How to run Appache Spark on Windows(in SBT console)

ANKITKUMAR KANERI
EMAIL: KANERI776@GMAIL.COM

Agenda

- Requirements
- Creating a Maven compliant folder structure
- Testing
- Useful Links

Requirements:

- Computer system with windows
- OS 7/8/10
- SBT console installed
- Scala(I used 2.10.6)
- Java(version used 1.8.0_181)

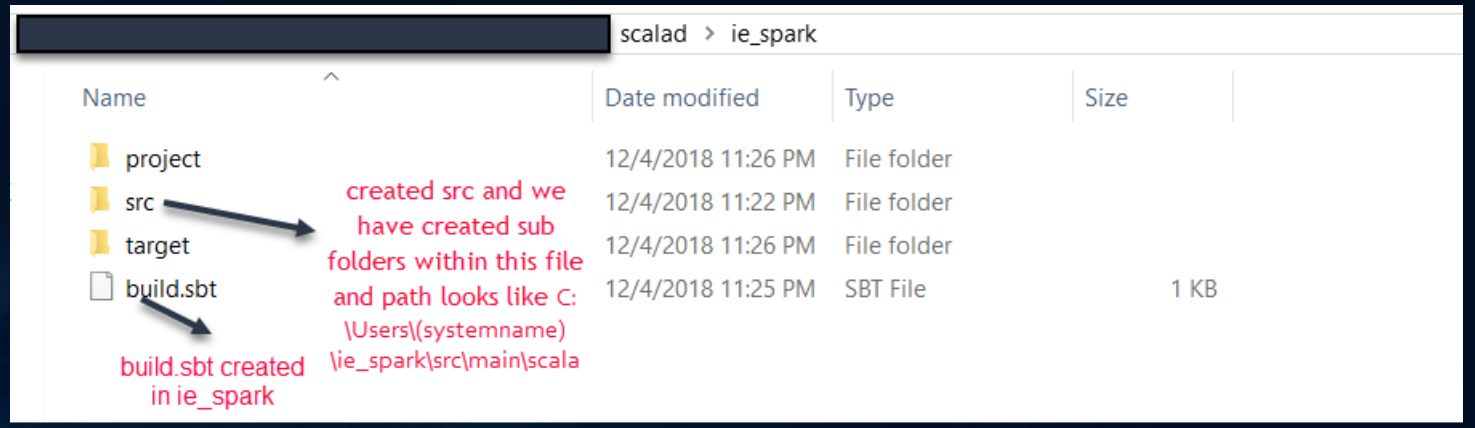
Creating a Maven compliant folder structure

Please follow the below steps:

1. Go to any directory in windows and create below folder structure

Ex: C:\User\(systemname)\

2. Create folder of any name(I created as ie_spark) and create sub-folder shown below strutucre



Ex: C:\Users\(systemname)\ie_spark\src\main\scala

3. Create file called build.sbt and below content in that file and place that file in the following path

Ex: C:\Users\(systemname)\ie_spark

- Content of build.sbt file is as follows:

```
name := "ie_spark_example"

version := "0.1.1"

scalaVersion := "2.10.6“ //(my scala version is 2.10.6)

libraryDependencies += "org.apache.spark" % "spark-core_2.10" %
"1.6.2“
```

- Basically we are specifying instructions for SBT to resolve the library dependencies that you mentioned in the build.sbt.

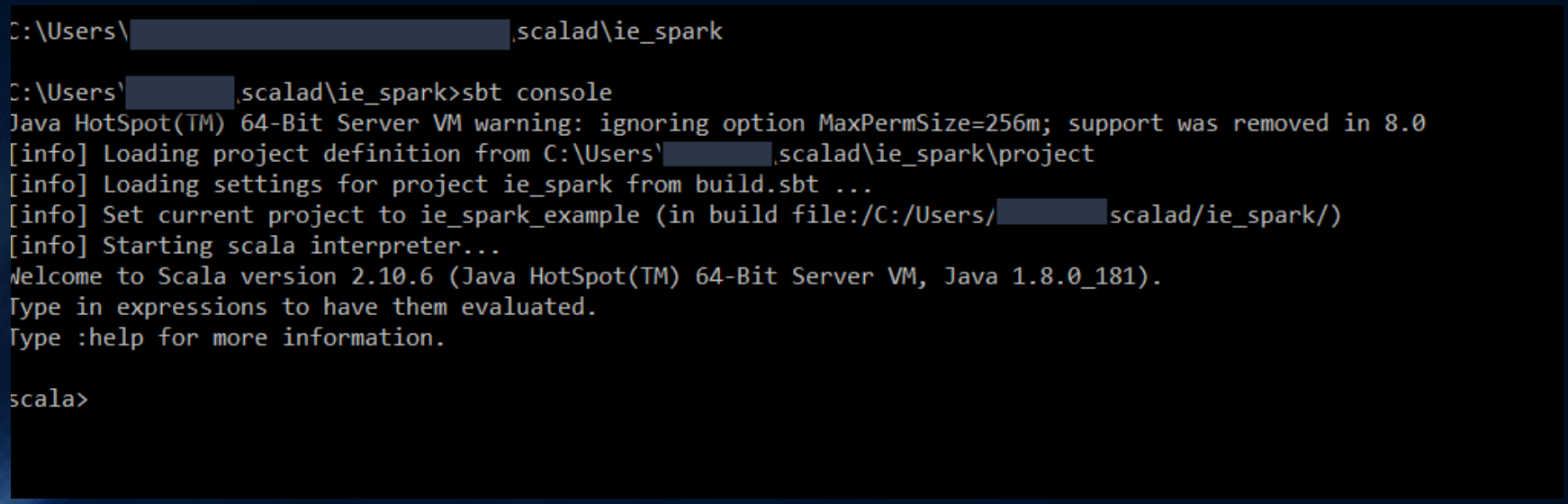
- When you “compile” scala code using SBT, it looks into build.sbt to see what dependencies are needed and it will go the specific repositories and will download them so that when your program needs them, they are there.
- So basically we are saying that your scala program that you’re gonna use will require Apache Spark Core libraries. So SBT will then go to maven repository to grab this library along with all of its dependencies.

- Now go to command prompt and perform the following steps:

1. Change directory to ie_spark . Use following command to change the directory.

cd C:\Users\(systemname)\scalad\ie_spark

2. Use command sbt console and enter



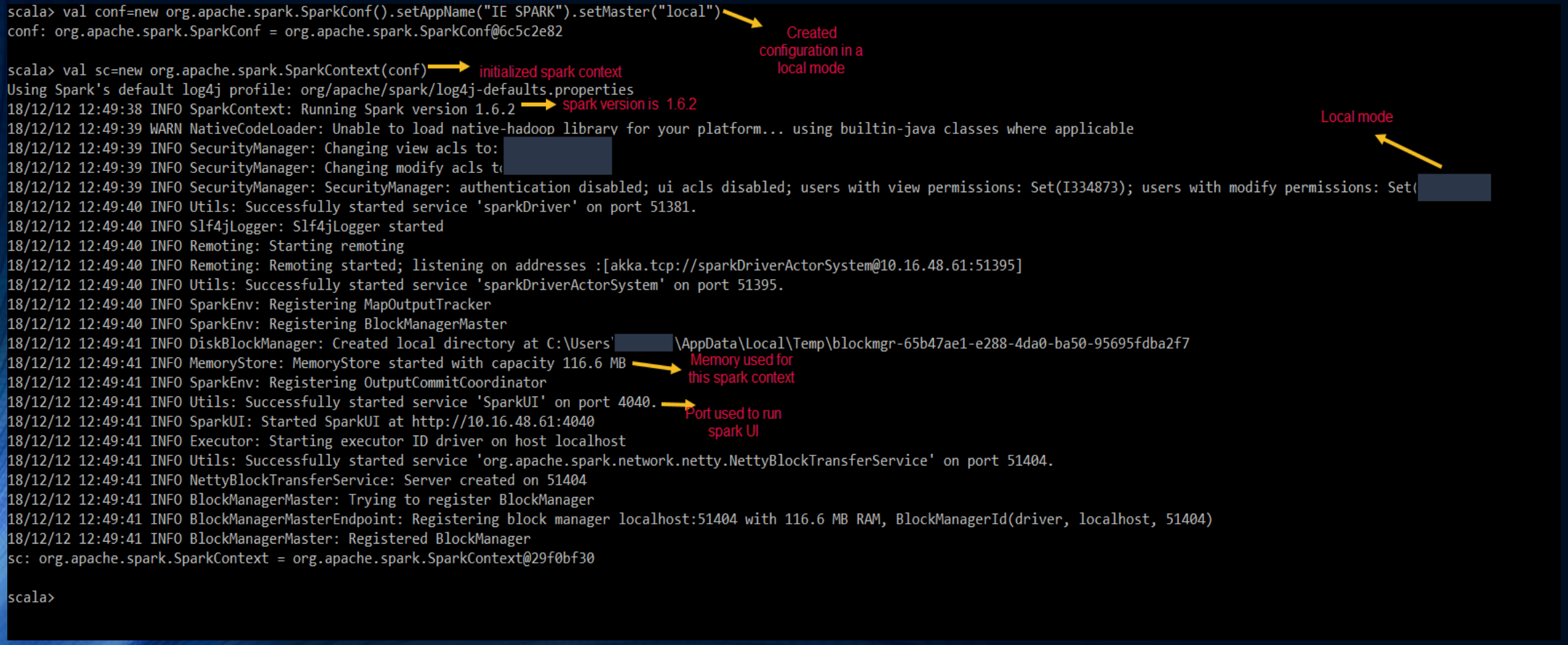
- You first create a Spark Conf object where you also specify the Spark mode. As you are executing it on your lonely laptop, so you will specify “local” mode here.

```
val conf=new org.apache.spark.SparkConf().setAppName("IE SPARK").setMaster("local")
```

and the next step is to create Spark Context (the entry point to use Spark Cluster and its APIs) using the conf object:

```
val sc=new org.apache.spark.SparkContext(conf)
```

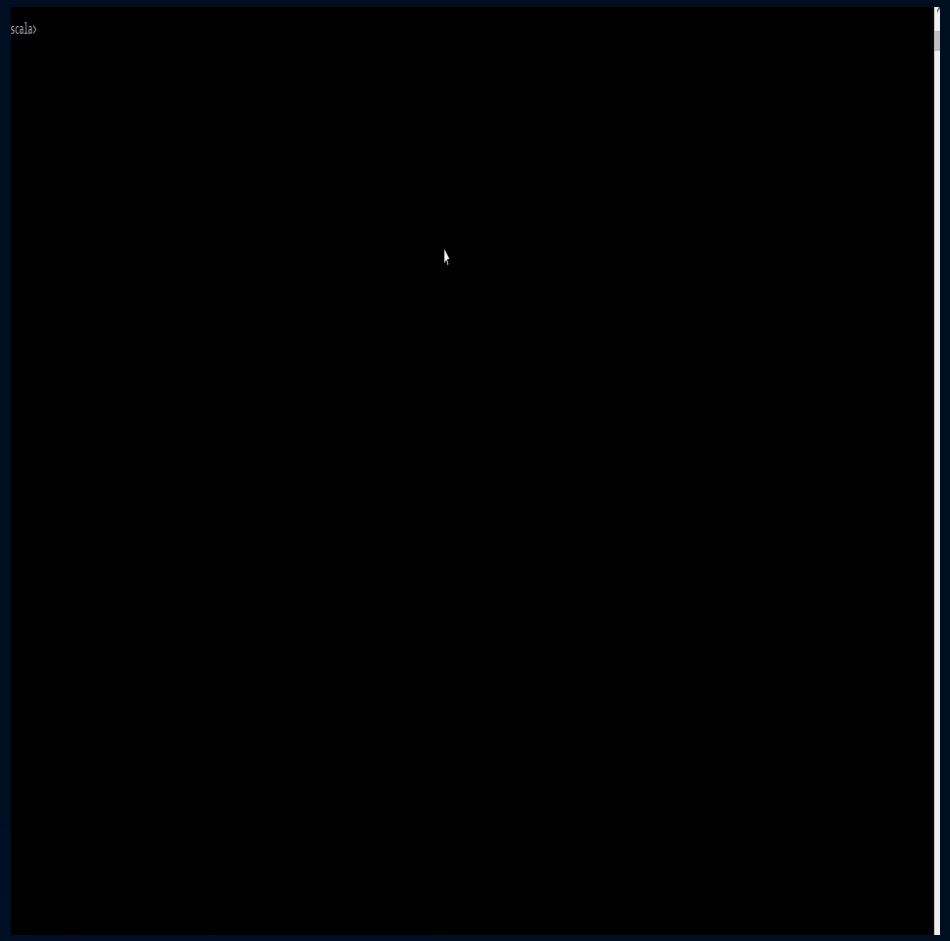
Figure shows the creating configuration and initializing spark context



Testing:

- Attached video for your reference, where it shows Spark Context is working as expected.
- You can also check by using the below code:

```
scala> val parallel = sc.parallelize(Array("Ankit", "Vinay")) // one input

parallel: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[27] at parallelize at <console>:24

scala> val par2 = sc.parallelize(Array("praveen", "Raju", "Ankit")) // sec input

par2: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[28] at parallelize at <console>:24

scala> parallel.union(par2).collect

res21: Array[String] = Array(Ankit, Vinay, praveen, Raju)//union output

scala> parallel.intersection(par2).collect

res21: Array[String] = Array(Ankit)//intersection output
```



Useful Links:

- Youtube Channel: https://www.youtube.com/channel/UCRzsq7k4-kT-h3TDUBQ82-w
- Apache Spark: https://spark.apache.org/documentation.html
- Hortonworks: https://hortonworks.com/apache/spark/
- GitHub: https://github.com/apache/spark
- Cloudera: https://www.cloudera.com/products/open-source/apache-hadoop/apache-spark.html
- Databrick: https://databricks.com/spark/about

Thank you