

10/10/2018

BDLE

Indications pour utiliser l'outil IDEA dans les salles de TME

Les étudiants BDLE ont un quota (5G) leur permettant d'utiliser idea

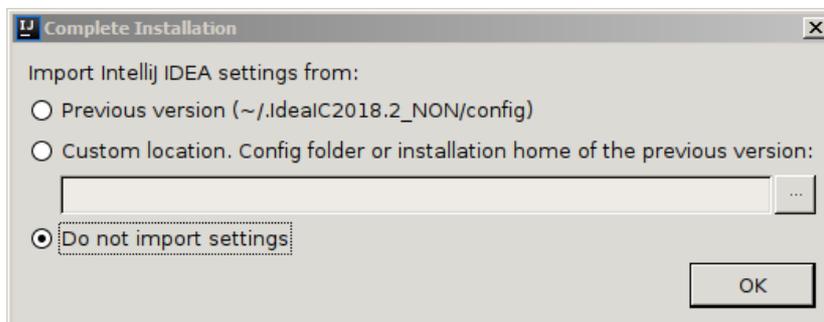
Configurer l'environnement

```
source /Infos/bd/spark-config
```

Lancer l'IDE idea

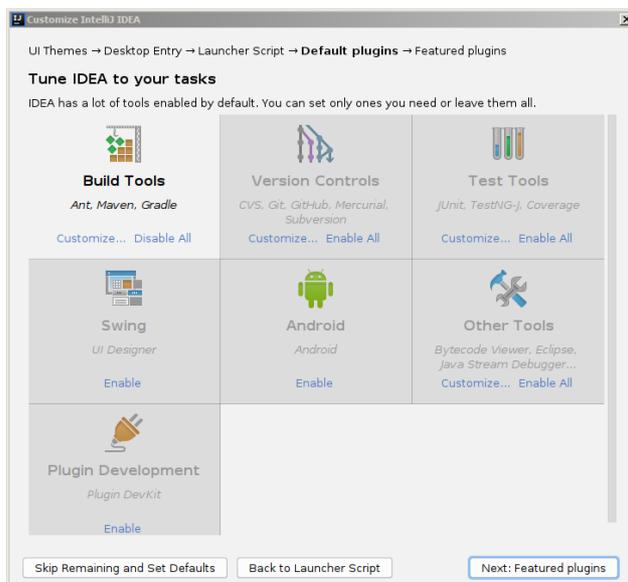
```
idea.sh &
```

Cocher **do not import settings**



Choix de l'apparence: IntelliJ (mais les autres apparences peuvent être utilisées)

Choix des Default Plugins : disable tous SAUF Build Tools qui doit resté sélectionné

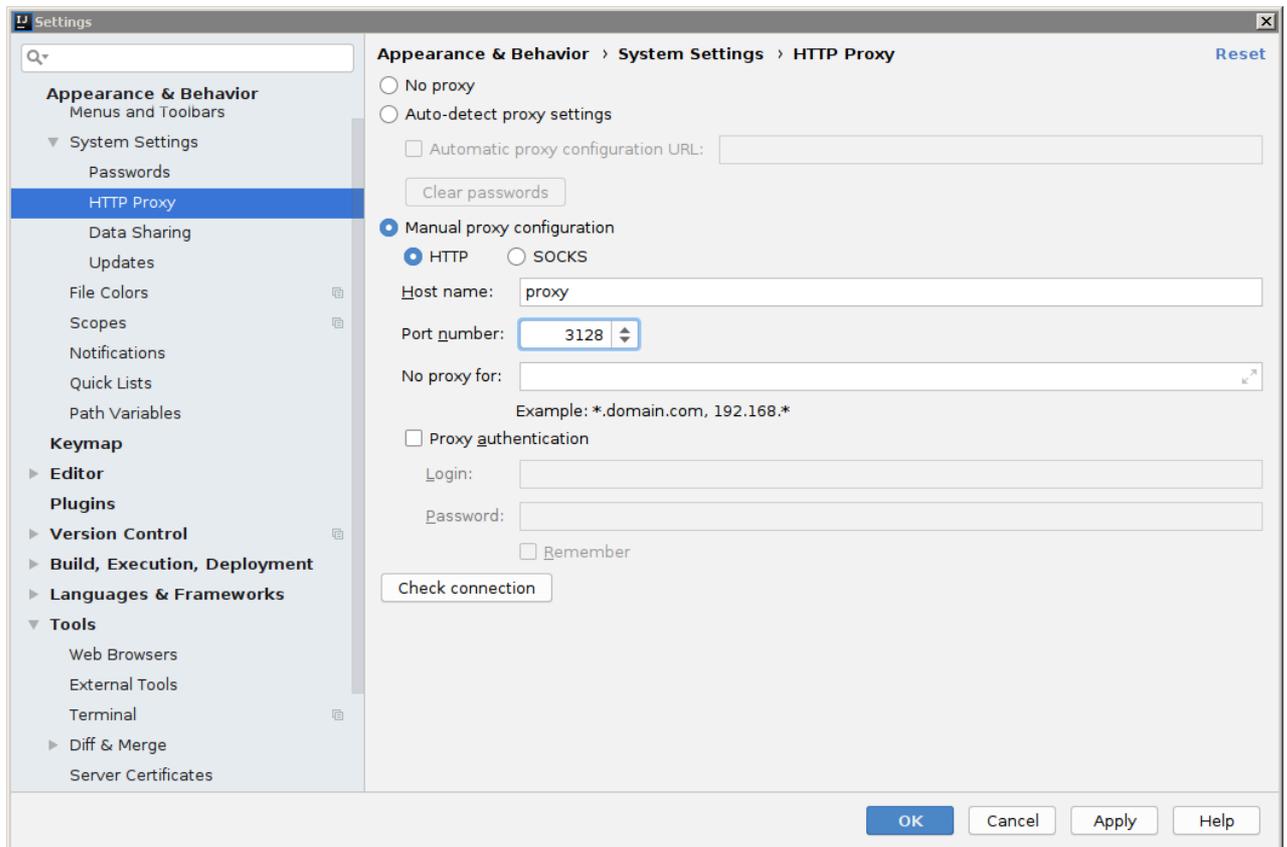


Configurer le proxy dans IDEA : File > Settings > Appearance > System Settings > HTTP Proxy:

Cocher *Manual proxy configuration* et saisir

Host name: **proxy**

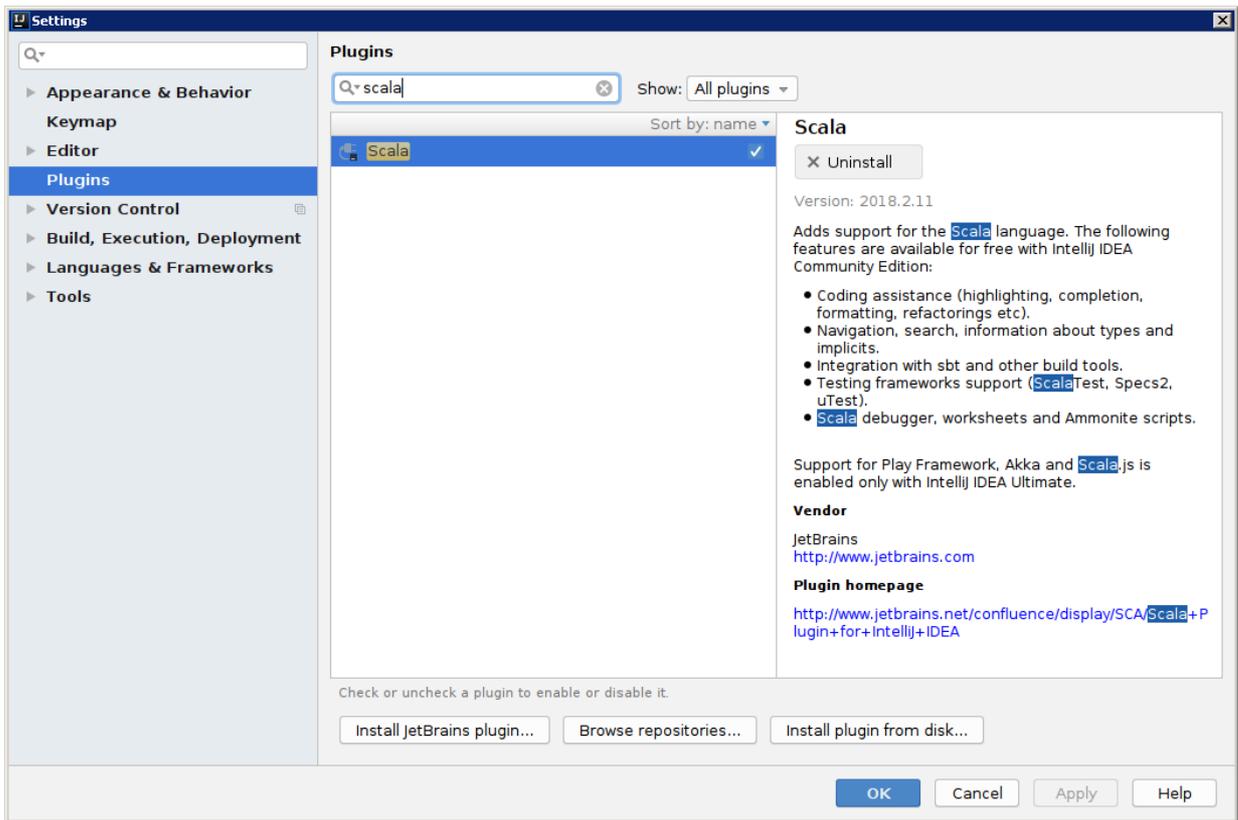
Port number: **3128**



Télécharger le plugin scala :

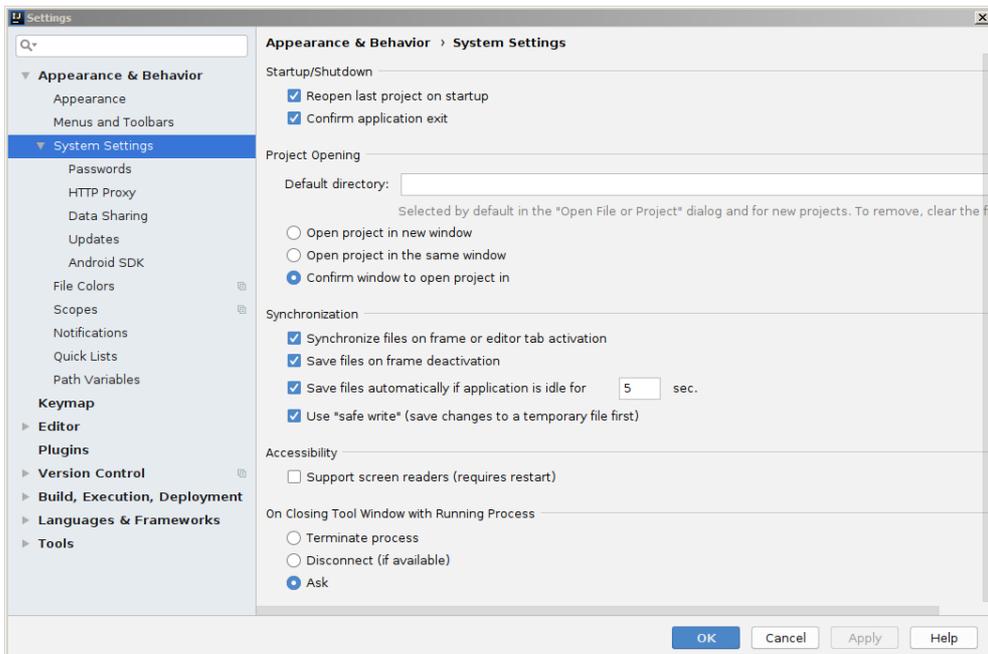
File > Settings > Plugins. Saisir *scala* dans la barre de recherche des plugins

Choisir le plugin nommé **Scala**



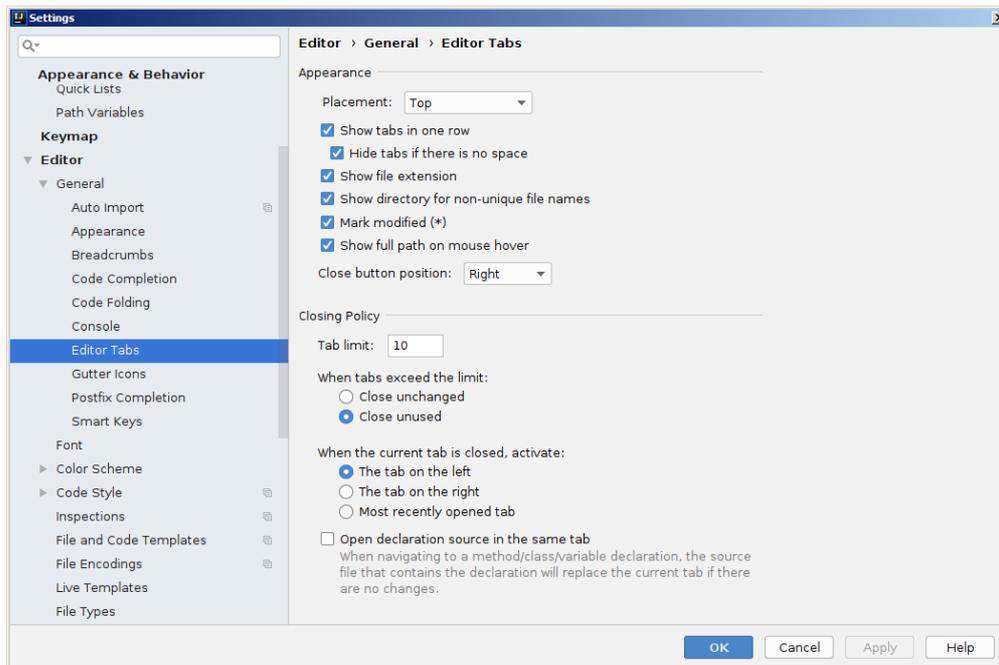
Activer la **sauvegarde automatique** des fichiers toutes les 5 secondes:

File > Settings > Appearance & behavior > System Settings > Dans la section Synchronization, cocher la case “Save files automatically if application is idle for” **5** sec.



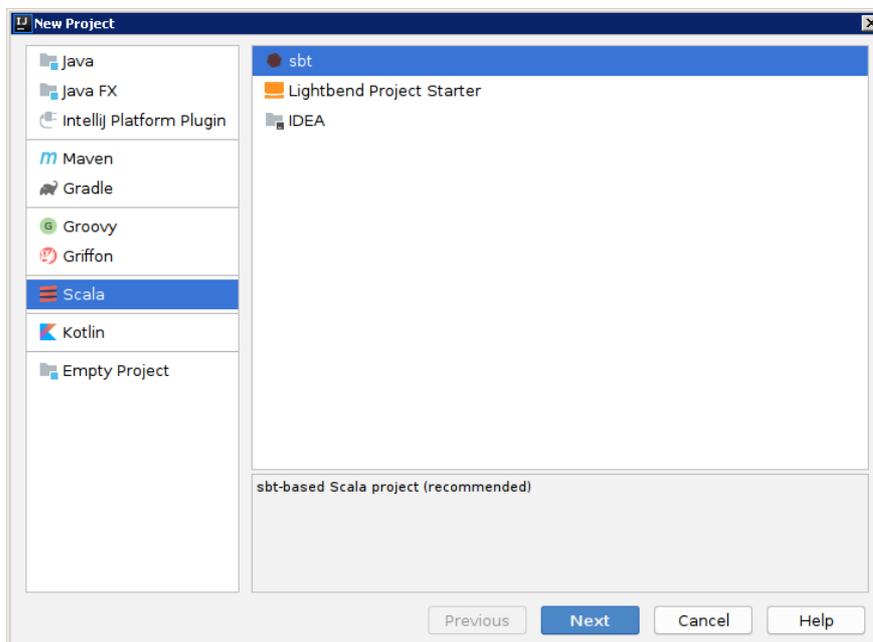
Marquer par une astérisque les onglets non sauvegardés :

File > Settings > Editor > General > Editor Tabs puis Cocher la case **Marks modified (*)**

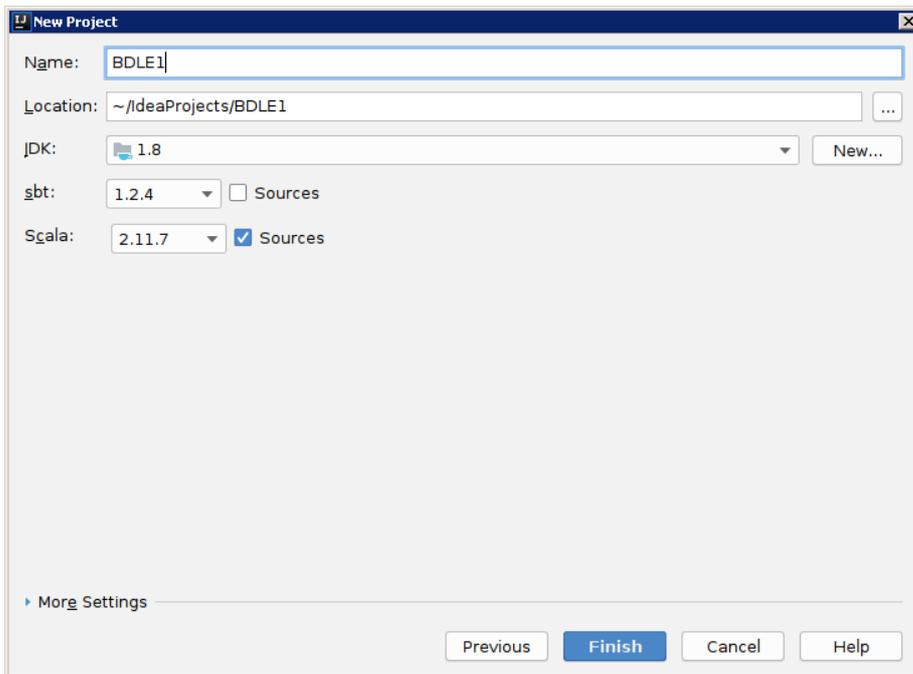


Créer un nouveau projet de type Scala **SBT**

File > New > Project . Sélectionner **Scala** à gauche et **SBT** en haut au centre

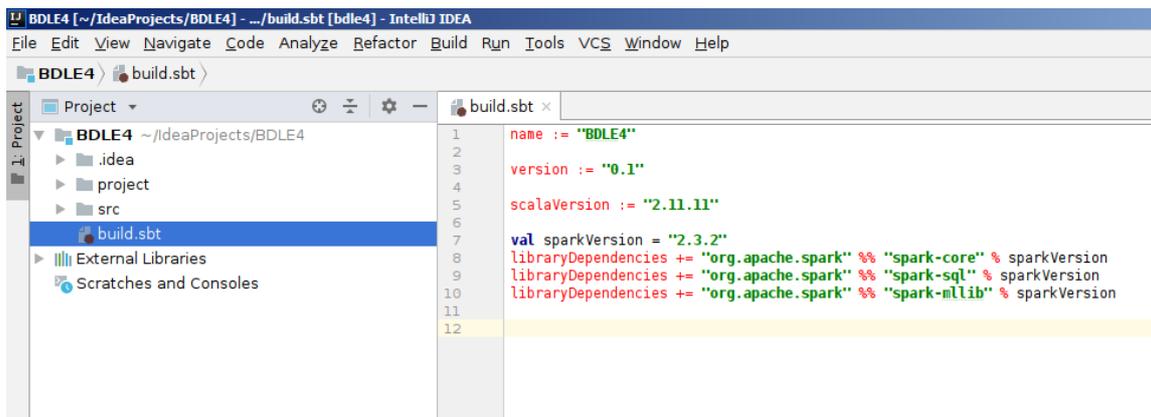


Choisir la version 11 de scala: **2.11** (ne **pas** garder la version 12 par défaut incompatible avec spark)



Lorsque le projet est créé:

Dans le panneau de gauche, sélectionner le fichier build.sbt qui se trouve déjà dans le projet que vous venez de créer et compléter le.



Ajouter:

```
scalaVersion := "2.11.11"
```

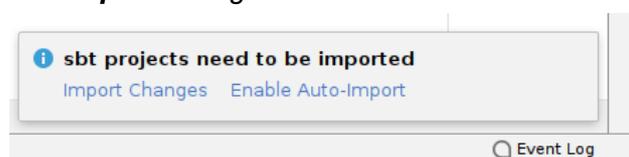
```
val sparkVersion = "2.3.2"
```

```
libraryDependencies += "org.apache.spark" %% "spark-core" % sparkVersion
```

```
libraryDependencies += "org.apache.spark" %% "spark-sql" % sparkVersion
```

```
libraryDependencies += "org.apache.spark" %% "spark-mllib" % sparkVersion
```

Cliquer sur **Import changes** en bas à droite

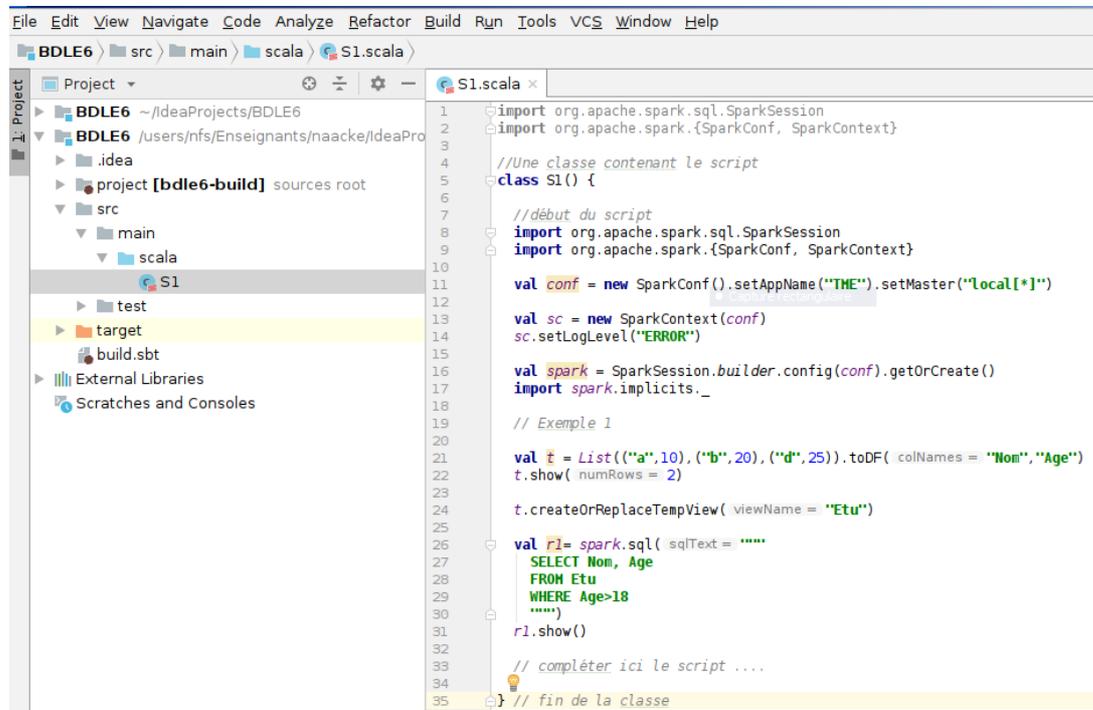


Dans le panneau de gauche, si 2 projets avec le même nom s'affichent dans le panneau de gauche, choisir le 2ème qui contient un sous projet avec suffixe **build**.

Puis aller dans le dossier src > main > scala

Attention le dossier scala doit être **coloré en BLEU**.

Créer un **script** scala : File > New > **Scala script** et lui donner un nom



```
1 import org.apache.spark.sql.SparkSession
2 import org.apache.spark.{SparkConf, SparkContext}
3
4 //Une classe contenant le script
5 class S1() {
6
7     //début du script
8     import org.apache.spark.sql.SparkSession
9     import org.apache.spark.{SparkConf, SparkContext}
10
11     val conf = new SparkConf().setAppName("TME").setMaster("local[*]")
12
13     val sc = new SparkContext(conf)
14     sc.setLogLevel("ERROR")
15
16     val spark = SparkSession.builder.config(conf).getOrCreate()
17     import spark.implicits._
18
19     // Exemple 1
20
21     val t = List(("a",10),("b",20),("d",25)).toDF( colNames = "Nom", "Age")
22     t.show( numRows = 2)
23
24     t.createOrReplaceTempView( viewName = "Etu")
25
26     val r1= spark.sql( sqlText = """
27     SELECT Nom, Age
28     FROM Etu
29     WHERE Age>18
30     """)
31     r1.show()
32
33     // compléter ici le script ....
34
35 }
```

Saisir les ligne suivantes dans le script

```
import org.apache.spark.sql.SparkSession
import org.apache.spark.{SparkConf, SparkContext}

//Une classe contenant le script
class S1() {

    //début du script
    import org.apache.spark.sql.SparkSession
    import org.apache.spark.{SparkConf, SparkContext}

    val conf = new SparkConf().setAppName("TME").setMaster("local[*]")
    val spark = SparkSession.builder.config(conf).getOrCreate()
    spark.sparkContext.setLogLevel("ERROR")
    import spark.implicits._

    // Exemple 1

    val t = List(("a",10),("b",20),("d",25)).toDF("Nom", "Age")
    t.show(2)
```

```
t.createOrReplaceTempView("Etu")
```

```
val r1= spark.sql("""  
  SELECT Nom, Age  
  FROM Etu  
  WHERE Age>18  
  """)  
r1.show()
```

```
// compléter ici le script ....
```

```
} // fin de la classe
```

Bouton droit : **run scala console**

puis sélectionner les lignes à partir du commentaire *//début du script*

et envoyer la sélection dans la console avec Ctrl-Maj-X

ainsi de suite : possibilité d'envoyer un paragraphe quelconque vers la console

(même fonctionnalité qu'avec le spark-shell).

Eviter autant que possible de saisir directement vos réponses dans la console. La console sert à voir le résultat de vos commandes saisies dans le fichier script scala.

Pour **commenter**/décommenter une région : Ctrl-/ (utiliser la touche "diviser" du pavé numérique).

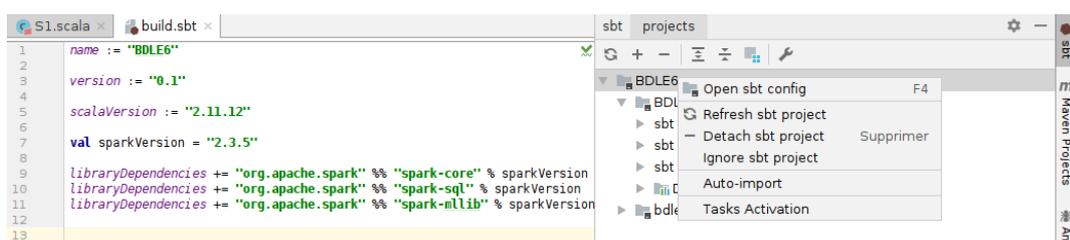
Avantages de l'outil idea : complétion automatique, indentation, info contextuelle, etc... Cela devrait vous permettre d'avancer plus rapidement sur votre projet.

bon projet!

Questions fréquentes

1. Impossible d'activer sbt

Cliquer sur **sbt** sur le panneau vertical de droite puis choisir "**refresh sbt project**"



2. Dans l'éditeur scala, la sélection d'une région de plusieurs lignes ne fonctionne pas. Vérifier que l'indentation est correcte.
3. Erreur au démarrage de idea: librairies java non trouvées : vérifier avant de lancer idea.sh, que la variable JAVA_HOME est correctement fixée. Le programme \$JAVA_HOME/bin/java doit exister.

Doc diverse

<https://fr.hortonworks.com/tutorial/setting-up-a-spark-development-environment-with-scala/>