Hadoop: Hands-On:
Spark WordCount

SPARK WORDCOUNT

Start Spark Session:

```
$spark-shell
```

Step 1: Import Spark packages

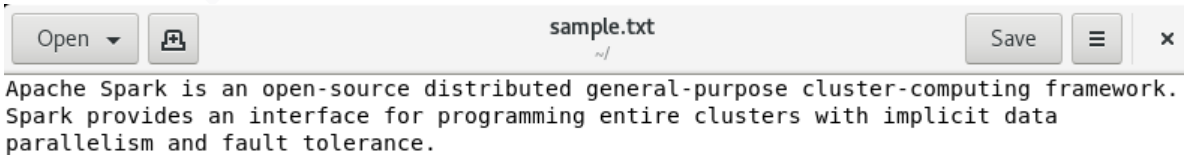
```
valsparkDummy = spark import
  sparkDummy.implicits._
```

```
scala> val sparkDummy = spark
sparkDummy: org.apache.spark.sql.SparkSession = org.apache
.spark.sql.SparkSession@11a0c708

scala> import sparkDummy.implicits._
import sparkDummy.implicits._
```

Step 2: Create a text file named ‘sample.txt’ in your system with the following content:

```
Apache Spark is an open-source distributed general-purpose cluster-computing framework. Spark
  provides an interface for programming entire clusters with implicit data parallelism and fault
  tolerance.
```



Step 3: Import the dataset

```
val text = (spark.read.text("sample.txt").as[String])
```

```
scala> val text = (spark.read.text("sample.txt").as[String])
text: org.apache.spark.sql.Dataset[String] = [value: string]
```

Step 4: Split the data using the 'flatMap' function

```
val counts = (text.flatMap(line =>line.split("\\s+"))
.groupByKey(_.toLowerCase)
.count)
```

```
scala> val counts = (text.flatMap(line => line.split("\\s+"))
     |   .groupByKey(_.toLowerCase)
     |     .count)
counts: org.apache.spark.sql.Dataset[(String, Long)] = [value: string
, count(1): bigint]
```


Step 5: Count the appearance of a particular value

```
counts.orderBy($"count(1)" desc).show
```

```
scala> counts.orderBy($"count(1)" desc).show
warning: there was one feature warning; re-run with -feature
 for details
+-----------------+--------+

|            value|count(1)|
+-----------------+--------+
|            spark|       2|
|               an|       2|
|  general-purpose|       1|
|              for|       1|
|         provides|       1|
|      open-source|       1|
```

