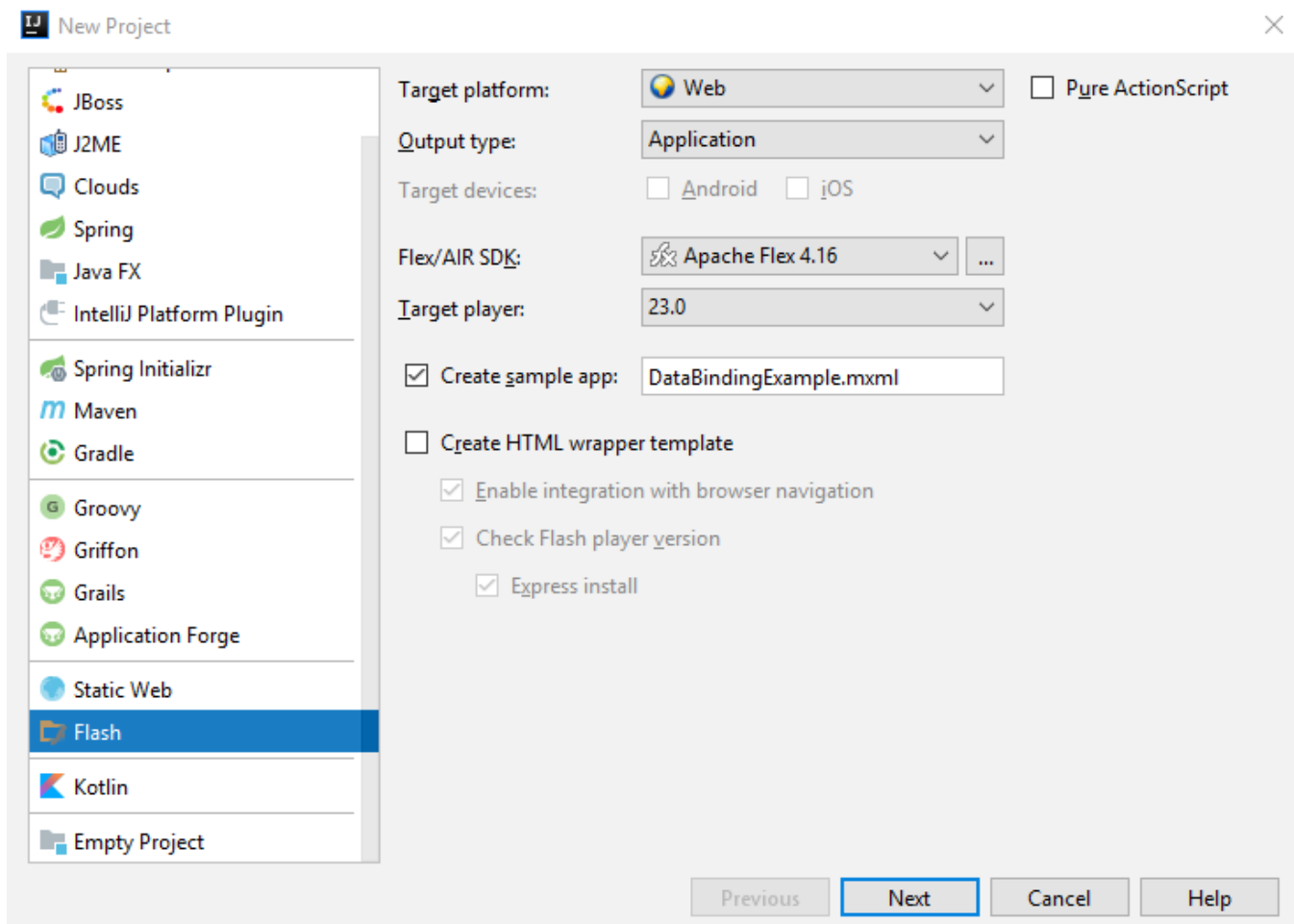


# Using FlexJS with IntelliJ IDEA

There is no particular enhancement made by JetBrains to support FlexJS but the Apache Flex Team has enhanced the Falcon compiler to make believe the IntelliJ Flex Plugin is using the legacy one but the configuration requires a little bit more tweaking than a classic Apache Flex application.

I will assume you know how to use IntelliJ and focus on the FlexJS configuration only, I will also assume, you installed the FlexJS SDK in a location that we will call <FLEXJS\_INSTALL\_FOLDER>.

The first gotcha is that you can't import an existing project, you will need to create a new one and copy your sources into the folder where you created your IntelliJ FlexJS project, here I chose a Web Application for this example but you can also choose a Web / Desktop / Mobile + Application / Library depending on what you are targeting.

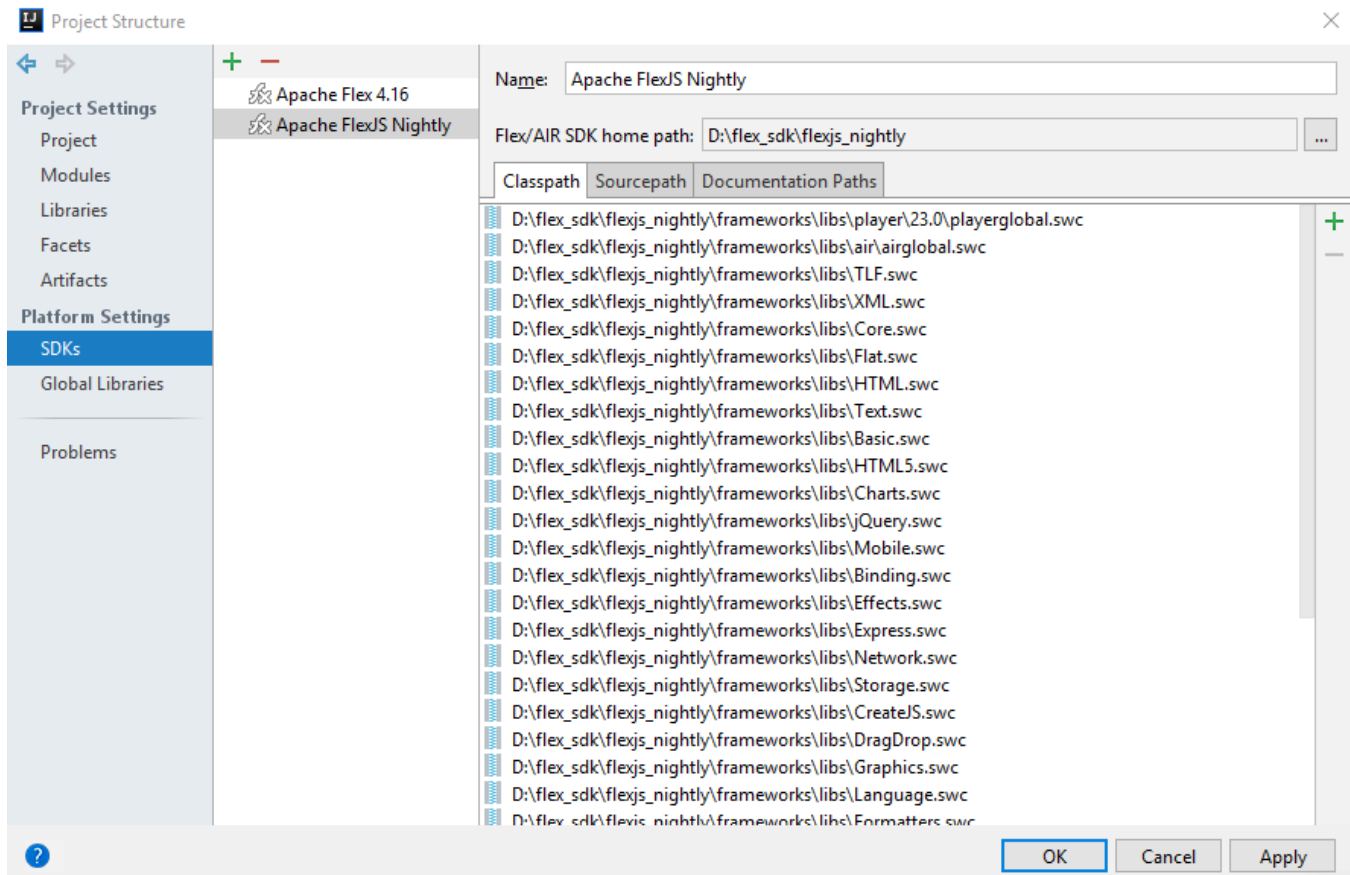


(You should have at least one Flex SDK installed and configured in IntelliJ to do that)

To start to play with, you can use the examples located in <FLEXJS\_INSTALL\_FOLDER>/examples. Actually, I will use DataBindingExample for the needs of this tutorial.

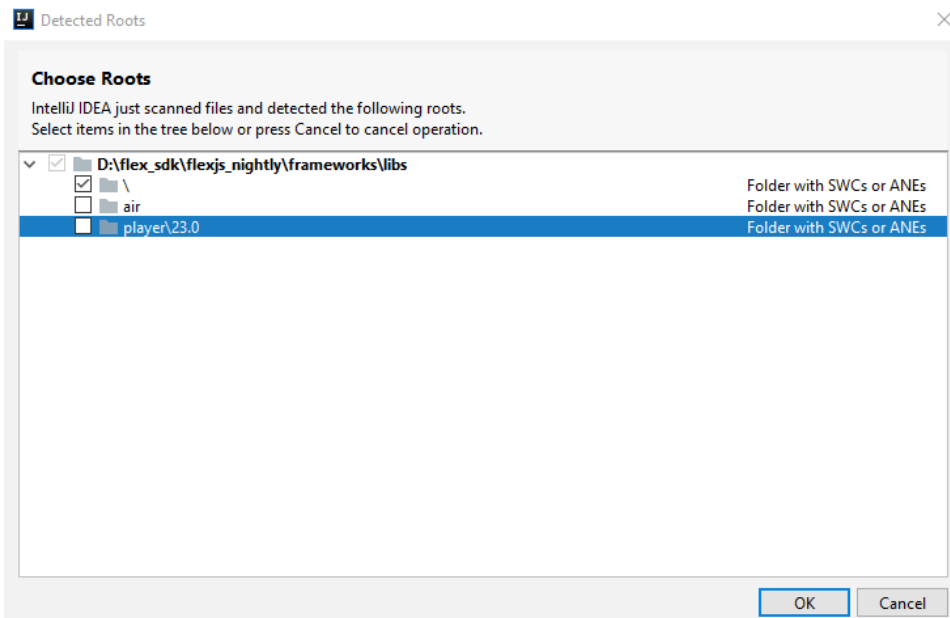
**The first thing you will need to do will be to add the FlexJS SDK to IntelliJ:**

1. Go to File -> Project Structure.. -> SDKs -> Add new SDK
2. Navigate to and select the FlexJS SDK from <FLEXJS\_INSTALL\_FOLDER>. (I'm using a nightly build here)

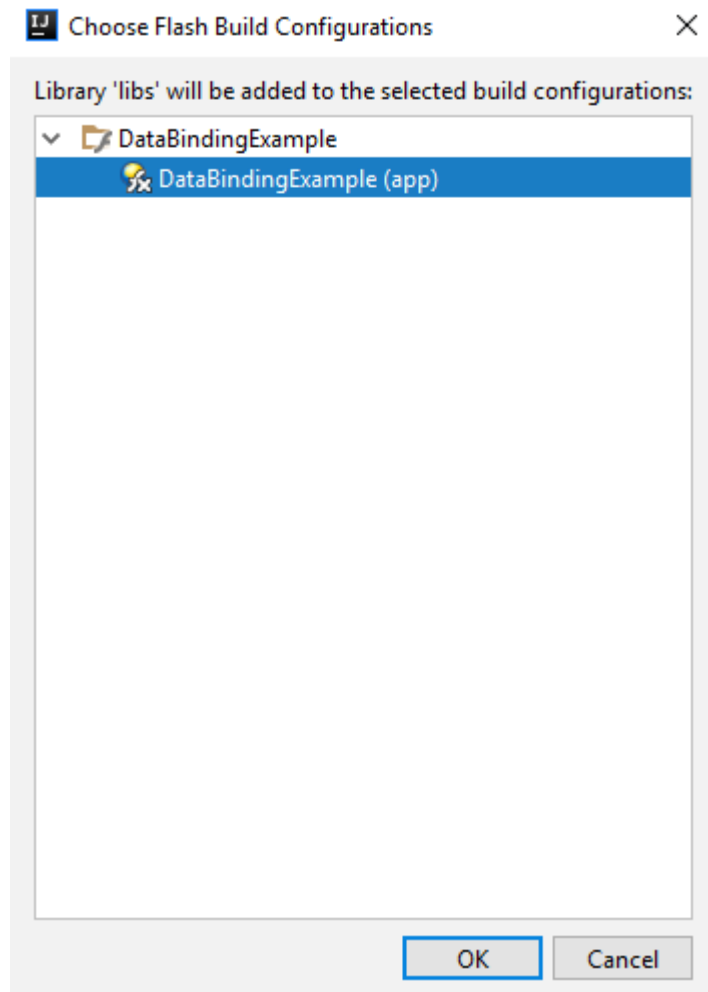


While this window is still open, click on Global Libraries, we will add the FlexJS SDK also as a library, that's the second gotcha:

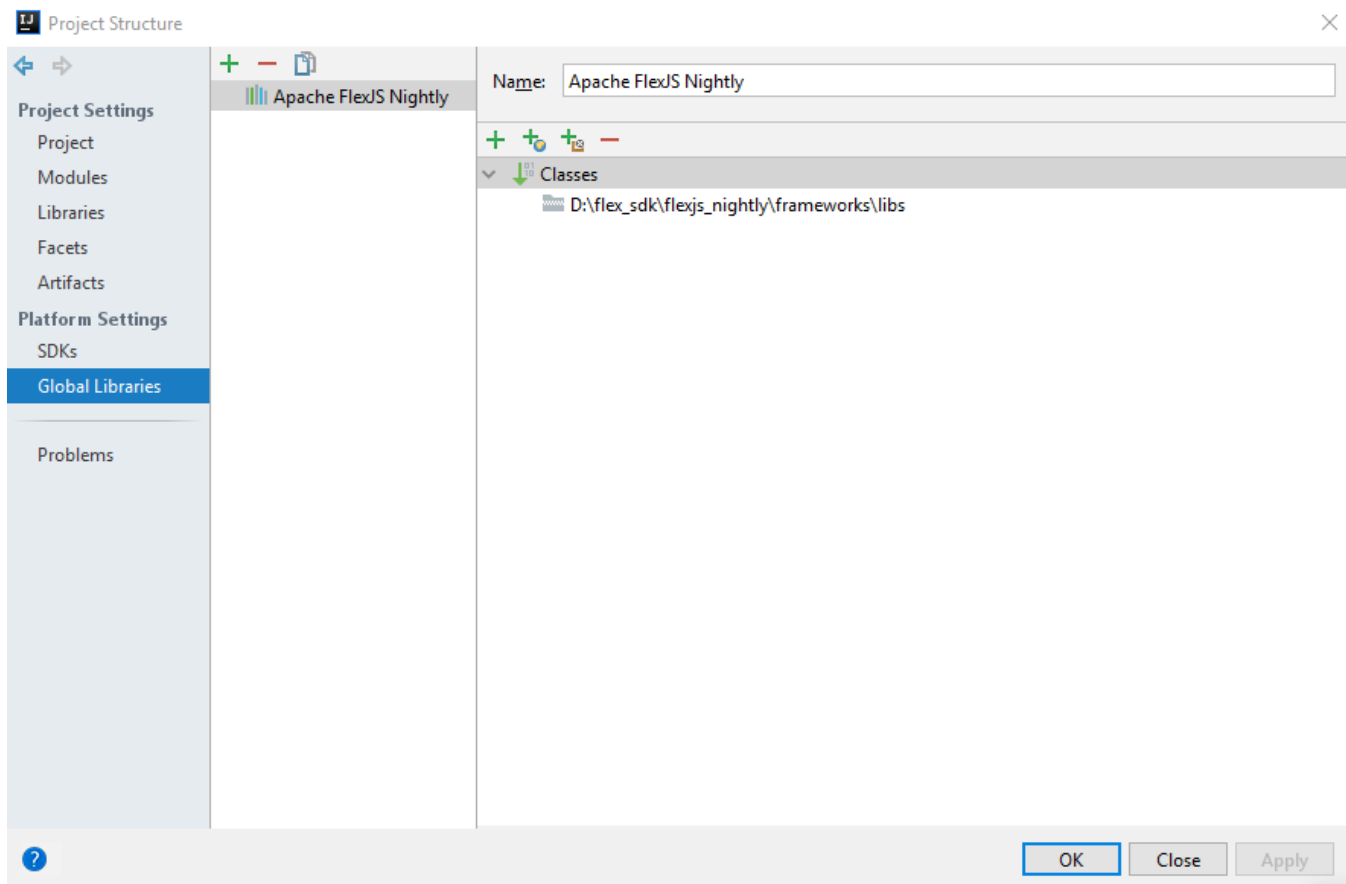
1. Click on Global Libraries, select <FLEXJS\_INSTALL\_FOLDER>/frameworks/libs. Then, deselect "air" and "player23.0" in my case.



2. In the next step IntelliJ will ask for choose project where library should be added

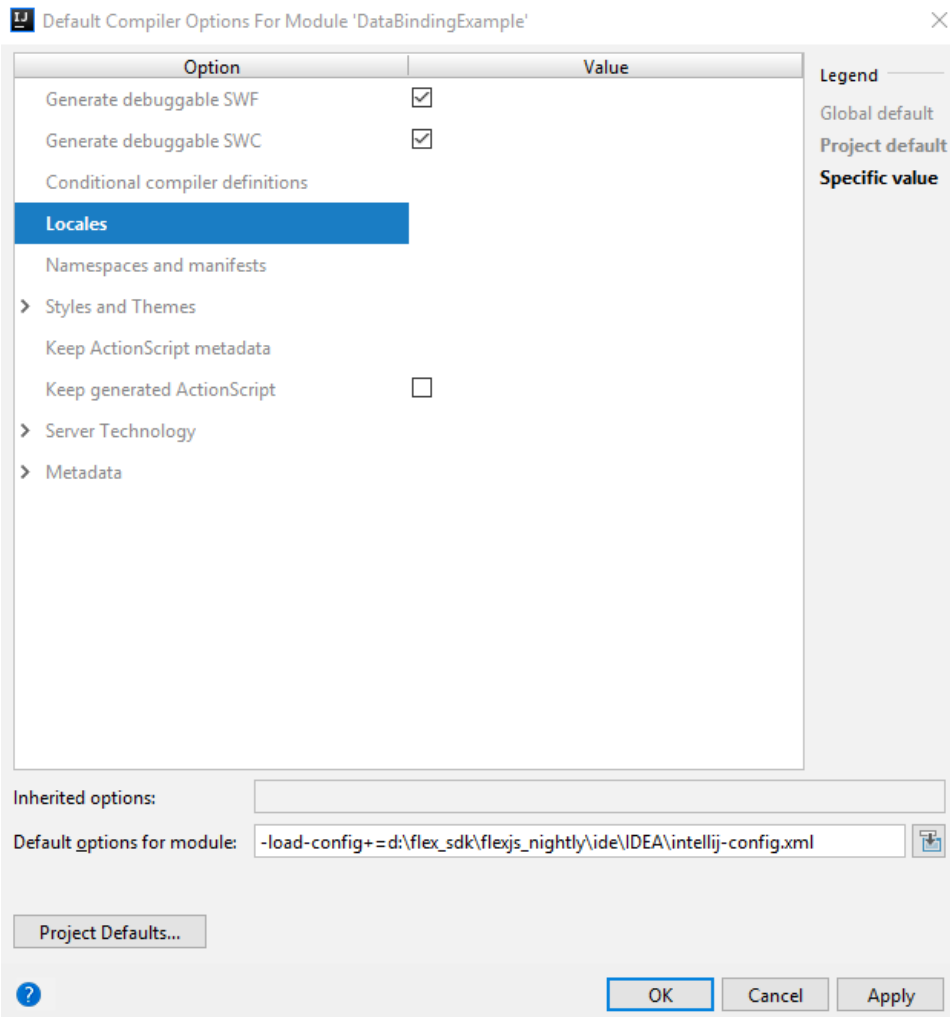


3. Rename the library to something meaningful, I choose to name mine "Apache FlexJS Nightly".

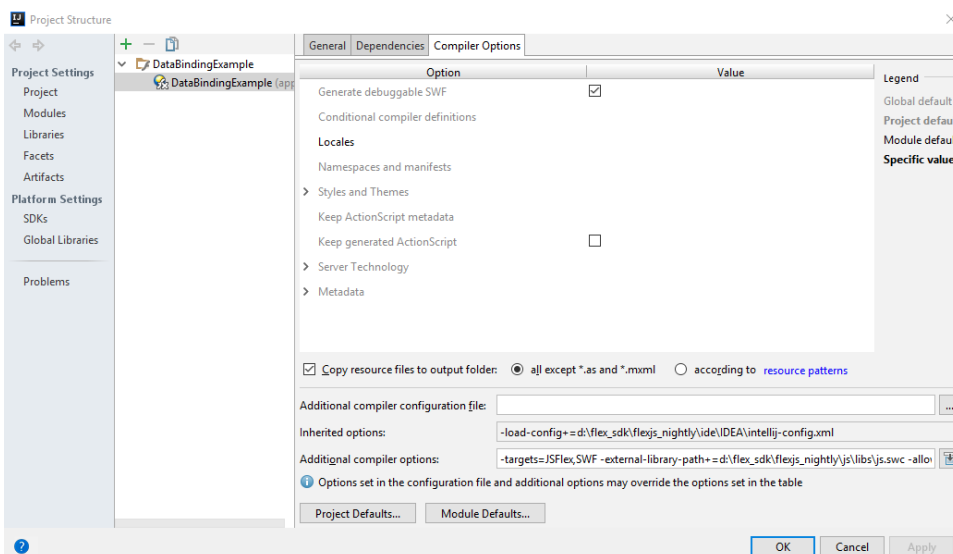


**We will now configure our module to use the freshly added SDK and library along with configuring the compiler options:**

1. Click on Modules, select the module you want to configure and click on Dependencies.
2. Select the FlexJS SDK you just added.
3. Add the FlexJS global library you just added too, if you have not added it in the previous steps.
4. Click on Compiler Options
5. Click on Module Defaults (If you have more than one module, it makes sense you apply this and the following to Project Defaults instead).
6. Remove the default en\_US local
7. In Default options for modules, add: `-load-config+=<FLEXJS_INSTALL_FOLDER>/ide/IDEA/intellij-config.xml`



8. Click OK to validate.
9. In Additional compiler options, add: `-targets=JSFlex,SWF -external-library-path+=<FLEX_JS_INSTALL_FOLDER>/js/libs/js.swc -allow-subclass-overrides=true`  
(The `-targets=JSFlex,SWF` will tell the compiler to build both the SWF / SWC and the JS version of your module. If you want to build the JS only, use `-targets=JSFlex`)

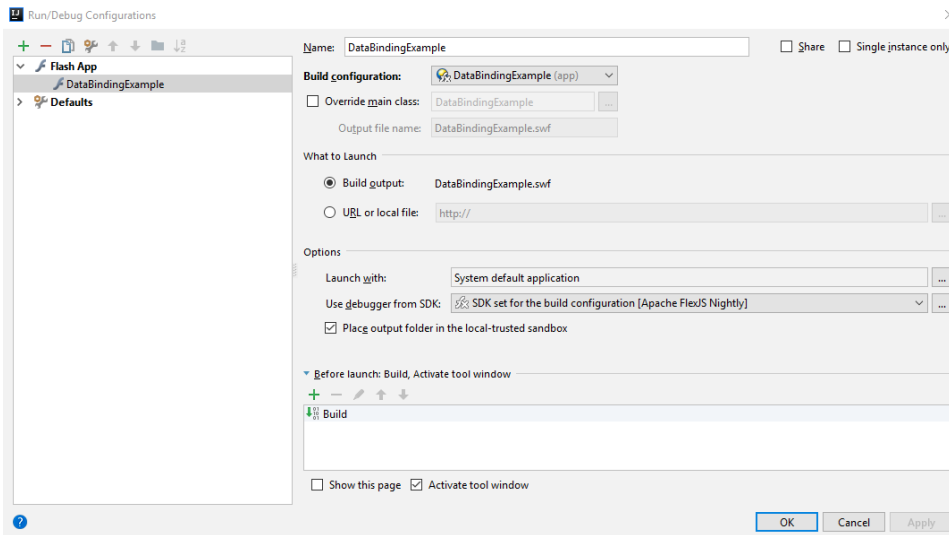


10. Click OK to validate.

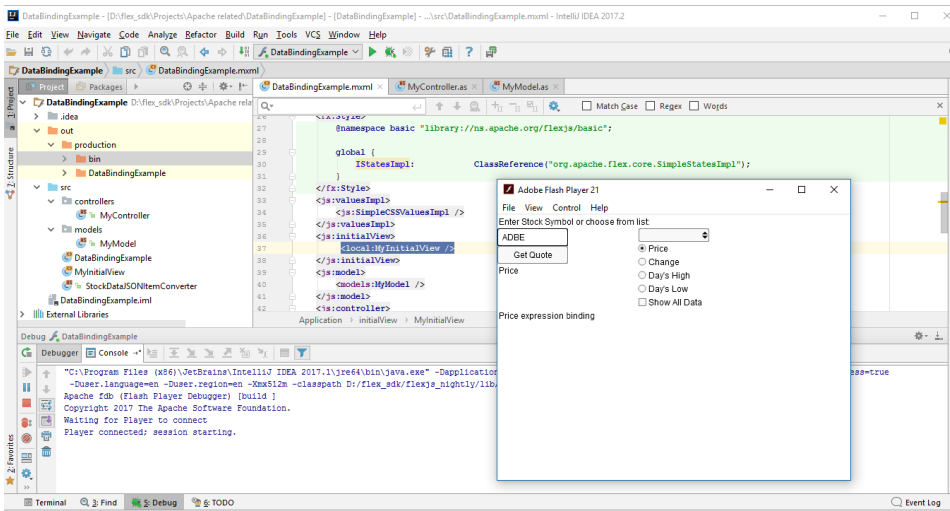
You should be able to build your project now, click on Build -> Rebuild project. (Fix the eventual error the compiler found and use Build -> Make project)

### In order to launch and debug SWF version of application:

1. Create a Launch Configuration for your SWF as you would do normally for a classic Flex application.



2. Click on Run or Debug for launch SWF version of application



### In order to launch and debug JS version of application:

1. In the Project pane, navigate to and right click on `/out/<Your module name>/js/out/bin/js-debug/index.html`
2. Click on Run or Debug 'index.html'. It will open the index.html in your default browser, in order to be able to debug your module in the browser, you will need to debug your application in Chrome.

I use to create another Chrome instance.

The `--disable-hang-monitor` option will allow you to be able to continue to debug even after the script time out limit in case of a SWF / SWC.  
The `--disable-bundled-ppapi-flash` will allow to use the NNAPI Flash debugger plugin instead of the PPAPI one, if it is not installed,  
you can download and install it from <https://www.adobe.com/support/flashplayer/downloads.html>

Once done, you will need to launch once the Debug configuration, it will open Chrome, close it, you see a popup appear in IntelliJ asking you to install the JetBrains IDE support for Chrome, do it, take care to configure the port as describe in the popup.

Set a break point in one of the JS file and launch the Debug configuration again, it will now break at it.

