How to Read Apache Kylin from Apache Flink using Scala

Date: August 2016

Author: Ramón Portolés, Alberto a.ramonportoles@gmail.com Linkedin

Intro

There are several Attempts to use this in Scala and JDBC Attempt1 Attempt2 Attempt3 Attempt4 … but none works ...

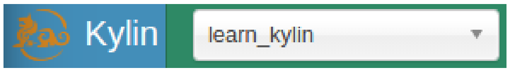
Problem 1: There aren't any doc about connect Kylin with Flink
Problem 2: We will try use CreateInput and JDBCInputFormat in batch mode and access via JDBC to Kylin. But isn't implemented in Scala is only in Java MailList

Then , We will go step by step solving problems

Pre-requisites

- We need an instance of Kylin, with a cube: Quick Start with Sample Cube, will be enough

  You can check:



learn_kylin

- Scala and Apache Flink Installed
- IntelliJ Installed and configured for Scala / Flink (See Flink IDE setup guide)

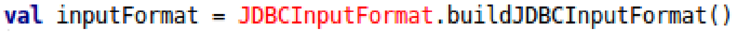
Used Software:

- Apache Flink v1.2-SNAPSHOT
- Apache Kylin v1.5.2
- IntelliJ v2016.2
- Scala v2.11

Starting point:

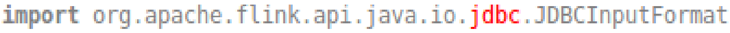
This can be out initial skeleton:

```
import org.apache.flink.api.scala._

val env = ExecutionEnvironment.getExecutionEnvironment

val inputFormat = JDBCInputFormat.buildJDBCInputFormat()
  .setDrivername("org.apache.kylin.jdbc.Driver")
  .setDBUrl("jdbc:kylin://172.17.0.2:7070/learn_kylin")
  .setUsername("ADMIN")
  .setPassword("KYLIN")
  .setQuery("select count(distinct seller_id) as sellers from kylin_sales
group by part_dt order by part_dt")
  .finish()
  val dataset =env.createInput(inputFormat)
```

The first error is: 

```
val inputFormat = JDBCInputFormat.buildJDBCInputFormat()
```

We add to Scala: `import org.apache.flink.api.java.io.jdbc.JDBCInputFormat`

Next error is 

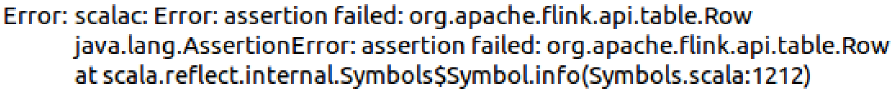
```
import org.apache.flink.api.java.io.jdbc.JDBCInputFormat
```

We can solve dependencies (mvn repository: jdbc)
Add this to your *pom.xml*:

```
<dependency>
   <groupId>org.apache.flink</groupId>
   <artifactId>flink-jdbc</artifactId>
   <version>${flink.version}</version>
</dependency>
```

Solve dependencies of row

Similar to previous point we need solve dependencies of Row Class (mvn repository: Table):



```
Error: scalac: Error: assertion failed: org.apache.flink.api.table.Row
       java.lang.AssertionError: assertion failed: org.apache.flink.api.table.Row
       at scala.reflect.internal.Symbols$Symbol.info(Symbols.scala:1212)
```

- In POM.XML

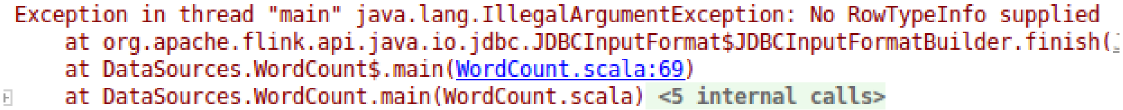
```
<dependency>
      <groupId>org.apache.flink</groupId>
      <artifactId>flink-table_2.10</artifactId>
      <version>${flink.version}</version>
</dependency>
```

- In Scala:

```
import org.apache.flink.api.table.Row
```

Solve RowTypeInfo property (and their new dependencies)

This is the new error to solve



```
Exception in thread "main" java.lang.IllegalArgumentException: No RowTypeInfo supplied
    at org.apache.flink.api.java.io.jdbc.JDBCInputFormat$JDBCInputFormatBuilder.finish(
    at DataSources.WordCount$.main(WordCount.scala:69)
    at DataSources.WordCount.main(WordCount.scala) <5 internal calls>
```

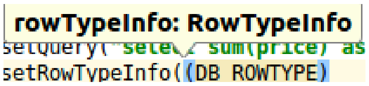
- If we check the code of JDBCInputFormat.java, we can see this new property (and mandatory) added on Apr 2016 by FLINK-3750 Manual JDBCInputFormat v1.2 in Java

Add the new Property: **setRowTypeInfo**

```
val inputFormat = JDBCInputFormat.buildJDBCInputFormat()
  .setDrivername("org.apache.kylin.jdbc.Driver")
  .setDBUrl("jdbc:kylin://172.17.0.2:7070/learn_kylin")
  .setUsername("ADMIN")
  .setPassword("KYLIN")
  .setQuery("select count(distinct seller_id) as sellers from kylin_sales
group by part_dt order by part_dt")
  .setRowTypeInfo(DB_ROWTYPE)
  .finish()
```

- ¿How can configure this property in Scala? In Attempt4 , there is an incorrect solution

We can check the types using the intellisense:



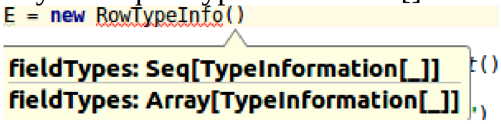
```
rowTypeInfo: RowTypeInfo
setRowTypeInfo((DB ROWTYPE)
```

Then we will need add more dependences :(

Add to scala:

```
      import org.apache.flink.api.table.typeutils.RowTypeInfo
      import org.apache.flink.api.common.typeinfo.{BasicTypeInfo,
TypeInformation}
```

We need create a Array or Seq of TypeInformation[]



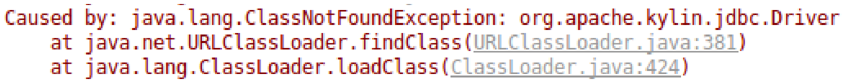
```
E = new RowTypeInfo()
fieldTypes: Seq[TypeInformation[_]]
fieldTypes: Array[TypeInformation[_]]
```

My solution:

```
var stringColum: TypeInformation[String] = createTypeInformation[String]
val DB_ROWTYPE = new RowTypeInfo(Seq(stringColum))
```

Solve Class Not Found

We need find the kylin-jdbc-x.x.x.jar and expose to flink

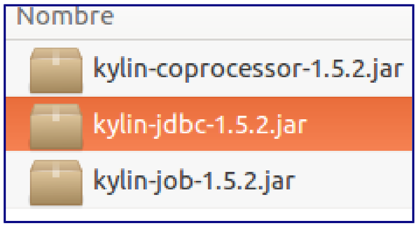


```
Caused by: java.lang.ClassNotFoundException: org.apache.kylin.jdbc.Driver
    at java.net.URLClassLoader.findClass(URLClassLoader.java:381)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
```

1. We need to find the JAR Class for the JDBC Connector

   From Kylin Download Choose **Binary** and the **correct version of Kylin and HBase**

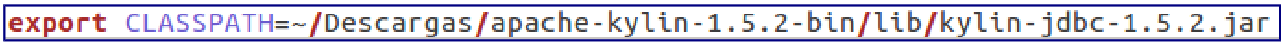
   Download & Unpack: in ./lib:



Nombre
kylin-coprocessor-1.5.2.jar
kylin-jdbc-1.5.2.jar
kylin-job-1.5.2.jar

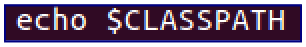
2. Make this JAR accessible to Flink

   If you execute like service you need put this JAR in you Java ClassPATH using your .bashrc



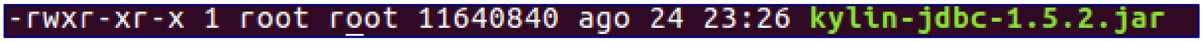
```
export CLASSPATH=~/Descargas/apache-kylin-1.5.2-bin/lib/kylin-jdbc-1.5.2.jar
```

Check the actual value:



```
echo $CLASSPATH
```

Check the permission for this file (Must be accessible for you):

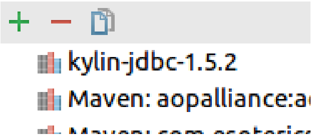


```
-rwxr-xr-x 1 root root 11640840 ago 24 23:26 kylin-jdbc-1.5.2.jar
```

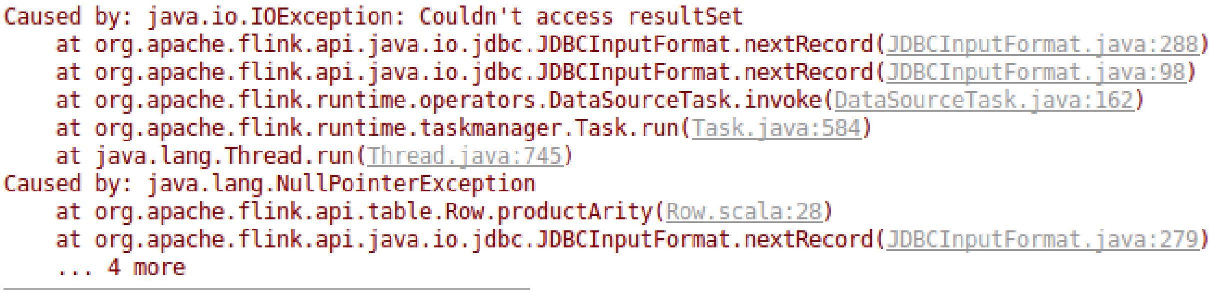
If you are executing from IDE, you need add your ClassPath manually:

On IntelliJ: File > Project Structure... > Libraries > + −

The result, will be similar to:



Solve Couldn’t access resultSet

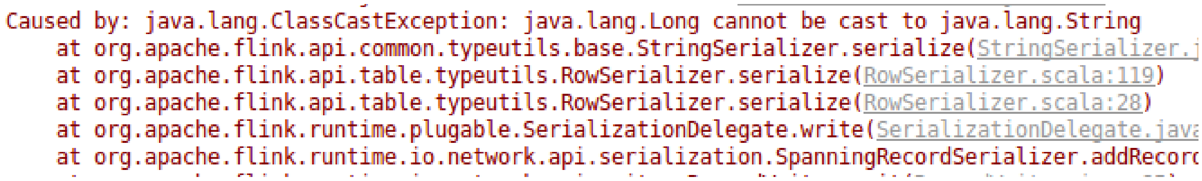


```
Caused by: java.io.IOException: Couldn't access resultSet
    at org.apache.flink.api.java.io.jdbc.JDBCInputFormat.nextRecord(JDBCInputFormat.java:288)
    at org.apache.flink.api.java.io.jdbc.JDBCInputFormat.nextRecord(JDBCInputFormat.java:98)
    at org.apache.flink.runtime.operators.DataSourceTask.invoke(DataSourceTask.java:162)
    at org.apache.flink.runtime.taskmanager.Task.run(Task.java:584)
    at java.lang.Thread.run(Thread.java:745)
Caused by: java.lang.NullPointerException
    at org.apache.flink.api.table.Row.productArity(Row.scala:28)
    at org.apache.flink.api.java.io.jdbc.JDBCInputFormat.nextRecord(JDBCInputFormat.java:279)
    ... 4 more
```

Is related with Flink 4108 (MailList) and Timo Walther make a PR

If you are <= Flink 1.2 you will need apply this path and *make clean install*

Solve the casting

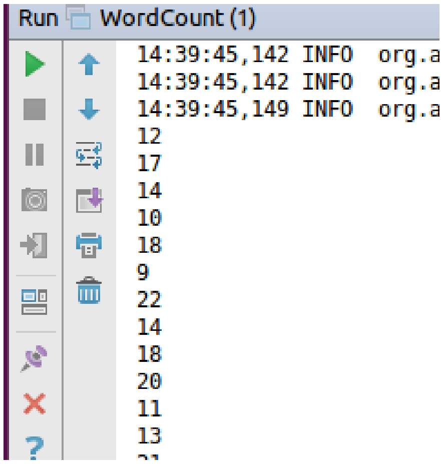


```
Caused by: java.lang.ClassCastException: java.lang.Long cannot be cast to java.lang.String
    at org.apache.flink.api.common.typeutils.base.StringSerializer.serialize(StringSerializer.j
    at org.apache.flink.api.table.typeutils.RowSerializer.serialize(RowSerializer.scala:119)
    at org.apache.flink.api.table.typeutils.RowSerializer.serialize(RowSerializer.scala:28)
    at org.apache.flink.runtime.plugable.SerializationDelegate.write(SerializationDelegate.java
    at org.apache.flink.runtime.io.network.api.serialization.SpanningRecordSerializer.addRecord
```

In the error msg you have the problem and solution …. nice ;) ¡¡

The result

The out must be similar to this, print the result of query by standard output:



```
14:39:45,142 INFO  org.a
14:39:45,142 INFO  org.a
14:39:45,149 INFO  org.a
12
17
14
10
18
9
22
14
18
20
11
13
```

Now more complex

We can try with multi-colum and multi-type query:

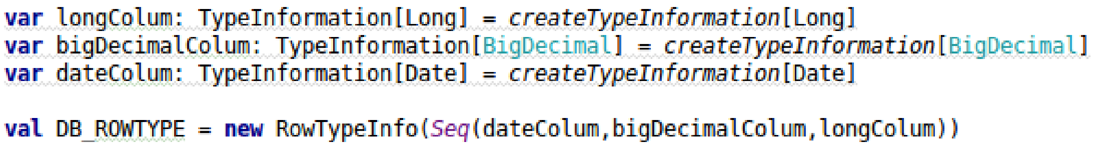
select part_dt, sum(price) as total_selled, count(distinct seller_id) as sellers

from kylin_sales

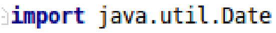
group by part_dt

order by part_dt

We will need changes in DB_ROWTYPE:

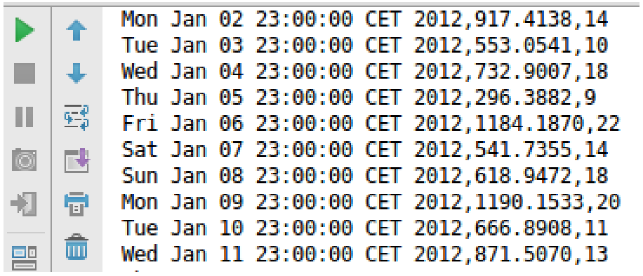


```
var longColum: TypeInformation[Long] = createTypeInformation[Long]
var bigDecimalColum: TypeInformation[BigDecimal] = createTypeInformation[BigDecimal]
var dateColum: TypeInformation[Date] = createTypeInformation[Date]

val DB_ROWTYPE = new RowTypeInfo(Seq(dateColum,bigDecimalColum,longColum))
```

And import lib of Java, to work with Data type of Java 

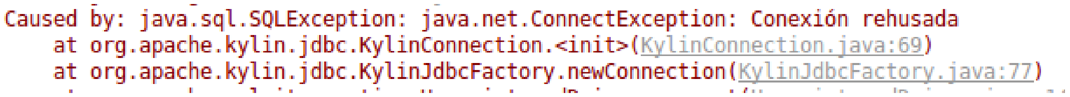
```
import java.util.Date
```

The new Result will be:



```
Mon Jan 02 23:00:00 CET 2012,917.4138,14
Tue Jan 03 23:00:00 CET 2012,553.0541,10
Wed Jan 04 23:00:00 CET 2012,732.9007,18
Thu Jan 05 23:00:00 CET 2012,296.3882,9
Fri Jan 06 23:00:00 CET 2012,1184.1870,22
Sat Jan 07 23:00:00 CET 2012,541.7355,14
Sun Jan 08 23:00:00 CET 2012,618.9472,18
Mon Jan 09 23:00:00 CET 2012,1190.1533,20
Tue Jan 10 23:00:00 CET 2012,666.8908,11
Wed Jan 11 23:00:00 CET 2012,871.5070,13
```

Error: Reused Connection



```
Caused by: java.sql.SQLException: java.net.ConnectException: Conexión rehusada
    at org.apache.kylin.jdbc.KylinConnection.<init>(KylinConnection.java:69)
    at org.apache.kylin.jdbc.KylinJdbcFactory.newConnection(KylinJdbcFactory.java:77)
```

Check if your HBase and Kylin is working

Also you can use Kylin UI for it

Final Words

Now we can read Kylin’s data from Apache Flink, great News ¡¡

We solved all integration problems, and tested with different types of data (Long, BigDecimal and Dates)

Today (12 Oct 2016) Flink 1.2-SnapShot, you need download, apply path, compile and make install …. but in new releases will not necessary

For any suggestions, feel free to contact me

Thanks, Alberto