

How to run Apache 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)

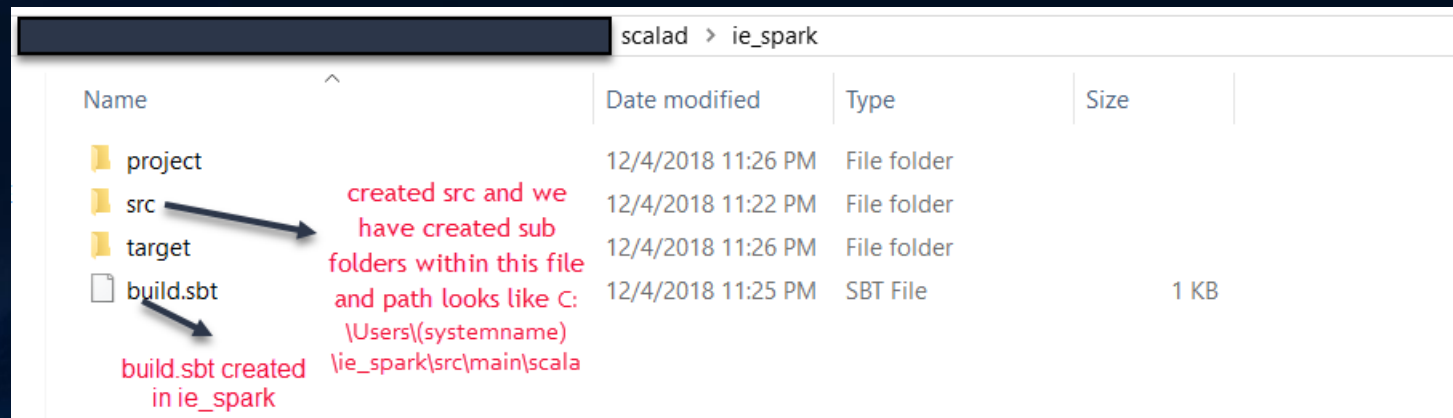
Creating a Maven compliant folder structure

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 strutucure



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 to 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

```
C:\Users\██████████\scalad\ie_spark

C:\Users\██████████\scalad\ie_spark>sbt console
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=256m; support was removed in 8.0
[info] Loading project definition from C:\Users\██████████\scalad\ie_spark\project
[info] Loading settings for project ie_spark from build.sbt ...
[info] Set current project to ie_spark_example (in build file:/C:/Users/██████████\scalad/ie_spark/)
[info] Starting scala interpreter...
Welcome to Scala version 2.10.6 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_181).
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

- 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

```
scala> val conf=new org.apache.spark.SparkConf().setAppName("IE SPARK").setMaster("local")
conf: org.apache.spark.SparkConf = org.apache.spark.SparkConf@6c5c2e82

scala> val sc=new org.apache.spark.SparkContext(conf)
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
18/12/12 12:49:38 INFO SparkContext: Running Spark version 1.6.2
18/12/12 12:49:39 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/12/12 12:49:39 INFO SecurityManager: Changing view acls to:
18/12/12 12:49:39 INFO SecurityManager: Changing modify acls to:
18/12/12 12:49:39 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(I334873); users with modify permissions: Set(
18/12/12 12:49:40 INFO Utils: Successfully started service 'sparkDriver' on port 51381.
18/12/12 12:49:40 INFO Slf4jLogger: Slf4jLogger started
18/12/12 12:49:40 INFO Remoting: Starting remoting
18/12/12 12:49:40 INFO Remoting: Remoting started; listening on addresses :[akka.tcp://sparkDriverActorSystem@10.16.48.61:51395]
18/12/12 12:49:40 INFO Utils: Successfully started service 'sparkDriverActorSystem' on port 51395.
18/12/12 12:49:40 INFO SparkEnv: Registering MapOutputTracker
18/12/12 12:49:40 INFO SparkEnv: Registering BlockManagerMaster
18/12/12 12:49:41 INFO DiskBlockManager: Created local directory at C:\Users\ \AppData\Local\Temp\blockmgr-65b47ae1-e288-4da0-ba50-95695fdb2f7
18/12/12 12:49:41 INFO MemoryStore: MemoryStore started with capacity 116.6 MB
18/12/12 12:49:41 INFO SparkEnv: Registering OutputCommitCoordinator
18/12/12 12:49:41 INFO Utils: Successfully started service 'SparkUI' on port 4040.
18/12/12 12:49:41 INFO SparkUI: Started SparkUI at http://10.16.48.61:4040
18/12/12 12:49:41 INFO Executor: Starting executor ID driver on host localhost
18/12/12 12:49:41 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 51404.
18/12/12 12:49:41 INFO NettyBlockTransferService: Server created on 51404
18/12/12 12:49:41 INFO BlockManagerMaster: Trying to register BlockManager
18/12/12 12:49:41 INFO BlockManagerMasterEndpoint: Registering block manager localhost:51404 with 116.6 MB RAM, BlockManagerId(driver, localhost, 51404)
18/12/12 12:49:41 INFO BlockManagerMaster: Registered BlockManager
sc: org.apache.spark.SparkContext = org.apache.spark.SparkContext@29f0bf30

scala>
```

Created configuration in a local mode

initialized spark context

spark version is 1.6.2

Local mode

Memory used for this spark context

Port used to run spark UI

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
```

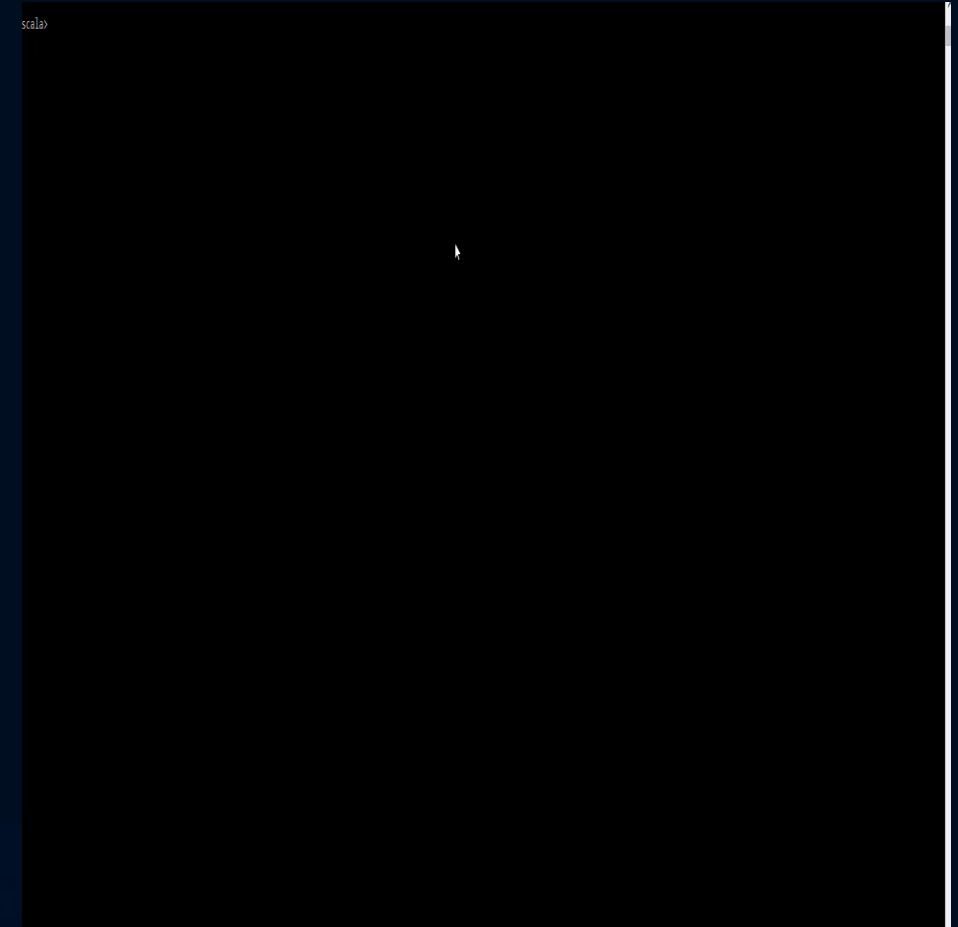
```
par2: org.apache.spark.rdd.RDD[String] =  
ParallelCollectionRDD[28] at parallelize at <console>:24
```

```
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/UCRzs7k4-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