

```
student@student:~$ echo $SHELL
/bin/bash
student@student:~$ echo $SHELL
/bin/bash
student@student:~$ nano ~/.bashrc
student@student:~$ nano ~/.bashrc
student@student:~$ source ~/.bashrc
student@student:~$ echo $PATH
/home/student/DSBDA/spark-3.5.1-bin-hadoop3/bin:/home/student/spark-3.5.1-bin-hadoop3/bin:/home/student/DSBDA/me/student/DSBDA/spark-3.5.1-bin-hadoop3/bin:/home/student/spark-3.5.1-bin-hadoop3/bin:/home/student/DSBDA/spa
tudent/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/sn
k-3.5.1-bin-hadoop3/bin:/opt/spark/spark-3.5.1-bin-hadoop3/bin:/opt/spark/spark-3.5.1-bin-hadoop3/bin:/opt/spa
student@student:~$ cd DSBDAL
student@student:~/DSBDAL$ spark-shell< WebLog_Processing.scala
25/04/07 15:30:04 WARN Utils: Your hostname, student resolves to a loopback address: 127.0.1.1; using 10.11.5.
25/04/07 15:30:04 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/04/07 15:30:13 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using built
Spark context Web UI available at http://10.11.5.57:4040
Spark context available as 'sc' (master = local[*], app id = local-1744020016589).
Spark session available as 'spark'.
Welcome to

      _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_
     /_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_\
    /_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_\_
   /_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/__\
  /_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_\_
 /_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/__\
/_/_/_/_/_/_/_/_/_/_/_/_/_/_/__\

version 3.5.1
```

```
scala> //let's look at some of the data

scala> base_df.show(3,false)
+-----+
|value|
+-----+
|IP,Time,URL,Staus|
|10.128.2.1,[29/Nov/2017:06:58:55,GET /login.php HTTP/1.1,200|
|10.128.2.1,[29/Nov/2017:06:59:02,POST /process.php HTTP/1.1,302|
+-----+
only showing top 3 rows

scala>
```

```
scala> /*
|       Parsing the log file
|     */
| val parsed_df = base_df.select(regex_extract($"value","""^([\s|,)+"""",1).alias("host"),
|   regex_extract($"value","""^.*\[(\d\d/\d{3}/\d{4}:\d{2}:\d{2}):(\d{2})"""",1).as("timestamp"),
|   regex_extract($"value","""^.*\w+\s+([\s|,)+\s+HTTP.*"""",1).as("path"),
|   regex_extract($"value","""^.*,([\s|,)+$"""",1).cast("int").alias("status"))
parsed_df: org.apache.spark.sql.DataFrame = [host: string, timestamp: string ... 2 more fields]

scala> parsed_df.show(5,false)
+-----+-----+-----+-----+
| host      | timestamp          | path              | status |
+-----+-----+-----+-----+
| IP        |                    |                   | NULL   |
| 10.128.2.1 | 29/Nov/2017:06:58:55 | /login.php       | 200    |
| 10.128.2.1 | 29/Nov/2017:06:59:02 | /process.php     | 302    |
| 10.128.2.1 | 29/Nov/2017:06:59:03 | /home.php        | 200    |
| 10.131.2.1 | 29/Nov/2017:06:59:04 | /js/vendor/moment.min.js | 200    |
+-----+-----+-----+-----+
only showing top 5 rows
```

```

| logs_df.describe("status").show()
+-----+-----+
|summary|      status|
+-----+-----+
|  count|      15789|
|   mean|230.19469250744189|
| stddev| 50.05853522906924|
|    min|         200|
|    max|         404|
+-----+-----+

scala> /*
|      HTTP status analysis
|      */
| logs_df.groupBy("status").count().sort("status").show()
+-----+-----+
|status|count|
+-----+-----+
|  200|11330|
|  206|   52|
|  302| 3498|
|  304|  658|
|  404|   251|
+-----+-----+

```

```

scala> not_found_df.withColumn("day", dayofyear($"time")).withColumn("year", year($"time")).groupBy("year", "day").count().show(10)
+-----+-----+
|day|year|count|
+-----+-----+
|312|2017|    8|
|313|2017|   10|
|314|2017|    6|
|315|2017|   12|
|316|2017|    6|
|317|2017|   10|
|318|2017|   18|
|319|2017|    8|
|320|2017|   10|
|321|2017|    5|
+-----+-----+
only showing top 10 rows

scala>
|
| /* To run the program
| scala> :load WebLog_Processing.scala
| */
|
scala> :quit

```