Getting Started with IntelliJ IDEA Scala Plugin

Redirection Notice
This page will redirect to https://www.jetbrains.com/idea/help/creating-and-running-your-scala-application.html.

This getting started guide is intended for those who want to start working with the Scala IntelliJ IDEA plugin. Hope it gives you a good kick to start exploring the plugin for yourself, and, of course, reading this space - you can find more useful information about other helpful plugin features, like Java/Scala cross-language interoperability which includes refactorings, coding assistance and much more.
In this guide:

Before You Begin

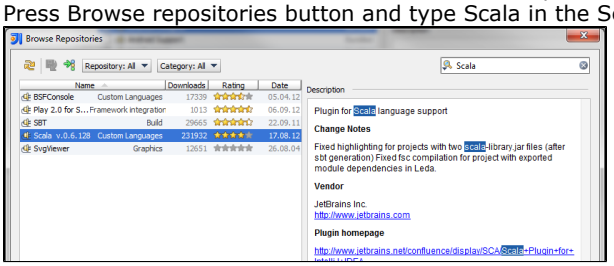
Currently, IntelliJ IDEA Scala plugin is in beta stage and is being actively developed. Here is the list of what you will need to try it:

- One of the latest IntelliJ IDEA EAP builds (can be downloaded from IntelliJ IDEA EAP page).
- Scala SDK should be downloaded and installed on your computer.

Setting up the Environment

After you have downloaded, installed and configured all the prerequisites, follow this simple procedure to get the Scala plugin up and running.

1. Run IntelliJ IDEA. On the Welcome Screen, click Open Plugin Manager.
2. Press Browse repositories button and type Scala in the Search box to quickly locate plugin in the list.



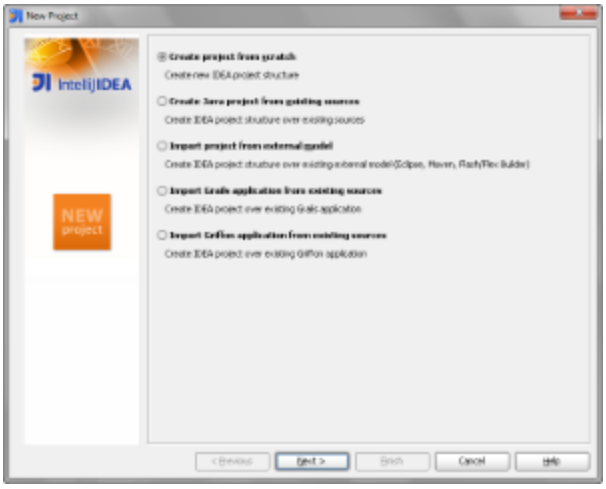
3. Right-click plugin name, select Download and Install from the menu, and restart IntelliJ IDEA to apply the changes.
4. After IntelliJ IDEA is back, open Plugin Manager again to examine the list of installed plugins. If everything is OK, there should be Scala plugin, listed in black, with a tick mark next to its name. All this means the plugin is installed, enabled and is working OK. If there's no tick mark, or plugin name is listed in red, or you can't find it at all - there may be some problem you need to address, like getting the latest IntelliJ IDEA EAP version.

There is an alternative way to install the Scala plugin.Download the Scala plugin from the Nightly Builds for Leda page and unzip the archive to the directory <user home>/.IntelliJIdea<xx>/config/plugins. After that, run IntelliJ IDEA.

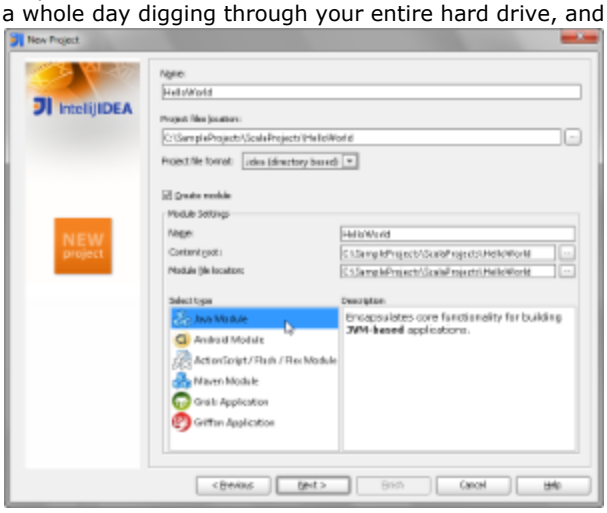
Creating a Project

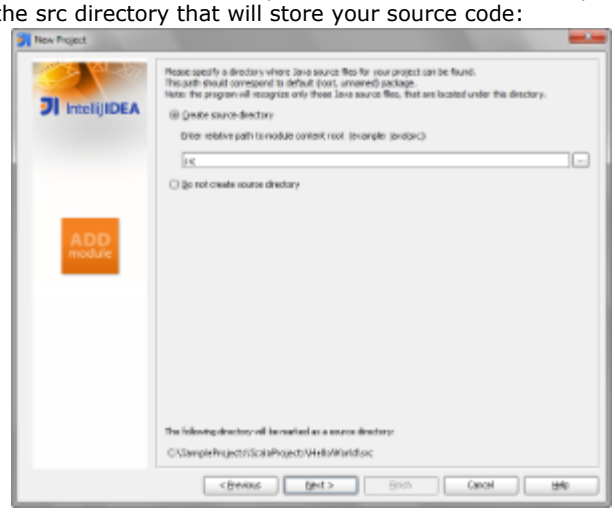
If everything is fine, you're ready to create the Scala project with IntelliJ IDEA.

1. Back to the Welcome Screen, click Create New Project. The good old IntellIJ IDEA New Project wizard apperas. Here you choose to create a new project from scratch:

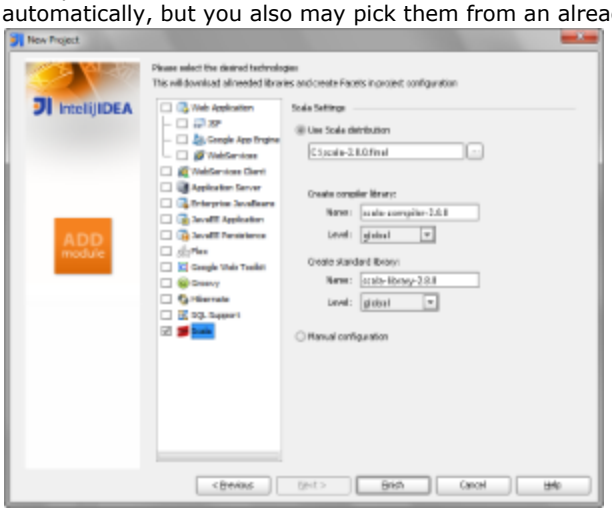


2. All you need is a Java Module with a name that makes sense, located somewhere you can find it later without spending a whole day digging through your entire hard drive, and the src directory that will store your source code:





3. On top of that, select Scala on the Select Desired Technologies. By default all the necessary jars will be downloaded automatically, but you also may pick them from an already existing IDEA library or a secret place on your hard drive.



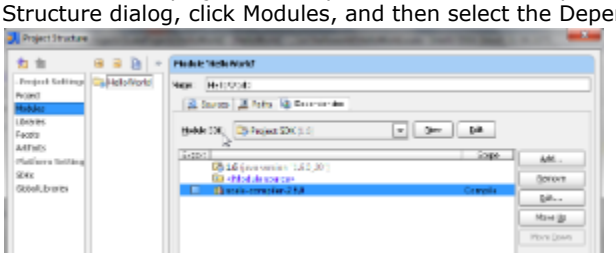
4. It's possible to adjust the level of a library you want to create. Select Global or Project from the Level drop-down list.

- If you plan to create other projects that involve this library, select Global - thus you will be able to share the library anywhere on the current machine.
- If you select Project level, then all the jar archives will be included in your project. This makes your project portable between the different computers and independent from the Scala SDK.

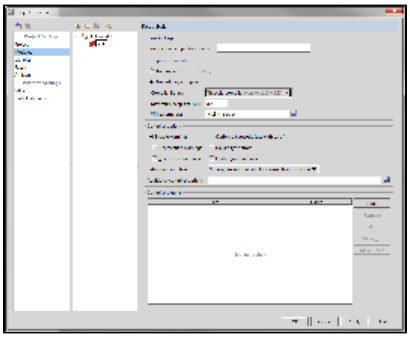
To significantly speed-up project compilation you may configure the Fast Scala Compiler (FSC).

Exploring Project Structure

Now, when the project is ready, it is advisable to explore its settings. To do that, on the main toolbar, click  . In the Project Structure dialog, click Modules, and then select the Dependencies tab:



Next, pay attention to the Scala facet added to our module:

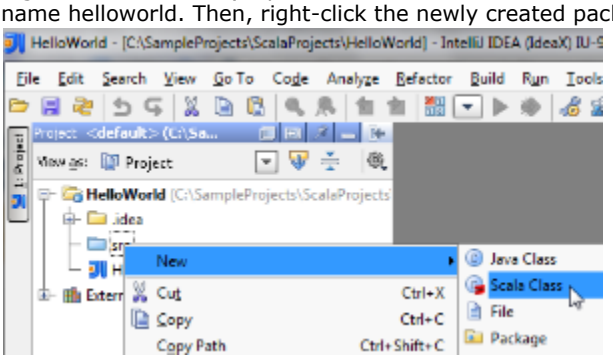


As you see, IntelliJ IDEA has automatically specified the Scala compiler library, and added it to the list of module dependencies. If you want to learn more about project configuration, refer to http://devnet.jetbrains.net/thread/290032.

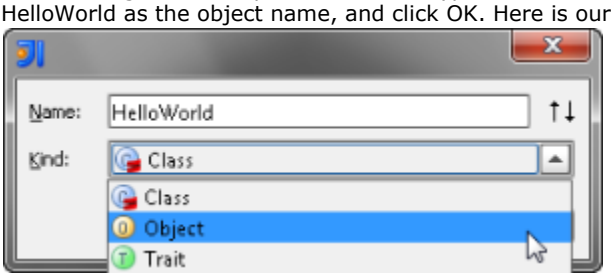
Creating Source Code

Now everything is ready for the first HelloWorld application.

1. Right-click src folder (or press Alt-Ins in default Windows IntelliJ IDEA keymap) and select Package. Type the package name helloworld. Then, right-click the newly created package and choose Scala Class on the context menu:

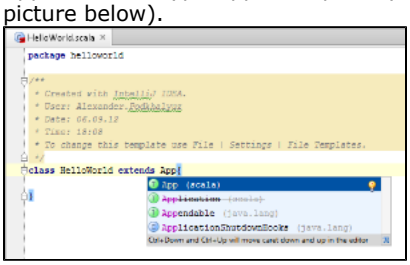


2. In the dialog box that opens, select the type of the Scala entity to be created - in our case, let it be Scala Object, type HelloWorld as the object name, and click OK. Here is our first Scala object.

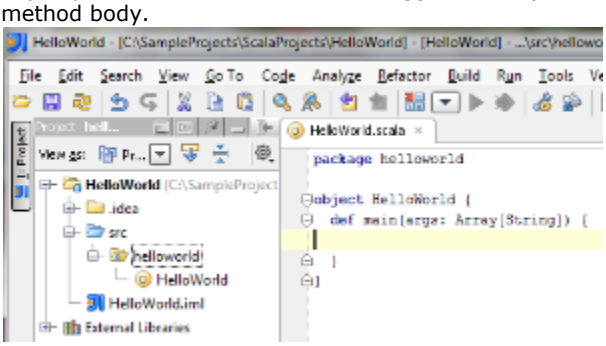


> You can skip first step. Instead of HelloWorld you can type helloworld.HelloWorld then appropriate package will be created automatically.
> To choose 'Object' in the combobox you can use up and down arrows on your keyboard, just to do it slightly faster.

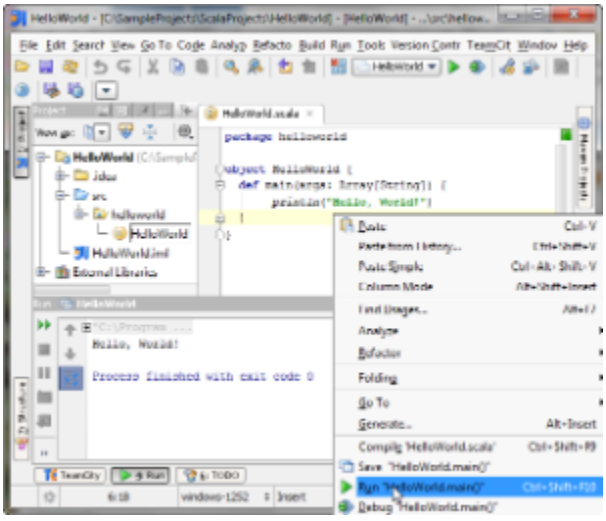
3. Now, we're going to make it runnable. There're several ways to achieve that. First off, we can extend the default scala.App trait. Just type App and space, you even don't need to use code completion, which will work automatically (see the picture below).



Or, we can use a more Java-like approach: create main() method inside of object body. Again, IntelliJ IDEA lends us a hand with Live Templates. All we do is type main and press Tab (or press Ctrl+J in the default Windows IntelliJ IDEA keymap, and select main from the suggestion list). IntelliJ IDEA expands this macro to a complete well-formed main() method body.



4. Let's teach the application to do some work. If you're familiar with IntelliJ IDEA, you should probably be aware of that running a simple Java application is just a matter of pressing Ctrl+Shift+F10 (on default Windows keymap). Good news is that it works equally well for Scala applications.
All we have to do is to type the code that displays the trivial Hello World message and then press Ctrl+Shift+F10 - or right-click the editor background and choose Run HelloWorld.main on the context menu. The application output is displayed in the console.



You can also turn your Scala object into a script. Just remove all declarations from the file, leave the executable statement only - println("Hello, World!"), and run the script as described above.