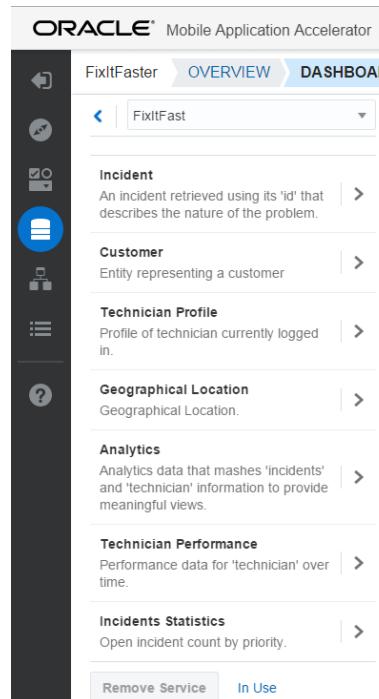
## Query Parameters

Id \*

You can override this mapping if you want a different way of filtering the returned data.



If you've mapped values, then the Mobile Designer notes the application as In Use. You can't remove this application without removing the mappings first. When you map a component to a business object, the Mobile Designer creates an action that loads the business object when app users open the screen. For more information, see [Use Case: Mapping Data to a Line Chart](#).



## Mapping Field Values to Forms

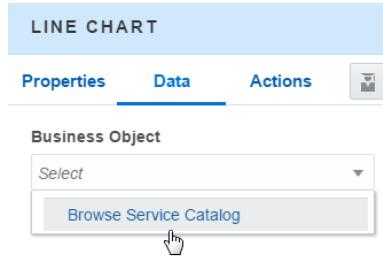
You can add as many fields as you need to a form. After you map a value to a field, the Mobile Designer adds a new field for you to populate. You can continue adding fields this way, or by clicking the **Add All Fields** button (  ) , which generates

fields using all of the field values that belong to the selected business object. If you've already mapped one value from the object, then clicking **Add All Fields** button adds the remaining values. Use the **Move** icon (↔) to reorder of the fields. For more information about adding and reordering fields using the Properties Inspector, see [Forms](#).

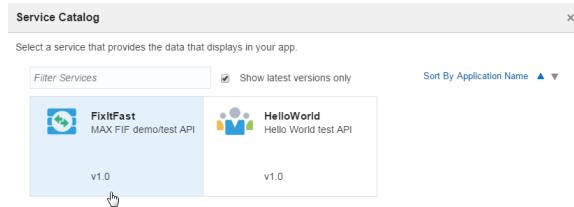
## Use Case: Mapping Data to a Line Chart

Say that you'd like to add a line chart to the dashboard of the appliance repair technician app that compares the number of incident reports (IRs) assigned to an individual repair technician (who is also the app's user) against the number of IRs assigned the other repair technicians on a monthly basis. As described in [Working with Tiles](#), you need to both add and size a tile before you can add a UI component to a dashboard. After you drag the Line Chart component in to the tile, you're ready to wire the chart to data using a business object. To do this:

1. Open the Data tab of the Properties Inspector.
2. Click **Browse Service Catalog** from the Business Object menu.



3. In the Service Catalog, click FixItFast, the service created by the development team to back the repair technician app with data. Then click **Next**.

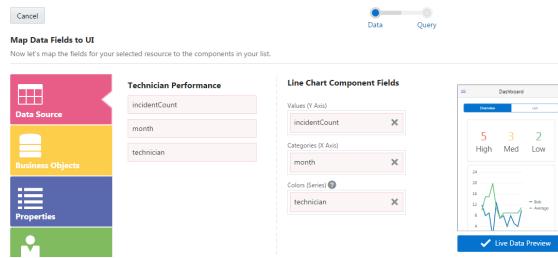


Page 1 of 1 (1-2 of 2 items) < 1 > X Select

4. The FixItFast service includes a set of business objects that provide different types of data. To find out which of these is best for this particular facet of the app, click **Show object descriptions** ( ⓘ Show object descriptions), to read the descriptions provided by your development team.
5. Because the line graph plots IRs by month, chose the Technician Performance business object. (In the Service Catalog its description in the reads "Performance data for technician over time".) Click **Next** to go the Data Mapper. The Data Mapper's Data Source category is open, showing the fields available from the business object currently in use, Technician Performance. The Data Mapper color

codes the fields to the category. In this case, they're pink because they belong to the Data Source Category.

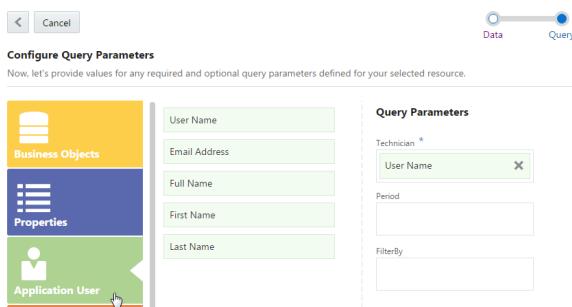
6. Use the Data Mapper to first map the technician performance-related values to the line chart and then filter the data that's returned from the FixItFast application.
  - a. To populate the chart's x and y axes, and color code its plotted lines to differentiate the individual technician, you'd pick the following values:
    - Drag the *Incident Count* value into to Values (Y Axis) field.
    - Drag the *month* value to into the Categories (X Axis) field.
    - Drag the *technician* value into the Colors (Series) field.



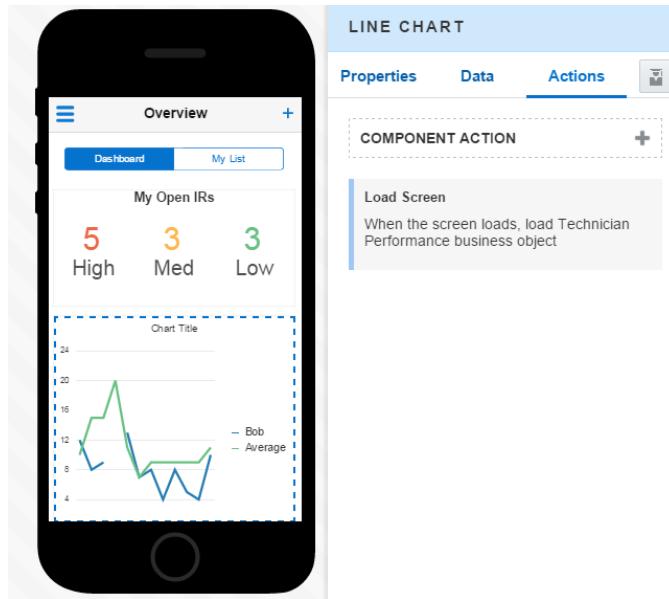
 **Note:**

While you can add more than one value for list and form fields, you can't do this for graphs and charts. For these components, you can only map one value.

6. Click **Next** to move the Query page of the Data Mapper.
  - c. To ensure that the app's user (the repair technician) sees only the IRs, customer information, and in this case, performance data that's specific to her, you'd filter the returned data by user-specific criteria. To return this data, you need use the technician's user name as a query parameter. To do that, you'd select the Application User category and then drag **username** into the Technician field (located under Query Parameters). Because this is the Application User category, the fields are green.



7. Now that you've completed the field mapping and set the query parameter, you can click **Finish**. Clicking the Actions tab in the Properties Inspector shows that the Mobile Designer has added the action to load the Technician Performance business action.



8. See [How Do I Create a Line Chart?](#) to find out about configuring line type, orientation, and other display properties for a line chart.

## Adding Actions

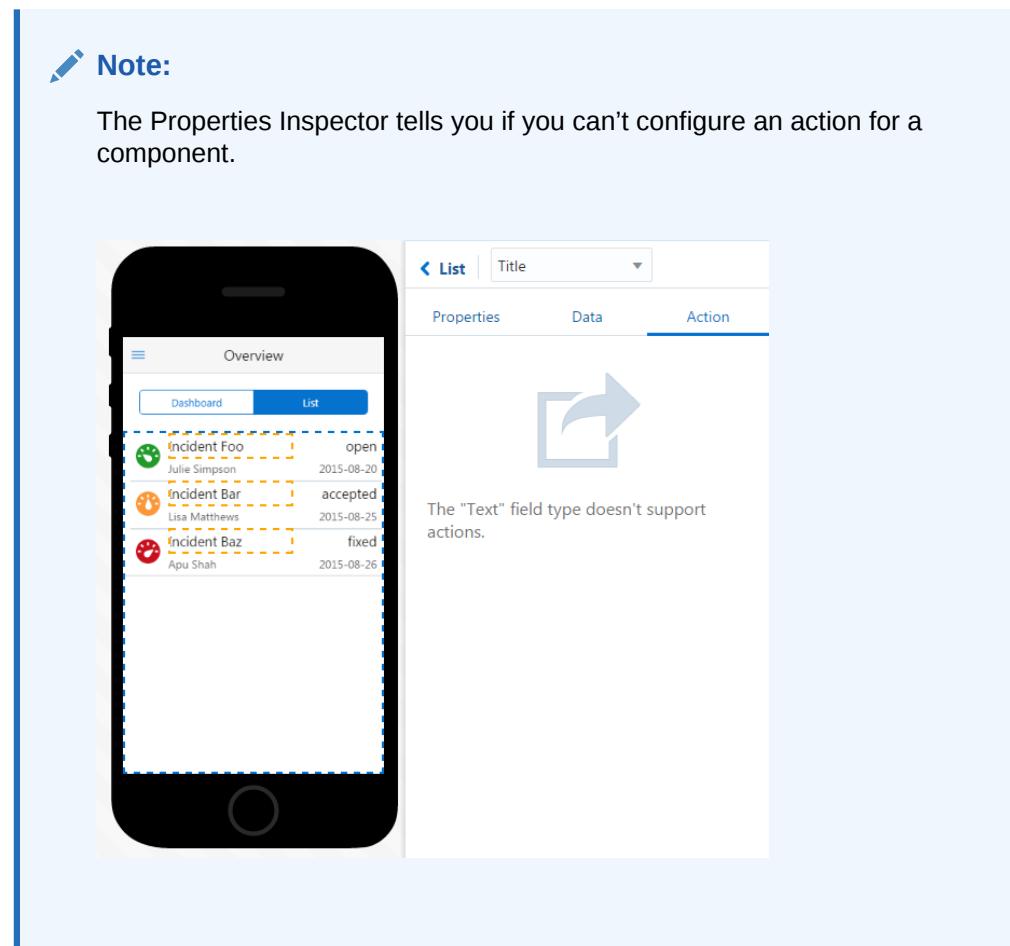
Use the Properties Inspector's Actions tab to configure gestures and actions for a selected component.

As mentioned in [The Properties Inspector](#), the Actions tab allows you to link your screens together in a flow. Using the Actions tab, you can choreograph how your screens interact with one another by configuring how your app responds to changes in field values or to user interactions like tap and swipe gestures. For example, you can create a tap gesture that triggers a business object action and screen navigation. Some actions are created for you automatically when you create the footprint or your app or when you complete the Add Data or Add a Detail Screen QuickStarts.

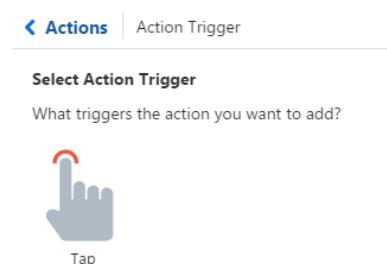
Action	Created by
Set Current Object	Mapping list or gauge components to a business object. This action, which appears at the start of the action chain, names the business object currently in use. The screen flows created using the list-related QuickStarts all have this action.
Load the screen.	Mapping a business object (using the Data Mapper or the Add Data QuickStart).
Open e-mail with a tap gesture.	Choosing Email as a field type for a Form component.
Open phone dialer with a tap gesture.	Choosing Phone as a field type for a Form component.
Navigate to the details screen.	Completing the Add a Detail Screen QuickStart
The application menu icon navigation	Completing the first page of your app. You can't change this configuration.

You can add your own gestures and actions by:

1. Selecting the component.



2. Selecting a tap gesture from the Actions panel.



Not seeing the trigger you're looking for? That may be because it is already in use.

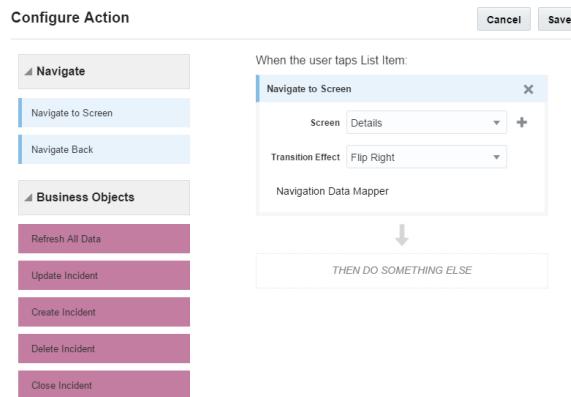


A swipe gesture is already created for you if you're configuring swipe tiles. To find out more, see [Adding Swipe Gestures](#).

3. Use the Configure Action panel to choose from the business object actions that apply to your components. These might include actions like refresh, create, update, and delete. To find out more, see [Getting to Know Business Objects](#).

Build an action chain that's triggered by the tap gesture by dragging and dropping actions into the *DRAG AN ACTION HERE* field. By combining these actions with the **Navigate to Screen** and **Navigate Back** options, you link your screens into a flow. You can enhance the navigation by choosing transition effects like flip right, flip left, or fade. If your app doesn't yet have the destination screen you need, click

**Add** (+) and then complete the wizard that prompts you through creating a landing, detail, create, or edit page. You can continue to expand the action chain by clicking **Add** and then by dragging and dropping the navigation options or business actions into the *THEN DO SOMETHING ELSE* fields that appear after each action definition. You can add as many actions as you need. Click **Delete** (X) to remove the actions that you don't need.



 **Tip:**

You can reorder the actions using the **Move** icon (≡).

Along with business actions and screen navigation, your app might require actions that update or affect values in another screen. To enable one screen to pass a value to another, click **Navigation Data Mapper** ([Navigation Data Mapper](#)). Using the Configure Navigation Data Mapping screen, you can select the value and add it to the target screen. You can choose a value from the current screen, a fixed value, and values related to the application user or the device (like the system date). For more information, see [Taking a Look at the Data Mapper](#). You can drag more than one value into a target field. You can't delete a default value that's populated in a target field, but you can replace it by dragging a new value on top of it. An asterisk (\*) indicates a required value. Clicking **Finish** takes you back to the Configure Action panel.

Configure Navigation Data Mapping

Map the data values that you want to pass from the current screen to the target screen when your user triggers this navigation action.

Cancel

Form-Opportunity

UI Fields

Region

Closing

Created

Products

Hidden Fields

aid

createddate

desc

expectedclose

oid

products

region

revenue

salesstage

Create Note Screen Parameters

Oid \*

oid

Where are these parameters used?

Business Object

Opportunity Notes

Component Form

Oid \*

oid

Where are these parameters used?

Business Object

Opportunity Notes

Component Form

To pass a value from the screen to an application, click **Business Action Mapper**

( [Business Action Mapper](#) ). In the Configure Action Business Action Parameters screen, you can not only select values from the current screen, but from the parent screen as well. Like the Data Mapper and Configure Navigation Data Mapping screens, you can also add a fixed value, an environment value, or an application user value.

Configure Business Action Parameters

Map the data values that you want to pass from the current screen to the business action's parameters.

Cancel

Business Objects

User Name

Email Address

Properties

Full Name

Application User

First Name

Current Screen

Last Name

Action Parameters

Oid \*

Default

Comment

Parent Screen

Username

Device Services

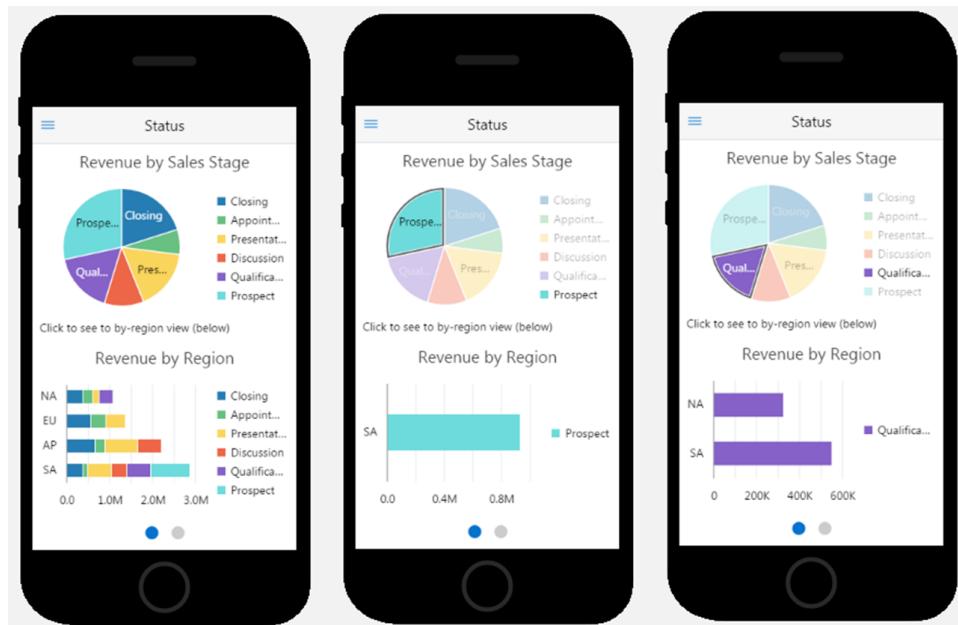
User Name

Fixed Value

# Application Properties

Setting an application property allows the components throughout your application flow to share the same data and collaborate.

These properties set application-wide values for your business object's query parameters. Unlike the query parameter values that you override to produce a single navigation instance when you use the Navigation Data Mapper, these values persist. They are global values. Components that once coexisted independently of one-another within the same application flow, or even the same screen, can interact in meaningful ways when you set the parameter that's passed between them as an application property. This way, your app can let users see the same data display simultaneously in different contexts: when a user taps one component, another reacts by redrawing its display of the selected data set. Users can see individual pie chart segments as raw data in a bar chart, for example, or drill down to a detail screen when they tap a series in a bar chart.



## How Do I Set Application Properties?

By including Set Application Property in the action chain, you can add dynamic displays to the various chart, graph, and list components in your app. But before you begin, keep in mind that application properties set values for the query parameters used by the business objects that belong to the service that backs your app. You can't use a random name for your application properties, so be sure to name them after the query parameters. Check the Query page of the Data Mapper for these names, or ask your developer.

To set an application property:

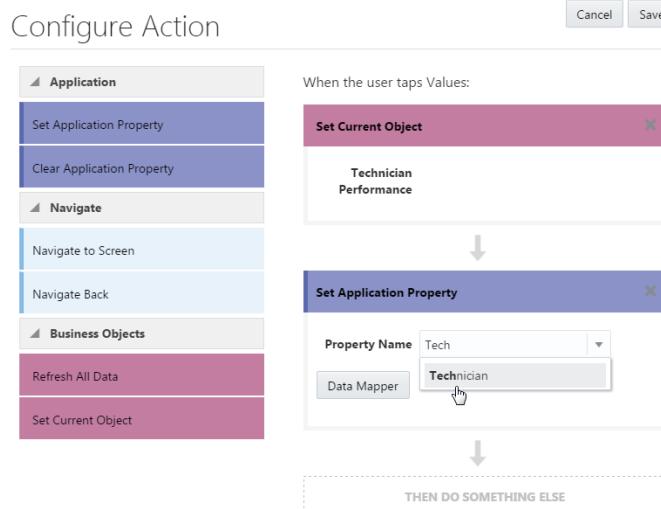
1. First, add a tap action to the source component that passes the parameter:

- a. Select a component in the Preview (one that's been mapped to a business object).
- b. In the Properties Inspector, click **Actions**.

- c. Add a tap action by first clicking **Add Component Action** and then **Tap** (  ).

2. Add the application property to the action chain:

- a. In the Configure Action panel, drag and drop the **Set Application Property** action.
- b. Enter the name of the query parameter, or, if you've already added application properties in the Data Mapper, select one from the list. These properties must match up with the query parameters for the target component (the consumer of the parameter).

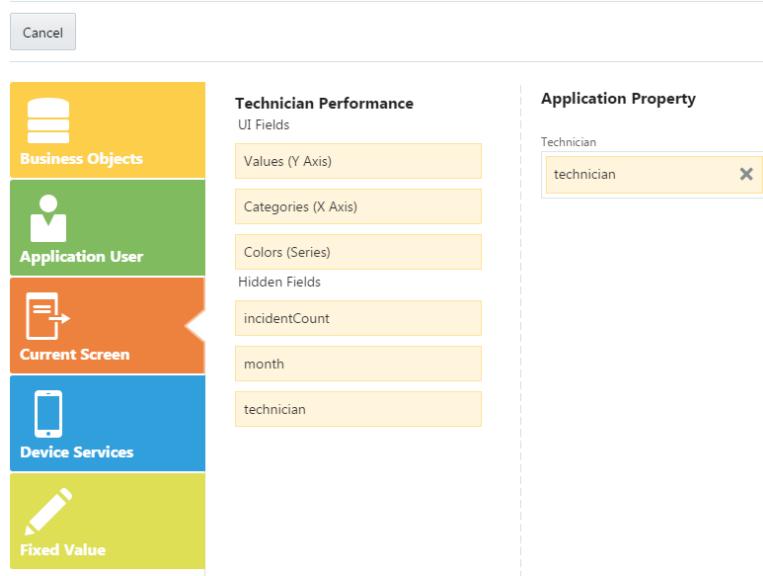


3. Next, set the parameter value that this component will pass:

- a. Click **Data Mapper** (  ).
- b. In the Map Data page, choose the value mapping from among the data source categories.

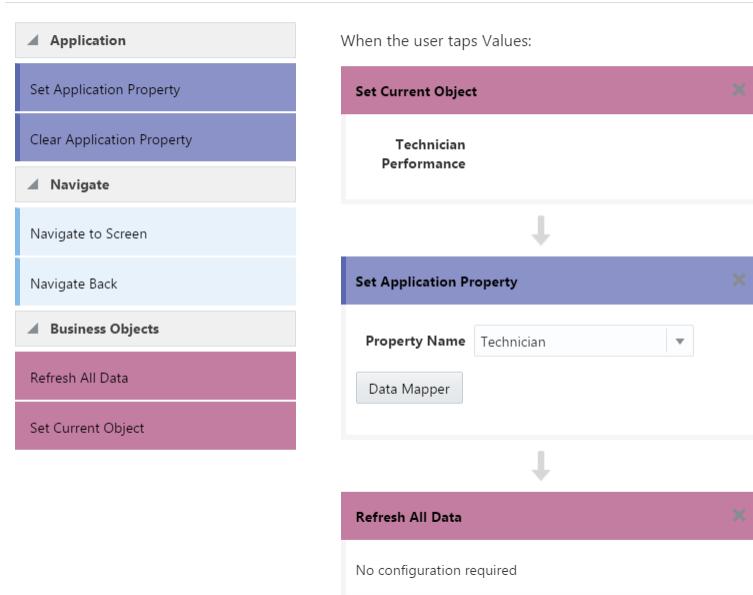
### Map Data

Map the application property value to the data source you want to use.



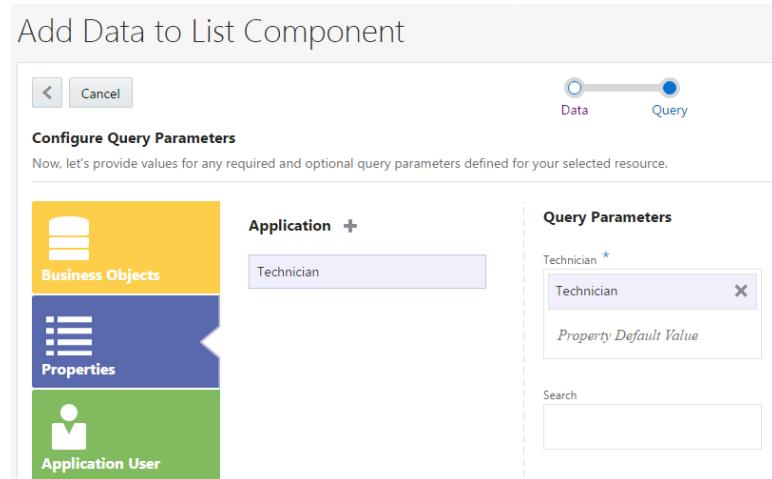
- c. Click **Finish**.
- d. If this component passes the parameter to another component within the same page, then drag and drop the **Refresh All Data** action and then click **Save**. Otherwise, you can omit this action and add a navigation option to the action chain. When you've completed the chain, click **Save**.

### Configure Action



- 4. With the property value set, you now need to configure the component to consume the parameter. To do this:

- a. Select the component in the Preview or breadcrumb trail and then open the Data tab in the Properties Inspector.
- b. Click **Data Mapper**.
- c. In the Data Mapper, open the **Properties** category.
- d. Open the Query page.
- e. Drag and drop the application property.



You can add a default value for property by choosing a field from any of the data source categories. In the Fix-It-Fast scenario, you can set the name of the currently logged-in technician user to be passed as the default value when the component can't pass a technician value.



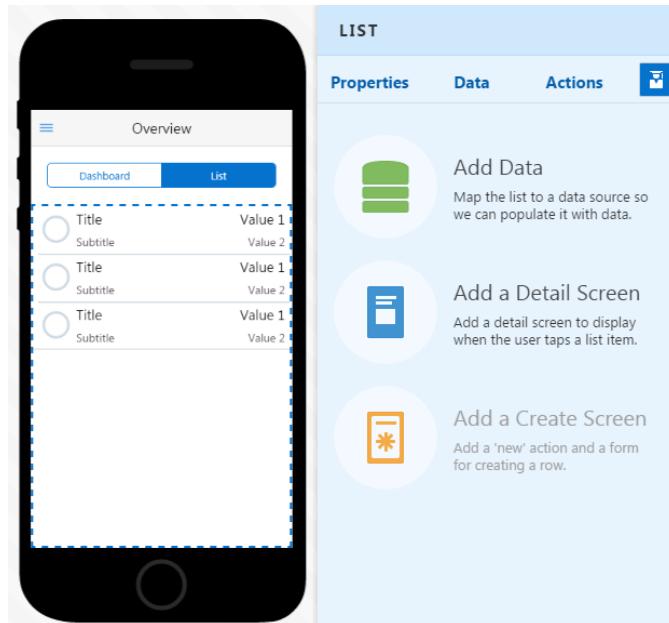
- f. Click **Finish**.

## Using QuickStarts

The QuickStart wizards help you out with your development tasks, like adding data to a UI component, or creating edit and detail screens.

The QuickStarts vary depending on what you're building. If you start off by choosing a List template, for example, the Mobile Designer will help you complete that list by providing wizards that prompt you through adding the data that displays in the list, creating a detail screen that users see when they tap a list item, and adding other functions users expect, like updating an item or creating a new one. You can find out

more by taking a look at [Use Case: Creating a List Using QuickStarts](#). You don't have to follow these QuickStarts – you can dismiss them by clicking the QuickStart icon  if they don't apply or if you can complete the task on your own.



Some of the QuickStarts that you might use include:

If you create...	...Use these QuickStarts
Lists	<ul style="list-style-type: none"><li>• Add Data</li><li>• Add a Detail Screen</li><li>• Add a Create Screen</li></ul>
Forms	<ul style="list-style-type: none"><li>• Add Data</li><li>• Add Edit Mode</li></ul>
Charts, Gauges, Cards and Metrics	Add Data

If you create...	...Use these QuickStarts
Maps	<ul style="list-style-type: none"> <li>• Add Data</li> <li>• Add a Detail Screen</li> </ul> <p><a href="#">Use Case: Adding a Map Component</a> describes how these QuickStarts help you to create a map with tap-to-zoom gestures.</p>

## Use Case: Creating a List Using QuickStarts

You can create a list that not only drills down to a detail screen, but also has create and update screens. This scenario illustrates using the QuickStarts to create a list of open incident reports (IRs) that are assigned to a repair technician. It shows you how to map data to a list that includes:

- Images that identify incident's priority (high, medium, and low).
- A short description of the incident.
- The status of the incident (open or closed).
- The date that the IR was created.
- The first and last name of the customer.

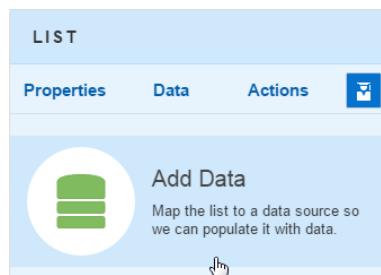
The repair technicians can create a new incident report from this screen. Clicking a row in this list opens a detail page, which provides a longer description of the incident and additional details. From this screen, users can update the IR using an edit screen.

To get started with this list, you'd add a List component to the Preview. After adding this component (or having chosen a list template as your app's first page) the Mobile Designer presents a task list of QuickStarts in the order that you'd need them. Because this scenario demonstrates how to create a navigable list, it demonstrates how to use the List component Quickstarts:

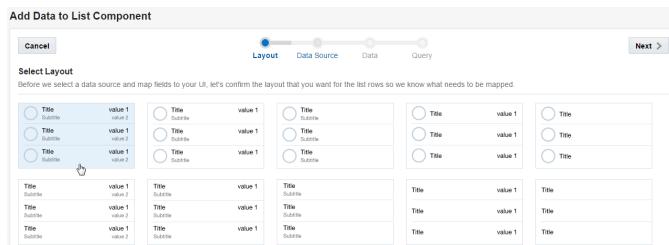
- Add Data
- Add a Detail Screen
- Add a Create Screen

### Step 1: Bind the List to Data

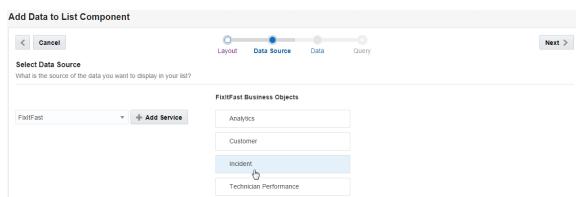
1. Click the **Add Data** QuickStart to open the wizard.



- For a List component, the wizard's first page presents list layout options. Because the mapping for the priority image, short description, and other information for the list requires all of the available fields described in [List Layouts](#), the layout option with the Graphic, Title, Subtitle, Value 1 and Value 2 fields (which is the default layout) is your best choice. Clicking **Next** opens the Data Source page of the wizard.

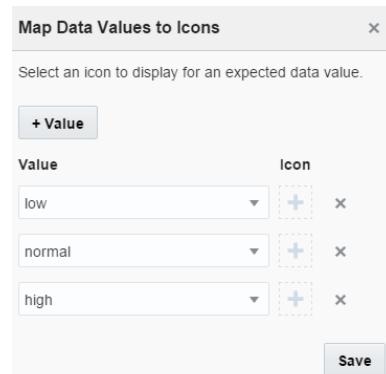


- Clicking **Add Service** (+) opens the Service Catalog. In this case, the service that provides the business objects needed to return data to the list and its child screens is *FixItFast*. The *Incidents* business object backs the field mappings of the list. This business object uses a collection action to return its records. It does not include all of the fields from service, or even all of the fields that describe an incident. Instead, it provides a subset of fields that is appropriate to the overview nature of a top-level list. With the service and business object selected, click **Next** to map the values using the Data page. The Incident object's fields display in the Data page's Data Source category, because it is the currently selected business object.

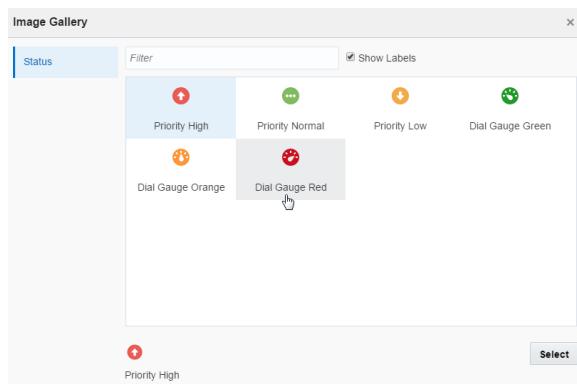


- Drag values from the Incident section of the page to the List Component Fields section of the page. Start off by dragging the priority value into the Graphic Source field:

- Click **Map values to icons** to open the Map Data Values to Icons dialog.

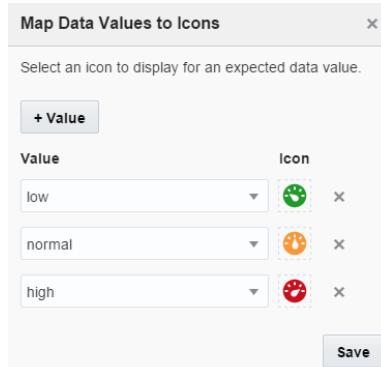


b. Clicking Add (  ) opens the Image Gallery.



As noted in the [Using the Image Gallery](#), the images available in the Gallery depend on context. Within the Data Mapper, the Image Gallery displays images provided by the business object.

Pick the images for the high, medium, and low values:



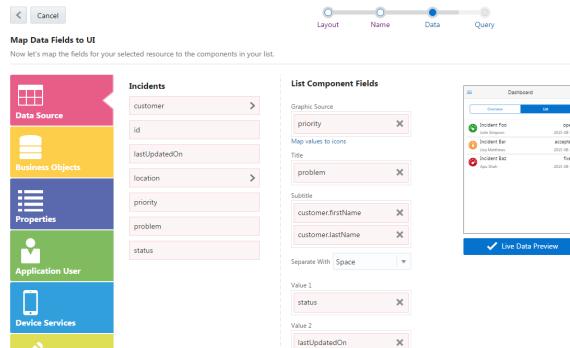
c. The *Incidents* business object provides most of the values for the list. Within the FixitFast service, this business object also has some reference objects: *Customer* and *Location*, which are displayed in the Data page (and the Data Mapper). As described in [Getting to Know Business Objects](#), these objects provide complementary information (in this case, the customer's first and last name) and are peers.

 **Tip:**

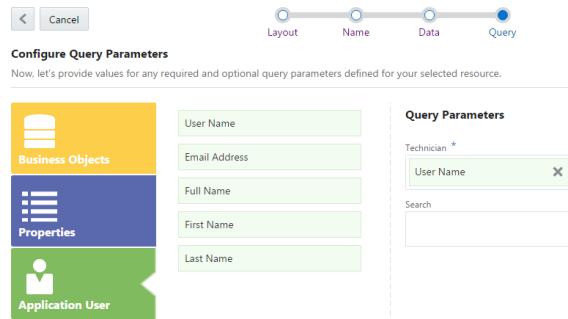
In this use case, the referenced objects provide the fields that we need to complete the screen from a single business object. However, this might not always be true of the business object that you're currently using to populate your screen's data. Because you can choose fields from other business objects (and other categories), your field mappings aren't limited to the business object that you've chosen for the screen. Or, in the case of detail or other child screens, you don't need to use the business object fields available to the parent screen.



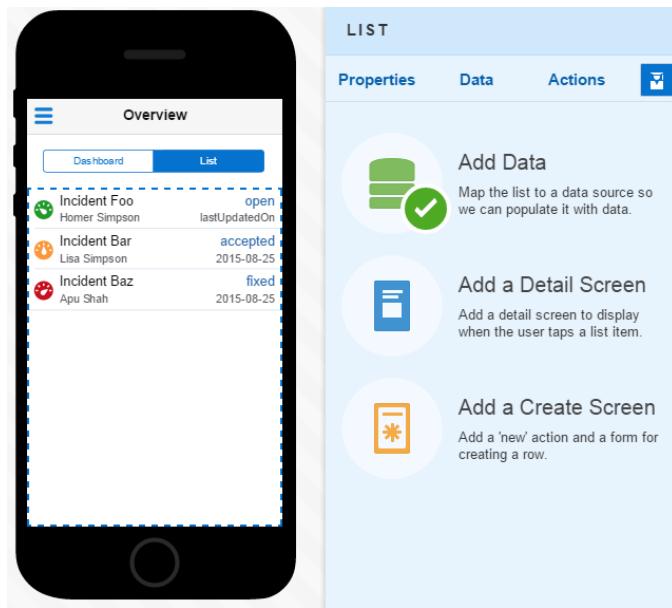
Business object	Value	Field
Incidents	problem	Title
Incidents	status	Value 1
Incidents	lastUpdatedOn	Value 2
Customer (a referenced business object)	firstname, lastname (two values, separated by a space)	Subtitle



- With the field mapping complete, click **Next** to choose the filter criteria. In this case, it's user name, because the list returns IRs assigned to a particular field technician.

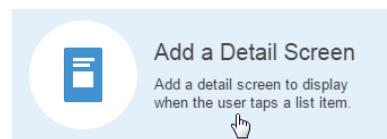


5. Clicking **Finish** completes the wizard. Back in the Properties Inspector, the Add Data QuickStart is checked off. You're now ready for the next QuickStart wizard, Add a Detail Screen.

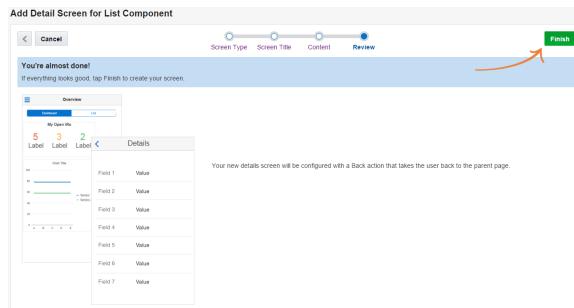


## Step 2: Create the Detail Screen

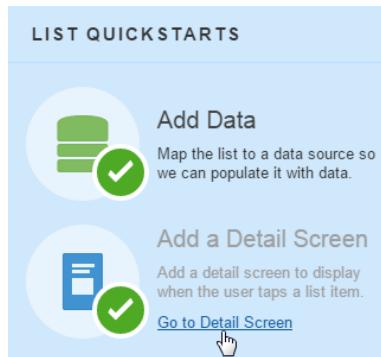
1. Click the **Add a Detail Screen** QuickStart.



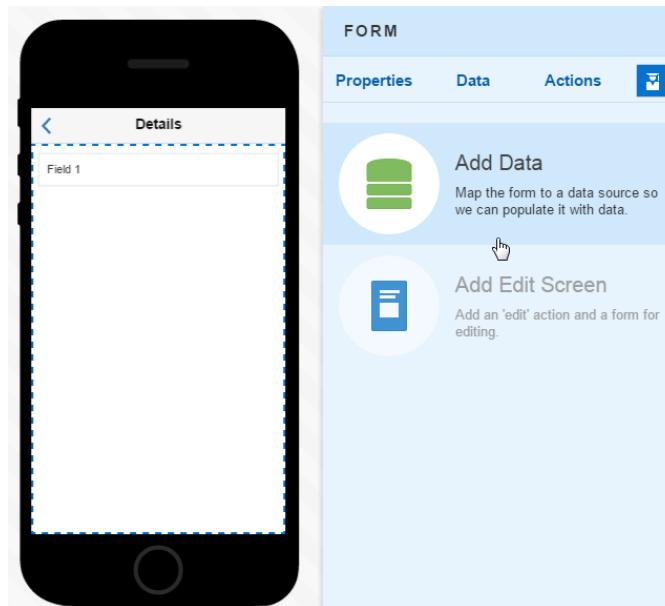
2. The QuickStart for a detail screen is a four-step wizard, similar to the one that you used to create the first screen of your app. This wizard steps you through creating the basic structure of the detail page by adding a name, setting the basic layout, and choosing how your content displays. You can choose from among a simple screen, a list, a dashboard, or a form. Because this detail page not only displays the full record of the item selected in the list (which wouldn't work well in a list layout), but also serves as the source for the child create and edit pages, choose the Form layout. Click **Finish**.



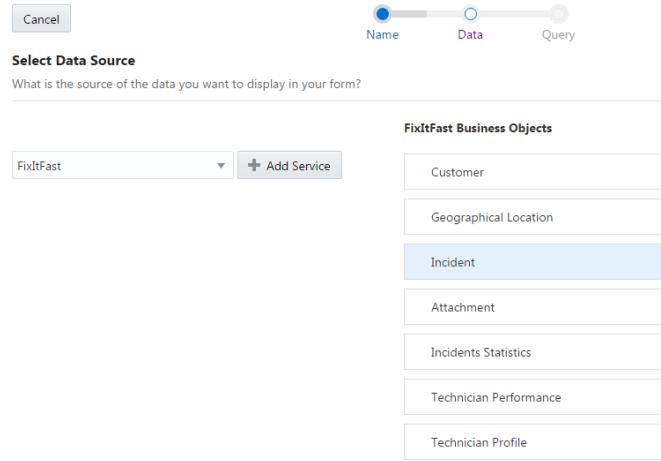
- With the skeleton of the form completed, click **Go to Detail Screen**.



- Because you need to add the fields to your form that app users will view and update, click the **Add Data** QuickStart to add these fields using the Add Data wizard.



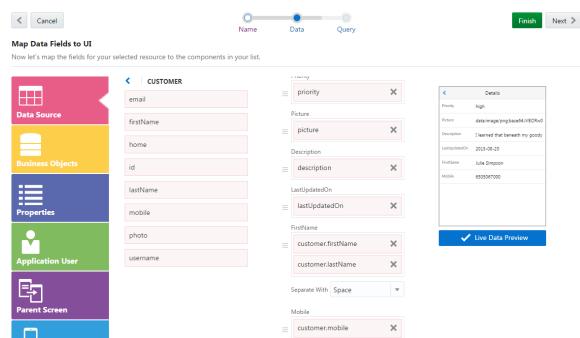
- Since you already chose the FixitFast service when you created the list in the previous step, you can go straight to choosing a business object for the form. In this case, you're going to choose additional fields from the Incident business object. While some of these fields will be read-only, others can be updated.



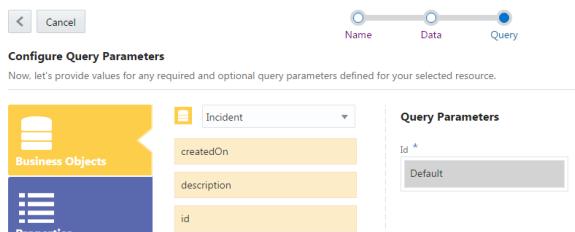
6. Map the Incident field values to the Form component fields. You can re-order these fields here or later on in the Properties Inspector. Keep in mind that you enable the fields to display properly by choosing the field type. When you're done with the field mapping, click **Next** to complete the Add Data wizard by selecting a filter.

 **Tip:**

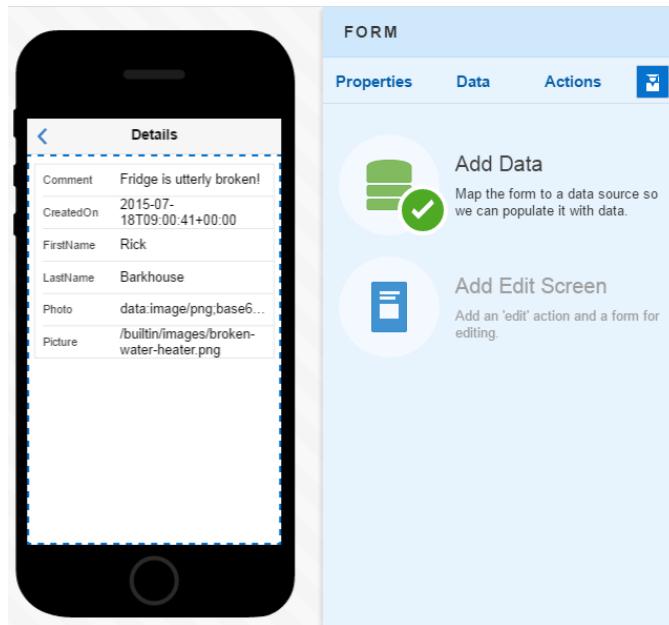
You can drag the properties individually to add as many fields to a Form component as you need, or click **Add All Fields** (). You can click **Delete** () to remove the fields that you don't need. For more information on form fields, see [Forms](#).



7. MAX volunteers the ID as the filter. In this case, the ID query parameter is provided by the service, so MAX notes it as *Default*. You could replace this value, but since the ID ties the incident fields that display in this screen to the row that users click in the list screen, you can accept this value by clicking **Finish**.



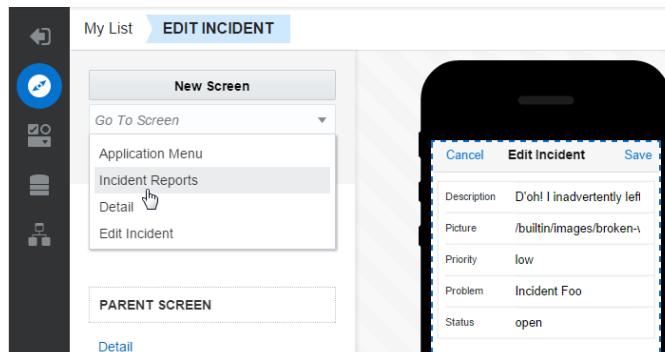
The details screen, complete with a Back action (◀), has been created. With the Add Data QuickStart checked off, you're ready to create an edit page.



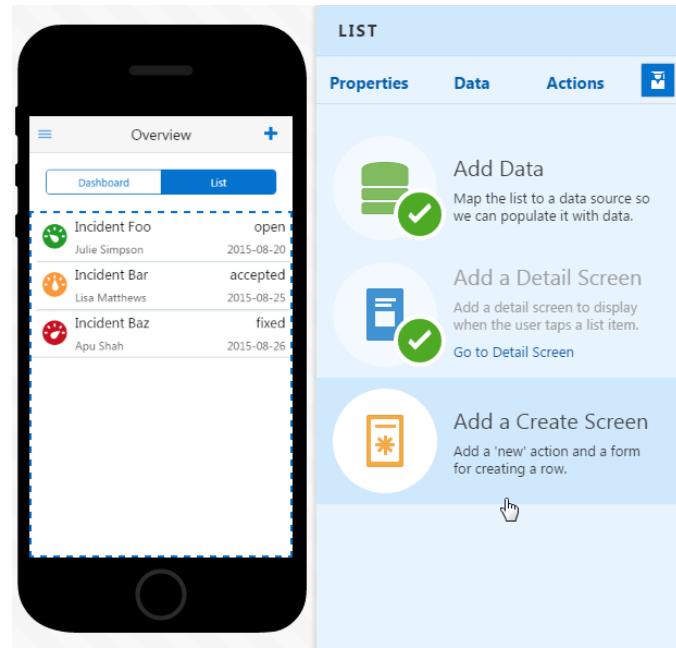
### Step 3: Adding a Create Screen

With your detail page created, you can add create and edit pages to the list page. Before you can add the create page, you need to get back to the list page and its QuickStarts. To do this:

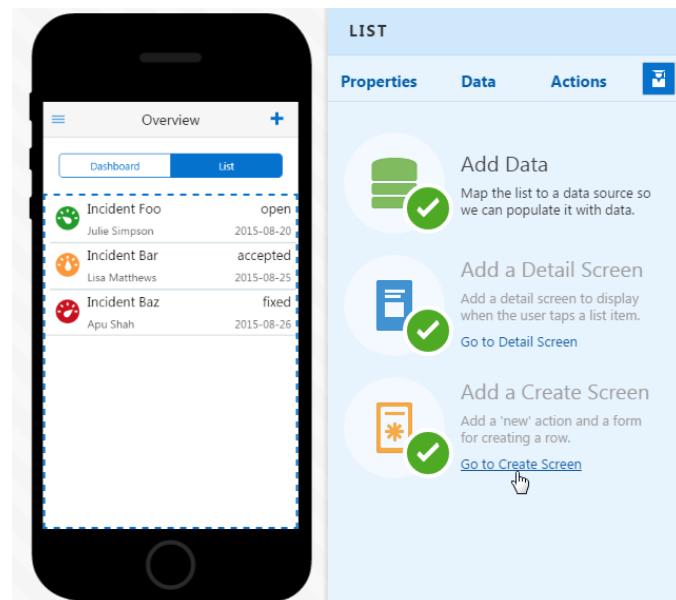
1. Click **Application Screens** (➕) in the left navbar.
2. From the Go to Screen menu, pick the list page. In the following illustration, the list page is called Incident Reports.



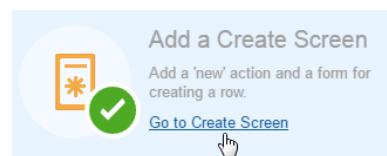
3. Now that you're back to the list page, click the **Add a Create Screen** QuickStart.



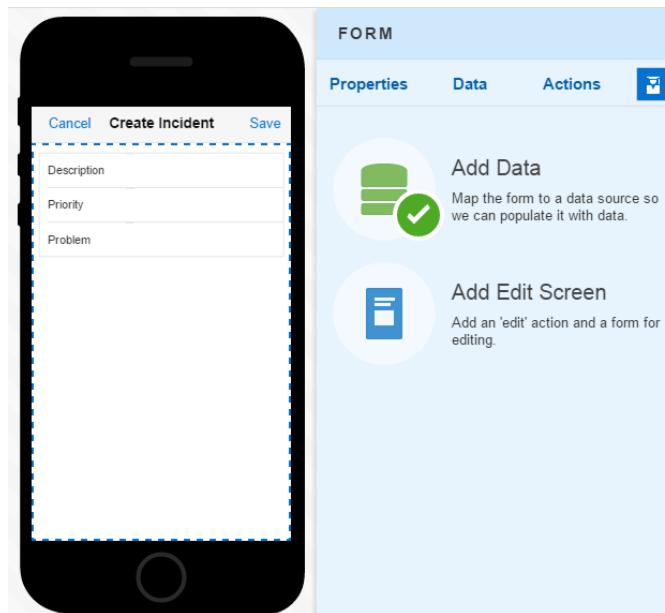
The list page now has a create action (+).



4. To open the create page, click **Go to Create Screen**.



The create screen appears in the Preview, complete with Cancel and Save actions. The create screen is a Form component because it allows user input. You don't have to add map data for this screen, because the Mobile Designer populates it with the fields from the Incident Activities business object that accept user input, like *Description*, *Priority*, and *Problem*.



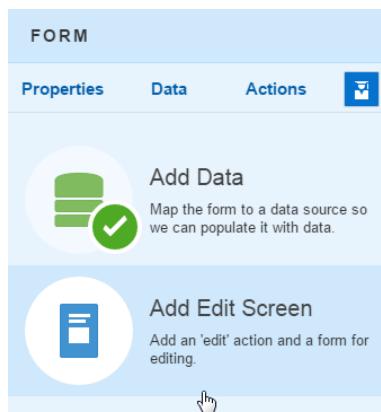
 **Note:**

Because this screen is used for inputting data, there's no need to wire it up to any data.

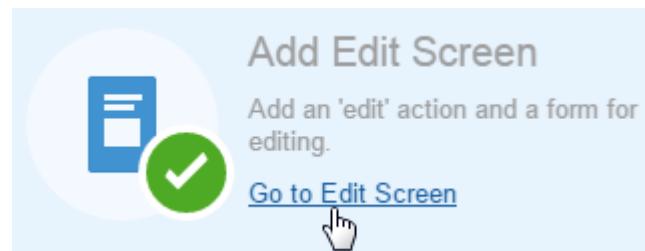
#### Step 4: Create the Edit Screen

With the create page added to the application flow, you can now add a corresponding edit page, a form that allows user input.

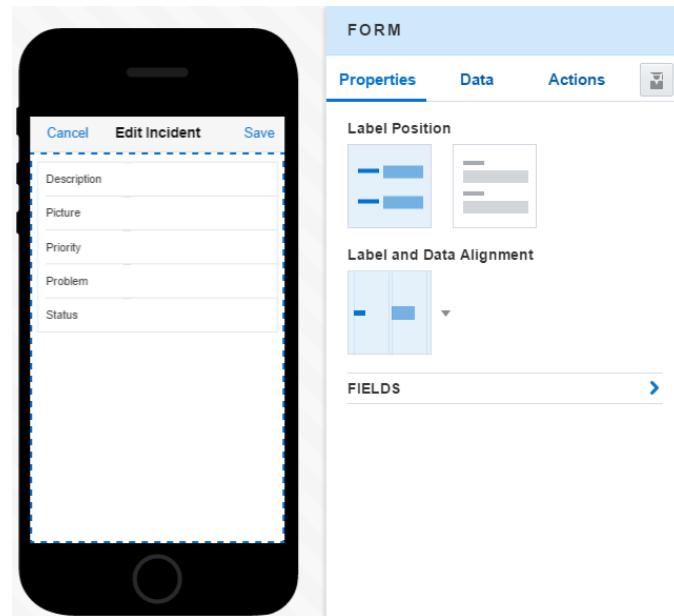
1. Add the page by clicking **Add Edit Screen**.



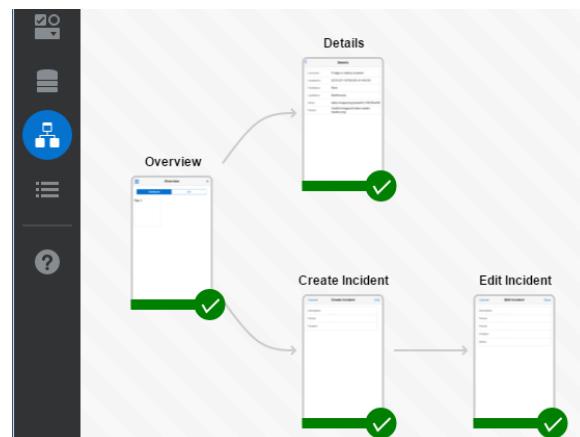
2. Open the edit screen by clicking **Go to Edit Screen**.



The Mobile Designer creates your edit page, which has Cancel and Save actions at the top. The edit page includes the fields selected for the details page, but only the ones that accept user input.



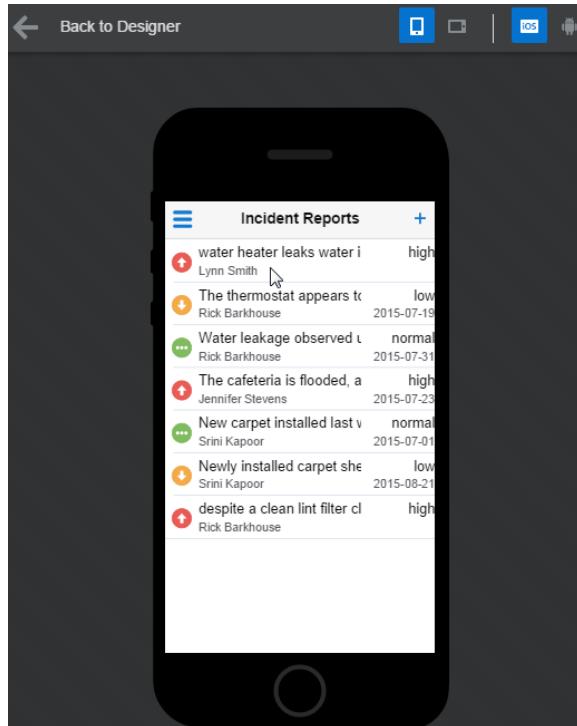
3. Take a look at the page flow you've just created by clicking **Screen Flow** (  ) in the left nav bar.



 **Tip:**

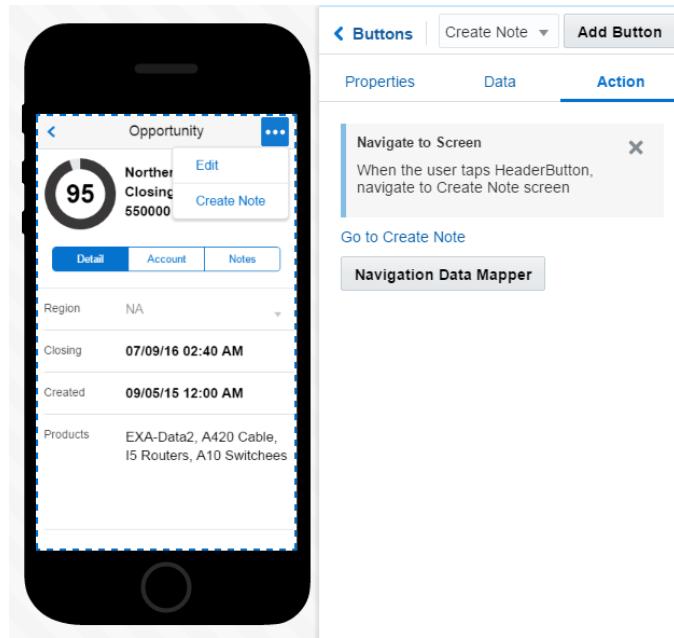
You can open a page in the Preview and Properties Inspector by double-clicking it.

4. Click **Test** (▶) to test your app with live data and see how the Back, Edit, and Create actions work. To find out more, including logging in as a test user, see [Previewing Your App](#)



## How Do I Add Another Create or Edit Screen?

Your app flow may include more than one set of create and update screens. If you've already completed the QuickStart wizards for these screens, you can add new ones when you create a button with a Navigate to Screen action. By adding a new action, you can equip your screen's subtabs with their own create and update functions that are based on a business object that's different from the one that's already been used. [Adding Screen Actions](#) describes adding a button to the screen header along with an overview of mapping the navigation.



## Using the Image Gallery

The Image Gallery is your repository for custom and default images.

The Image Gallery lets you replace the default images for the icons used throughout your app. You can add your own images or select from the ones that the Mobile Designer provides to change the following:

- The application launch icon
- The application menu icon
- The tab icons

You can select these images by clicking the tabs of the Image Gallery. Your image selection depends on where you open the Image Gallery: when you open the Image Gallery from the Application Settings page, for example, you'll see the Default tab that

 has the default image that's used for the launch icon (). You can also see the Custom tab, which opens the Custom page. You can use this page to add an image to replace the default launch icon image. When you open the Image Gallery from the Properties Inspector, you see only the Custom tab.

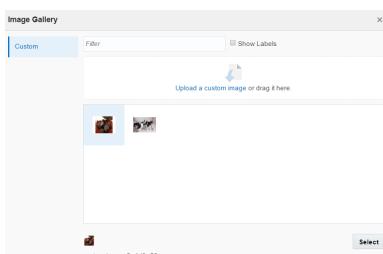
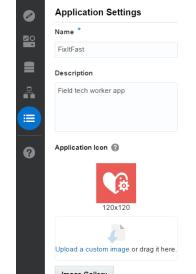
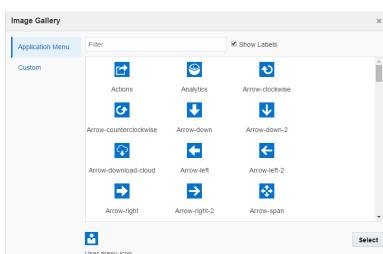
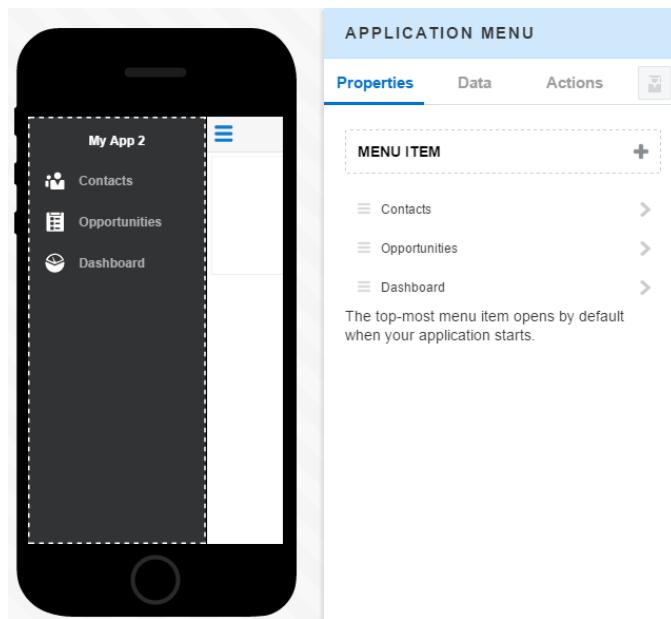
Image Gallery Tab	Images	Where Can I Access These Images?
Custom	<p>These are images that you can add.</p> 	<p>You can access the Custom tab by:</p> <ul style="list-style-type: none"> <li>Clicking the Image Gallery button (  ) that appears in the Data tab for the Properties Inspector for the Image component.</li> <li>Clicking the Application Settings (  ) icon in the left navbar to open the Application Settings page.</li> </ul> 
Application Menu	<p>Images used for the application menu.</p> 	<p>This page also lets you change the display name of your app.</p> <ul style="list-style-type: none"> <li>The Custom tab also accompanies the Application Menu tabs and Functional tabs. You can add new application menu and Functional Tab images by uploading them (or by dragging them in) to the Custom page.</li> </ul> <ol style="list-style-type: none"> <li>Select Application Menu in the Preview.</li> <li>In the Properties Inspector, click the menu item label icon (  ).</li> </ol> <p>You can replace the application menu with an image that you've added to the Custom page.</p>

Image Gallery Tab	Images	Where Can I Access These Images?
Functional	Images used for the tabs lining the bottom of the Screen with Bottom Tabs template.	<ol style="list-style-type: none"> <li>1. Select Screen with Bottom Tabs in the Preview.</li> <li>2. In the Properties Inspector, click the image icon and then choose <b>Select Icon</b> (  ).</li> </ol> <p>You can replace the application menu with an image that you've added from the Custom page.</p>
Default	Shows the image for the launch icon that's added when you created the first page of your app. For more information, see <a href="#">Creating the First Screen of Your App</a>	From the Application Settings page. You can replace the default icon image with an image that you've added from the Custom page
Business Object-Specific Images	The images returned by a business object.	From the Data Mapper. To find out more, see <a href="#">Use Case: Creating a List Using QuickStarts</a> .

## Managing Your Screens

Use the Properties Inspector for the Application Menu component to add screens to your app and reorder them.

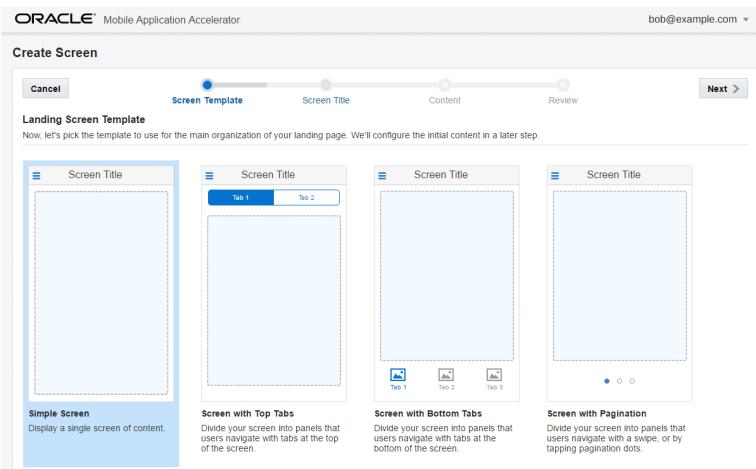
You can open the Properties Inspector for this component by clicking the menu icon (≡) in the Preview.



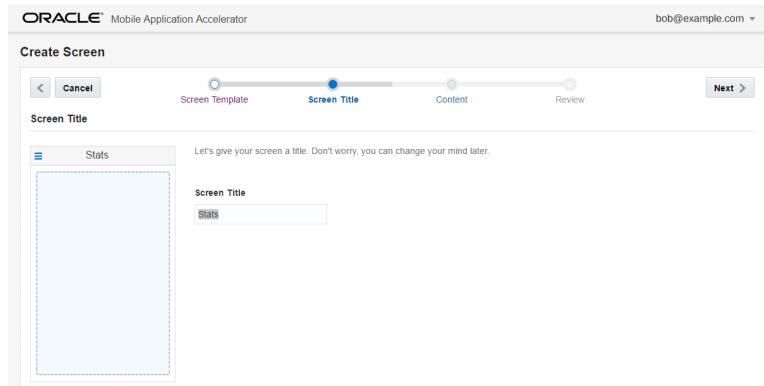
## Adding Landing Screens to Your App

To add a screen to your app:

1. Click the menu (≡) in the Preview to open the Application Menu component.
2. Click **Add** (+).
3. Complete the screen creation wizard:
  - a. Pick the basic format for the screen and then click **Next**.



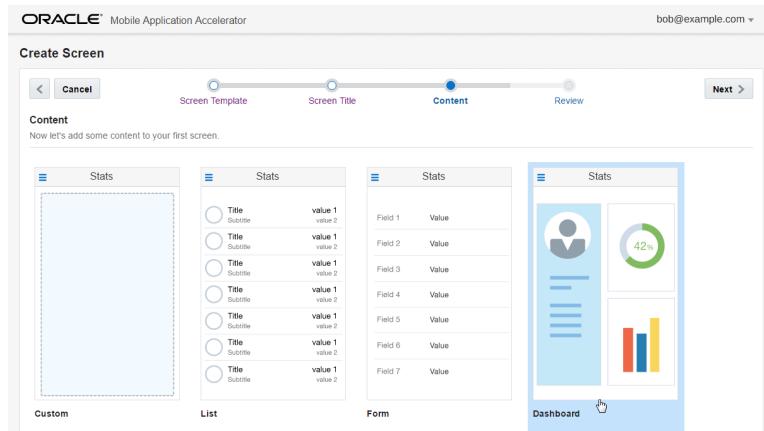
- b. Next, enter a title and then click **Next**.



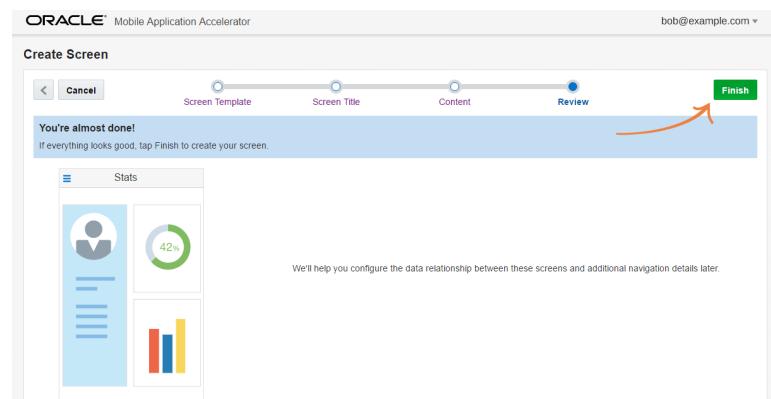
**Tip:**

If your landing screen includes tabs, you can create them by replacing the *Tab 1* , *Tab 2*, and *Tab Title* text in the *Tabs* fields with your own names. You can add up to five bottom tabs or four top tabs.

c. Select the layout option for your content (that is, for the UI components you want to add and ultimately, for the data that they display). Click **Next**



4. Click **Finish** to complete the screen.

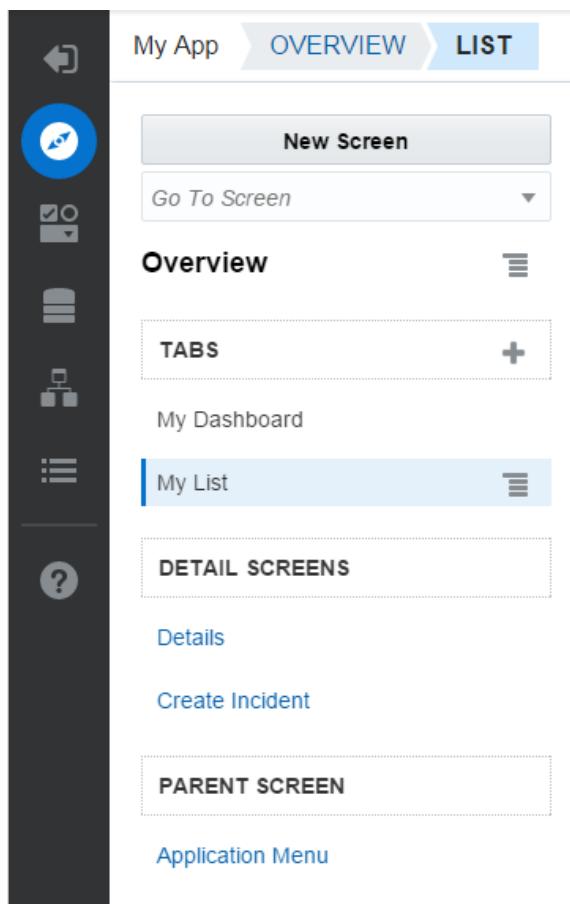


 **Tip:**

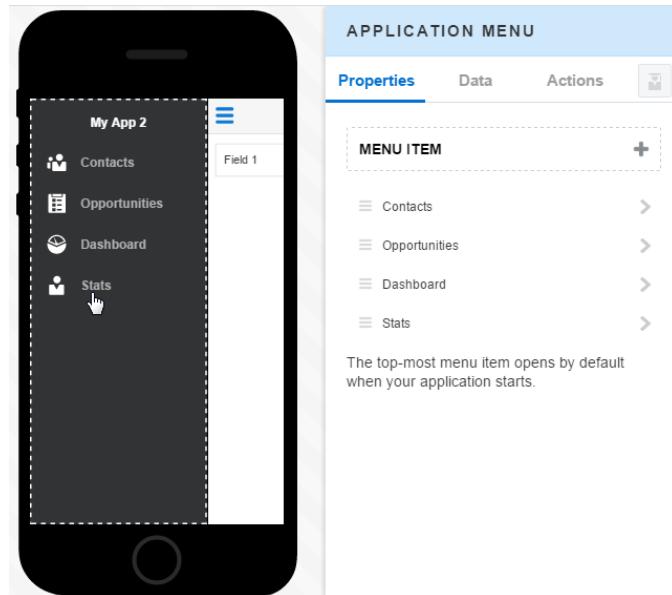
If you want to change something, click **Previous** () or click one of the stops along the progress trail at the top of the page.

You can also add new landing screens to the Application Menu component by

clicking Application Screens () icon in the left navbar to open the Application Screens page. Click **New Screen** and then complete the screen creation wizard.



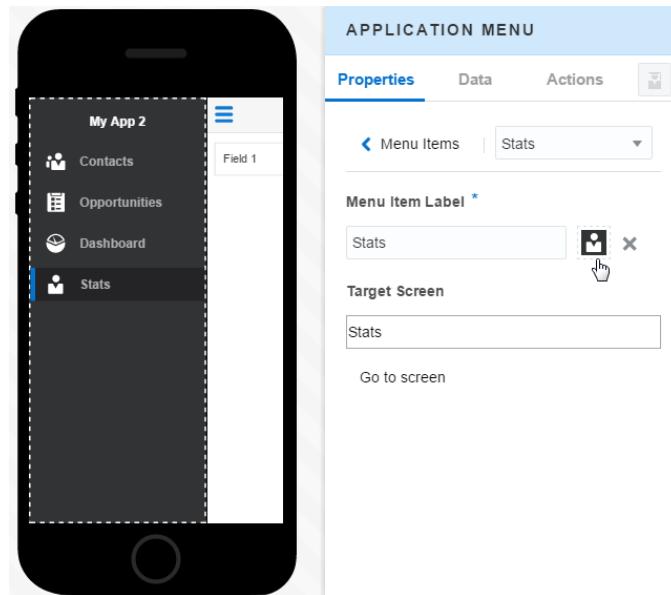
5. After you complete the screen and return to the Mobile Designer, you can access the display options that let you change the name and icon for your new screen by either:
  - Selecting the screen in the Preview.
  - Selecting the screen in the Properties Inspector. You can also click **Next** ()



6. To change the display name of the screen, enter a new name in the Menu Item Label field. This is a required field.

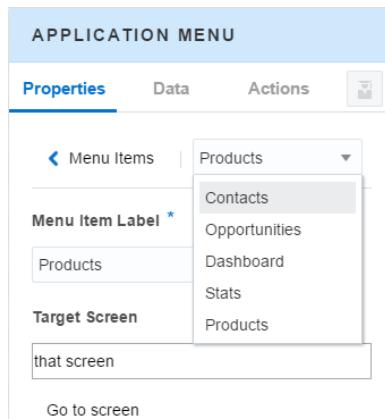


7. To change the icon, click the menu item label icon ( ) and then choose a new one from the Image Gallery. To find more, see [Using the Image Gallery](#).



8. If needed, enter a new name in the Menu Item Label field. This is a required field.

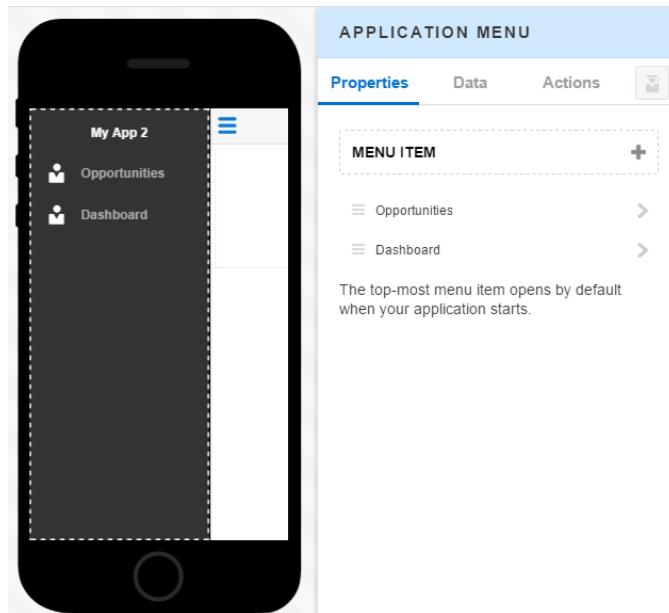
Clicking **Go to screen** ( [Go to screen](#) ) opens the screen in the Preview and the Properties Inspector. From there, you can develop it further by adding components, data, and actions. You can change the icon and display name for other screens in your app by first selecting them from the Menu Items list and then using the options in this page.



## Rearranging Screens

To change the first (or landing) screen of your app:

1. In the Preview, click the menu icon (≡) to open the Application Menu component in the Properties Inspector.



2. Use the move icon (≡) to reorder your screens.

## Video: Mobile Application Accelerator (MAX) Demo

To see the MAX Designer in action, take a look at this video:



# 4

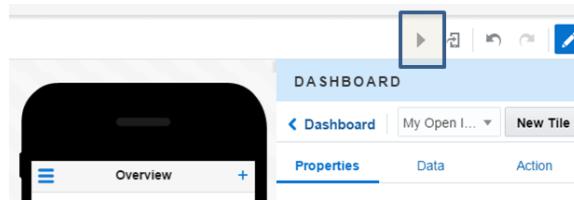
## Building and Testing Your Apps

You can test your apps using a Web-based simulator or on a phone.

### Previewing Your App

You can preview your app in a hosted simulator or on a device at any point after you've created its first screen.

The Test tool lets you use the simulator to see how your UI displays live data. You can also test out screen navigation and the actions that you've configured for UI components.

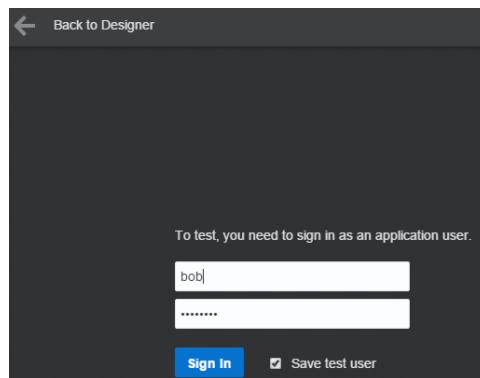


#### Note:

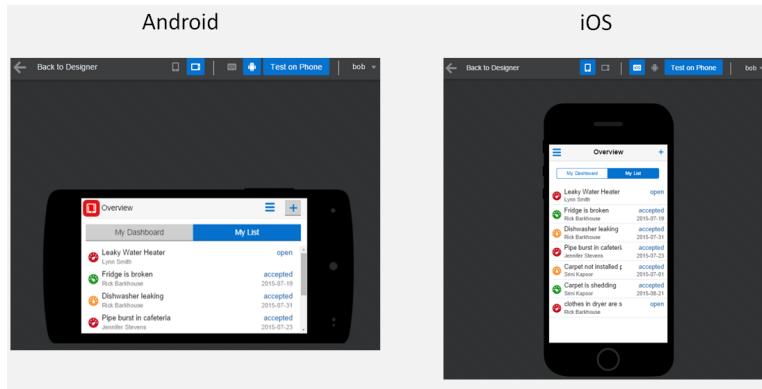
You need a test user account to preview your app with live data. Contact your administrator if you don't have one.

To preview your app:

1. Click **Test** (▶).
2. Enter the user name and password for your test user account and then click **Sign In**.

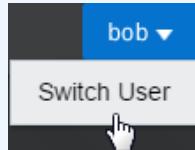


3. Test your app's functions, such as the overall page flow, the behavior of a single field, or the create and back actions. While you're testing, toggle the orientation and the iOS and Android options on and off. When you're done, click **Back to Designer** ( [Back to Designer](#)) to resume your work.



**Note:**

Clicking **Switch User** logs you out.



4. If you want to continue to work on your app, click **Back to Designer** ( [Back to Designer](#)). Otherwise, you're ready to see how your app works on an actual phone. See [Testing Your App](#) to find out more.

## Testing Your App

Testing your app on an actual device gives you a better picture of how your app functions on different platforms, operating systems, and device models. It's where the rubber hits the road, so to speak. Unlike an emulator, which approximates the user experience using a mouse, you can use actual tap and swipe gestures to navigate through your app's screens when you run it on a real smartphone. You can also find out how your app interacts with actual device services, like the phone or the contact list. You can test your app on a device using the MAX App, or you can also test it as a native, standalone app on an iPhone or Android phone.

 **Note:**

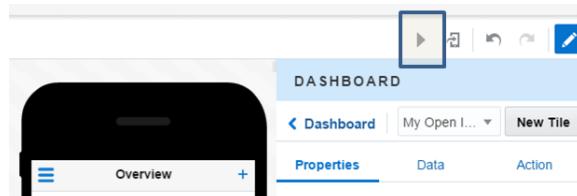
- Before you can begin this real-world testing using the MAX App, you need to install the MAX App onto any device that you're planning to use. Your test group (that is, anyone who's participating in the testing) must also have the MAX App installed. See [Installing the MAX App](#)
- All app users (app testers and app users alike) must have user accounts.
- You need a test user account to generate the QR code associated with the test version of the app.

To test the app:

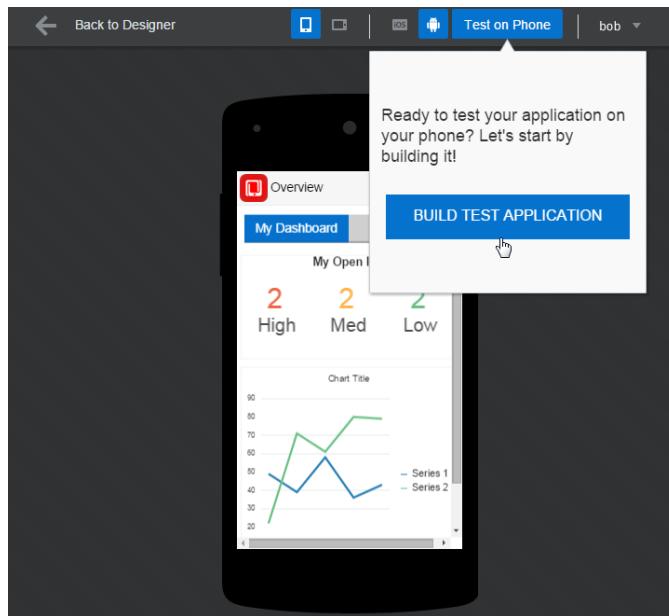
1. In the left navbar, click **Screen Flow** () to make sure that there are no problems.



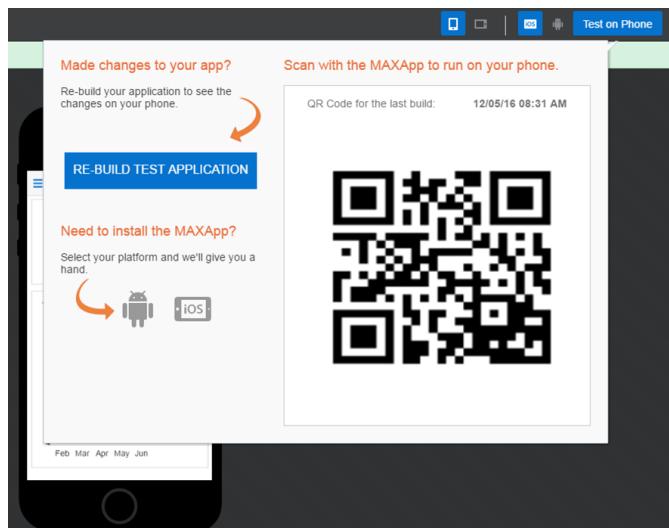
2. Click **Test** ()



3. If you signed out when you last previewed your app, enter the user name and password for your test user account.
4. Click **Test on Phone** () and then **Build Test Application**.



The Mobile Designer generates a QR code when the build finishes.

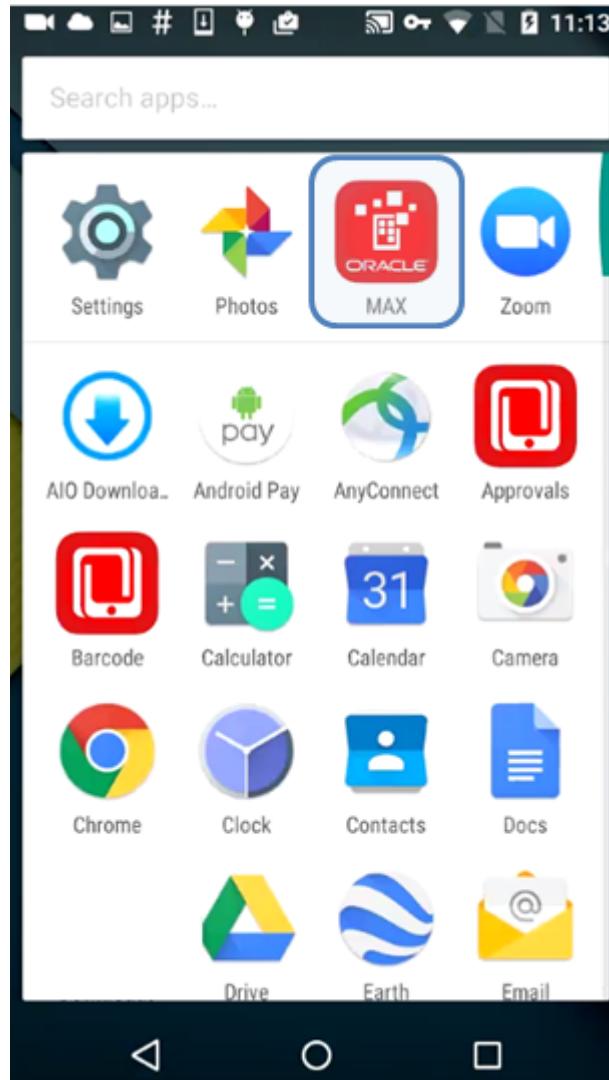


5. Scan the QR code using the MAX App.

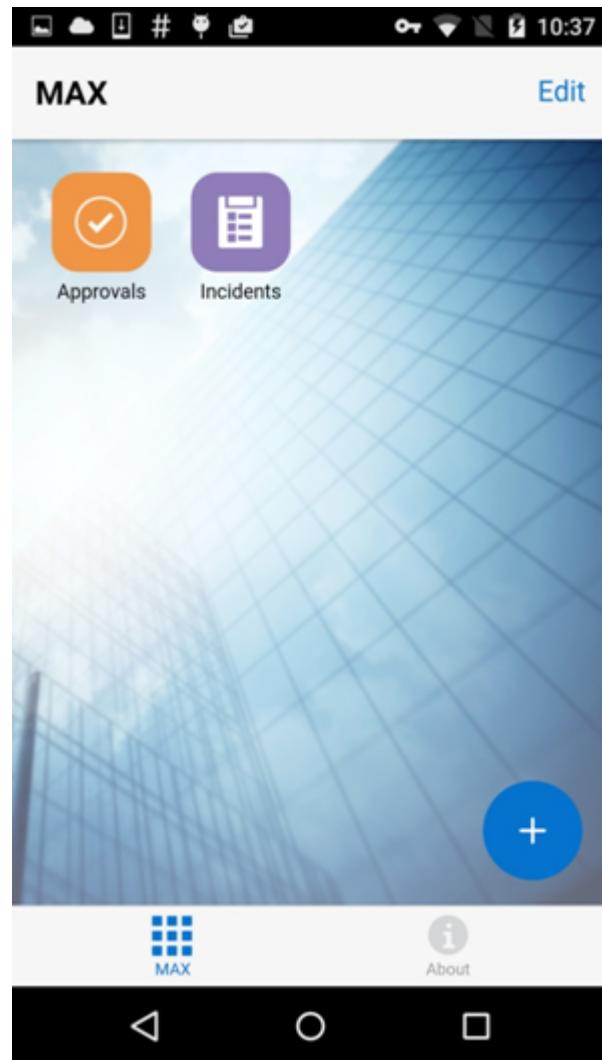
**! Important:**

This QR code is for testing apps that are under development. Share it with your test group. When you publish the app later on, MAX will generate a different QR code, one associated with the published (or finished) version of your app. It's the one that you'll use to distribute all versions of your app to its users. The test and production versions of the app can't coexist within the MAX App: when you scan the QR code for the published version of your app, the published app replaces the test version that's currently in the MAX App.

6. On your device, open the MAX App (  ).



7. Click Scan (  ).



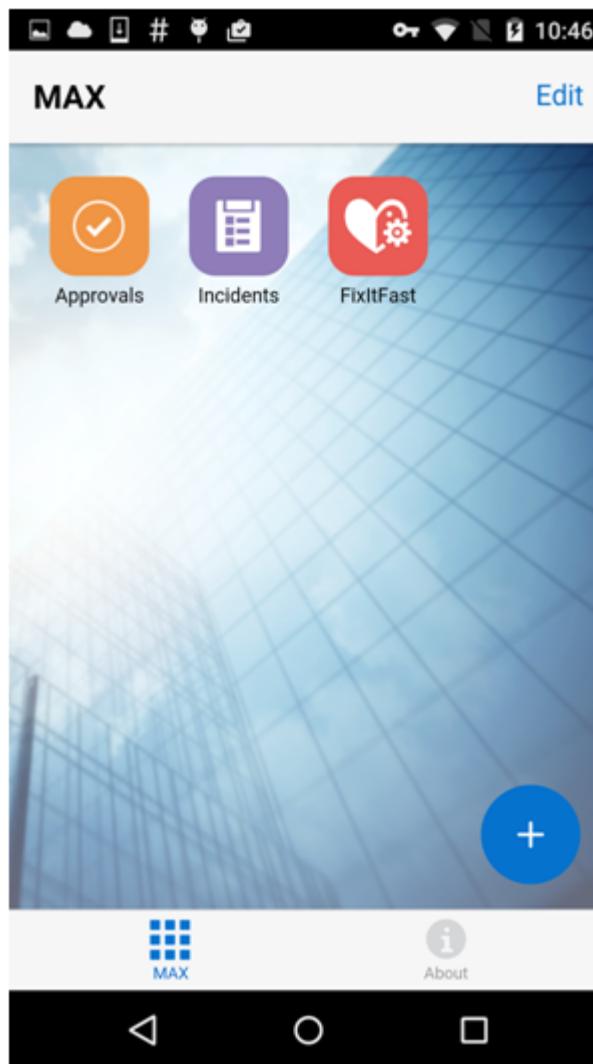
8. Point your phone at the test QR code to download the app.



9. Log into the app by entering your user name and password. The test app appears in the MAX App.

 **Tip:**

You can also login using your test user name and password.



10. Repeat these steps when you're ready for another round of testing.

 **Tip:**

You might see errors if plan to run the both the test and production version of the app on the same phone. To avoid this, delete and then reinstall the MAX App before running the production version of the app. If you're going to revise the app, then you need to once again delete and reinstall the MAX app before you test a new version of the app.

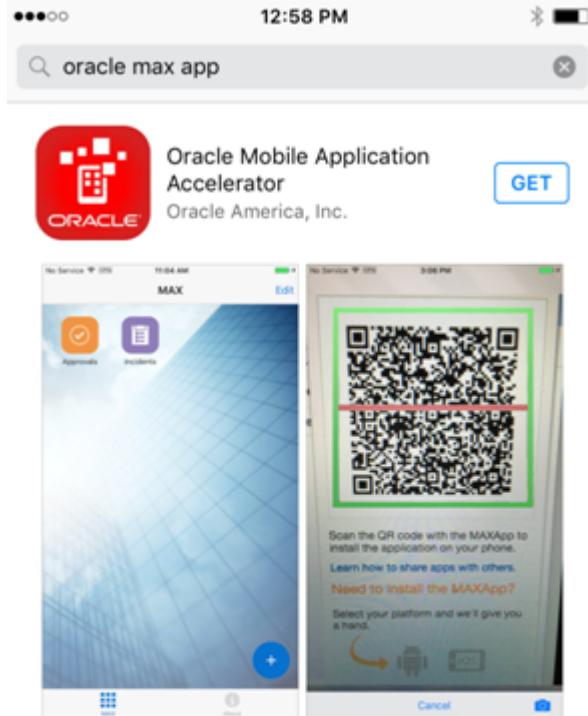
## Installing the MAX App

You can only run your mobile app on an actual device using the MAX App, a container that allows your app to not only display its content and leverage device services, but also behave identically to other apps that were developed specifically for iOS or Android devices. While the MAX App provides a consistent user experience, it also allows users to install your app. Because of this, all of your users (app testers and actual users alike) need to install the MAX app on their devices. You can install the MAX App from both the App Store and from Google Play.

## Installing the MAX App on iPhones

To install the MAX App from the App Store:

1. On your iPhone, open the App Store.
2. Search for Oracle MAX App.



3. Tap **Get** and then **Install**.

 **Note:**

You might be prompted to sign into iTunes if you're not already signed in.

4. When the download completes, tap **Open**.
5. When prompted, allow MAX to send you notifications.
6. Read through the end user terms and conditions and then tap **Agree**.

7. Enter your login information:
  - Host
  - Username
  - Password
8. Tap **Sign In**.

## Installing the MAX App on Android Devices

To install the MAX App from Google Play:

1. On your phone, open Google Play.
2. In the Play Store, tap **Apps**.
3. Search for Oracle MAX App.
4. Tap **Install**.
5. When prompted, allow MAX to send you notifications.
6. Read through the end user terms and conditions and then tap **Agree**.
7. Enter your login information:
  - Host
  - Username
  - Password
8. Tap **Sign In**.

## MAX App Quick Reference

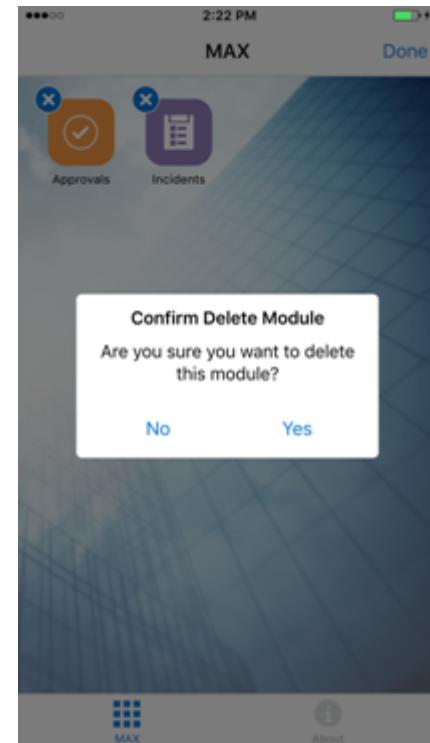
---

To do this...	Do this...
Download an app	 <ol style="list-style-type: none"><li>1. Click <b>Scan</b> ( ).</li><li>2. Point your phone at the QR code to download the app.</li><li>3. Enter your user name and password. See <a href="#">Testing Your App</a> to find out more.</li></ol>

---

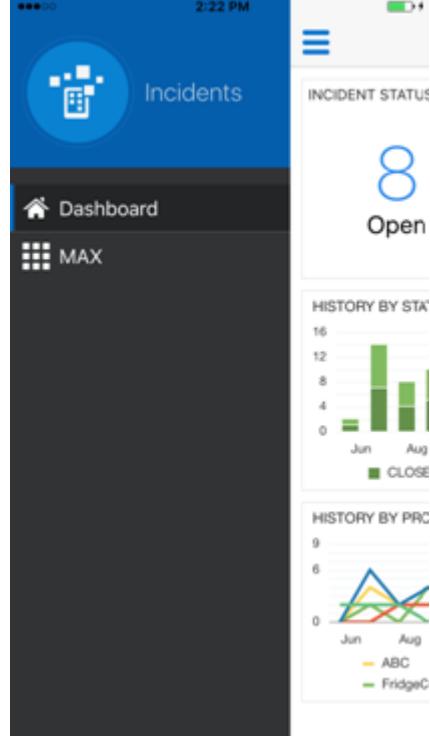
To do this...	Do this...
Delete an app	<ol style="list-style-type: none"><li>1. In the springboard, click <b>Edit</b>.</li></ol>

1. In the springboard, click **Edit**.
2. Tap the app (  ).
3. Click **Yes** in the Confirm Delete Module dialog.



4. Click **Done**.

---

To do this...	Do this...
Switch apps	<ol style="list-style-type: none"><li>1. From within an app, click menu (≡).</li><li>2. Click the MAX springboard icon (grid) and then open the app.</li></ol>  <p>The screenshot shows the MAX app's dashboard. At the top, there's a blue header with the text 'Incidents' and a circular icon. Below the header, the word 'Dashboard' is visible. On the right side of the dashboard, there's a summary card with the number '8' and the word 'Open'. Below this, there are two charts: 'HISTORY BY STA' (a bar chart showing data for Jun and Aug) and 'HISTORY BY PRC' (a line chart showing data for Jun and Aug for categories ABC and FridgeC).</p> <p>Navigate back to the springboard</p> <p>Click the MAX springboard icon (grid).</p>

---

## Sharing Your App

To share your app, you need to distribute a QR code to your users along with the name of your app.

For each app, you'll have two QR codes: one used for testing and one used distributing the published version of your app. These codes reflect your app in its unpublished or published state, respectively. They are dedicated to your app, meaning that they don't change. Depending on your intent (testing or distributing), you'll always use one or the other. Your app can run within the MAX App, or as a standalone, native app on iOS and Android phones.

### ! Important:

If you distribute your app through the MAX App, be sure that each user has a user account and that each user has the MAX App installed on their devices. Your mobile app can only run in the MAX App. Both your test group and your user group will install your app and run it using the MAX App. To find out more, see [Installing the MAX App](#).

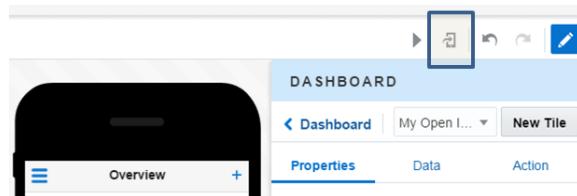
### Sharing Your App with Your Testing Group

You don't need to publish an app to test it on a device. As described in [Testing Your App](#), the Mobile Designer generates a QR code that you use for testing purposes. You can distribute this code by sending a picture of it to your testing group. They will scan this code every time they need to install a new iteration of this app.

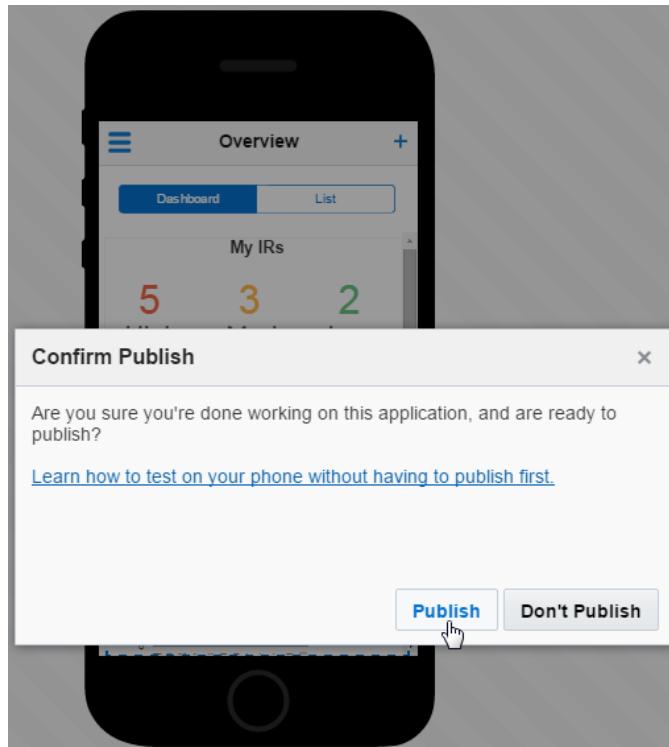
### Sharing Your App with Your User Group

To get the QR code used for distributing the release version of your app to your user group, you must first publish it. To do this:

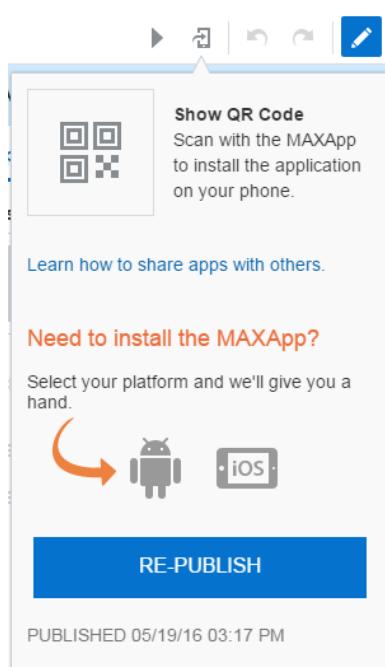
1. Click **Publish** (  ).



2. Click the Publish button ( **PUBLISH** ).
3. If you're done testing are now ready to publish the app, click **Publish** in the confirmation dialog.

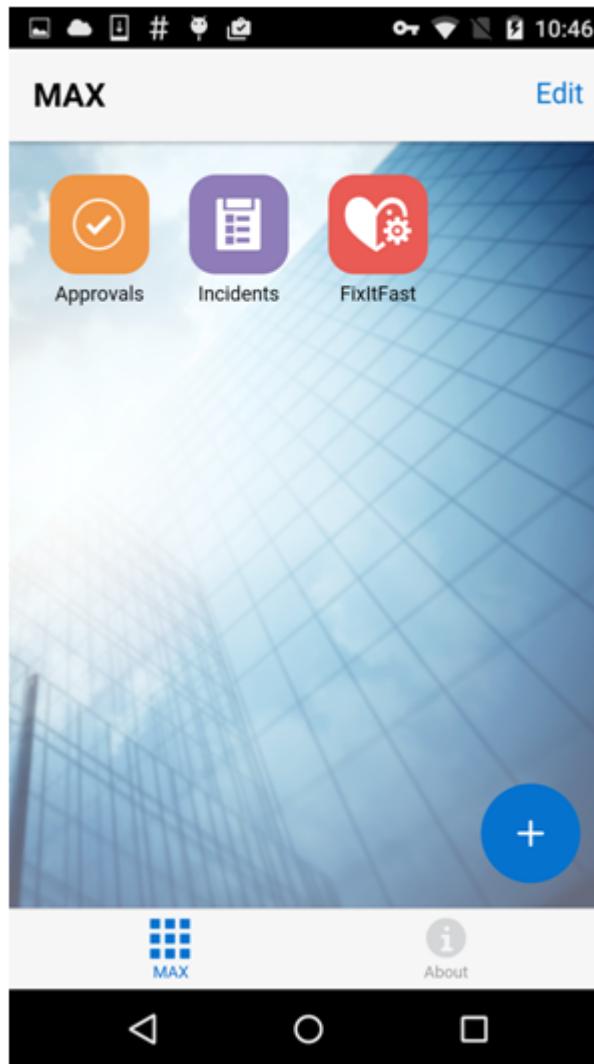


The Mobile Designer generates a QR code for the published app.



4. Click **Show QR Code** (  ) to display the QR code.
5. Send a picture of this QR code to your user group. They will scan this code to install any new versions of this app.

6. After they launch the MAX APP, your app users click **Scan** (  ). When prompted, they allow the MAX app access to the device camera and then focus the scanner to read the QR code. The MAX app then downloads the app to their smartphones. All downloaded apps display in the MAX App springboard.



# 5

## Managing Your Apps

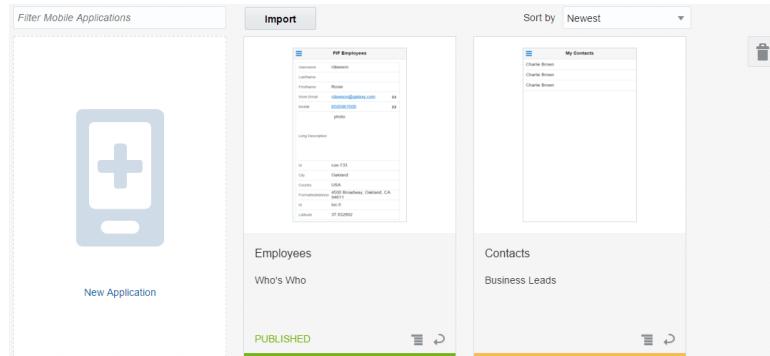
You can manage the entire lifecycle of your app from the Applications page. You can create apps from here, revise them, and remove them. If you need someone to troubleshoot your app, you can use the page's Export option to package your app in a ZIP file that you send to your designated expert. When you get the ZIP file back, you can use the Import function to unpack it.

After your first visit to MAX, the Applications page becomes your landing page. You can open this page by clicking **Back to Applications** (  ) in the Designer.

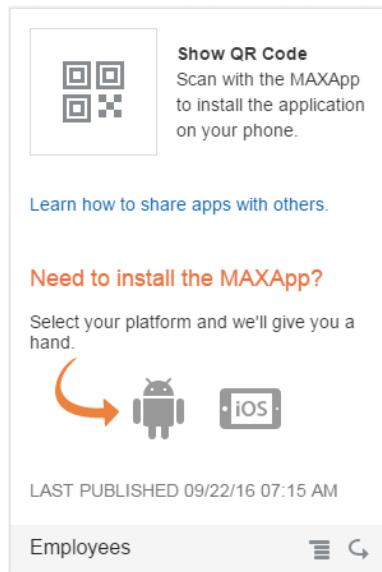
### Tip:

You can find an app using the Filter Mobile Applications field.

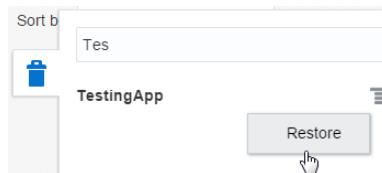
The Applications page lists all of your apps and identifies the ones that have been published with a green border. The ones that you haven't yet published have an orange border.



By flipping the card of a published app (  ) you can find out the last time you published it. By clicking **Show QR Code** (  ), you can display the distribution version of the QR code that belongs to the app.



To find out more about generating this code, see [Sharing Your App](#). When you want to revisit an app, update it, and republish it, you can open it in the Designer by either selecting **Open** from the menu (≡) or by double-clicking the snapshot of the app. When your app is no longer needed, you can send it to the trash by choosing **Move to Trash** from the menu. If you reconsider (or inadvertently throw an app away), you can fish an app out of the trash and then return it to the Applications page by first clicking the trash can icon (trash) to open the trash drawer and then by selecting **Restore** from the menu.



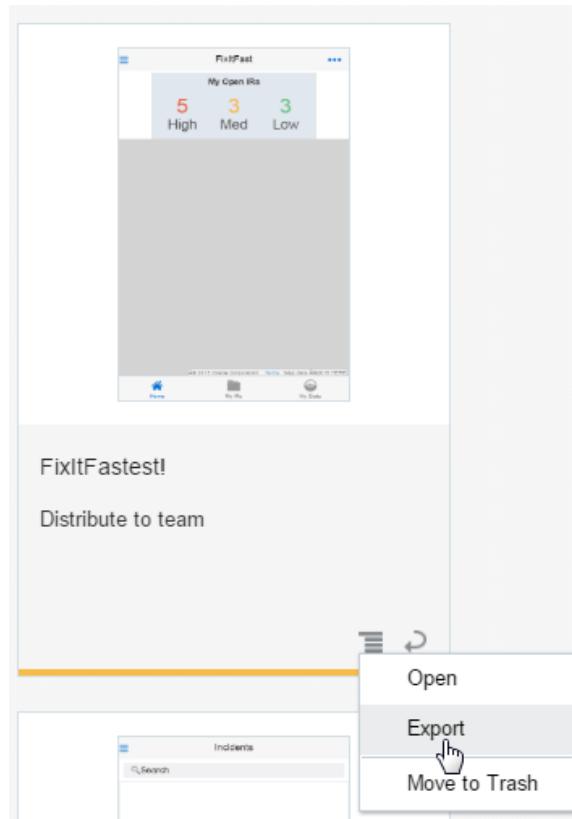
 **Tip:**

Find your app by entering its name (or part of it) in the Filter Trash field.

## Importing and Exporting Your Apps

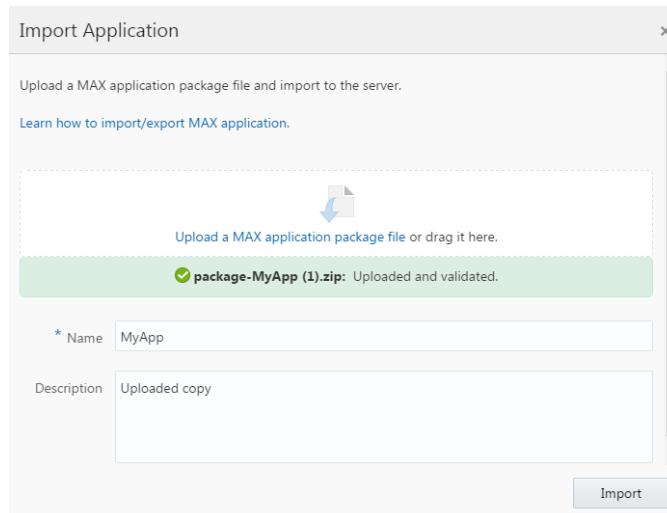
You can export your app to your local system if you want to keep a backup copy when you upgrade the service, or if you need to send it to your administrator for debugging.

Choosing **Export** from the Actions menu packages your app into a ZIP file. This ZIP contains the `application.json` and `services.json` files that store the metadata about your app's menus, screens, images and the service that backs your app with data.



When your admin gets this ZIP, she can take a look at these files and then import the ZIP into her instance of MAX to work on the app in the Designer. When she's done, she can export your revised app as a ZIP before sending it back to you. By clicking

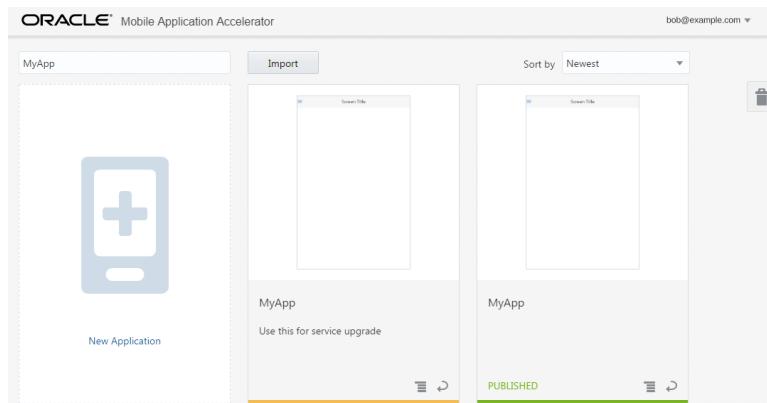
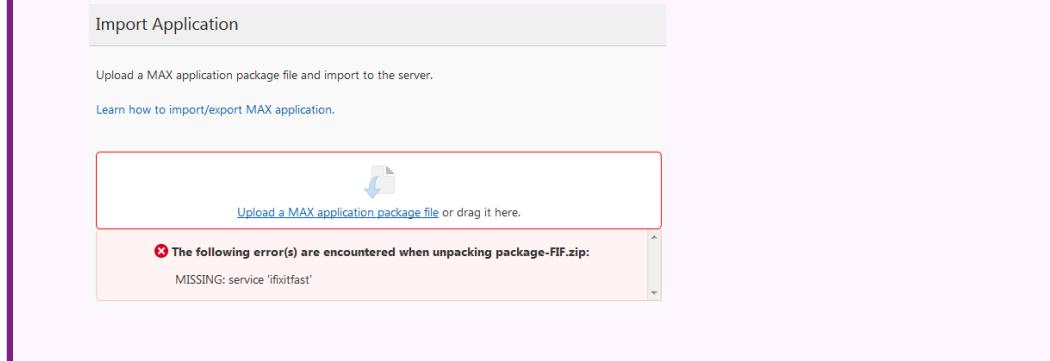
**Import** (  ), you can unpack the ZIP and add the revised version of the app to your MAX instance.



You can make duplicate, “working copies” of your apps by importing and exporting them. Duplicates (which are new versions of your app) will always be in the draft state, even when you've imported and exported a published app. See [Service Upgrades](#).

**! Important:**

You can't import an app into an instance where its service doesn't exist. Check with your administrator or service developer if a missing service prevents you from importing your app.



**💡 Tip:**

Open the screen flow (  ) to check for warnings or errors. Be sure to do this after you upgrade the service because business objects, fields, and query parameters might be changed or removed. See [Service Upgrades](#).

