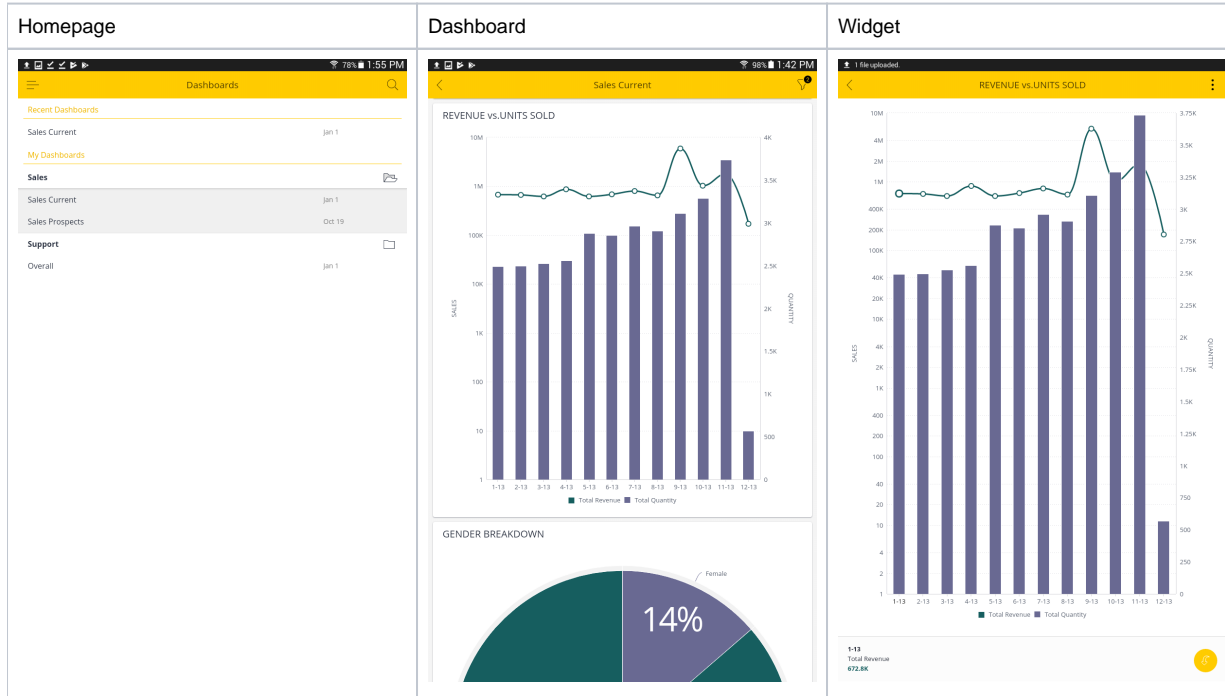


# Developing Plugins for Sisense Mobile

You can develop plugins for Sisense Mobile similar to how you develop plugins for the Sisense Web Application with the JavaScript API. For more information about developing plugins, see [JavaScript API & Plugins](#).

There are a few differences that you should note when developing plugins for Sisense Mobile.

First, the navigation flow for Sisense Mobile is fixed, homepage that includes your dashboard list, the dashboard, and the widget. This means that a widget that is opened as part of a dashboard behaves differently than a widget that is opened in Widget Mode.



To keep Sisense Mobile's navigation flow and look and feel, modifying or creating DOM elements outside the widget container is not supported.

Widgets that are rendered as part of a dashboard have the same screen width as the other widgets within the dashboard. Widgets that are rendered in Widget Mode, can occupy the entire screen minus the space of the Navigation bar and tooltip pans for chart widgets. You should develop your widgets to behave responsively to fit various screen sizes in landscape and portrait views.

Sisense Mobile can render a single widget instance in different containers. For each render call, the plugin should use unique IDs for its DOM elements.

Another difference between working with plugins in the Sisense Web Application and Sisense Mobile is that in mobile plugins, users can only interact with a widget in Widget Mode. The user must select the widget first from the dashboard, then they can interact with it.

## Supporting Plugins across Multiple Platforms

Some plugins are only suitable on the Web, while others are only suitable on mobile devices. Sisense has added a new property to the widget manifest (widget.js), called

```
hideOnMobileApp:true
```

This property should be included in the registerWidget of the prism object, for example:

### Example of a possible response

```
prism.registerWidget("WidgetsTabber", {
  hideOnMobileApp:true,
  name : "WidgetsTabber",
  family : "WidgetsTabber",
  title : "Tabber",
  ...
})
```

When this property is set to true, Sisense Mobile ignores the plugin. This property is useful for plugins that should be used only on the Sisense Web Application and not Sisense Mobile.

You can also set your plugins to behave differently on the Sisense Web Application and Sisense Mobile with a conditional that checks for the presence of the `window.sisenseMobileApp` object in the plugin script. For example,

#### Example of a possible response

```
if(window.sisenseMobileApp){
    // define mobile behavior here
} else {
    // define Sisense Web Application behavior here
}
```

**Note:** Navigation links based on `<a href="">` tags are not supported within Sisense Mobile. To open a dashboard in Sisense Mobile, use the helper function `window.sisenseMobileApp.navigateToDashboard(dashboardId)`. For other types of navigation links, use `window.open`.

## Example Sisense Mobile Plugin

This section provides a simple example of a plugin leveraging Sisense's JavaScript API to change the order of your widgets in a dashboard.

You can download the plugin [here](#).

This example contains three files:

**Config.js:** This file contains the dashboard ID to be modified in Sisense Mobile and an array of widget IDs in the order of the widget's are to be displayed.

**Plugin.json:** This file contains the plugin's metadata so Sisense Mobile knows which files to associate with the plugin and what the plugin is called. This file is also important so your Sisense Administrator can manage the plugin from the Admin console in the Sisense Web Application.

**mobileDashboardReorder.js:** This file contains the logic for reordering the dashboard and widgets when the dashboard is displayed in Sisense Mobile.

To implement this MobileDashboardReorder plugin, download the file above and unzip the contents into the following directory as you would with any Sisense plugin:

```
C:\Program Files\Sisense\PrismWeb\plugins\
```

Then, restart the web server.

Next, you need to modify the plugin files and add your Dashboard ID and Widget IDs as described below.

You can add multiple Dashboards and order their widgets as you like. Any dashboards not defined in the `dashboardsMap` variable will be rendered as a regular dashboard in the Sisense Mobile app.

For Sisense 7.2 Beta Users



In Sisense V7.2 Beta, the location has been changed to `C:\Program Files\Sisense\app\plugins`.

Your existing plugins will be migrated to the new location when upgrading to v7.2 Beta.

For more information, please see the [v7.2 Developer Release Notes](#)

### Config.js

The `config.js` defines the dashboard to be modified in Sisense Mobile and the order of the widgets according to their Widget ID and position in an array of widgets.

Replace the dashboard ID example with your ID and the widget IDs with yours. The order of the widget IDs determines which widgets are displayed at the top of dashboard in Sisense Mobile. You can add or remove widgets from the array as needed.

For more information on how to obtain your dashboard and widget IDs, see [Embedding Dashboard Widgets](#).

### Example of a possible response

```
var dashboardsMap = {
  '1234ad242d86760b24f1adb4': // Dashboard ID to be modified. Replace this example ID with your Dashboard ID.
  [ /* Array of widget IDs to display when loading the dashboard in Sisense Mobile. Replace the example IDs
with your widget IDs paying attention to the order. You can remove or add widget IDs as necessary. */
    {widgetid: '1234ad242d86760b24f1adba'},
    {widgetid: '1234ad242d86760b24f1adb7'},
    {widgetid: '1234ad242d86760b24f1adb9'},
    {widgetid: '1234ad242d86760b24f1adb5'},
    {widgetid: '1234ad242d86760b24f1adb8'}
  ]
};
```

## Plugin.json

Sisense uses the plugin.json to implement the plugin. This file contains the following properties:

**Name:** The name of the plugin.

**Source:** The files associated with the mobileDashboardReorder plugin.

**Lastupdate:** Indicates the last time the plugin was updated.

**FolderName:** The name where the plugin files are stored within the plugin directory.

### Example of a possible response

```
{
  "name": "mobileDashboardReorder",
  "source": [
    "js/config.js",
    "js/mobileDashboardReorder.js"
  ],
  "lastUpdate": "2017-11-19T09:42:47.118Z",
  "folderName": "mobileDashboardReorder"
}
```

## mobileDashboardReorder.js

This file contains the logic for determining when the MobileDashboardReorder plugin should be implemented. The logic is a simple condition that when the sisenseMobileApp object is rendered, run the MobileDashboardReorder plugin.

### Example of a possible response

```
if(window.sisenseMobileApp){ // Apply the plugin when running inside Sisense Mobile BI app environment
  prism.on('dashboardloaded', function (e, args) {
    if(dashboardsMap[args.dashboard.oid]){ // Do we need to manipulate the layout of this loaded dashboard
      defined in config.js?
      /* under the mobile app the list of dashboard widgets is flatten based on a layout scan of columns ->
      cells -> subcells
      based on that, we can push our mobile designated widgets to a flat widget list in a single subcell
      while trimming the original layout */
      var layout = args.dashboard.layout;
      layout.columns.length = 1;
      layout.columns[0].cells.length = 1;
      layout.columns[0].cells[0].subcells.length = 1;
      layout.columns[0].cells[0].subcells[0].elements = dashboardsMap[args.dashboard.oid];
    }
  })
}
```