

# TP Apache Spark Dataframes

v1.1

## Table of Contents

### Intro

Preparation de l'environnement

**Spark notebook**

### Ressources

## Intro

Pour ce TP vous allez utiliser **spark-notebook** qui a été précédemment installé dans votre VM.

## Preparation de l'environnement

Dans nos exemples l'adresse IP de la VM est **192.168.56.101**. *N'oubliez pas de remplacer partout dans les exemples cette adresse par l'adresse affichée dans la console VirtualBox:*

Pour accéder à l'interface spark-notebook il faudra rediriger également le port 9001 (en plus des ports habituels).



```
[andrei@desktop ~]$ ssh -L 9080:127.0.0.1:8080 \
                        -L 8081:127.0.0.1:8081 \
                        -L 8082:127.0.0.1:8082 \
                        -L 4040:127.0.0.1:4040 \
                        -L 9001:127.0.0.1:9001 \
                        bigdata@192.168.56.101
bigdata@192.168.56.101's password:
Last login: Sun Jan  4 14:53:32 2015 from pc12.home
```

BASH

## Spark notebook

- (1) lancer le spark-notebook (depuis le bon repertoire !)

```
[bigdata@bigdata ~]$ cd /home/bigdata/spark-notebook-0.7.0/ 1
[bigdata@bigdata spark-notebook-0.7.0]$ bin/spark-notebook 2
Play server process ID is 4582
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/bigdata/spark-notebook-
0.7.0/lib/ch.qos.logback.logback-classic-
1.1.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/bigdata/spark-notebook-
0.7.0/lib/org.slf4j.slf4j-log4j12-1.7.16.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type
[ch.qos.logback.classic.util.ContextSelectorStaticBinder]
[info] play - Application started (Prod)
[info] play - Listening for HTTP on /0:0:0:0:0:0:0:0:9001 3
```

- (2) suivez les exercices sur l'interface du spark-notebook

([http://localhost:9001/notebooks/telecom2016/TP6\\_Dataframes.snb](http://localhost:9001/notebooks/telecom2016/TP6_Dataframes.snb)) Vous pouvez editer une cellules en cliquant. Vous pouvez executer le code d'une cellule via le menu/Cell/Run ou via `Shift + Enter`. Pour avoir la completion automatique du code vous pouvez utiliser `TAB`. Si le notebook ne reponds pas vous pouvez redemarer le kernel spark via Kernel/Restart ou redemarer le notebook (`Ctrl + C` dans le terminal puis relancer `bin/spark-notebook`).

**SPARK NOTEBOOK** TP6 Dataframes (autosaved)

File Edit View Insert Cell Kernel Help | Scala [2.10.6] Spark [2.0.2] Hadoop [2.7.2] (Hive v)

Cell Toolbar: None

### Description des bases de données annuelles des accidents corporels de la circulation routière

#### Intro

Pour les exercices suivant on va utiliser une base de donnees publique du gouvernement qui contient des donnees sur les accidents de la route (<https://www.data.gouv.fr/fr/datasets/base-de-donnees-accidents-corporels-de-la-circulation/>)

Nous allons utiliser des dataframes et datasets pour manipuler ces donnees.

#### Environnement

Vous pouvez faire ces exercices dans le spark shell.

#### Description des donnees

Pour chaque accident corporel (soit un accident survenu sur une voie ouverte à la circulation publique, impliquant au moins un véhicule et ayant fait au moins une victime ayant nécessité des soins), des saisies d'information décrivant l'accident sont effectuées par l'unité des forces de l'ordre (police, gendarmerie, etc.) qui est intervenue sur le lieu de l'accident.

Ces saisies sont rassemblées dans une fiche intitulée bulletin d'analyse des accidents corporels (BAAC). Cela comprend des informations de localisation de l'accident, telles que renseignées ainsi que des informations concernant les caractéristiques de l'accident et son lieu, les véhicules impliqués et leurs victimes.

Les bases de données de 2005 à 2015 sont désormais annuelles et composées de 4 fichiers (Caractéristiques – Lieux – Véhicules – Usagers) au format csv.

La description des différents fichiers se trouve ici: [https://www.data.gouv.fr/fr/ressources/base-de-donnees-accidents-corporels-de-la-circulation/20160926-173908/Description\\_des\\_bases\\_de\\_donnees\\_ONISR\\_-Annees\\_2005\\_a\\_2015.pdf](https://www.data.gouv.fr/fr/ressources/base-de-donnees-accidents-corporels-de-la-circulation/20160926-173908/Description_des_bases_de_donnees_ONISR_-Annees_2005_a_2015.pdf)

#### Documentation

Documentation dataframes: <http://spark.apache.org/docs/latest/sql-programming-guide.html#creating-dataframes> Documentation API scala: <http://spark.apache.org/docs/latest/api/scala/index.html#org.apache.spark.sql.Dataset>

**Dans les exercices suivants remplacez les TODO par le code necessaire**

#### Lecture et exploration des donnees CSV

```
import org.apache.spark.sql.{SparkSession, Dataset}

def TODO = ??? // ceci est juste un marquer d'une valeur que vous devez remplacer dans les lignes qui suivent

// on cree une session Spark avec un nom particulier pour la retrouver plus facilement dans le SparkUI
val mySession = SparkSession
  .builder()
  .config(new SparkConf()
    .setAppName(TODO))//rajouter comme parametre le nom de votre application
  .getOrCreate()
```



Identifiants de connexion:

- utilisateur: ***bigdata***
- password: ***bigdatafuret***

## Ressources

Spark Programming Guide (<https://spark.apache.org/docs/latest/sql-programming-guide.html>)

SQL Dataframes Tutorial (<http://spark.apache.org/docs/latest/sql-programming-guide.html>)

Scala API (<http://spark.apache.org/docs/latest/api/scala/index.html#org.apache.spark.package>)

Scala Cheat-Sheet

(<http://homepage.cs.uiowa.edu/~tinelli/classes/022/Fall13/Notes/scala-quick-reference.pdf>)

Last updated 2018-12-10 17:10:59 CET