Article
Niyaz Khafizov · Jul 6, 2018 3m read

The way to launch Apache Spark + Apache Zeppelin + InterSystems IRIS

Hi all. Yesterday I tried to connect Apache Spark, Apache Zeppelin, and InterSystems IRIS. During the process, I experienced troubles connecting it all together and I did not find a useful guide. So, I decided to write my own.

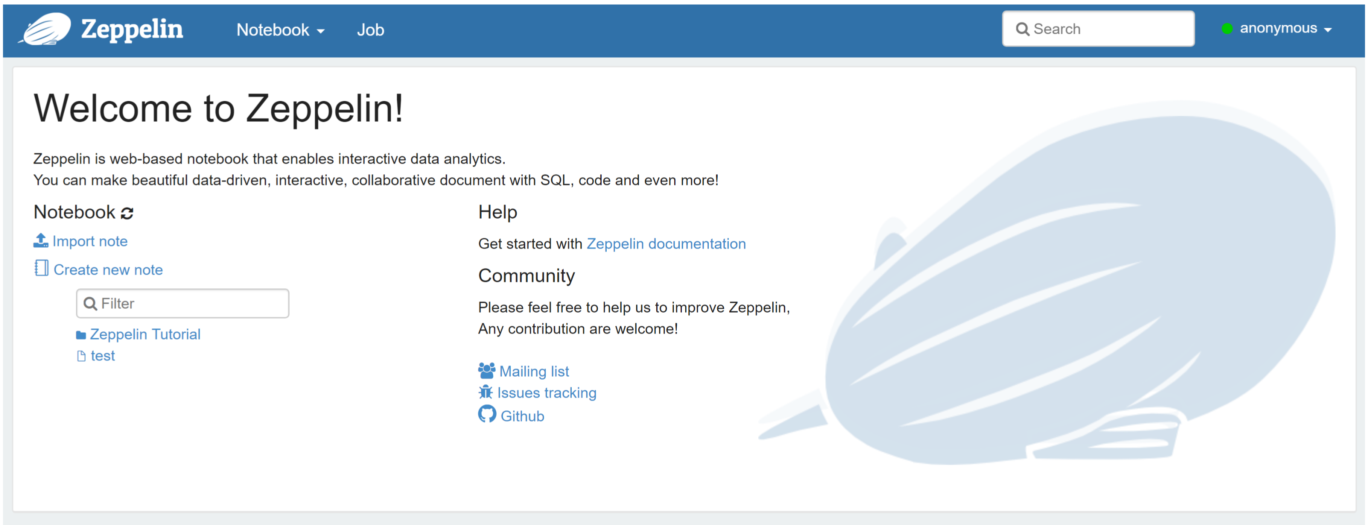
Introduction

What is Apache Spark and Apache Zeppelin and find out how it works together. Apache Spark is an open-source cluster-computing framework. It provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. So, it is very useful when you need to work with Big Data. And Apache Zeppelin is a notebook, that provides cool UI to work with analytics and machine learning. Together, it works like this: IRIS provides data, Spark reads provided data, and in a notebook we work with the data.

Note: I have done the following on Windows 10.

Apache Zeppelin

Now, we will install all the necessary programs. First of all, download apache zeppelin from the official site of apache zeppelin. I have used **zeppelin-0.8.0-bin-all.tgz**. It includes **Apache Spark**, **Scala,** and **Python.** Unzip it to any folder. After that you can launch zeppelin by calling \bin\zeppelin.cmd from the root of your Zeppelin folder. Wait until the **Done, zeppelin server started** string appears and open http://localhost:8080 in your browser. If everything is okay, you will see **Welcome to Zeppelin!** message.



Welcome to Zeppelin!

Zeppelin is web-based notebook that enables interactive data analytics.
You can make beautiful data-driven, interactive, collaborative document with SQL, code and even more!

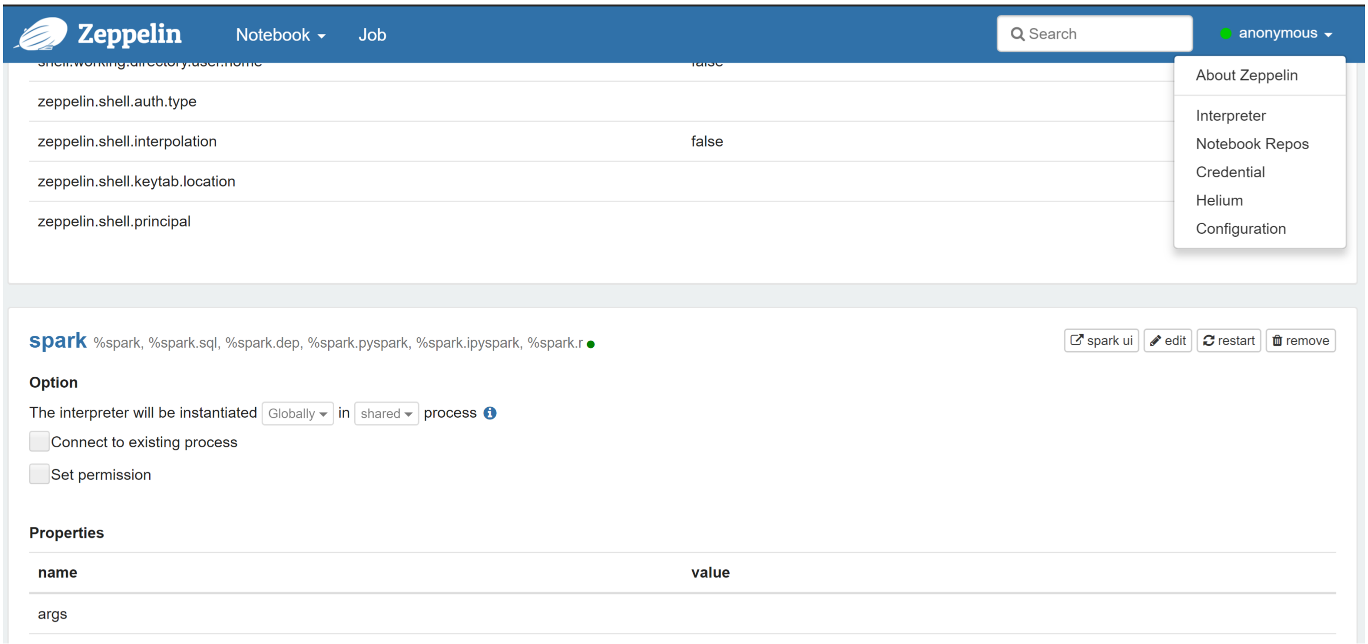
Get started with Zeppelin documentation

Please feel free to help us to improve Zeppelin,
Any contribution are welcome!

Note: I assume, that InterSystems IRIS already installed. If not, download and install it before the next step.

Apache Spark

So, we have the browser's open window with Zeppelin notebook. In the upper-right corner click on **anonymous** and after, click on **Interpreter**. Scroll down and find **spark**.



zeppelin.shell.auth.type	
zeppelin.shell.interpolation	false
zeppelin.shell.keytab.location	
zeppelin.shell.principal	

spark %spark, %spark.sql, %spark.dep, %spark.pyspark, %spark.ipyspark, %spark.r ●

Option

The interpreter will be instantiated Globally in shared process

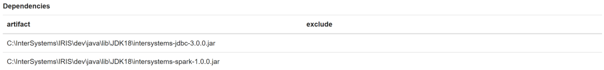
□ Connect to existing process

□ Set permission

Properties

name	value
args	

Next to the spark find **edit** button and click on it. Scroll down and add dependencies to **intersystems-spark-1.0.0.jar** and to **intersystems-jdbc-3.0.0.jar**. I installed InterSystems IRIS to the

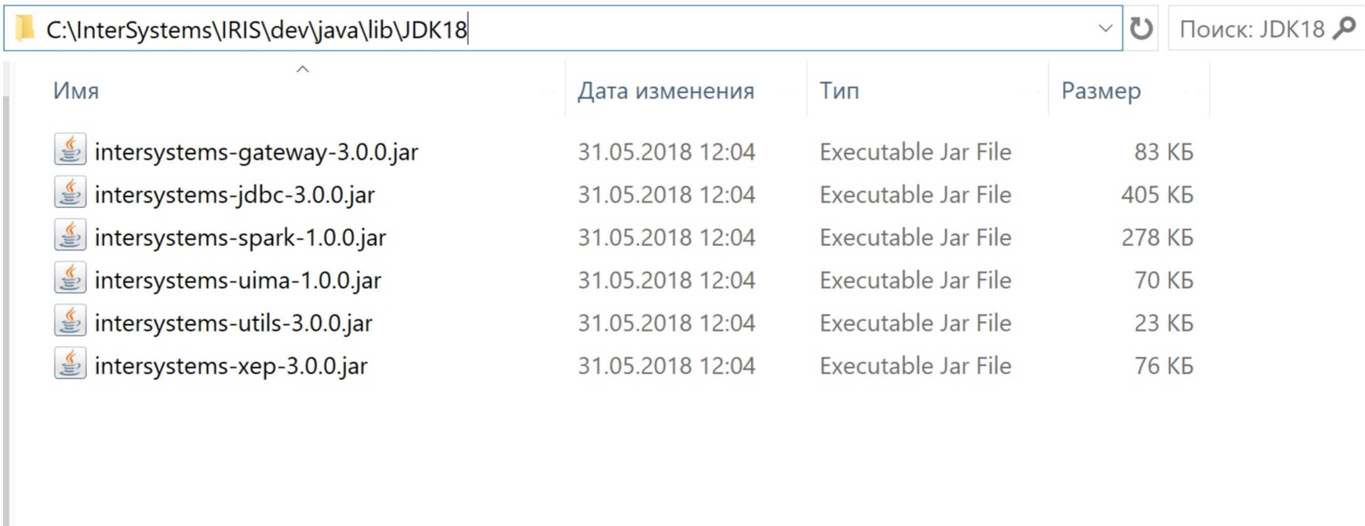


Dependencies

artifact	exclude
C:\InterSystems\IRIS\dev\java\lib\JDK18\intersystems-jdbc-3.0.0.jar	
C:\InterSystems\IRIS\dev\java\lib\JDK18\intersystems-spark-1.0.0.jar	

C:\InterSystems\IRIS\directory, so artifacts I need to add are at:

My files are here:



C:\InterSystems\IRIS\dev\java\lib\JDK18

Имя	Дата изменения	Тип	Размер
intersystems-gateway-3.0.0.jar	31.05.2018 12:04	Executable Jar File	83 КБ
intersystems-jdbc-3.0.0.jar	31.05.2018 12:04	Executable Jar File	405 КБ
intersystems-spark-1.0.0.jar	31.05.2018 12:04	Executable Jar File	278 КБ
intersystems-uima-1.0.0.jar	31.05.2018 12:04	Executable Jar File	70 КБ
intersystems-utils-3.0.0.jar	31.05.2018 12:04	Executable Jar File	23 КБ
intersystems-xep-3.0.0.jar	31.05.2018 12:04	Executable Jar File	76 КБ

And save it.

Check that it works

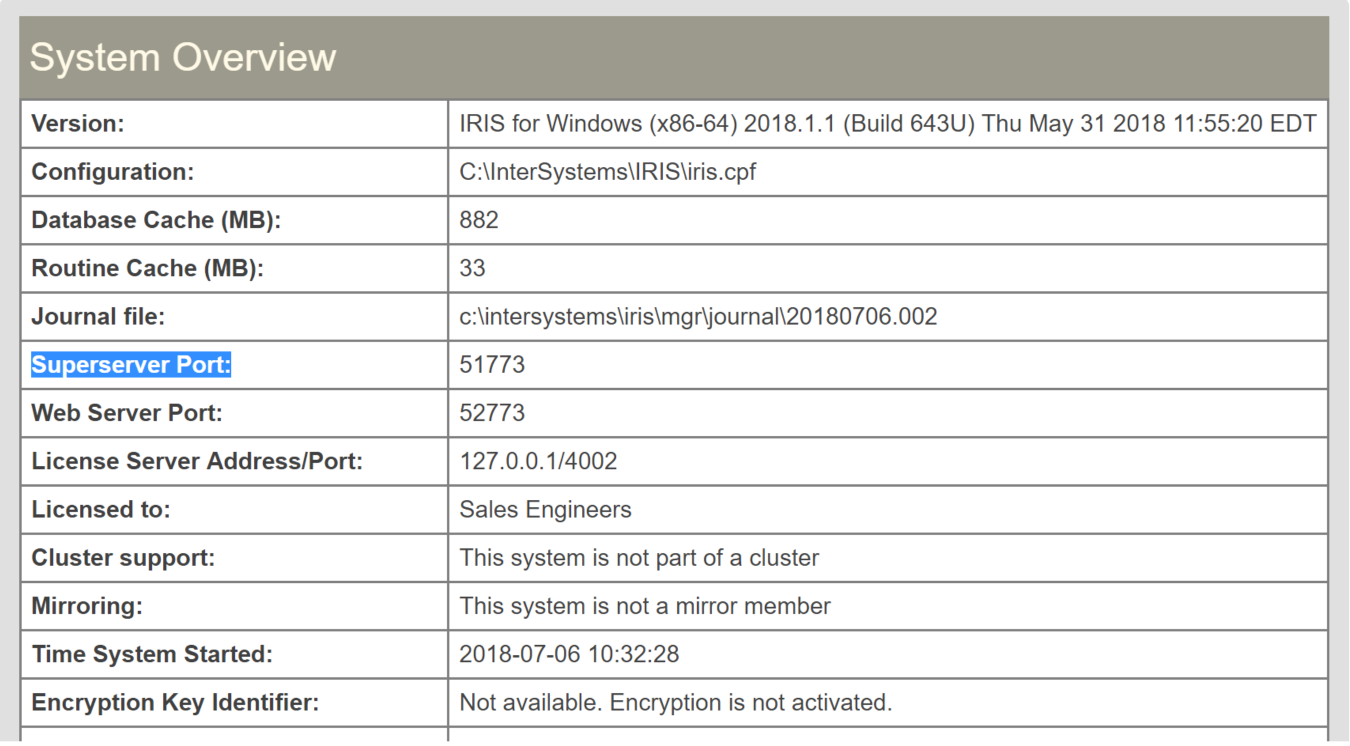
Let us check it. Create a new note, and in a paragraph paste the following code:

```
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 - name of your table
```

URL - IRIS address. It is formed as follows IRIS://ipAddress:superserverPort/namespace:

- protocol IRIS is a JDBC connection over TCP/IP that offers Java shared memory connection;
- ipAddress — The IP address of the InterSystems IRIS instance. If you are connecting locally, use 127.0.0.1 instead of localhost;
- superserverPort — The superserver port number for the IRIS instance, which is not the same as the webserver port number. To find the superserver port number, in the Management Portal, go to System Administration > Configuration > System Configuration > Memory and Startup; namespace — An existing namespace in the InterSystems IRIS instance. In this demo, we connect to the USER namespace.

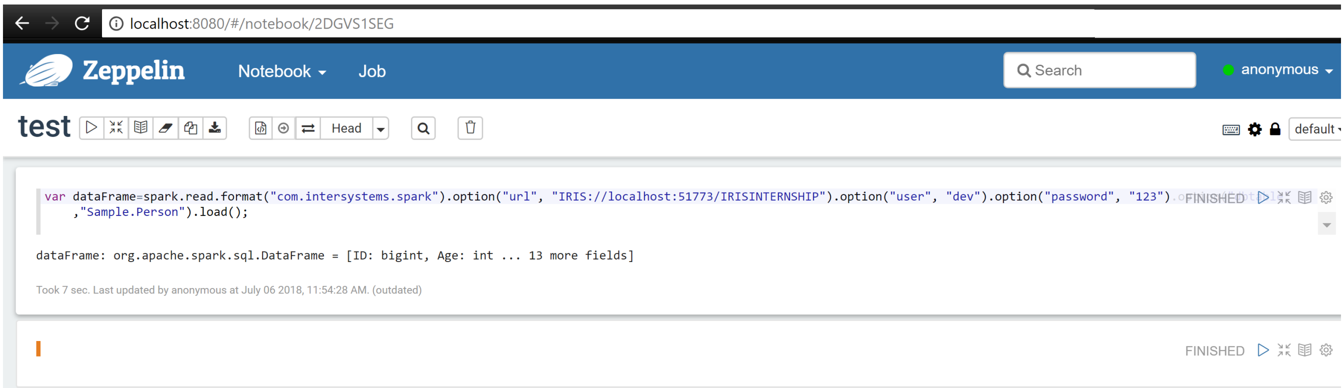


System Overview

Version:	IRIS for Windows (x86-64) 2018.1.1 (Build 643U) Thu May 31 2018 11:55:20 EDT
Configuration:	C:\InterSystems\IRIS\iris.cpf
Database Cache (MB):	882
Routine Cache (MB):	33
Journal file:	c:\intersystems\iris\mgr\journal\20180706.002
Superserver Port:	51773
Web Server Port:	52773
License Server Address/Port:	127.0.0.1/4002
Licensed to:	Sales Engineers
Cluster support:	This system is not part of a cluster
Mirroring:	This system is not a mirror member
Time System Started:	2018-07-06 10:32:28
Encryption Key Identifier:	Not available. Encryption is not activated.

Run the paragraph. If everything is okay, you will see **FINISHED**.

My notebook:



```
var dataFrame=spark.read.format("com.intersystems.spark").option("url", "IRIS://localhost:51773/IRISINTERNSHIP").option("user", "dev").option("password", "123")
    ,"Sample.Person").load();
```

```
dataFrame: org.apache.spark.sql.DataFrame = [ID: bigint, Age: int ... 13 more fields]
```

Took 7 sec. Last updated by anonymous at July 06 2018, 11:54:28 AM. (outdated)

Conclusion

In conclusion, we found out how Apache Spark, Apache Zeppelin, and InterSystems IRIS can work together. In my next articles, I will write about data analysis.

Links

- The official site of Apache Spark
- Apache Spark documentation
- IRIS Protocol
- Using the InterSystems Spark Connector

#Artificial Intelligence (AI) #Beginner #Best Practices #Big Data #Machine Learning (ML) #InterSystems IRIS

Source URL:https://community.intersystems.com/post/way-launch-apache-spark-apache-zeppelin-intersystems-iris