
Artículo

[Alberto Fuentes](#) · 14 mayo, 2019 · Lectura de 3 min

Guía rápida para conectar Apache Spark y Apache Zeppelin con InterSystems IRIS

Hola a todos, me gustaría compartir una guía rápida publicada en el Developer Community sobre cómo conectar Apache Spark + Apache Zeppelin con InterSystems IRIS.

Introducción

Apache Spark es un framework opensource para computación en cluster. Proporciona una interfaz para desarrollar sobre clusters incluyendo paralelismo y tolerancia a fallos. Por ello es muy utilizado en Big Data.

Apache Zeppelin es un bloc de notas, que cuenta con una excelente interfaz de usuario para trabajar con diversos tipos análisis y machine learning.

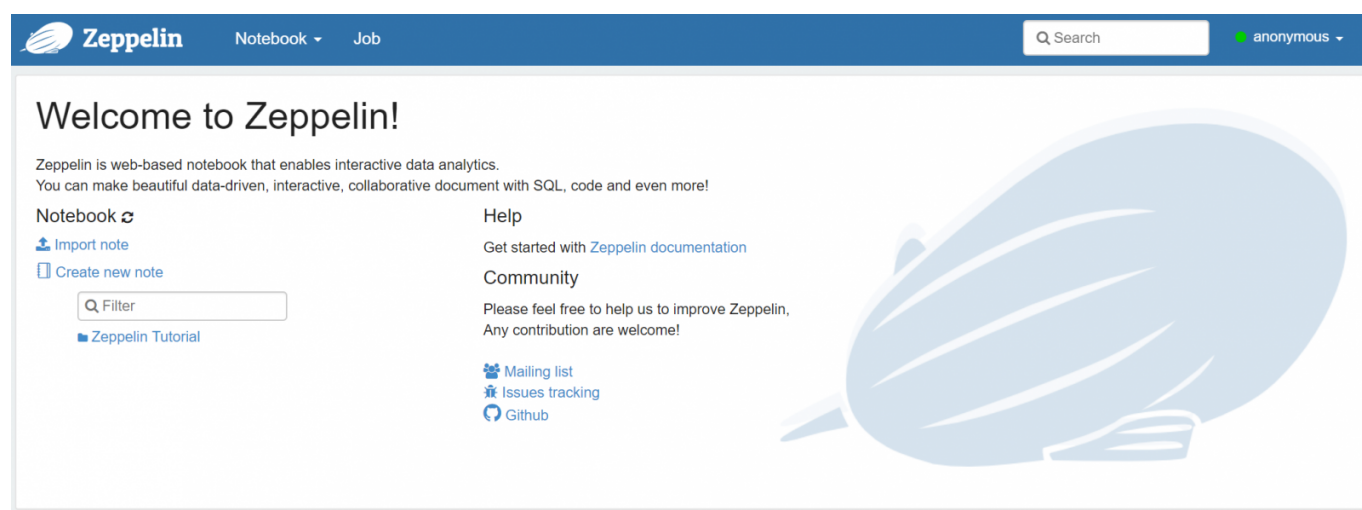
Combinándolo con InterSystems IRIS, el conjunto funciona de tal manera que: IRIS proporciona los datos, Spark los lee y los trabajamos con en un bloc de notas de Apache Zeppelin.

Nota: la guía se ha confeccionado ejecutando los pasos sobre Windows 10.

Apache Zeppelin

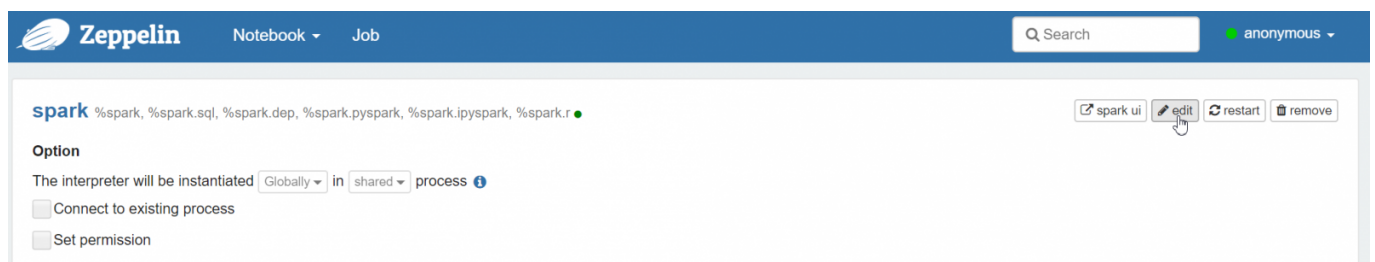
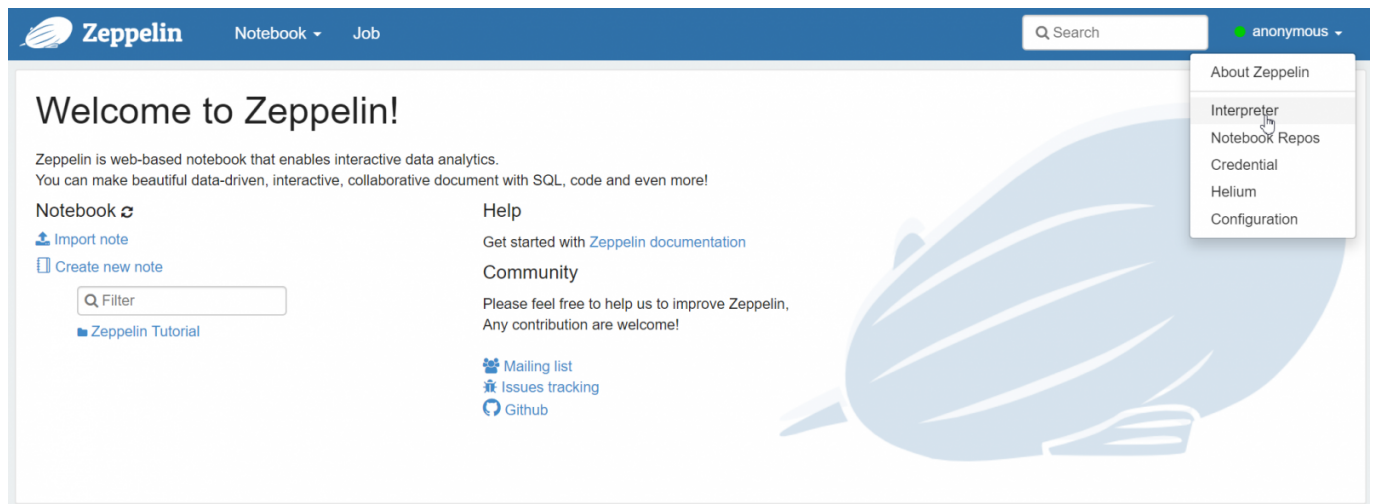
Ahora instalaremos los programas necesarios. Antes que nada, descargamos Apache Zeppelin de su [sitio oficial](#). Hemos utilizado zeppelin-0.8.0-bin-all.tgz que incluye Apache Spark, Scala, y Python.

Tras descomprimirlo, podemos ejecutar Zeppelin con el script con `/bin/zeppelin.cmd`, que se encuentra en la raíz de su carpeta Zeppelin. Tras aparecer "Done, zeppelin server started" ya podemos abrir la interfaz <http://localhost:8080> en el navegador. Si todo se ejecutó de manera correcta, aparecerá lo siguiente:



Apache Spark

En la interfaz de Zeppelin, hacemos click en la esquina superior derecha de la pantalla en `anonymous > Interpreter`. Nos desplazamos hacia abajo y buscamos spark.



Hacemos click en el botón edit que aparece junto a spark y nos desplazamos hacia abajo para añadir las dependencias a las librerías jar que necesitamos (se encuentran en el directorio `dev\java\lib\JDK` de la instalación de IRIS).

Dependencies

These dependencies will be added to classpath when interpreter process starts.

artifact	exclude	action
C:\InterSystems\IRIS\dev\java\lib\JDK18\intersystems-jdbc-3.0.0.jar	(Optional) comma separated groupId:artifactId list	✕
C:\InterSystems\IRIS\dev\java\lib\JDK18\intersystems-spark-1.0.0.jar	(Optional) comma separated groupId:artifactId list	+

[Save](#) [Cancel](#)

Comprobar el funcionamiento

Vamos a comprobar que funciona. Creamos una nueva nota en Zeppelin, y en un párrafo nuevo escribimos el siguiente código para recuperar datos de una tabla `Sample.Person`:

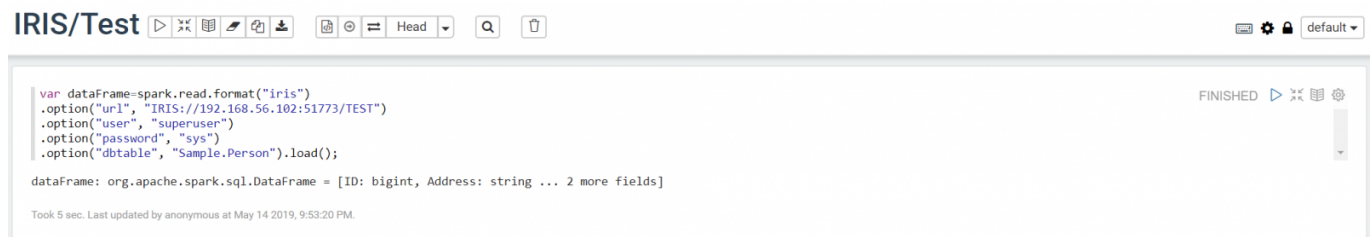
```
var dataframe=spark.read.format("com.intersystems.spark").option("url", "IRIS://localhost:51773/NAMESPACE").option("user", "UserLogin").option("password", "UserPassword").option("dbtable", "Sample.Person").load()

// dbtable - pasar el nombre de la tabla a cargar (e.g. Sample.Person)
```

La URL de IRIS tiene la siguiente forma: `IRIS://ipAddress:superserverPort/namespace`:

- El protocolo IRIS es una conexión JDBC hecha sobre TCP/IP que ofrece una conexión de memoria compartida con Java.
- ipAddress — La dirección IP de la instancia InterSystems IRIS.
- superserverPort — El número de puerto del superservidor para la instancia IRIS (no es el puerto web). Recuerda que para consultar el número puerto del superservidor de IRIS puedes ir al Portal de Administración del sistema System Administration > Configuration > System Configuration > Memory and Startup (SuperServer port)
- Namespace - Namespace de la instancia al que vamos a conectarnos.

Al ejecutarlo, si todo va bien aparecerá como FINISHED



The screenshot shows the Apache Zeppelin web interface. At the top, there's a toolbar with icons for play, stop, refresh, and other actions. Below the toolbar, the job is titled "IRIS/Test". The main area displays the executed code in Scala, which connects to an IRIS instance and loads data from a table named "Sample.Person". The output shows the resulting DataFrame structure: "org.apache.spark.sql.DataFrame = [ID: bigint, Address: string ... 2 more fields]". The status "FINISHED" is visible in the top right corner of the code block. At the bottom, a small note indicates the job took 5 seconds and was last updated by an anonymous user on May 14, 2019.

```
var dataframe=spark.read.format("iris")
.option("url", "IRIS://192.168.56.102:51773/TEST")
.option("user", "superuser")
.option("password", "sys")
.option("dbtable", "Sample.Person").load();

dataframe: org.apache.spark.sql.DataFrame = [ID: bigint, Address: string ... 2 more fields]

Took 5 sec. Last updated by anonymous at May 14 2019, 9:53:20 PM.
```

Conclusión

En esta guía descubrimos cómo hacer que Apache Spark, Apache Zeppelin e InterSystems IRIS trabajen en conjunto. En próximos artículos hablaremos sobre las distintas formas que existen para analizar datos.

Enlaces

- [El sitio oficial de Apache Spark](#)
- [Documentos disponibles sobre Apache Spark](#)
- [El protocolo IRIS](#)
- [Cómo utilizar InterSystems Spark Connector](#)

[#Análítica](#) [#Big Data](#) [#Artificial Intelligence \(AI\)](#) [#Mejores prácticas](#) [#Python](#) [#InterSystems IRIS](#)

URL de
fuente: <https://es.community.intersystems.com/post/gu%C3%ADa-r%C3%A1pida-para-conectar-apache-spark-y-apache-zeppelin-con-intersystems-iris>