SPAnalytics Installation Guide

|  |  |
| --- | --- |
| Primary Author: | Alex Kwon, Nathan Soundappan |
| Contributors: | Alex Kwon, Nathan Soundappan, Matthew Murch |

**ACCEPTANCE:**

|  |  |  |
| --- | --- | --- |
| Sponsor/Client: | **Jackson Polish** | **5/7/2020** |
| Team Representative: | **Colin Evans** | **5/7/2020** |

**Table of Contents**

[**Preface [Alex Kwon]**](#_m5501j9fnso6) **3**

[**System Requirements [Alex Kwon]**](#_wu02qg9brsnx) **4**

[**Installation Instructions [Alex Kwon, Nathan Soundappan]**](#_iadev7fo7k9j) **4**

[Step 1:](#_glatfp8xkran) **Retrieving a JSON Key from Firebase** 4

[Step 2:](#_ogsoz3sm1p1p) **Running the Java Project cloned from Gitlab in Eclipse**  4

[Step 3:](#_ogsoz3sm1p1p) **Downloading JavaFX and SceneBuilder**  9

[**Verification [**](#_ft520uxbdmzi)[**Alex Kwon, Nathan Soundappan**](#_iadev7fo7k9j)[**]**](#_ft520uxbdmzi) **13**

## Preface [Alex Kwon]

**Introduction**:

SPAnalytics is an application founded by our client Jackson Polish. It aims to record accurate and detailed hockey game data in real time or through recorded games and using those data it will generate analytical functions that output meaningful player and team statistics for hockey teams using this application.

There are 4 main features of this application. The first feature is the shot chart which records different types of shots taken during the game. The second feature is the possession diagram which allows live coverage of who has possession of the puck. The third feature is the passing diagram which records the different types of passes occuring during the game. The fourth feature is the time on ice diagram which keeps track of an individual player’s total time on ice.

**Purpose**:

This is a detailed guide to all necessary installations and setting configurations that are needed to be accomplished prior to running and editing the source code of the application in Eclipse. There are three main steps. The first is retrieving a JSON Key from Google Firebase for application login purposes. The next step is setting up Eclipse before running the Java Project source code. The third step explains how to connect a visual tool called Scene Builder to your Eclipse Workspace so that you can upload your source code from Eclipse to Scene Builder and work on the front-end development using that tool.

**Audience**:

The main audience is the developers who will be working on the implementation of this application. After all of the installations are complete, developers will be ready to edit the Java source code.

## System Requirements [Alex Kwon]

There are no special system requirements needed before proceeding to the installation instructions.

## Installation Instructions [Alex Kwon, Nathan Soundappan]

### Step 1: Retrieving a JSON Key from Firebase

1. Skip this step if you do not need a new key.
2. The primary testing key used during development is already provided in the Gitlab repository given by the previous group. Since the next phase of this project involves transferring over to another type of database, there should be no reason to create another key from Firebase.
3. However, f the provided key from Gitlab is lost or if a developer needs a new key, instructions for making a new firebase key can be found in the link below:

<https://drive.google.com/file/d/1gTZX7Pw0mgElKVCAkl3vffsbbcNxgWsp/view?usp=sharing>

1. How to use the new firebase key will be explained in Step 2

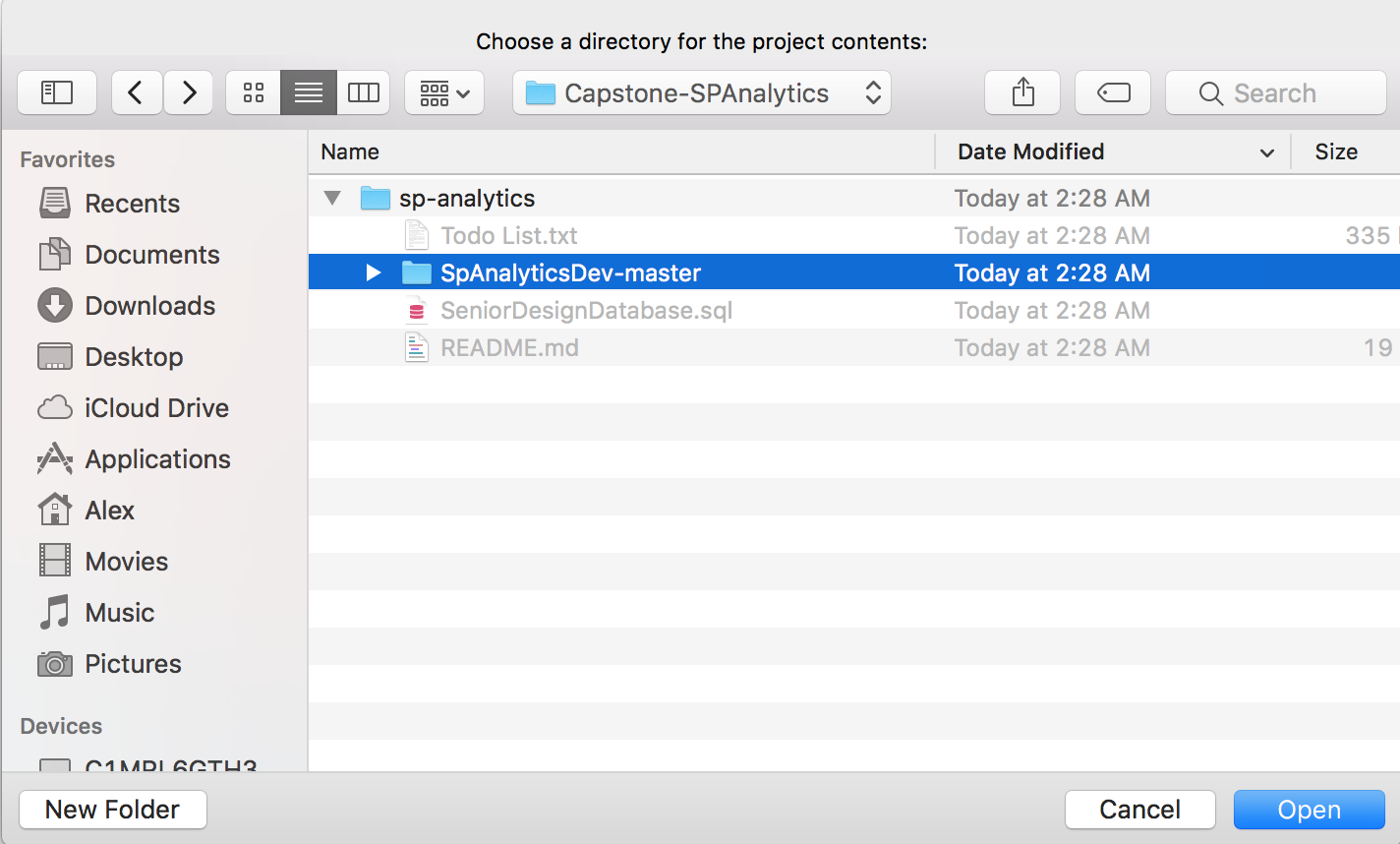
### Step 2: Running the Java Project cloned from Gitlab in Eclipse

**In Eclipse,**

1. Create new workspace
2. Create new java project inside the new workspace
   1. In Project Layout, uncheck “Use default location”, “Use project folder as root for sources and class files.” Then click “Browse”



* 1. Browse for the SPAnalyticsDev-master Project Folder (Not the SPAnalytics folder under the master folder)

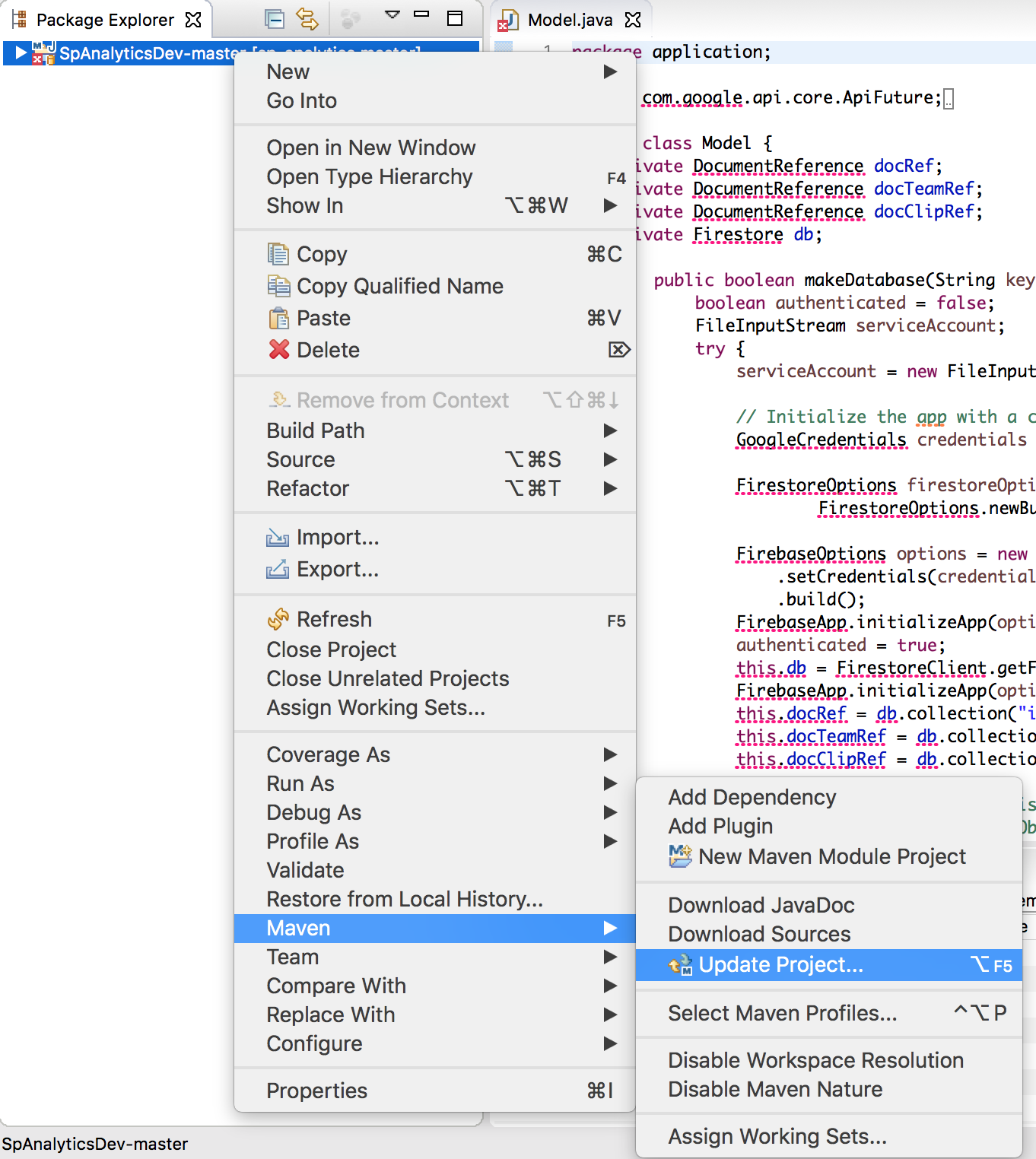


* 1. Click Finish



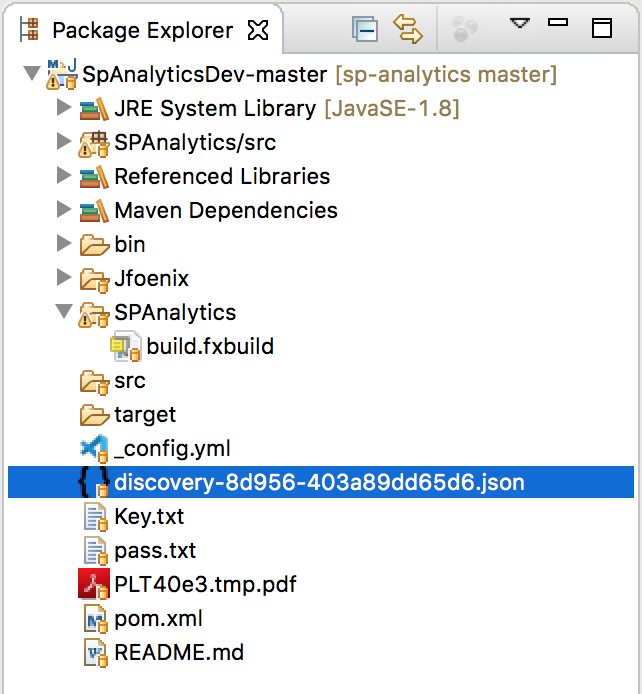
* + 1. If you get a pop-up talking about the compiler not being in compliance with the project, you do not have the right version JRE/JDK. The eclipse version you need is 1.8.0.
    2. Once you download the correct version, open eclipse and click Window -> Preferences.
       1. Open the Java tab and click compiler. In the top right you should see a drop down. Click it and select 1.8.
       2. You also need to uncheck the box that says “Use --release option”
       3. Click apply and close. Click window -> preferences.
       4. Open the Java tab and click Installed jre’s.
       5. If the one selected is not 1.8, click add and add it as a standard VM.
       6. Go to where you have it saved on your device (it is most likely C:/Program Files/Java)
       7. If you do not have one that is 1.8, then you need to install one and place it here. Then, go back to step 2.c.ii.3
       8. Add the 1.8 jre/jdk as a “Standard VM”.

1. You should have unsolved errors in the source code. To solve that issue, right click on the master folder in Package Explorer
   1. Hover over “Maven” > Click “Update Project”> Click “OK”



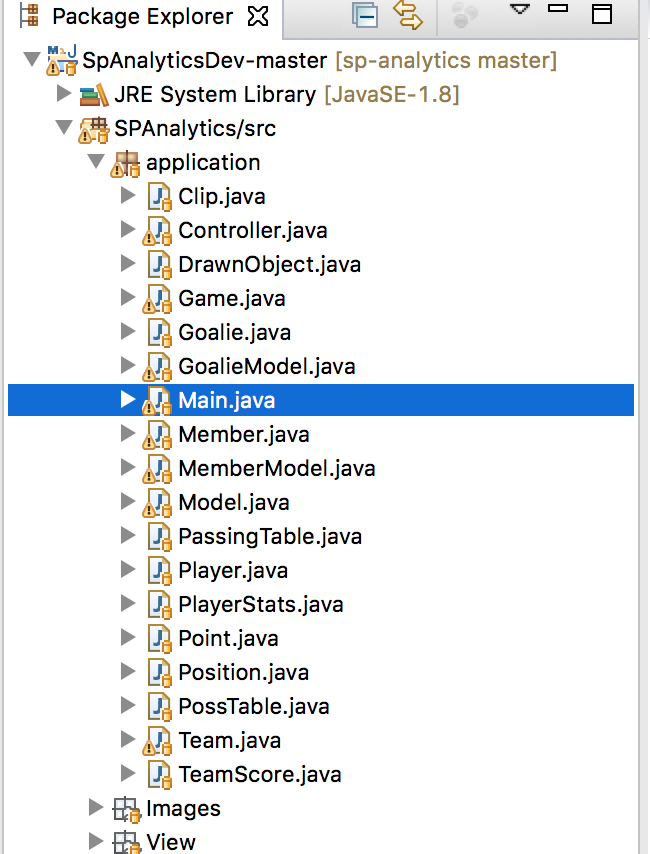


1. Check if there is a Firebase json key file under the master folder



* 1. If the cloned project does not have a json key as shown above, generate a new key and add the Firebase json key file (from the project Gitlab) under the master folder. Steps to generate the key can be found in the link from Step 1.

1. Run in eclipse the main.java file in the application folder



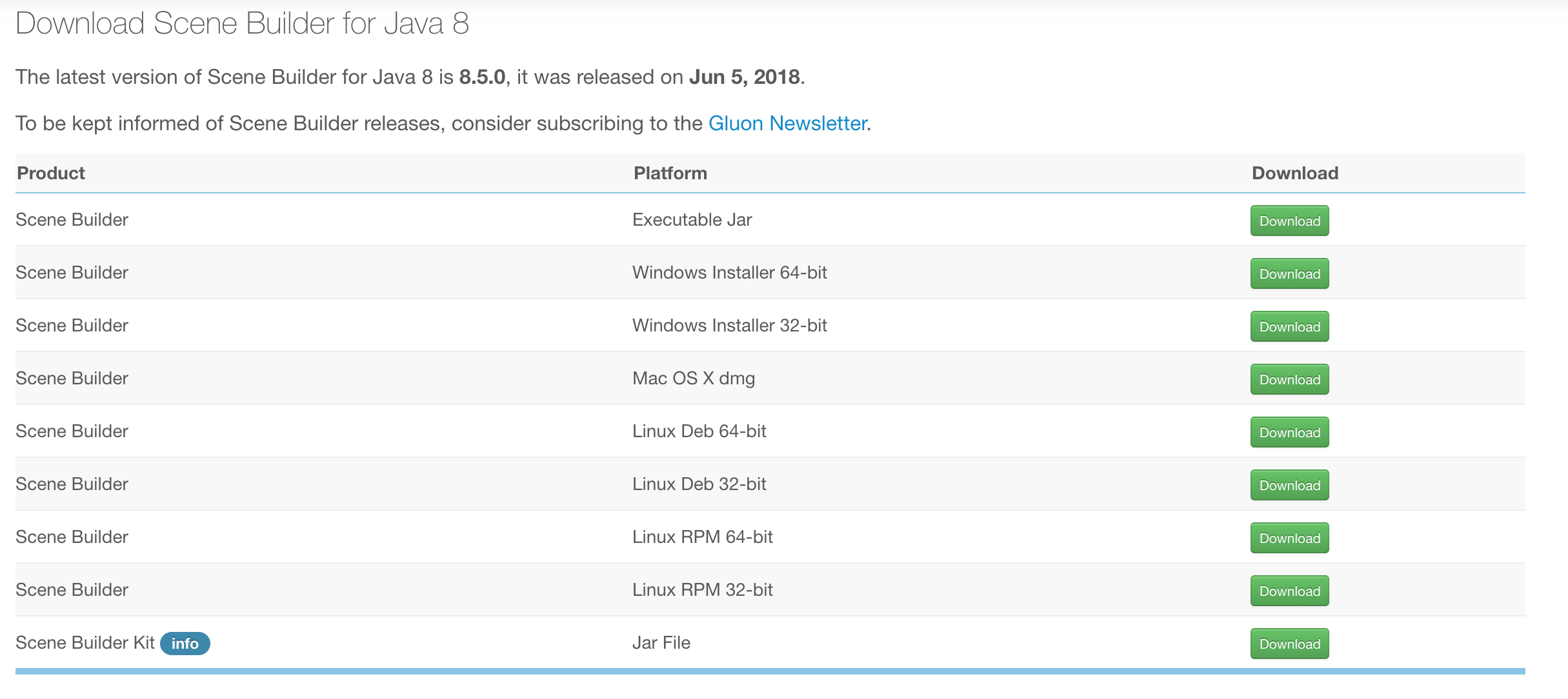
* 1. When you enter your key, copy and paste the full file name of the json file including “.json”



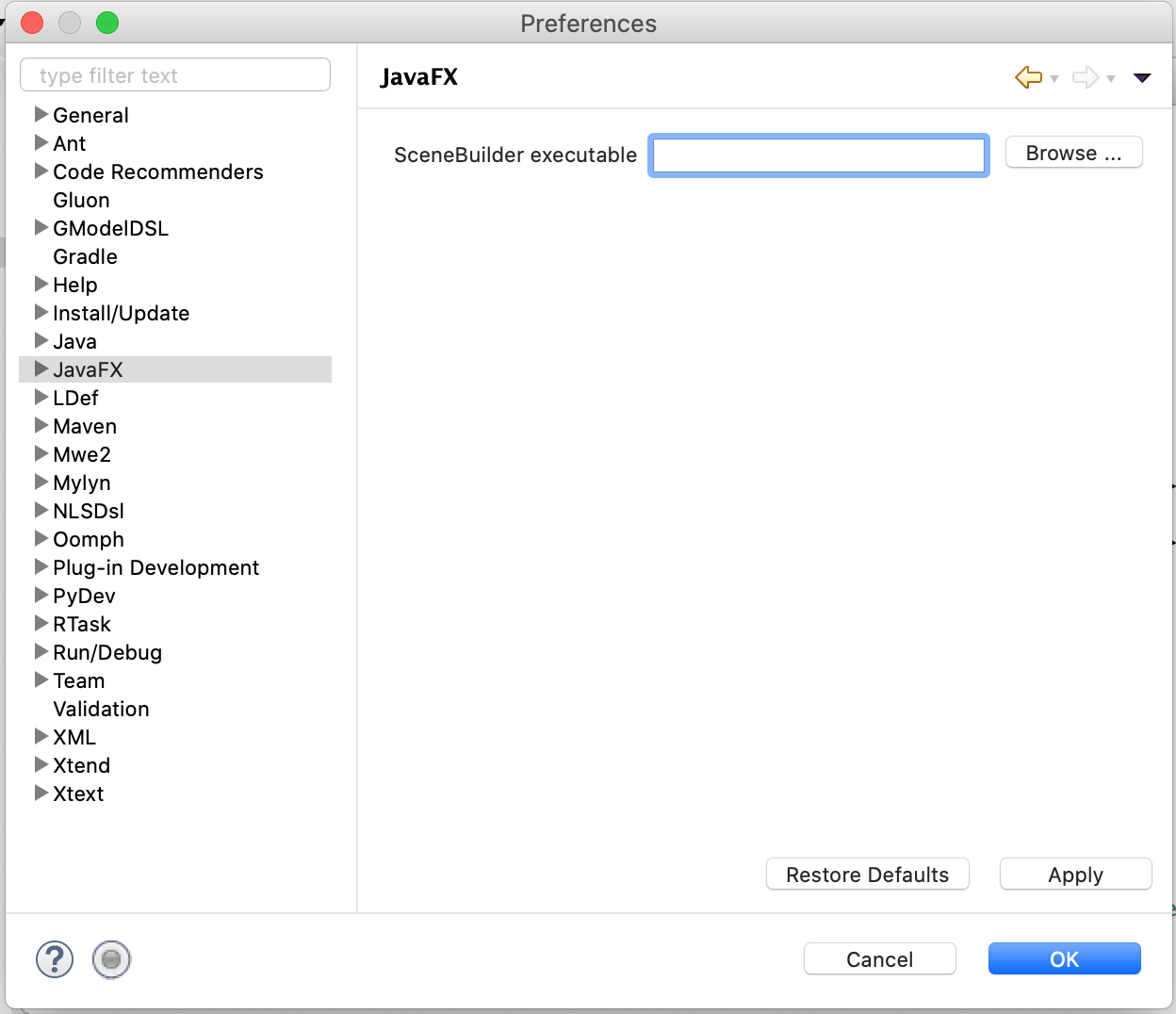
### Step 3: Downloading JavaFX and SceneBuilder

Eclipse doesn’t come with JavaFX, we can run our code without it because the library is included in our project. To run SceneBuilder with eclipse we will need to install JavaFX first.

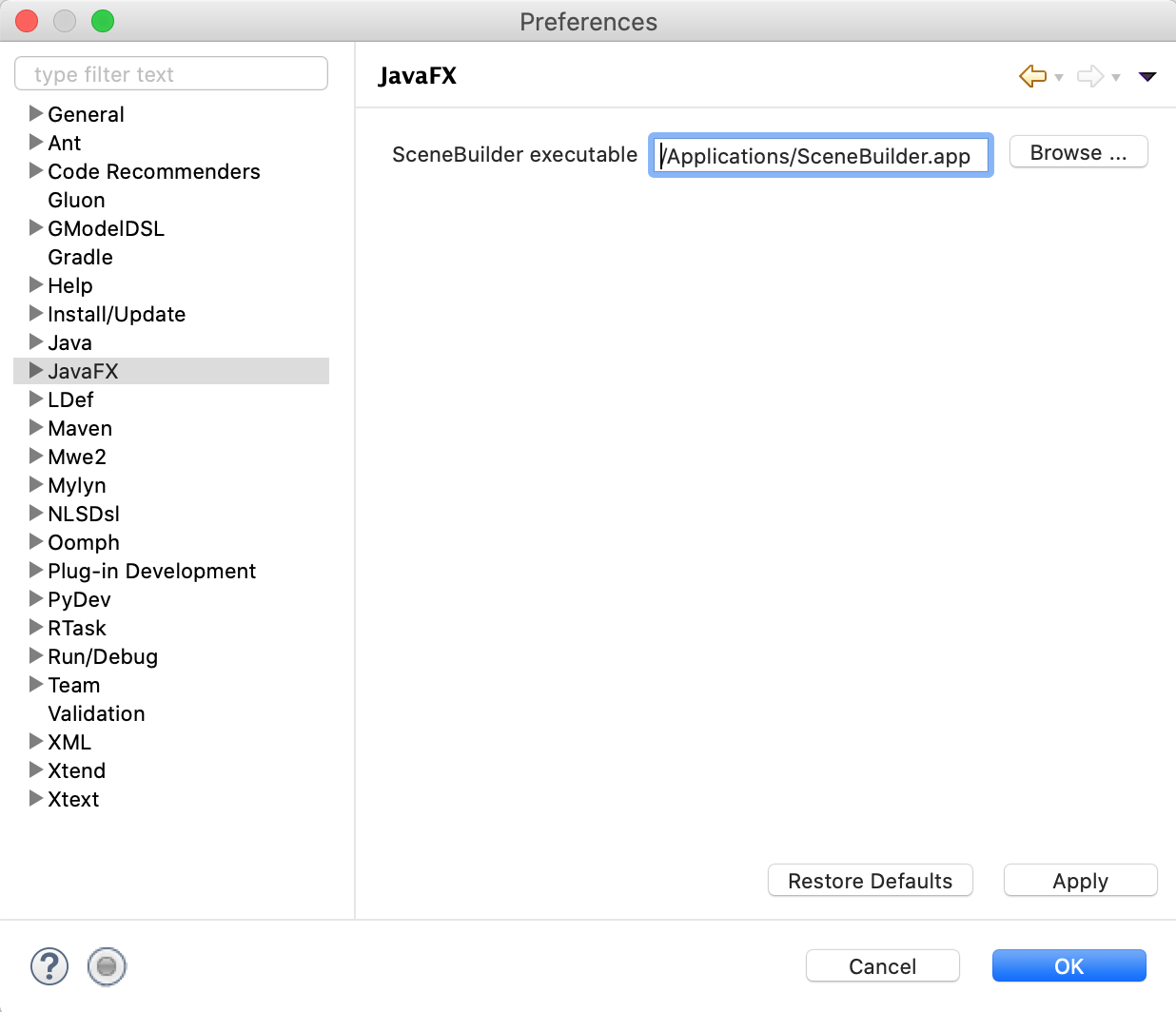
1. **JavaFX**
   1. Select **Help -> Eclipse Marketplace** from the Eclipse menu
   2. Search for “**JavaFX**”
      1. Go to **e(fx)clipse 3.6.0** (should be first result) and click install
   3. Go through the dialogue boxes (if there are any, I don’t remember)
      1. A loading bar will appear in the bottom right corner
   4. Restart eclipse when prompted to apply the changes
2. **SceneBuilder** 
   1. Go to [Gluon](https://gluonhq.com/products/scene-builder/#download) and download Scenebuilder
      1. Make sure you remember where you download it



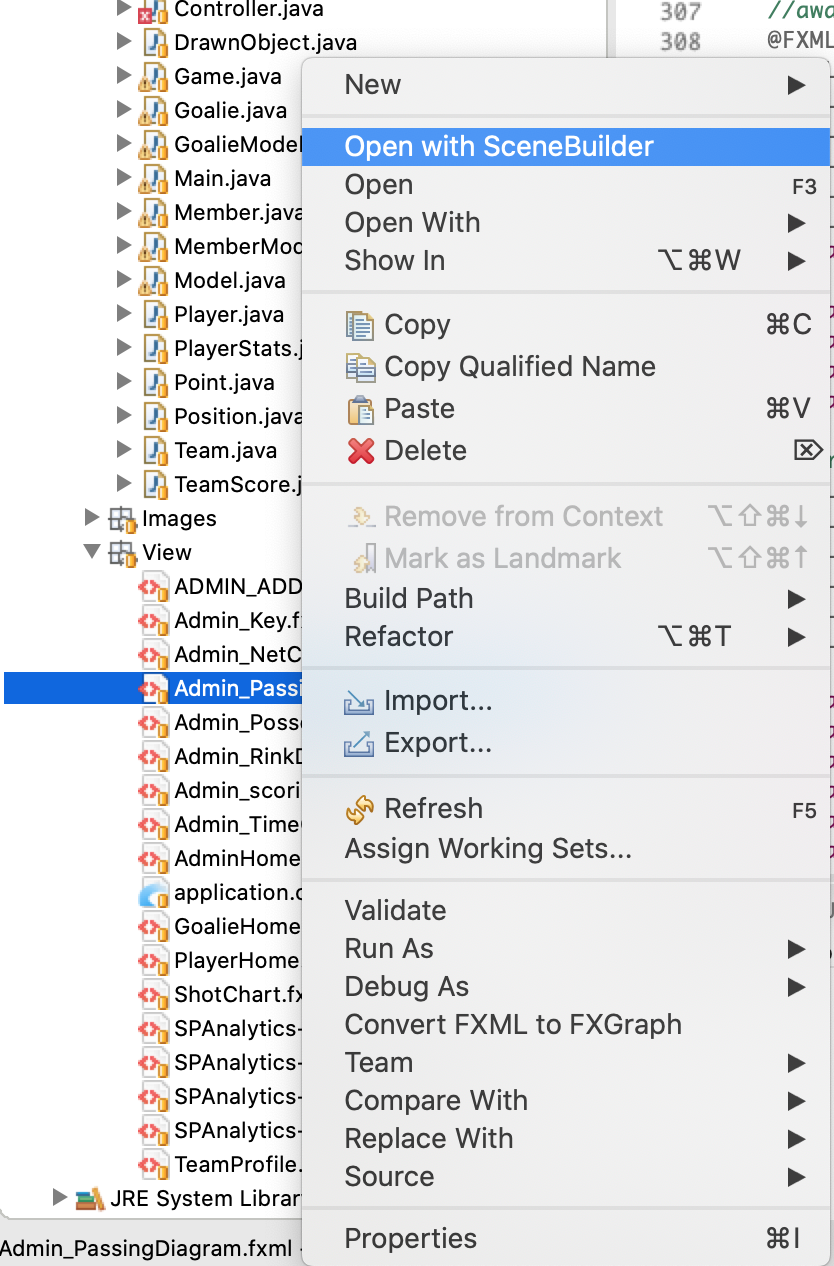
* 1. Open up your project in eclipse
  2. Click **Window -> Preferences -> JavaFX** 
     1. You will see “SceneBuilder Executable”
     2. We wouldn’t see this if we didn’t download JavaFX in the previous step



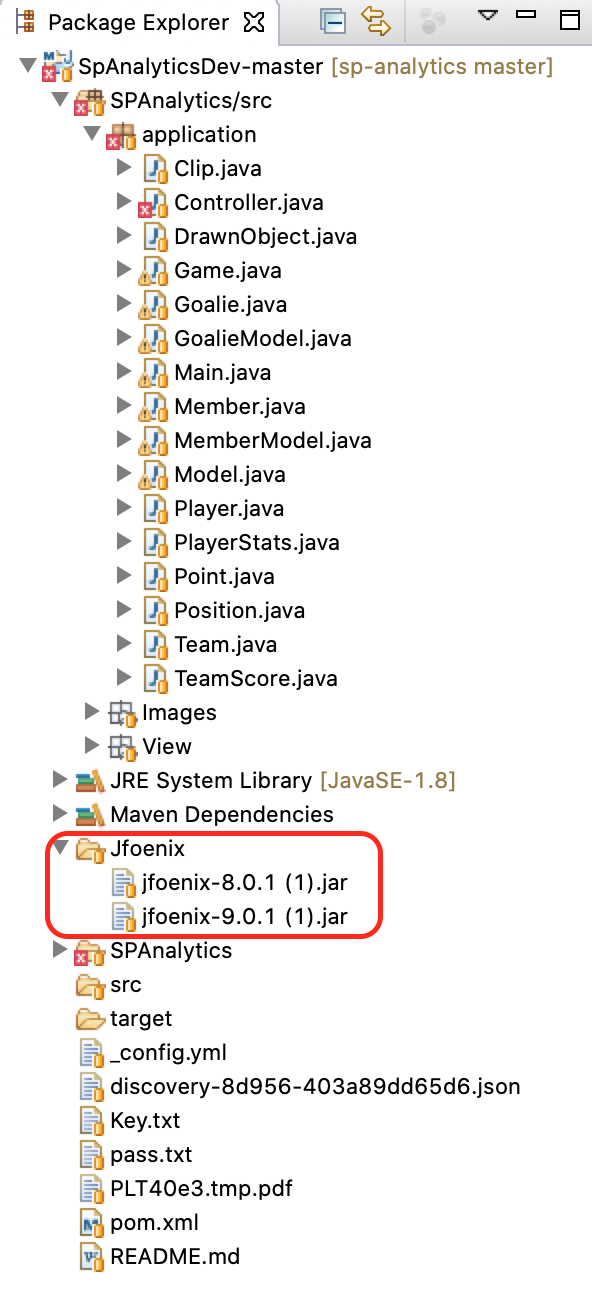
* 1. Click **Browse**, find your SceneBuilder folder, and select the SceneBuilder executable
  2. Click **Apply and Close**

****

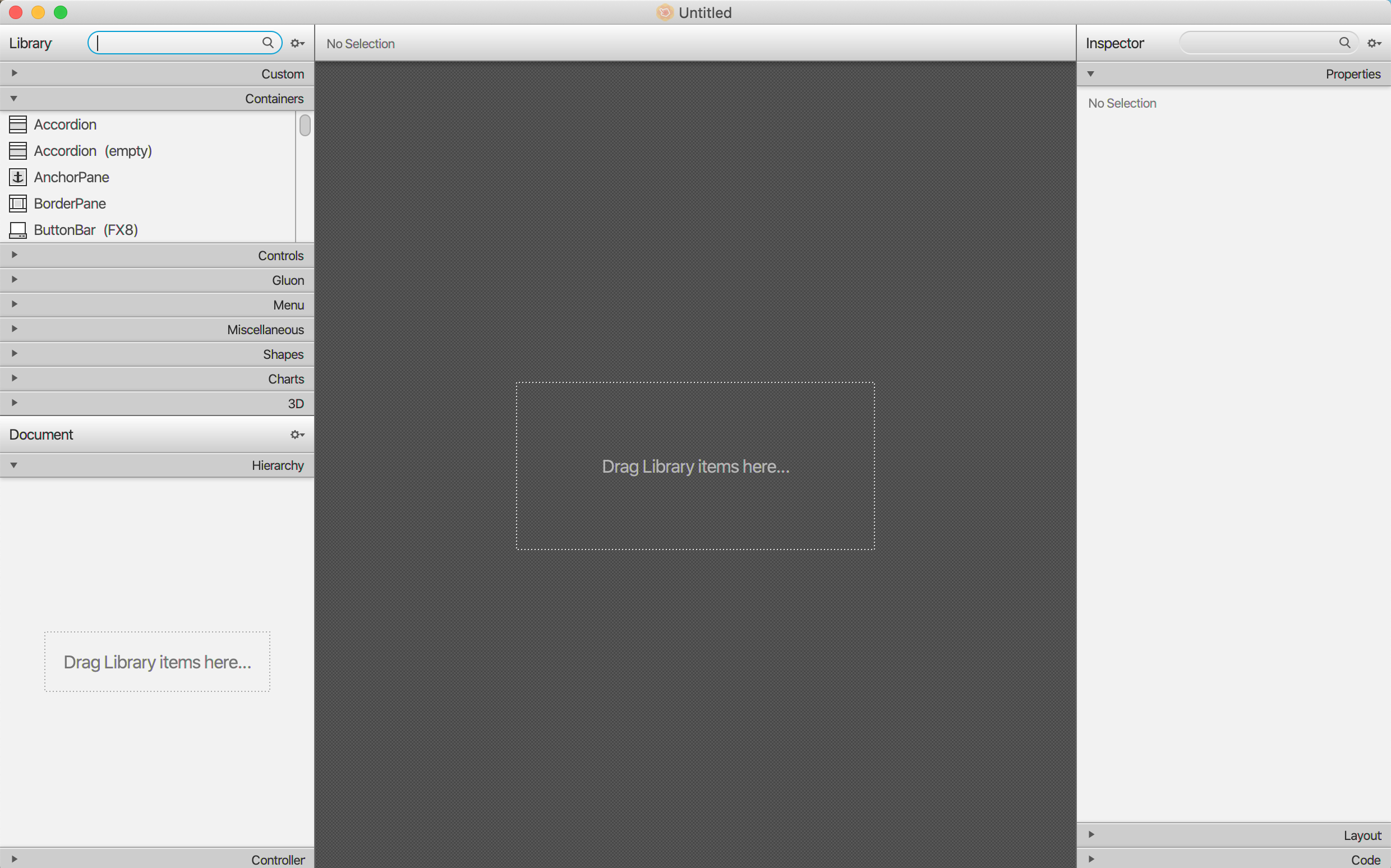
* 1. Navigate to **SpAnalyticsDev-Master -> SPAnalytics/src -> View** 
     1. You’ll find each page of the app as a fxml document
  2. Right click on any fxml file and Select **Open with SceneBuilder**
     1. You should get an **Error**
     2. Scene Builder does not recognize some components from our project, we need to add our Jfoenix JAR file to Scene Builder



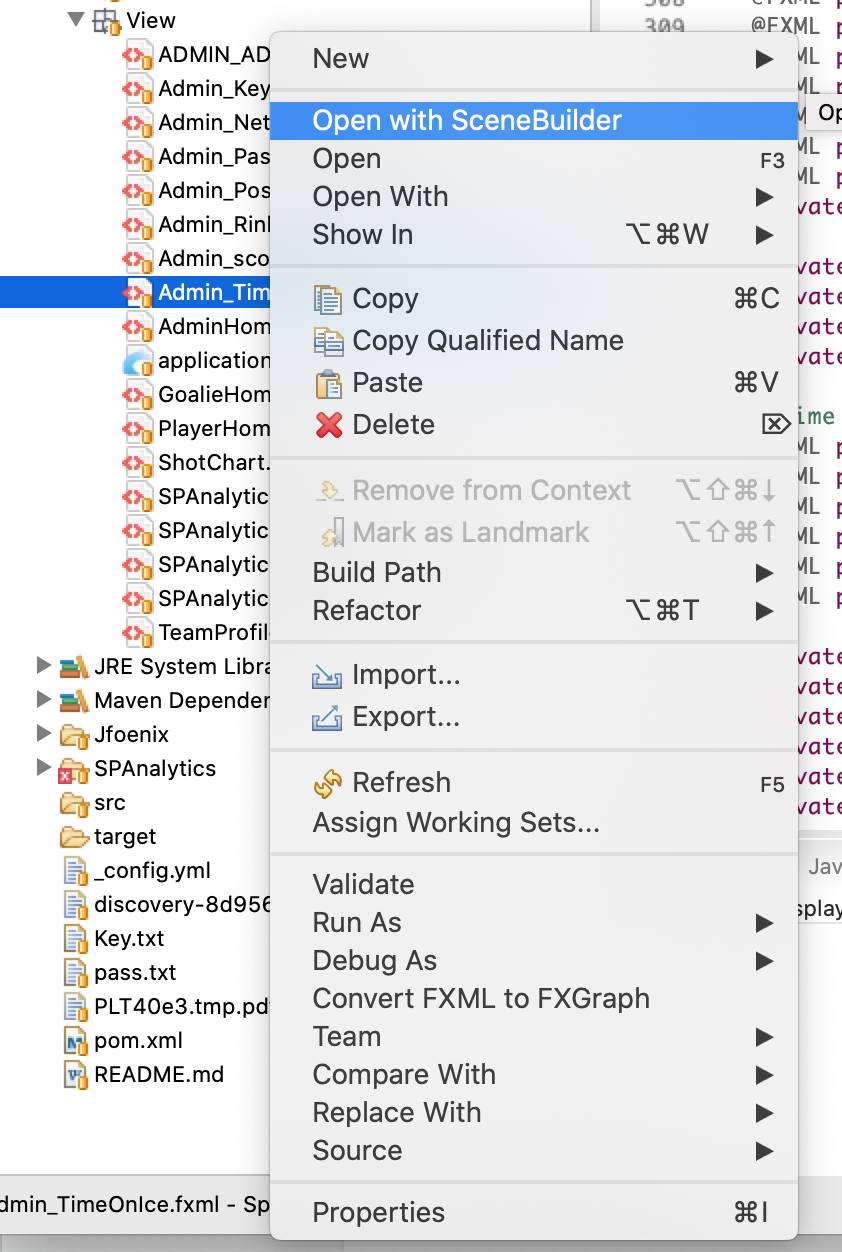
* 1. Navigate to **SpAnalyticsDev-Master -> Jfoenix** to find the necessary JARs

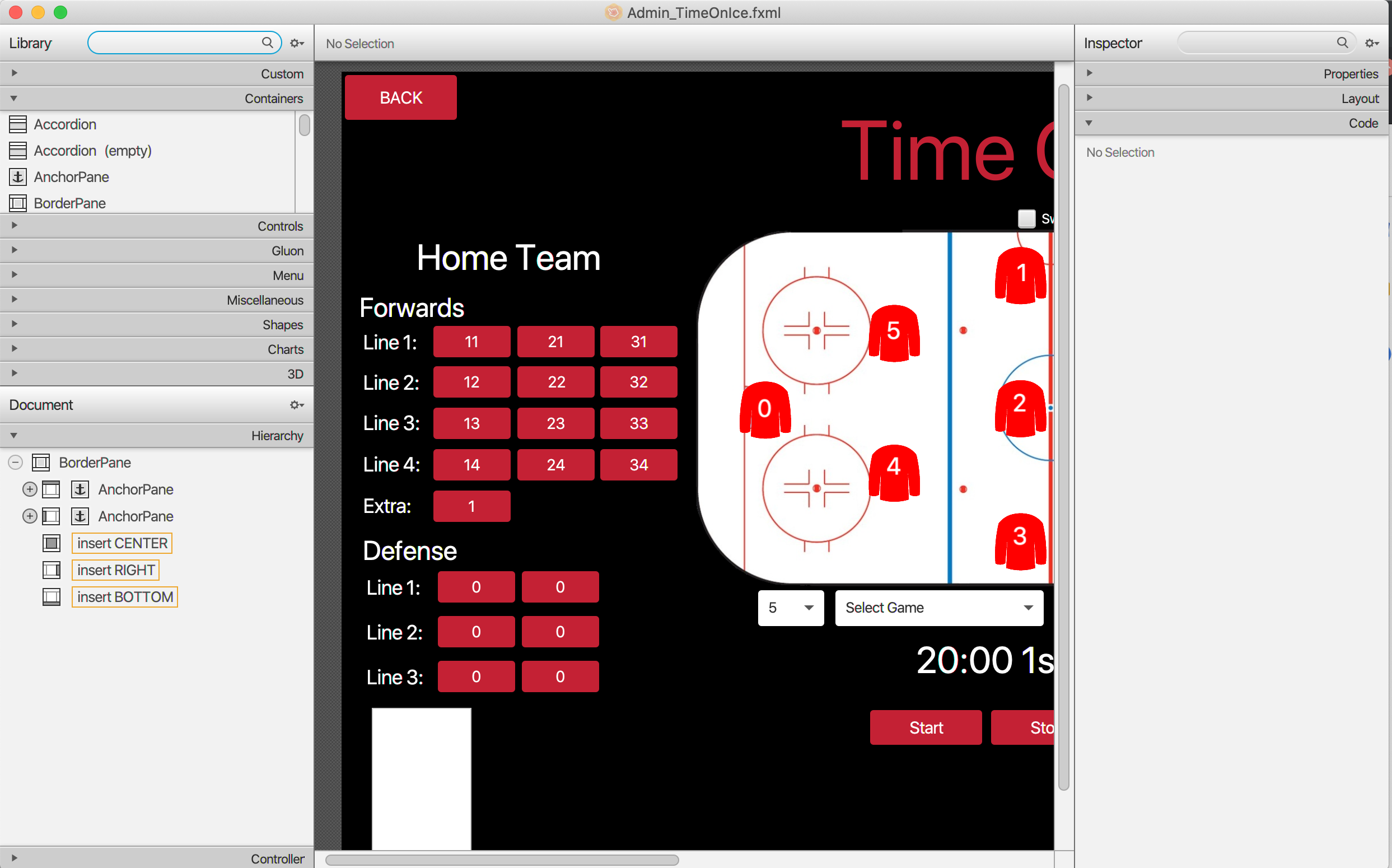


* 1. Drag and drop the JAR file to the **Library** section of SceneBuilder



* + 1. A dialog box will open asking which components you want to import, by default they are all selected
    2. After import, under the **Custom** tab you should find all the different components, Ex. JFXButton
  1. That’s it, you should be able to use SceneBuilder now with Eclipse by right clicking on an fxml and selecting **Open with SceneBuilder**

****

****

## Verification [Alex Kwon, Nathan Soundappan]

Installation is complete if the user is able to run the application from Eclipse and able to edit the front-end using Scene Builder.