Creating a Code Widget

- Overview
- Code widget sequence diagram
- Next steps

Overview

A code widget is a reusable bundle of code that provides flexible dashboard customization (see more details here).

To write a code widget:

1. Write the back-end code and check it works.
2. Write the front-end code and check it works.
3. Customize the Code Widget Properties panel (optional).
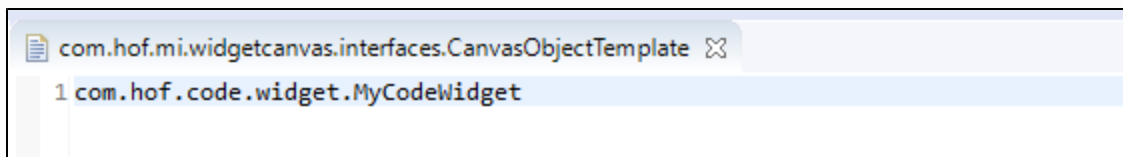4. Bundle the code into a jar file, then upload it to Yellowfin.

The code widget will now display in a list of code widgets within your dashboard canvas editor, where they can be dragged and dropped onto the canvas from the Code Widgets menu on the left in Dashboard Creation mode.

You can set up your IDE to develop Yellowfin plugins by following this guide.
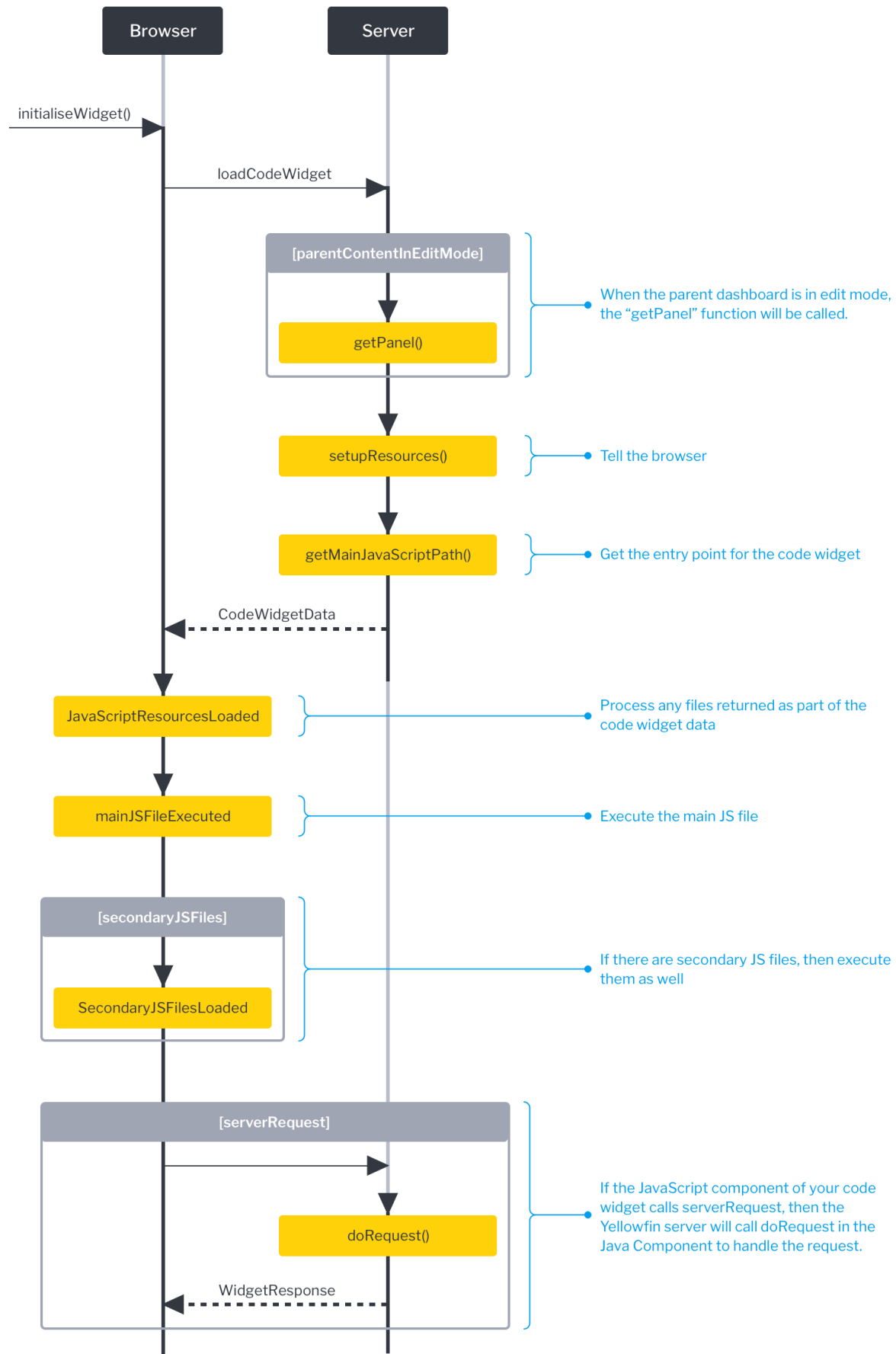
> ⓘ **Dev Tip**
>
> Code widgets are children of canvas widgets, so when you are using the META-INF/services feature mentioned in that guide, you will need to add your code widget as a *CanvasObjectTemplate*.
>
> ```
> 📄 com.hof.mi.widgetcanvas.interfaces.CanvasObjectTemplate ⊠
> 1 com.hof.code.widget.MyCodeWidget
> ```

Creating a Code Widget#top

Code widget sequence diagram

Browser

Server

initialiseWidget()

loadCodeWidget

[parentContentInEditMode]

getPanel()

When the parent dashboard is in edit mode, the "getPanel" function will be called.

setupResources()

Tell the browser

getMainJavaScriptPath()

Get the entry point for the code widget

CodeWidgetData

JavaScriptResourcesLoaded

Process any files returned as part of the code widget data

mainJSFileExecuted

Execute the main JS file

[secondaryJSFiles]

SecondaryJSFilesLoaded

If there are secondary JS files, then execute them as well

[serverRequest]

doRequest()

If the JavaScript component of your code widget calls serverRequest, then the Yellowfin server will call doRequest in the Java Component to handle the request.

WidgetResponse

Next steps

When you're ready to get started with code widgets, follow this process:

1. Write the back-end code.   Start here
2. Write the front-end code.
3. Customize the Code Widget Properties panel (optional).
4. Bundle the code into a jar file, then upload it to Yellowfin.

To further assist you during code widget development, click on the code widget reference page for samples, API links and detailed descriptions.

Creating a Code Widget#top