

PROJET LEE

PRÉSENTÉ PAR : DIA Amadou

ENCADRÉ PAR : Monsieur BOUDES Pierre

Sujet :

Le projet consiste à faire un programme spark avec scala.

Possédant un fichier csv nommé **Crimes_data_chicago_2017.csv** (qui se trouve dans la racine du dossier du projet) contenant des informations sur la liste des crimes qui ont été commis dans différentes villes au État Unis, le but est de déterminer le nombre de crime par mois et par type en ordre croissant par mois et décroissant par nombre de crime.

Pour le faire, nous allons procédé de 2 manières différentes. La première méthode consistera utilisé un **map-reduce** et la deuxième utilisera un **DataFrame**.

Lien du projet vers github :

Voici le lien pour accéder au projet.

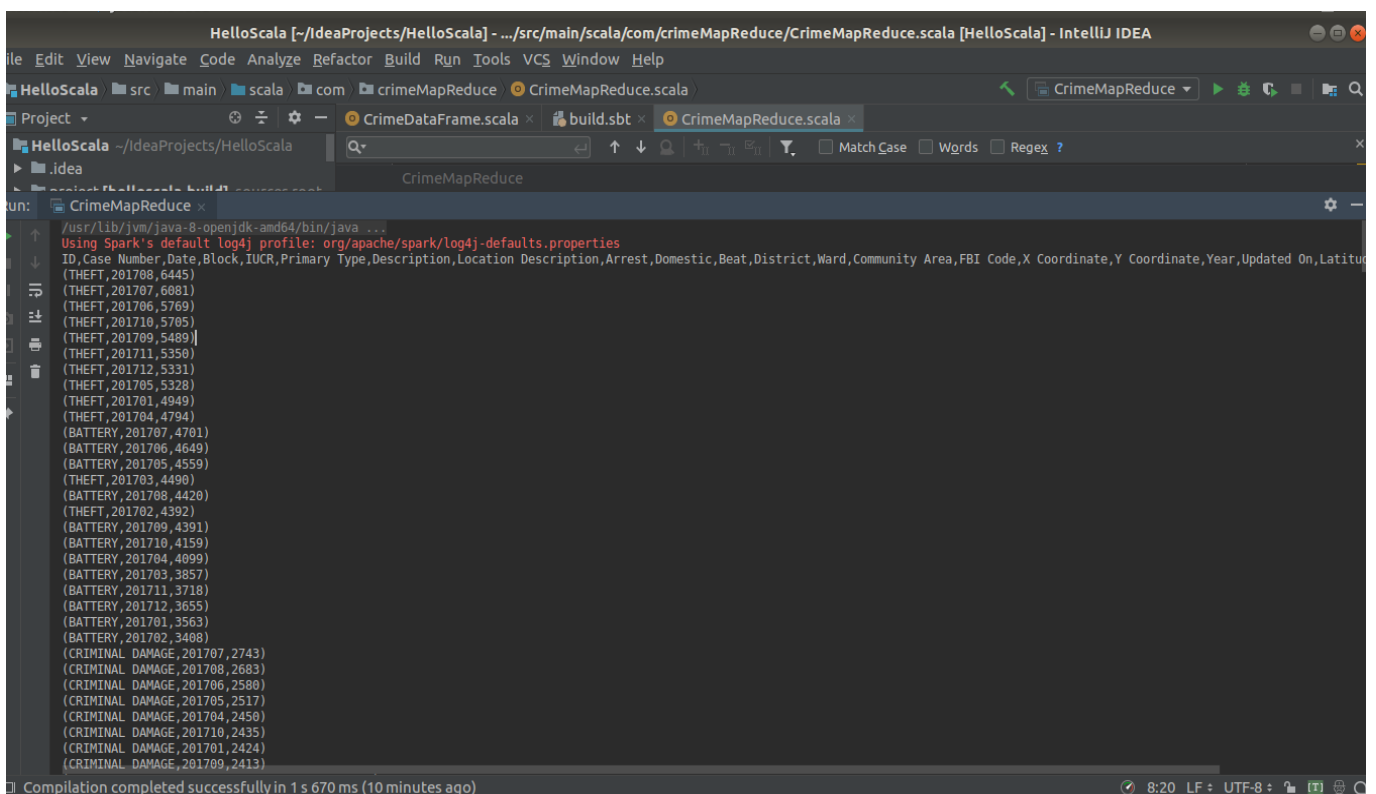
« <https://github.com/AmadouDIA/ProjetSparkScalaLEE> »

Méthode avec map-reduce :

Le fichier est nommé **<CrimeMapReduce.scala>** contenu dans le package **<crimeMapReduce>**.

Le code est bien documenté pas à pas.

Voici le résultats des quelques première lignes affichés après exécution de la class.



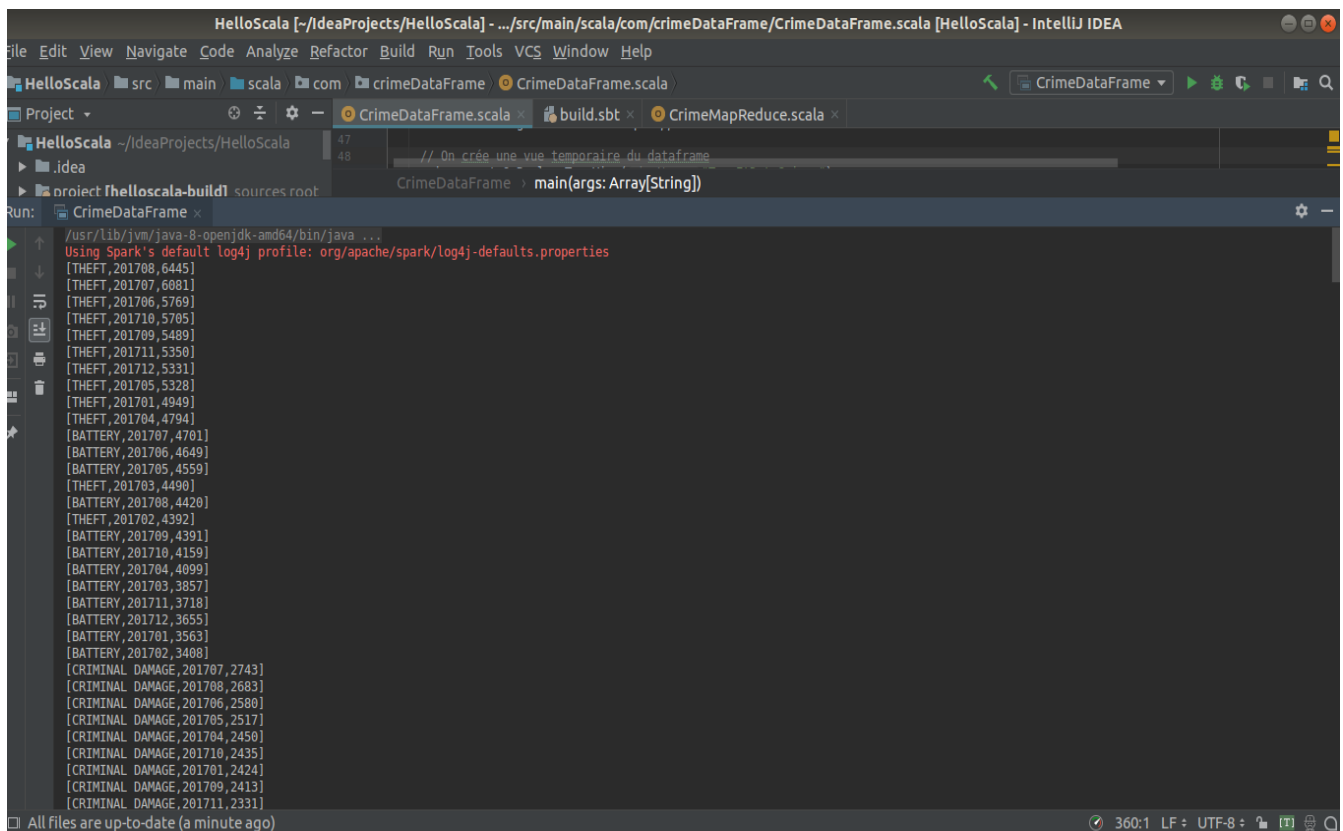
```
run: CrimeMapReduce x
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
ID, Case Number, Date, Block, IUCR, Primary Type, Description, Location Description, Arrest, Domestic, Beat, District, Ward, Community Area, FBI Code, X Coordinate, Y Coordinate, Year, Updated On, Latitude
(THEFT, 201708, 6445)
(THEFT, 201707, 6081)
(THEFT, 201706, 5769)
(THEFT, 201710, 5705)
(THEFT, 201709, 5489)
(THEFT, 201711, 5350)
(THEFT, 201712, 5331)
(THEFT, 201705, 5328)
(THEFT, 201701, 4949)
(THEFT, 201704, 4794)
(BATTERY, 201707, 4701)
(BATTERY, 201706, 4649)
(BATTERY, 201705, 4559)
(THEFT, 201703, 4490)
(BATTERY, 201708, 4420)
(THEFT, 201702, 4392)
(BATTERY, 201709, 4391)
(BATTERY, 201710, 4159)
(BATTERY, 201704, 4099)
(BATTERY, 201703, 3857)
(BATTERY, 201711, 3718)
(BATTERY, 201712, 3655)
(BATTERY, 201701, 3563)
(BATTERY, 201702, 3408)
(CRIMINAL DAMAGE, 201707, 2743)
(CRIMINAL DAMAGE, 201708, 2683)
(CRIMINAL DAMAGE, 201706, 2580)
(CRIMINAL DAMAGE, 201705, 2517)
(CRIMINAL DAMAGE, 201704, 2450)
(CRIMINAL DAMAGE, 201710, 2435)
(CRIMINAL DAMAGE, 201701, 2424)
(CRIMINAL DAMAGE, 201709, 2413)
Compilation completed successfully in 1 s 670 ms (10 minutes ago)
```

Méthode avec DataFrame :

Le fichier est nommé `<CrimeDataFrame.scala>` contenu dans le package `<crimeDataFrame>`.

Le code est bien documenté pas à pas.

Voici le résultats des quelques première lignes affichés après exécution de la class.



```
Run: CrimeDataFrame x
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
[THEFT,201708,6445]
[THEFT,201707,6081]
[THEFT,201706,5769]
[THEFT,201710,5705]
[THEFT,201709,5489]
[THEFT,201711,5350]
[THEFT,201712,5331]
[THEFT,201705,5328]
[THEFT,201701,4949]
[THEFT,201704,4794]
[BATTERY,201707,4701]
[BATTERY,201706,4649]
[BATTERY,201705,4559]
[THEFT,201703,4490]
[BATTERY,201708,4420]
[THEFT,201702,4392]
[BATTERY,201709,4391]
[BATTERY,201710,4159]
[BATTERY,201704,4099]
[BATTERY,201703,3857]
[BATTERY,201711,3718]
[BATTERY,201712,3655]
[BATTERY,201701,3563]
[BATTERY,201702,3408]
[CRIMINAL_DAMAGE,201707,2743]
[CRIMINAL_DAMAGE,201708,2683]
[CRIMINAL_DAMAGE,201706,2580]
[CRIMINAL_DAMAGE,201705,2517]
[CRIMINAL_DAMAGE,201704,2450]
[CRIMINAL_DAMAGE,201710,2435]
[CRIMINAL_DAMAGE,201701,2424]
[CRIMINAL_DAMAGE,201709,2413]
[CRIMINAL_DAMAGE,201711,2331]
All files are up-to-date (a minute ago)
```

Instructions pour compiler :

Si vous avez installé IntelliJ, allez sur **File** → **Open** ensuite vous sélectionnez le dossier du projet puis vous cliquez sur **OK**. Le projet sera alors importé.

Maintenant pour compiler la méthode map-reduce, vous ouvrez le fichier `src/main/scala/com/crimeMapReduce/CrimeMapReduce.scala` en faisant un double clic .

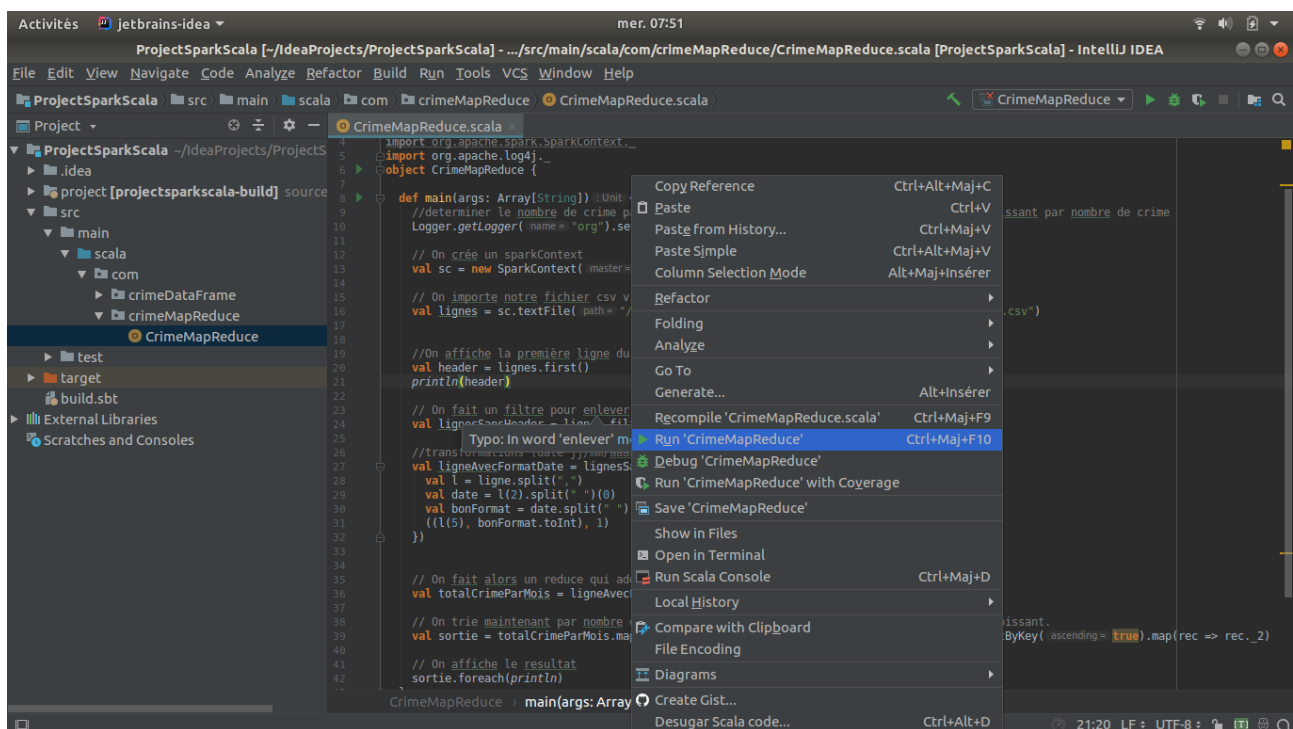
Il faudra ensuite modifier le répertoire contenant le fichier csv. Vous mettez le bon répertoire où se situe le fichier.

Voici la ligne en image qu'il faut modifier :

```
14
15 // On importe notre fichier csv via notre sparkContext
16 val lignes = sc.textFile( path = "/home/dia/Documents/hadoop_spark/Crimes_data_chicago_2017.csv")
17
```

Ensuite vous faite un clique doit puis vous cliquez sur **Run CrimeMapReduce**.

Voir image ci après :



Pour la méthode **DataFrame** vous allez ouvrir le fichier **src/main/scala/com/crimeMapReduce/CrimeMapReduce.scala** ensuite vous faite le même procédé qu'avant sauf qu'il faut modifier en plus du répertoire ou se situe le fichier csv mais aussi du répertoire de configuration du SparkSession. Pour ce dernier vous mettez n'importe quel répertoire de votre choix.

Voici les ligne qu'il faut modifier :

```
30 // On importe le fichier csv contenant les données avec la session spark
31 val ligne = spark.sparkContext.textFile( path = "/home/dia/Documents/hadoop_spark/Crimes_data_chicago_2017.csv")
```

```
33 //On importe le fichier csv contenant les données avec la session spark
34 val ligne = spark.sparkContext.textFile( path = "/home/dia/Documents/hadoop_spark/Crimes_data_chicago_2017.csv")
35
```

Ensuite vous faite un clique doit puis vous cliquez sur **Run CrimeDataFrame**.

Voir image ci-après :

