04. Setting up & getting started with sbt

Posted on October 20, 2018

SBT (i.e. **S**imple **B**uild **T**ool) for Scala requires Java & Scala. This tutorial is based on Java 8 (i.e. 1.8.0) and Scala 2.12.7. Scala runs on the JVM, so Java and Scala stacks can be freely mixed. You can call Java libraries from Scala.

Install sbt

Step 1: Install sbt on Windows/Mac.

To install sbt on Windows, download and run the Windows MSI from http://www.scala-

300+ Java Interview FAQs

300+ Java FAQs

16+ Java Key Areas Q&As

150+ Java Architect FAQs

80+ Java Code Quality Q&As

150+ Java Coding Q&As

300+ Big Data Interview FAQs

300+ Big Data FAQs

Tutorials - Big Data

TUT - Starting Big Data

TUT - Starting Spark & Scala

TOP

sbt.org/download.html.

To install sbt on Mac

```
brew install sbt
```

Creating a project & sbt commands on command-line

Step 2: Create a new folder "sbt-tutorial" inside a folder say "projects". Change directory "~/projects/sbt-tutorial".

```
$ cd ~/projects/sbt-tutorial
~/projects/sbt-tutorial]$ sbt
sbt:sbt-tutorial>
```

Step 3: On the sbt command-line you can use "show" command to display Scala and the project versions.

```
sbt:sbt-tutorial> show scalaVersion
[info] 2.12.7
```

```
sbt:sbt-tutorial> show version
[info] 0.1.0-SNAPSHOT
```

Step 4: You can modify the version with the "**set**" command.

Any changes need to be saved with "**session save**". The values that are modified with the "**set**" commands will be saved to the "**build.sbt**" file.

```
sbt:sbt-tutorial> session save

```

```
sbt:sbt-tutorial> set version := "0.1"

```

Step 5: You can open a Scala REPL within sbt with the "**console**" command.

```
sbt:sbt-tutorial> console
[info] Starting scala interpreter...
Welcome to Scala 2.12.7 (Java HotSpot(TM) 64-Bit S
Type in expressions for evaluation. Or try :help.

scala>

```

You can exit Scala REPL with ":q" or "sys.exit" commands.

Step 6: You can exit the sbt with the "exit" command. You should now have the directories project, target & the **build.sbt** file.

Creating the project artefacts in Scala

Step 7: In "/project/sbt-tutorial", create a new folder "src/main/scala".

```
~/projects/sbt-tutorial]$ mkdir -p "src/main/scala

```

Step 8: Create a package inside "src/main/scala" named "com/sbt-hello".

```
~/projects/sbt-tutorial/src/main/scala]$ mkdir -p

```

Step 9: Create a new Scala file in the “com.sbthello” package.

```
~/projects/sbt-tutorial/src/main/scala/com/sbt-hel

```

Step 10: Open a text editor or an IDE like “Visual Studio Code”, and type the following basic Scala code.

```
package com.sbthello

object HelloSbt {
   def main(args: Array[String]) = {
       print("Hello Sbt")
   }

}
```

Step 11: Type “**sbt**” from a command-line from the project folder (i.e. ~/projects/sbt-tutorial).

```
~/project/sbt-tutorial]$ sbt
```

Step 12: Type “**compile**” to compile the code.

```
sbt:sbt-tutorial> compile
[success] Total time: 1 s, completed 20/10/2018 1:

```

Step 13: Type “**package**” to package the code as a jar file

```
sbt:sbt-tutorial> package
[success] Total time: 0 s, completed 20/10/2018 1:

```

You can see the built jar file "**sbt-tutorial_2.12-0.1.jar**" in the "~/projects/sbt-tutorial/target/scala/2.12" folder.

Making it ready for Eclipse import

Step 14: Create the "**plugins.sbt**" file inside the "~/projects/sbt-tutorial/project", add the following line.

```
addSbtPlugin("com.typesafe.sbteclipse" % "sbteclip
```

Step 15: From the command-line type "**sbt**", and then when inside sbt type "**eclipse**" to generate the eclipse metadata files like like .project & .classpath.

```
~/project/sbt-tutorial]$ sbt
```

```
sbt:sbt-tutorial> eclipse
```

Step 16: Open Eclipse with Scala IDE installed as per the previous tutorial. Within Eclipse import "sbt-tutorial" with "**File** –> **Import** –> **General** –> **Existing Projects into Workspace**".

Step 17: Within Eclipse select the top level of the Scala project, right-click and select "**properties**" and select "**Scala compiler**". Make sure that the "**User project settings**" is ticked.

Step 18: Now select the folder "sbthello" right-mouse-click and select "**Build Path**" and click on "**Include**". Do the same for "**com**" as well. Now Eclipse recognises "com.sbthello" as a package. The symbol of both the package and the file “HelloSbt.scala” change.

Step 19: Right mouse click on “HelloSbt.scala” and select “**Run As**” and then “**Scala Application**“.

Outputs:

```
Hello Sbt
```

‹ 9+ CI/CD Docker compose DevOps interview Q&As

Docker DevOps Q&As with Big Data code snippets ›

Top