Getting Started with JavaFX

Author: Yida Tao

Background

If you are using Java 8, JavaFX is already bundled with your JDK so you do not need to take any further steps. With the advent of Java 11 in 2018, however, JavaFX was removed from the JDK so that it could evolve at its own pace as an independent open-source project guided by Oracle and others in the OpenJFX community.

In this tutorial, we assume that you're using Java 11 or later.

1. Create a "Hello World" JavaFX Application

Prerequisite

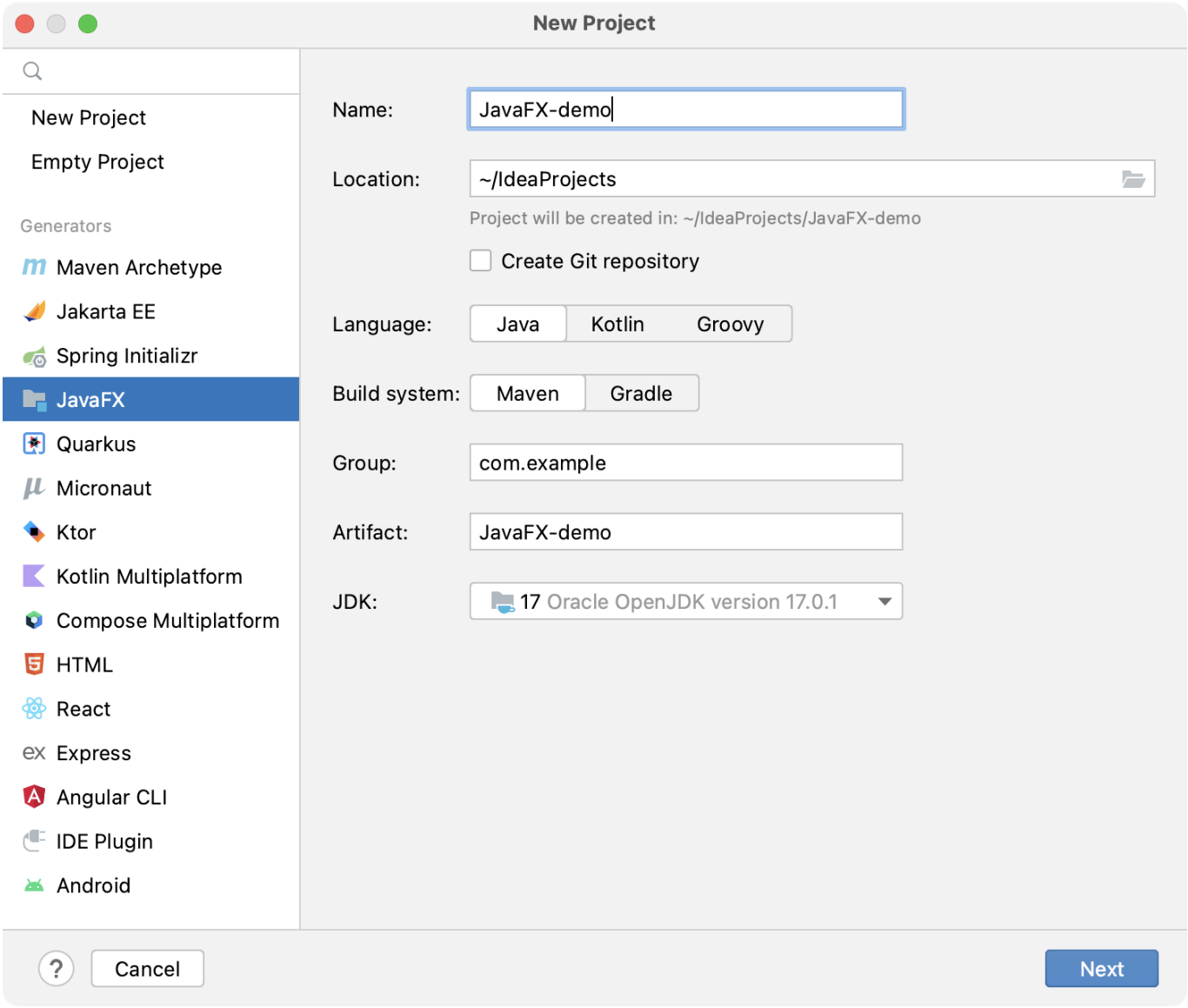
To be able to work with JavaFX in IntelliJ IDEA, the JavaFX bundled plugin must be enabled:

- In the `Settings/Preferences` dialog (Ctrl+Alt+S), select `Plugins`.
- Switch to the `Installed` tab and make sure that the JavaFX plugin is enabled. *If the plugin is disabled, select the checkbox next to it*.
- Apply the changes and close the dialog. Restart the IDE if prompted.

Create a new project

When you create a new JavaFX project, IntelliJ IDEA generates a fully configured sample application.

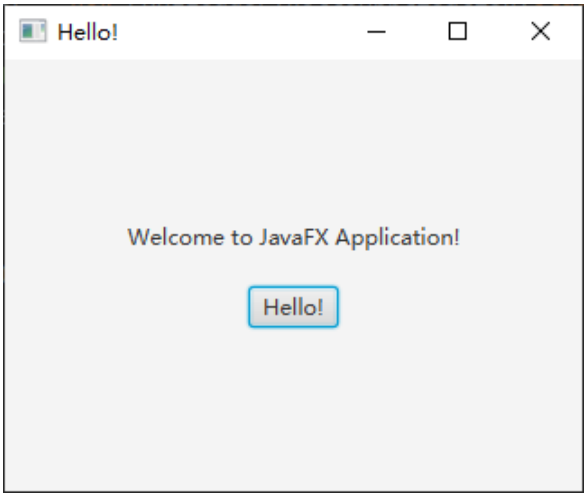
- Launch IntelliJ IDEA. Click `New Project`. Otherwise, from the main menu, select File | New | Project.
- From the Generators list on the left, select `JavaFX`.
- Name the new project, change its location if necessary, and select a language, and a build system.
- In the Group field, specify the name of the package that will be created together with the project.
- From the JDK list, select the JDK that you want to use in your project. If the JDK is installed on your computer, but not defined in the IDE, select Add JDK and specify the path to the JDK home directory. If you don't have the necessary JDK on your computer, select Download JDK.
- Click Next -> Create.



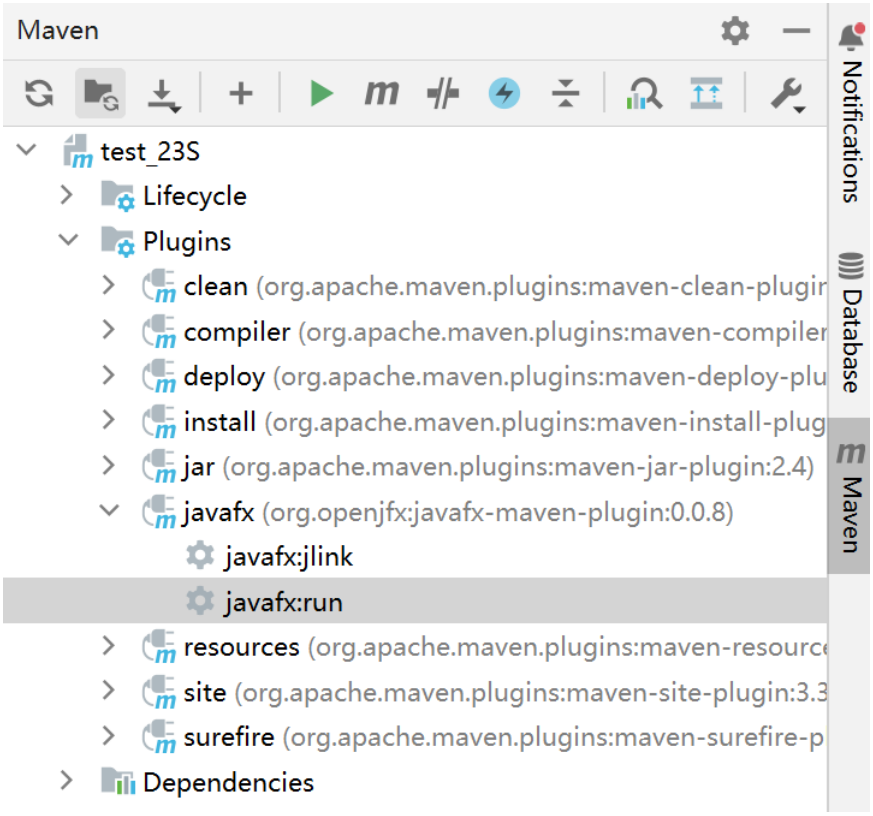
Run the application

Open the `HelloApplication.java` class, click the `Run application` icon in the gutter, and select `Run 'HelloApplication.main()'`. The IDE starts compiling your code.

When the compilation is complete, the application window appears. This means that the project is configured correctly and everything works as it should.



You may also open the "Maven" view, select the current project, then click "Plugins" -> "javafx" -> "javafx:run" to execute the program. Here, we are executing the javafx maven plugin's `run` goal, which is configured in `pom.xml` to execute the main class.

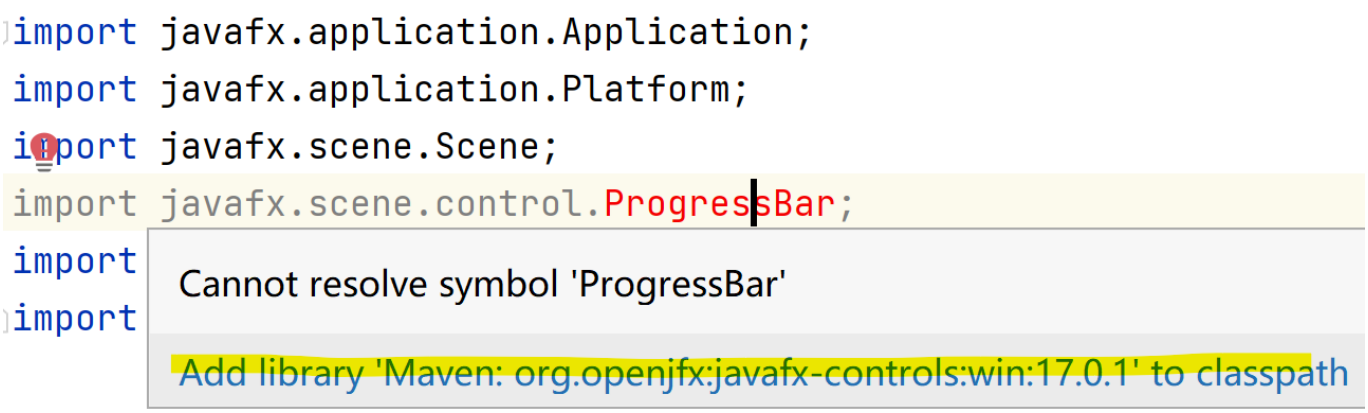


2. Add JavaFX Functionalities to an Existing Project

Suppose that you have an existing Java project, and you want to use JavaFX in it. In this case, you should add JavaFX support to your current project. You could achieve this using either of the following approaches.

2.1 Using Maven

You could let maven automatically download the required JavaFX dependencies by put your mouse on missing dependencies and click the suggested action.



Then, add a `module-info.java` to your project to declare a module. In this file:

- `requires`: specifies dependencies on other modules, such as `javafx.controls` and `javafx.fxml`
- `opens`: allows `javafx.fxml` to access the FXML controllers in our package.
- `exports`: specifies which packages in this module should be accessible to the JavaFX runtime.

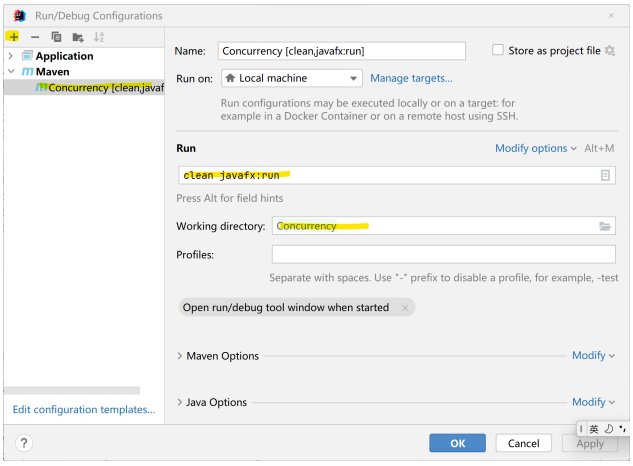
```
module Concurrency {
    requires javafx.controls;
    requires javafx.fxml;

    opens . to javafx.fxml; // . should be replace by your package name
    exports .;              // . stands for default package
}
```

Next, add the following to `pom.xml`:

```
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.8.0</version>
            <configuration>
                <release>17</release>
            </configuration>
        </plugin>
        <plugin>
            <groupId>org.openjfx</groupId>
            <artifactId>javafx-maven-plugin</artifactId>
            <version>0.0.8</version>
            <configuration>
                <mainClass>ConcurrencyExample</mainClass>
            </configuration>
        </plugin>
    </plugins>
</build>
```

Next, click the `Maven` tab on the right side of IDEA, choose `Plugins->javafx`, then click `javafx:run` to start your application. Or, you may click `Edit Configuration->+->Maven`, in `Run`, input `clean javafx:run` for your working directory.

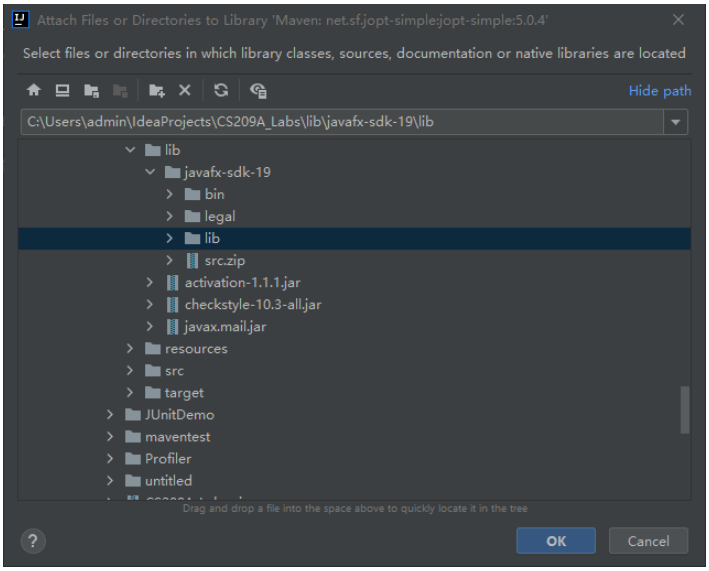


Click `Apply`. Now you can run this configured maven command to start your JavaFX application.

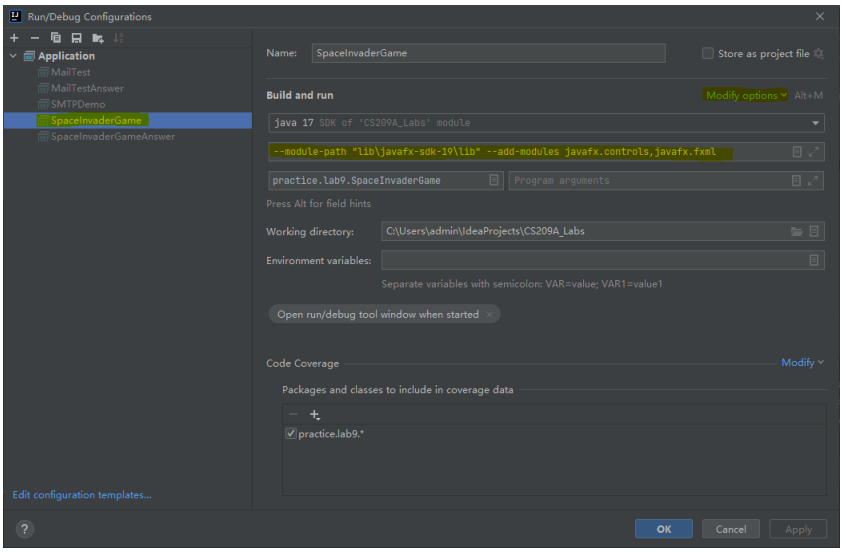
2.2 Manually

Alternatively, you could manually download JavaFX and configuring the classpath.

- Download JavaFX SDK from the official site. For example, download the Windows/x86/SDK and unzip the downloaded zip for Windows users.
- In the existing project, click `File` -> `Project Structures` -> `Project Settings` -> `Libraries`, click `+`, find your downloaded javafx-sdk folder and select the `lib` subfolder.



- Click `Run` -> `Edit Configurations`, select your JavaFX class on the left, click `Modify options` -> `Add VM options`, and add `--module-path "your-path-to-sdk\javafx-sdk-19\lib" --add-modules javafx.controls,javafx.fxml`



Now, you should be able to run your JavaFX code in this project.

3. Adding JavaDoc Support

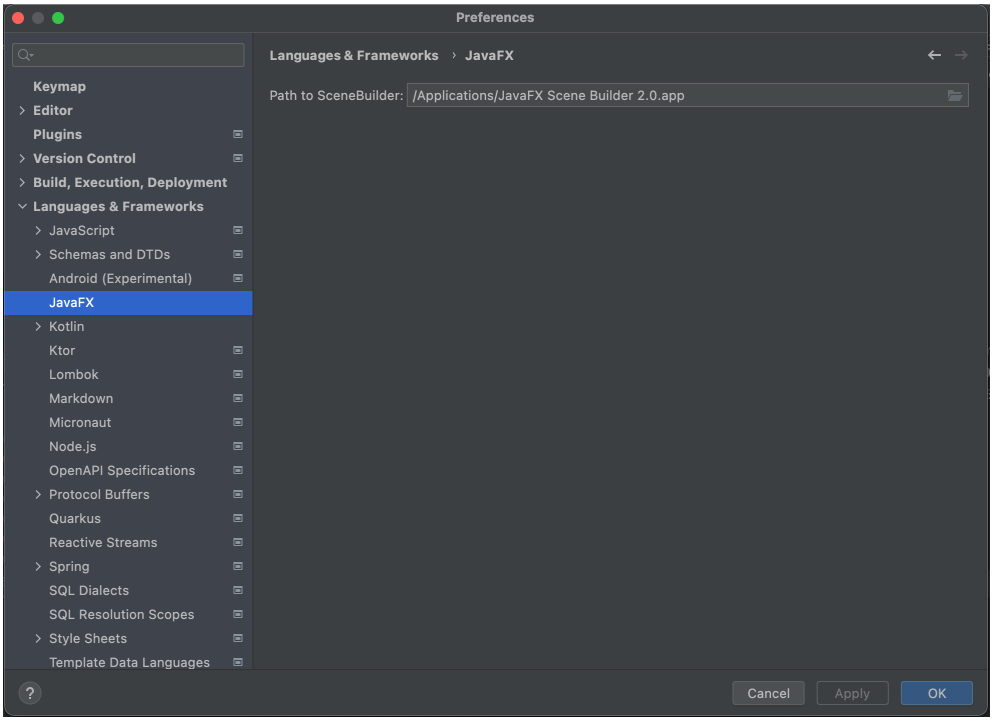
Documentation is super helpful for us to learn a library. If you want to bring up JavaFX's documentation in IDEA, right click `pom.xml` in your project, select "maven" -> "download documentation", which could automatically download corresponding documentation for your javafx dependency.

Once applying the above configuration, whenever you hover your mouse to a certain JavaFX entity in IDEA, you should be able to see its JavaDoc automatically pops up.

4. Using Scene Builder for UI Design

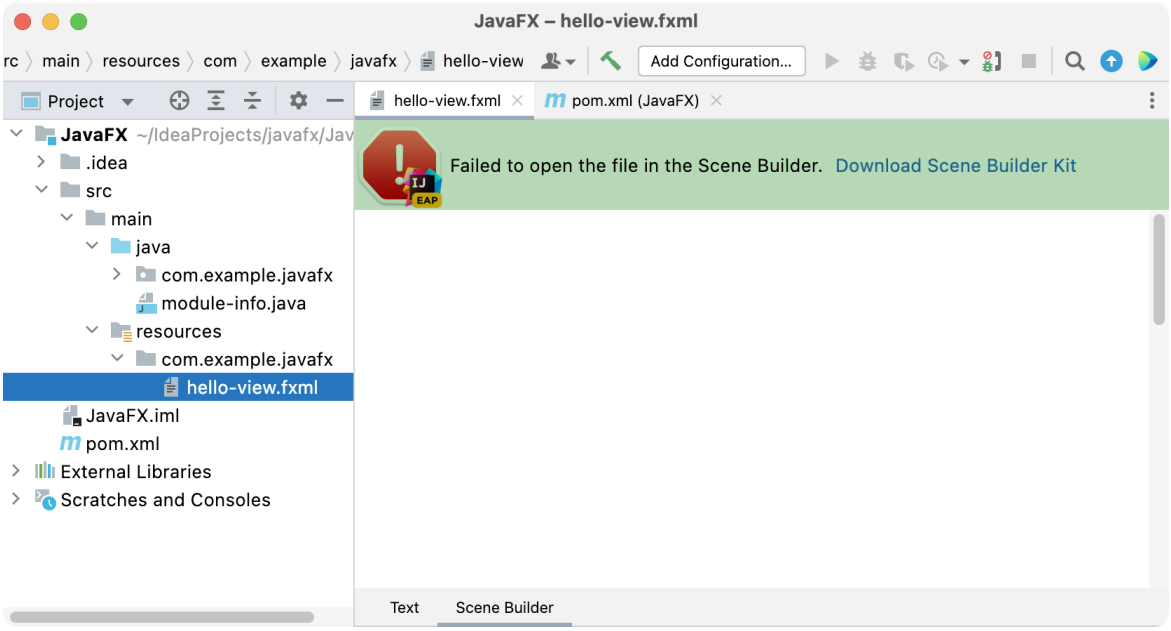
First, download Scene Builder here and install it.

Next, click "File" -> "Setting" -> "Language & Frameworks" -> "JavaFX", and specify the installation path for Scene Builder.

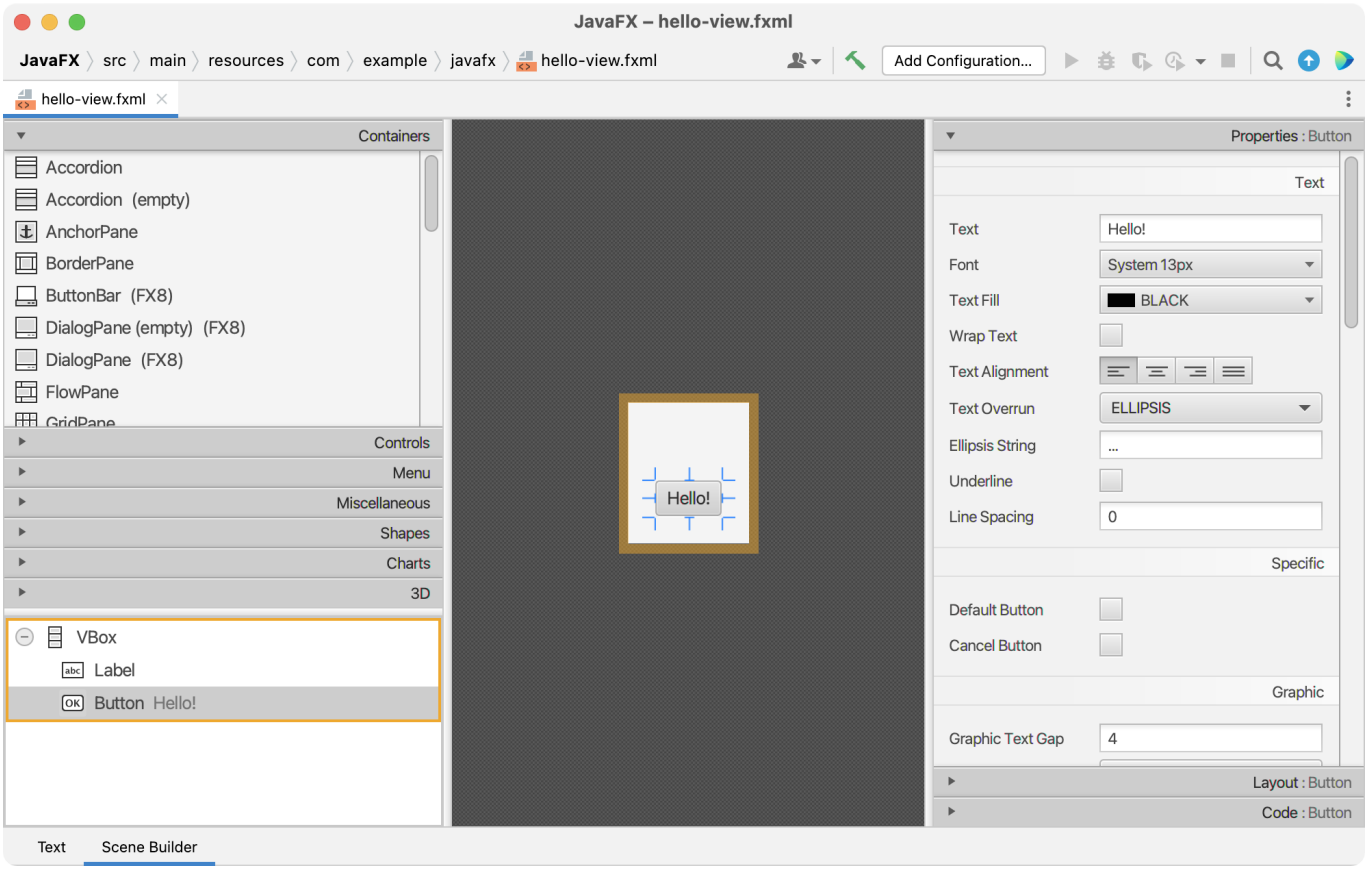


Now, when you open an `.fxml` file in the editor, there are two tabs underneath the editing area: the Text tab is for developing the markup, and the Scene Builder tab is for editing the file in Scene Builder.

If you cannot view `.fxml` in the Scene Builder tab, you'll see a notification like below. In that case, click `Download Scene Builder Kit` in the notification to download and install the tool.



Next, you could open the `.fxml` in the Scene Builder tab and design the UI visually.



If this still doesn't work, you could right-click .fxml and select "Open in Scene Builder", which will open the file in Scene Builder rather than inside IDEA.