Dataframe

Working Directory

dataframe

Depends On

None

Step 1 : Login to Sandbox

Follow instructions for your environment.

Step 2 : Stage JSON data in to HDFS

For this lab we are going to use clickstream data in JSON format.
Follow the steps to

```
#  be in the working directory
$   cd  hadoop-spark/dataframe

#  create an HDFS directory
$   hdfs   dfs -mkdir  -p clickstream/in-json

#  copy sample file
$   hdfs   dfs   -put   ../data/click-stream/clickstream.json    clickstream/in-json
```

Let's generate some more clickstream JSON data and stage into HDFS.

```
#  generate data
$  python ../data/clickstream/gen-clickstream-json.py

#  copy data to HDFS
$  hdfs dfs -put   *.json   clickstream/in-json/

#  verify data in HDFS
$  hdfs   dfs  -ls   clickstream/in-json/
```

Step 2 : Start Spark shell

```
$    spark-shell
```

And set the log level to WARN

```
scala>
        sc.setLogLevel("WARN")
```

Step 3 : Load JSON into Dataframe

Issue the following commands in Spark-shell

```
val clickstream = sqlContext.read.json("/user/root/clickstream/in-json/clickstream.json")

// inspect the schema
clickstream.printSchema

// inspect data
clickstream.show
```

Step 4 : Methods available on Dataframe

Use tab completion to explore methods available

```
clickstream.  HIT THE TAB KEY
```

Step 5 : Querying Dataframes

Find records that have cost > 100

```
val over100 = clickstream.filter(clickstream("cost") > 100)
over100.show

// count the records
over100.count
```

Count records where action = 'click'

```
clickstream.filter(clickstream("action") === "clicked").count
```

Count the number of visits from each domain

```
clickstream.groupBy("domain").count.show
```

Step 6 : Querying Using SQL

```
//  register the data frame as a temporary table
clickstream.registerTempTable("clickstream")

// try sql queries
sqlContext.sql("select * from clickstream").show
```

```
    // find all facebook traffic
    sqlContext.sql("select * from clickstream where domain = 'facebook.com'").show

    // count traffic per domain from highest to lowest
    sqlContext.sql("select domain, count(*) as total from clickstream  group by domain order by total
desc").show
```

Step 7 : Load all JSON data

Let's load all JSON files in `clickstream/in-json` directory

```
    val clicks = sqlContext.read.json("/user/root/clickstream/in-json/")
    clicks.count

    clicks.groupBy("domain").count.show

    // repeat instructions from step 6
```