



```
edureka@localhost:~
File Edit View Search Terminal Help
[edureka@localhost ~]$ sudo yum localinstall sbt-0.13.8.rpm
[sudo] password for edureka:
Loaded plugins: fastestmirror, refresh-packagekit
Setting up Local Package Process
Examining sbt-0.13.8.rpm: sbt-0.13.8-1.noarch
Marking sbt-0.13.8.rpm to be installed
Loading mirror speeds from cached hostfile
 * base: mirror.dhakacom.com
 * extras: mirror.dhakacom.com
 * updates: mirror.dhakacom.com
Resolving Dependencies
There are unfinished transactions remaining. You might consider running yum-complete-transaction first to finish them.
The program yum-complete-transaction is found in the yum-utils package.
--> Running transaction check
--> Package sbt.noarch 0:0.13.8-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                               Arch                               Version                               Repository
=====
Installing:
sbt                                  noarch                             0.13.8-1                             /sbt-0.13.8
=====
Transaction Summary
=====
Install      1 Package(s)

Total size: 1.2 M
Installed size: 1.2 M
Is this ok [y/N]: █
```

Now check the sbt version.

**sbt --version**

```
edureka@localhost:~
File Edit View Search Terminal Help
[edureka@localhost ~]$ sbt --version
sbt launcher version 0.13.8
[edureka@localhost ~]$ █
```

To import your project to Eclipse, you must add **sbtclipse** to your plugin definition file. You can use either the global one at **~/.sbt/plugins/plugins.sbt** or the project-specific one at **PROJECT\_DIR/project/plugins.sbt**:

We will add the plugin globally so that we need not add it in each project individually, follow below steps.

Make a plugin directory inside **.sbt/0.13/**

**mkdir -p .sbt/0.13/plugins**

Create a file plugins.sbt

**sudo gedit .sbt/0.13/plugins/plugins.sbt**

```
edureka@localhost:~  
File Edit View Search Terminal Help  
[edureka@localhost ~]$ mkdir -p .sbt/0.13/plugins  
[edureka@localhost ~]$ sudo gedit .sbt/0.13/plugins/plugins.sbt
```

```
addSbtPlugin("com.typesafe.sbteclipse" % "sbteclipse-plugin" % "4.0.0")
```

```
File Edit View Search Tools Documents Help  
Open Save Undo  
plugins.sbt  
addSbtPlugin("com.typesafe.sbteclipse" % "sbteclipse-plugin" % "4.0.0")
```

After installation, the next time we launch sbt we will be able to use the additional command eclipse. Below are the steps to run a project directly by sbt, and then we will see how you can run the project in eclipse.

## sbteclipse

```
mkdir helloworld
```

```
cd helloworld/
```

```
mkdir -p src/main/scala
```

```
sudo gedit src/main/scala/hello.scala
```

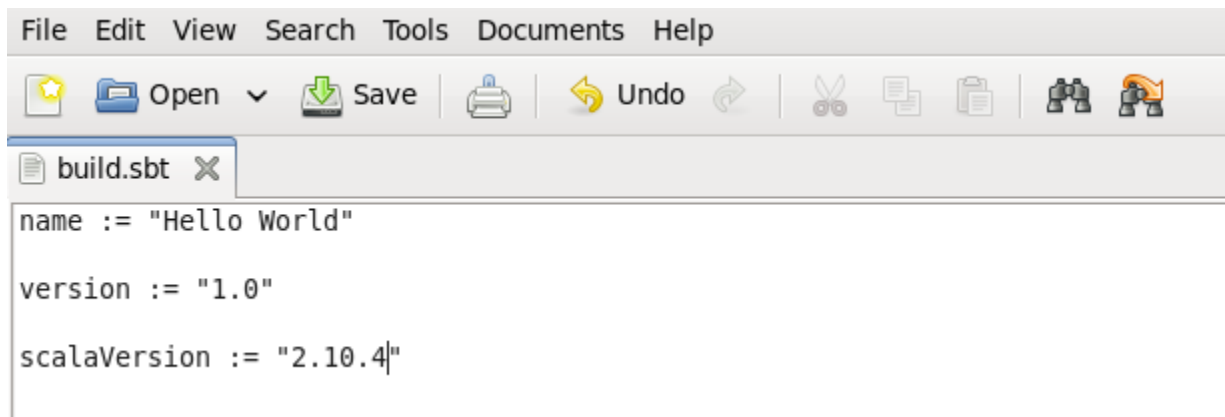
```
edureka@localhost:~/helloworld  
File Edit View Search Terminal Help  
[edureka@localhost ~]$ mkdir helloworld  
[edureka@localhost ~]$ cd helloworld/  
[edureka@localhost helloworld]$ mkdir -p src/main/scala  
[edureka@localhost helloworld]$ sudo gedit src/main/scala/hello.scala
```

Put the below code.

```
object Hello {  
  
  def main(args : Array[String]) = {  
    println("Hello World")  
  }  
}
```

Create a **build.sbt** file inside the **helloworld** directory.

**sudo gedit build.sbt**



Below is a very basic sbt file, you can add dependencies also to this file needed for your application.

## sbt package

```
edureka@localhost:~/helloworld
File Edit View Search Terminal Help
[edureka@localhost helloworld]$ sudo gedit build.sbt
[edureka@localhost helloworld]$ sbt package
[info] Loading global plugins from /home/edureka/.sbt/0.13/plugins
[info] Updating {file:/home/edureka/.sbt/0.13/plugins/}global-plugins...
[info] Resolving org.fusesource.jansi#jansi;1.4 ...
[info] downloading https://repo.scala-sbt.org/scalasbt/sbt-plugin-releases/com.typesafe.sbteclipse/sbteclipse-plugin/
r ...
[info] [SUCCESSFUL ] com.typesafe.sbteclipse#sbteclipse-plugin;4.0.0!sbteclipse-plugin.jar (20471ms)
[info] downloading https://repol.maven.org/maven2/org/scalaz/scalaz-core_2.10/7.1.0/scalaz-core_2.10-7.1.0.jar ...
[info] [SUCCESSFUL ] org.scalaz#scalaz-core_2.10;7.1.0!scalaz-core_2.10.jar(bundle) (18030ms)
[info] downloading https://repol.maven.org/maven2/org/scalaz/scalaz-effect_2.10/7.1.0/scalaz-effect_2.10-7.1.0.jar ..
[info] [SUCCESSFUL ] org.scalaz#scalaz-effect_2.10;7.1.0!scalaz-effect_2.10.jar(bundle) (2806ms)
[info] Done updating.
[info] Set current project to Hello World (in build file:/home/edureka/helloworld/)
[info] Updating {file:/home/edureka/helloworld/}helloworld...
[info] Resolving org.fusesource.jansi#jansi;1.4 ...
[info] Done updating.
[info] Compiling 1 Scala source to /home/edureka/helloworld/target/scala-2.10/classes...
[info] Packaging /home/edureka/helloworld/target/scala-2.10/hello-world_2.10-1.0.jar ...
[info] Done packaging.
[success] Total time: 1 s, completed Jan 4, 2016 4:40:08 PM
[edureka@localhost helloworld]$
```

This will create the jar file to run this application. Now run the below command to run your application.

```
spark-submit --class "Hello" --master local[2] target/scala-2.10/hello-world_2.10-1.0.jar
```

```
edureka@localhost:~/helloworld
File Edit View Search Terminal Help
[edureka@localhost helloworld]$ spark-submit --class "Hello" --master local[2] target/scala-2.10/hello-world_2.10-1.0.jar
Hello World
16/01/04 16:44:47 INFO util.Utils: Shutdown hook called
[edureka@localhost helloworld]$
```

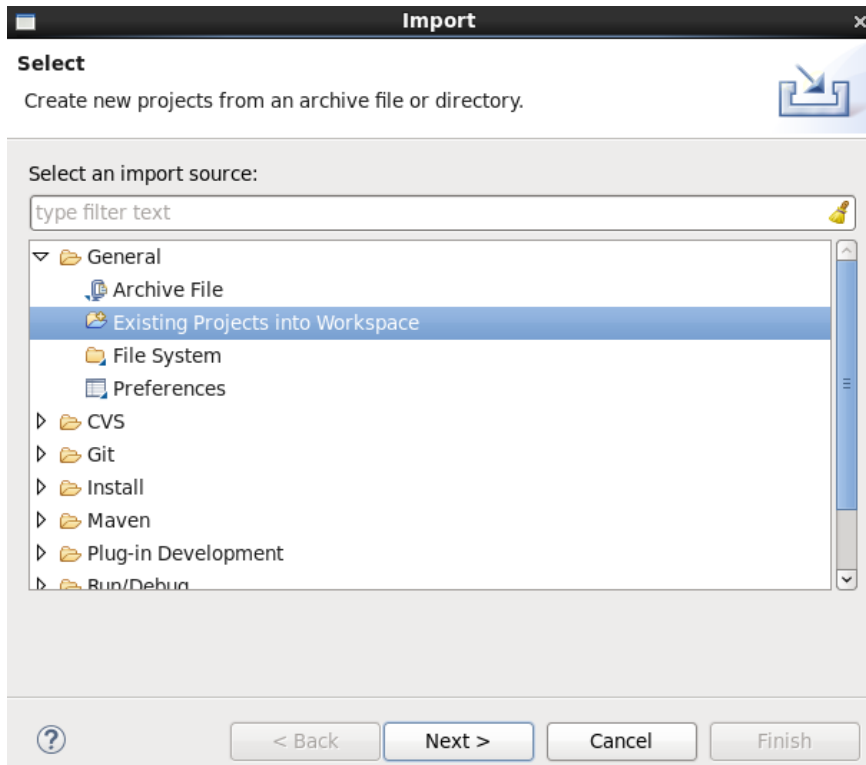
## sbt eclipse

Below command will make the project eclipse compatible and you will be able to import this project in eclipse and run it.

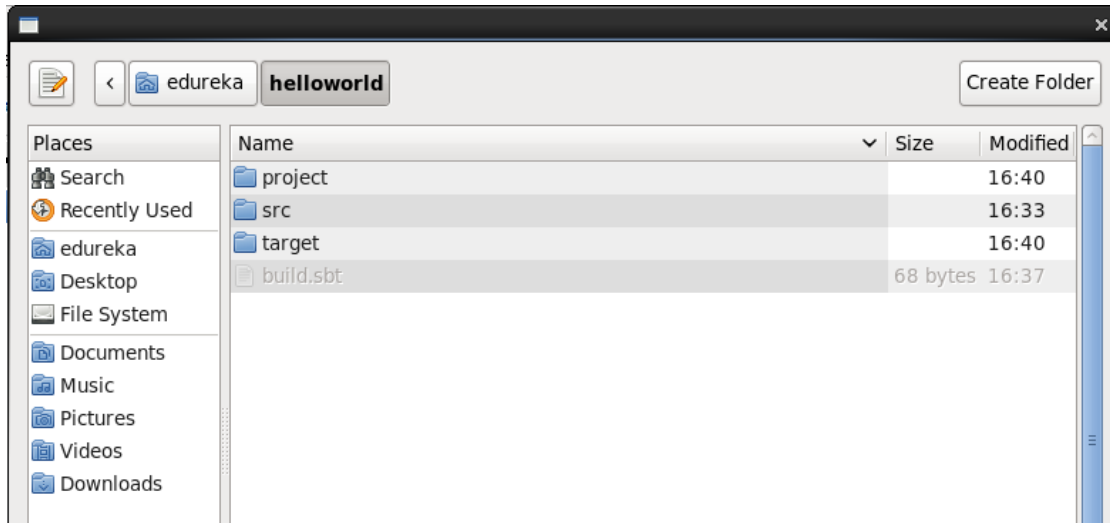
### sbt eclipse

```
edureka@localhost:~/helloworld
File Edit View Search Terminal Help
[edureka@localhost helloworld]$ sbt eclipse
[info] Loading global plugins from /home/edureka/.sbt/0.13/plugins
[info] Set current project to Hello World (in build file:/home/edureka/helloworld/)
[info] About to create Eclipse project files for your project(s).
[info] Successfully created Eclipse project files for project(s):
[info] Hello World
[edureka@localhost helloworld]$
```

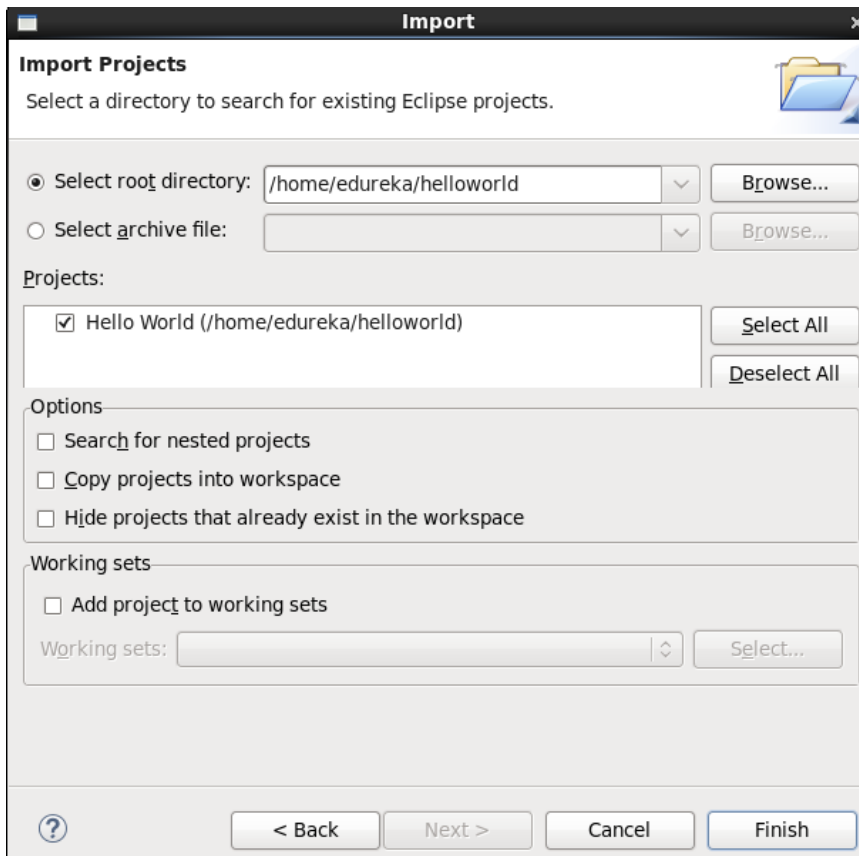
Go to **Scala IDE** , **File -> Import**



Select root Directory **helloworld**.



You will see, you can import this project, click on Finish.



Now run it as scala application.

```
File Edit Refactor Navigate Search Project Scala Run Window Help
Package Explorer
Hello World
  src/main/scala
    (default package)
      hello.scala
      hello.scala~
  Scala Library container [ 2.10.5 ]
  JRE System Library [jdk1.7.0_67]
  project
  src
  target
  build.sbt
  build.sbt~
  test

hello.scala
object Hello {
  def main(args : Array[String]) = {
    println("Hello World")
  }
}

Problems Tasks Console
<terminated> Hello$ [Scala Application] /usr/lib/jvm/jdk1.7.0_67/bin/java
Hello World
```

Just FYI, when you are setting **SparkConf**, always remember to set **Master** as done below.

```
hello.scala sample.scala
import org.apache.spark.SparkContext

object SampleApp {
  def main(args: Array[String]) {
    val txtFile = "/home/edureka/sample_app/src/main/scala/sample.scala"
    val conf = new SparkConf().setAppName("Sample Application").setMaster("local[2]")
    val sc = new SparkContext(conf)
    val txtFileLines = sc.textFile(txtFile, 2).cache()
    val numAs = txtFileLines.filter(line => line.contains("Hello")).count()
    println("Lines with val: %s".format(numAs))
  }
}
```

Then you can run your application.

```
Problems Tasks Console
<terminated> SampleApp$ [Scala Application] /usr/lib/jvm/jdk1.7.0_67/bin/java (Jan 4, 2016, 5:28:09 PM)
16/01/04 17:28:16 INFO DAGScheduler: ResultStage 0 (count at sample.scala:11) finished in 0.209 s
16/01/04 17:28:16 INFO DAGScheduler: Job 0 finished: count at sample.scala:11, took 0.351024 s
Lines with val: 1
16/01/04 17:28:16 INFO SparkContext: Invoking stop() from shutdown hook
```