

Getting strated with Scala and SBT from Command Line

Install SBT

1. Make sure you have the Java 8 JDK (also known as 1.8)
 - Run **javac -version** in the command line and make sure you see javac 1.8.____
 - If you don't have version 1.8 or higher, install the JDK
 - JDK URL: <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
2. Install SBT on windows
 - Reference Link: <https://www.scala-sbt.org/1.x/docs/Installing-sbt-on-Windows.html>
 - Download windows installer (msi installer) from the above link and run the file.

Create the project

- **cd** to an empty folder.
- Run the following command **sbt new scala/hello-world.g8**. This pulls the 'hello-world' template from GitHub. It will also create a target folder, which you can ignore.
- When prompted, name the application **hello-world**. This will create a project called "hello-world".
- Let's take a look at what just got generated:

```
- hello-world
  - project (sbt uses this to install and manage plugins and dependencies)
    - build.properties
  - src
    - main
      - scala (All of your scala code goes here)
        - Main.scala (Entry point of program) <-- this is all we need for now
      - build.sbt (sbt's build definition file)
```

- After you build your project, sbt will create more target directories for generated files. You can ignore these.

Running the project

- **cd** into hello-world.
- Run **sbt**. This will open up the sbt console.
- Type **~run**.
 - The ~ is optional and causes sbt to re-run on every file save, allowing for a fast edit/run/debug cycle. sbt will also generate a target directory which you can ignore.

Modifying the code

- Open the file `src/main/scala/Main.scala` in your favorite text editor.
- Change “Hello, World!” to “Hello, New York!”
- If you haven’t stopped the `sbt` command, you should see “Hello, New York!” printed to the console.
- You can continue to make changes and see the results in the console.

Adding a dependency

- Changing gears a bit, let’s look at how to use published libraries to add extra functionality to our apps.
- Open up `build.sbt` and add the following line:

```
libraryDependencies += "org.scala-lang.modules" %% "scala-parser-combinators" % "2.1.1"
```

- Here, `libraryDependencies` is a set of dependencies, and by using `+=`, we’re adding the `scala-parser-combinators` dependency to the set of dependencies that `sbt` will go and fetch when it starts up. Now, in any Scala file, you can import classes, objects, etc, from `scala-parser-combinators` with a regular import.
- You can find more published libraries on Scaladex, the Scala library index, where you can also copy the above dependency information for pasting into your `build.sbt` file.

