

Setting Up IntelliJ IDEA for Creating and Testing Scala Projects

- [Official Docs From Scala-Lang.org](#)
 - [IntelliJ Documentation Links](#)
 - [IMPORTANT Warning About sbt Version 1.3.0](#)
 - [Other Options](#)
 - [Using sbt Directly](#)
 - [Using VSCode + sbt](#)
- [Legacy Instructions](#)
 - [Installing Java 8](#)
 - [Installing IntelliJ](#)
 - [Creating a Project](#)
 - [Locating Scala SDK and the JDK](#)
 - [Writing Code](#)
 - [Writing Tests](#)
- [Running Tests](#)

Official Docs From Scala-Lang.org

The official Scala Language download page now recommends installing Scala by installing IntelliJ IDEA.

<https://www.scala-lang.org/download/>

IntelliJ Documentation Links

We recommend following the directions on the same pages linked to on the scala-lang.org download page:

1. [Getting Started with Scala in IntelliJ](#)
Just follow the "Installation" instructions from this page. To actually create a project and build Scala code, you should follow the directions on the next page.
The description of Scala Worksheets in "Experimenting with Scala" might be interesting to you, even though you won't create Worksheets for the homework.
2. [Building a Scala Project with IntelliJ and sbt](#)
You should use Scala version 2.13.0 and sbt version 1.2.8 for your projects.
3. Optional Reading: [Testing Scala in IntelliJ with ScalaTest](#)
The `build.sbt` file included with the `hw_1` provided code will pull in the correct version of the ScalaTest library, but this is still a nice reference.
We will only cover basic features of ScalaTest in class, but you're welcome to use more advanced features from its [FunSuite](#) module if you'd like.

IMPORTANT Warning About sbt Version 1.3.0

~~There is currently a bug in sbt (Scala Build Tool) version 1.3.0 that interacts poorly with the official Scala plugin for IntelliJ (see [SCL-16209](#)). You should use sbt version 1.2.8 (the previous stable version) until a new version is released. The bug has been patched in sbt, so we expect version 1.3.1 and onward will work correctly once available.~~

A patched version of the IntelliJ Scala plugin has been released. If you're using the newest version this should no longer be an issue.

Other Options

Using sbt Directly

If you don't like IntelliJ, the second official recommendation from scala-lang.org is to download sbt (Scala Build Tool) directly, then run commands from your shell to build and test your project. If you prefer using Vim or Emacs as your primary editor, this is a good option for you.

1. [Getting Started with Scala and sbt on the Command Line](#)
2. [Testing Scala with sbt and ScalaTest on the Command Line](#)

Using VSCode + sbt

If you still like the feel of a traditional IDE but IntelliJ feels too heavy-weight, you can also try Microsoft's open source [VSCode](#) IDE with its [Scala Metals plugin](#). The plugin gives you syntax highlighting, auto-complete, compiler error highlighting, usage location, etc.; however, it's not as fully-integrated as IntelliJ since you still have to compile and test from the shell by invoking sbt directly.

Legacy Instructions

Adapted from Professor Wallach's Comp 215 instructions for installing IntelliJ IDEA with Java 8.

This document walks through all the necessary steps to install IntelliJ properly, including the necessary libraries and plugins to use is with Scala.

Installing Java 8

You need to install the latest version of the Java Development Kit (JDK). You can get it here

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

You want the "Java SE Development Kit 8u60" for whatever your computer happens to be. Oracle supports you for Windows, Mac OS X, and Linux.

If you're using an "old" version of Java8 that you may have installed in prior months or years, this is a good time to update to the latest and greatest. Prof. Wallach started out writing code for Comp 215 with last year's "8u25" version and managed to get the Java compiler to blow up on his perfectly reasonable and beautiful code. (That said, something that worked great for Prof. Wallach with 8u51 totally failed in the latest 8u60. We'll work through these issues when they occur. Don't panic.)

Installing IntelliJ

Visit the IntelliJ IDEA download site: <https://www.jetbrains.com/idea/download/>

and download the free "community edition". (There's nothing in the "ultimate" edition that will make any difference to your work in Comp215, Comp322, Comp311, or (hopefully) most other undergraduate computer science courses.) Versions are available for Windows, Mac OS X, and Linux. IntelliJ is an example of an integrated development environment (IDE). The other popular Java IDE you may have heard of or used in the past is Eclipse.

If this is the first time you have installed IntelliJ IDEA on your computer, the software will ask you to customize it a bit before starting. You can pick a color scheme that you like, then it will ask you to customize different plugins. Go through these pages and follow these instructions:

- On the page "Tune IDEA to your tasks" click on "customize" for "Test Tools". You should see that JUnit is automatically installed. Click on "Save Changes and Go Back".
- When you get to the page "Download featured plugins" you will see a column for Scala. Click on "Install" for that plugin. The Scala plugin will install (this should take less than a minute on a reasonably responsive network).

Creating a Project

If you've got something old around, this is a menu entry, **File New Project...** Otherwise, if you're staring at the "**Welcome to IntelliJ IDEA**" tiny screen, there's a "**Create New Project**" button. Click it. Either way, you'll see many different kinds of projects to choose from. In the left column, near the bottom, it should offer you a "**Scala**" project as an option. Select that. It will give you sub-options for "**Scala**", "**SBT**", and "**Activator**". Click "**SBT**". Then click "**Next**".

Locating Scala SDK and the JDK

You will now need to tell IntelliJ where to find the Java 8 JDK as well as Scala.

For **Project SDK**, click "New" and locate the folder containing the JDK you recently downloaded. Its location will be platform-specific:

- On Mac OS: **/Library/Java/JavaVirtualMachines/jdk1.8.0_60.jdk**

For **Scala SDK**, click "Create". If **Scala 2.11.7** does not appear as an option, click "**Browse**" and navigate to the Scala folder you recently downloaded.

Writing Code

You will now have a new project with no open files. To create your first Scala file:

- On the left-hand side of the screen, below the word "Project" you will see a list of directories in a tree view. The first directory should have the same name as your project. Open this directory.
- Right click on the subdirectory **src**.
- Navigate to the **New** option and select **File**
 - **Warning:** If you select **Scala Class**, IntelliJ will create an ordinary class, not a case class for you. Be sure to edit the source code it creates accordingly.
- You will be prompted for a file name. Name it **your-file-name.scala**
- You can now enter source code for your program in this file

Writing Tests

The easiest way to start using JUnit with your project in IntelliJ is to:

- Place the cursor over the declared name of the class in the source file for the class
- Select **Navigate Test Create New Test**
- A dialog box will pop up. Select **JUnit 4**.
- Directly beneath your selection, you will see an error message: **JUnit 4 library not found in the module**. Sigh and click **Fix**. (Amazingly, this seems to be the easiest way to import JUnit into your project.)
- **Click Ok**.
- Bizarrely, this procedure will produce a Java source file, not a Scala one. Select **Refactor Rename File**, and rename the file to have a .scala suffix.
- Delete the auto-generated Java code in your test file and replace it with this template:

```
import junit.framework.TestCase
import junit.framework.Assert._

class YourTestClass(name: String) extends TestCase(name) {

}
```

- You can now fill this test class in with your test methods as discussed in class.

Running Tests

Run your tests in IDEA as follows:

- Right-click (or Ctrl-Click) on the name of your test file in your project's directory tree on the left.
- Select **Run your-test-class**
- IDEA will run JUnit on your test class and the progress will be reported at the bottom of the screen. A green bar indicates all tests passed, a red bar indicates some failed. You can click on the failures at the bottom of the screen to go to the corresponding location in the source code.