

Java-Success.com

Prepare to fast-track, choose & go places with 800+ Java & Big Data Q&As with lots of code & diagrams.

[Home](#) [Why? ▾](#) [300+ Java FAQs ▾](#) [300+ Big Data FAQs ▾](#) [Courses ▾](#)

[👤 Membership ▾](#) [Your Career ▾](#)

[Home](#) › [bigdata-success.com](#) › [Tutorials - Big Data](#) › [TUT - Starting Spark & Scala](#) ›

09. Getting started with Spark & Hadoop – client mode on local & cluster mode on yarn

09. Getting started with Spark & Hadoop – client mode on local & cluster mode on yarn

 Posted on [November 15, 2018](#)

This extends [Setting up & getting started with Spark local mode with Sbt & Scala](#) and [Setting up & getting started with Hadoop on Mac](#).

The Spark job written in Scala will be reading the data from HDFS.

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

1. CSV data in HDFS

```
1 Maths, 85
2 English, 94
3 Science, 75
```

2. Make sure HDFS & YARN services are up

You can check if the services are up with:

```
1 jps -lm
```

If the services are down start them with:

```
1 $ start-all.sh
```

3. Load the data on to HDFS from the local file system

```
1
2 $ hdfs dfs -mkdir -p /data/marks
3 $ hdfs dfs -put ./test.csv /data/marks
4
```

4. Spark job to read from HDFS

This is an extension to the SimpleSpark.java covered in the previous Spark tutorials in the series [setting-up-getting-started-with-scala-sbt-spark-hadoop](#)

```
1
2 package com.scalaproject
3
4 import org.apache.spark.sql.SparkSession;
5 import org.apache.spark.sql.Dataset;
6
```

TUT - Starting with Python

TUT - Kafka

TUT - Pig

TUT - Apache Storm

TUT - Spark Scala on Zeppelin

TUT - Cloudera

TUT - Cloudera on Docker

TUT - File Formats

TUT - Spark on Docker

TUT - Flume

TUT - Hadoop (HDFS)

TUT - HBase (NoSQL)

TUT - Hive (SQL)

TUT - Hadoop & Spark

TUT - MapReduce

TUT - Spark and Scala

TUT - Spark & Java

TUT - PySpark on Databricks

TUT - Zookeeper

800+ Java Interview Q&As

300+ Core Java Q&As



300+ Enterprise Java Q&As



150+ Java Frameworks Q&As



120+ Companion Tech Q&As



Tutorials - Enterprise Java



```

7 object SimpleSpark {
8
9     def main(args: Array[String]): Unit = {
10         val spark = SparkSession.builder.appName("SimpleSpark")
11             .config("spark.master", "local")
12             .getOrCreate()
13
14         spark.read
15             .option("delimiter", ",")
16             .option("header", "false")
17             .csv("hdfs://localhost:9000/data/marks.txt")
18
19         spark.stop()
20
21     }
22 }
23

```

Within Eclipse you can “Run As” -> “Scala Application”

Output:

```

1 //.....
2 18/11/14 23:35:01 INFO CodeGenerator: Code generated
3 18/11/14 23:35:01 INFO Executor: Finished task 0.0
4 18/11/14 23:35:01 INFO TaskSetManager: Finished task 0.0
5 18/11/14 23:35:01 INFO TaskSchedulerImpl: Removed
6 18/11/14 23:35:01 INFO DAGScheduler: ResultStage 1
7 18/11/14 23:35:01 INFO DAGScheduler: Job 1 finished
8 +-----+
9 |      _c0|_c1|
10 +-----+
11 | Maths| 85|
12 |English| 94|
13 |Science| 75|
14 +-----+
15
16 18/11/14 23:35:01 INFO SparkUI: Stopped Spark web
17 18/11/14 23:35:01 INFO MapOutputTrackerMasterEndp
18 18/11/14 23:35:01 INFO MemoryStore: MemoryStore c
19 //.....
20

```

5. Package with sbt

```

1 ~/projects/sbt-tutorial]$ sbt package
2

```

6. Spark-submit local & client mode

Spark-submit to run the master on the local & in client mode:

```
1 ~/projects/sbt-tutorial]$ spark-submit --class com
2
```

You will get the same output:

```
1 2018-11-14 23:50:03 INFO CodeGenerator:54 - Code
2 2018-11-14 23:50:03 INFO Executor:54 - Finished t
3 2018-11-14 23:50:03 INFO TaskSetManager:54 - Fin
4 2018-11-14 23:50:03 INFO TaskSchedulerImpl:54 - F
5 2018-11-14 23:50:03 INFO DAGScheduler:54 - Result
6 2018-11-14 23:50:03 INFO DAGScheduler:54 - Job 1
7 +-----+
8 |      _c0|_c1|
9 +-----+
10 | Maths| 85|
11 |English| 94|
12 |Science| 75|
13 +-----+
14
15 2018-11-14 23:50:03 INFO AbstractConnector:318 -
16 2018-11-14 23:50:03 INFO SparkUI:54 - Stopped Sp
17 2018-11-14 23:50:03 INFO MapOutputTrackerMasterE
18 2018-11-14 23:50:03 INFO MemoryStore:54 - MemoryS
19 2018-11-14 23:50:03 INFO BlockManager:54 - BlockM
20
```

6. Spark-submit yarn & cluster mode

Spark-submit to run the master on the yarn & in cluster mode:

Firstly, in the source code remove
".config("spark.master", "local")". The source code
should look like:

```
1 package com.scalaproject
```

```
2
3 import org.apache.spark.sql.SparkSession;
4 import org.apache.spark.sql.Dataset;
5
6 object SimpleSpark {
7
8     def main(args: Array[String]): Unit = {
9         val spark = SparkSession.builder.appName("SimpleSpark")
10             .getOrCreate()
11
12         spark.read
13             .option("delimiter", ",")
14             .option("header", "false")
15             .csv("hdfs://localhost:9000/data/marks.csv")
16
17         spark.stop()
18     }
19 }
20
21
```

Package it:

```
1 ~/projects/sbt-tutorial]$ sbt package
2
```

Set the following environment variables (e.g. `~/.bash_profile` or `~/.bashrc`):

```
1 export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
2 export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
```

Run it on yarn in cluster mode:

```
1 ~/projects/sbt-tutorial]$ spark-submit --class com
2
```

Check <http://localhost:8088/cluster> for the output.
Click on the id and then the logs.

Output:

```
1 +-----+---+
2 |      _c0|_c1|
3 +-----+---+
4 |   Maths| 85|
5 |English| 94|
6 |Science| 75|
7 +-----+---+
8
```

You can practice the Spark code that are covered in the [Spark tutorials on Apache Zeppelin on Docker](#).

◀ 08. Setting up & getting started with Hadoop on Mac

10. Setting up & getting started with Hive ▶

Disclaimer

The contents in this Java-Success are copyrighted and from EmpoweringTech pty Ltd. The EmpoweringTech pty Ltd has the right to correct or enhance the current content without any prior notice. These are general advice only, and one needs to take his/her own circumstances into consideration. The EmpoweringTech pty Ltd will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. Links to external sites do not imply endorsement of the linked-to sites. [Privacy Policy](#).