

Installing Jupyter Notebook for Spark

Arnaud.nauwynck@gmail.com

Step 1 : install python (simplest method, not using AnaConda)

The screenshot shows the Python.org downloads page. At the top, there's a navigation bar with links for Python, PSF, Docs, PyPI, Jobs, and Community. Below the navigation bar is the Python logo and a search bar with buttons for 'Donate', 'Search', 'GO', and 'Socialize'. A main menu bar below the logo includes links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The central content area features a large yellow button labeled 'Download the latest version for Windows' with a sub-link 'Download Python 3.11.5'. Below this, text provides links for Python for Windows, Linux/UNIX, macOS, and Other OSes. It also mentions Prereleases and Docker images. To the right of the text is a cartoon illustration of two boxes descending from the sky on parachutes.

← → ⌛ 🔒 python.org/downloads/

Python PSF Docs PyPI Jobs Community

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

Download the latest version for Windows

Download Python 3.11.5

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)

Want to help test development versions of Python 3.12? [Prereleases](#), [Docker images](#)



Step 2 : install jupyter

JupyterLab

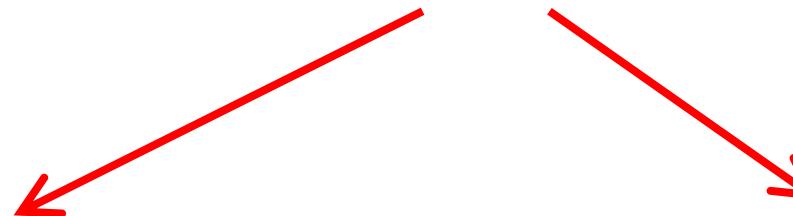
Install JupyterLab with [pip](#):

```
pip install jupyterlab
```

Note: If you install JupyterLab with conda or mamba, we recommend using [the conda-forge channel](#).

Once installed, launch JupyterLab with:

```
jupyter lab
```



Choose class Jupyter Notebook... OK

Jupyter Notebook

Install the classic Jupyter Notebook with:

```
pip install notebook
```

To run the notebook:

```
jupyter notebook
```

Star Jupyter Notebook

jupyter notebook

```
C:\Users\arnaud>jupyter notebook
[I 2023-09-06 23:09:00.357 ServerApp] Package notebook took 0.0000s to import
[I 2023-09-06 23:09:00.482 ServerApp] Package jupyter_lsp took 0.1187s to import
[W 2023-09-06 23:09:00.482 ServerApp] A `__jupyter_server_extension_points` function was not found in jupyter_lsp.
Instead, a `__jupyter_server_extension_paths` function was found and will be used for now. This function name will
be deprecated in future releases of Jupyter Server.
[I 2023-09-06 23:09:00.544 ServerApp] Package jupyter_server_terminals took 0.0598s to import
[I 2023-09-06 23:09:00.544 ServerApp] Package jupyterlab took 0.0000s to import
```

[[truncated logs Also contains errors??]]

Jupyter Notebook ..

```
[I 2023-09-06 23:09:01.879 ServerApp] jupyterlab | extension was successfully loaded.  
[I 2023-09-06 23:09:01.879 ServerApp] notebook | extension was successfully loaded.  
[I 2023-09-06 23:09:01.895 ServerApp] Serving notebooks from local directory: C:\Users\arnaud  
[I 2023-09-06 23:09:01.895 ServerApp] Jupyter Server 2.7.3 is running at:  
[I 2023-09-06 23:09:01.895 ServerApp] http://localhost:8888/tree?token=a9195d7e950f42e4e26a7b93bbebf3664d1794b94959c5e3  
[I 2023-09-06 23:09:01.911 ServerApp] http://127.0.0.1:8888/tree?token=a9195d7e950f42e4e26a7b93bbebf3664d1794b94959c5e3  
[I 2023-09-06 23:09:01.911 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).  
[C 2023-09-06 23:09:01.974 ServerApp]
```

To access the server, open this file in a browser:

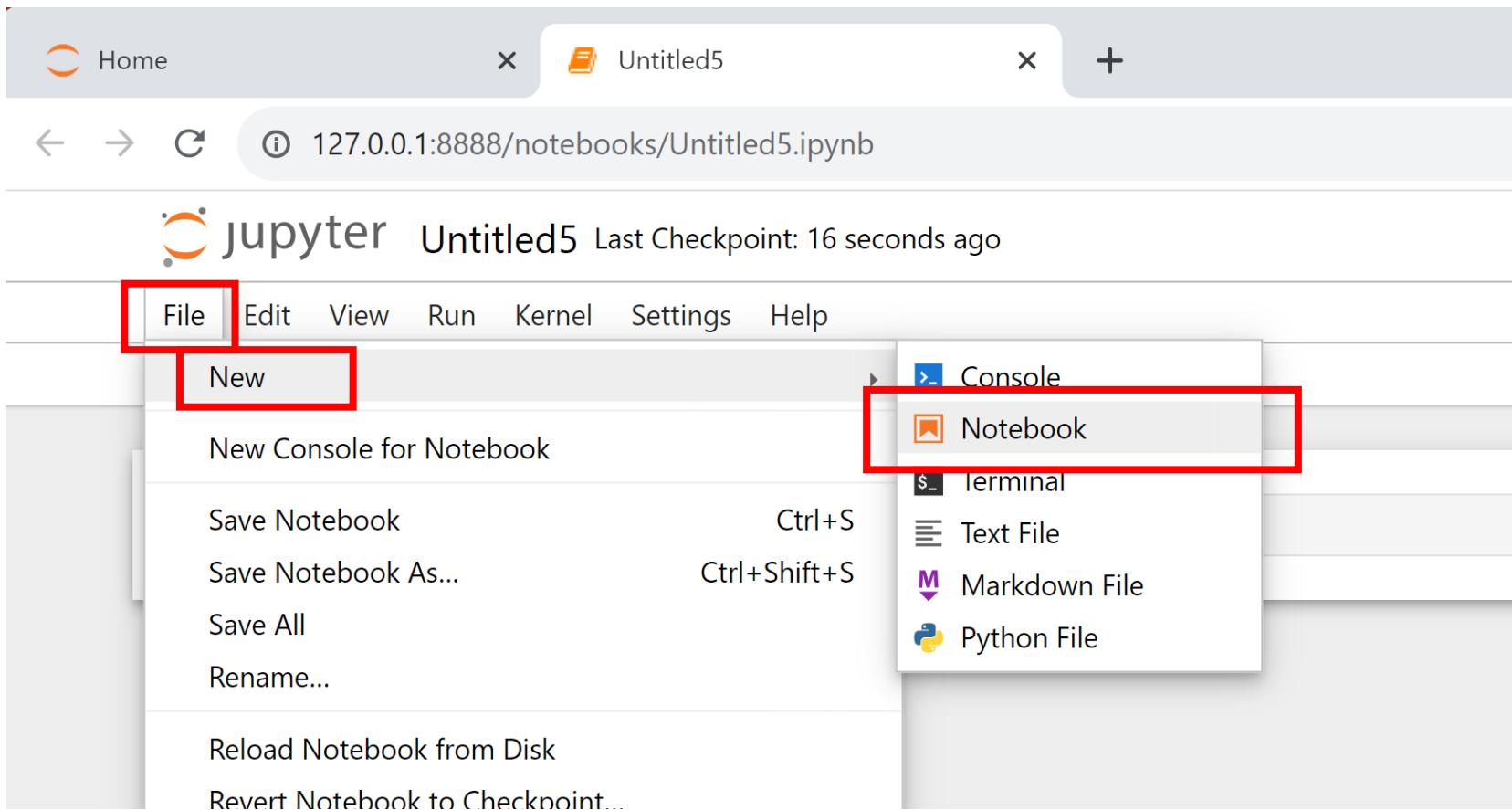
`file:///C:/Users/arnaud/AppData/Roaming/jupyter/runtime/jpserver-636-open.html`

Or copy and paste one of these URLs:

`http://localhost:8888/tree?token=a9195d7e950f42e4e26a7b93bbebf3664d1794b94959c5e3`

`http://127.0.0.1:8888/tree?token=a9195d7e950f42e4e26a7b93bbebf3664d1794b94959c5e3`

Creating a new notebook



Step 3: jupyter needs kernels

← → ⌂ docs.jupyter.org/en/stable/projects/kernels.html



Try Jupyter Usage [Projects](#) Community Contributing More ▾

Home > Projects > Kernels...

Section Navigation

Jupyter User Interfaces

[Kernels \(Programming Languages\)](#)

Education

Execution

Deployment and infrastructure

Formatting and Conversion

IPython

Core Building Blocks

Incubator Projects

Architecture

Project Documentation

Release Notes

Kernels (Programming Languages)

The Jupyter team maintains the [IPython](#) project which is shipped as a default kernel (as [ipykernel](#)) in a number of Jupyter clients. Many other languages, in addition to Python, may be used in the notebook.

The community maintains many other language kernels, and new kernels become available often. Please see the [list of available kernels](#) for additional languages and [kernel installation instructions](#) to begin using these language kernels.

Kernels

Kernels are *programming language specific* processes that run independently and interact with the Jupyter Applications and their user interfaces. [ipykernel](#) is the reference Jupyter kernel built on top of [IPython](#), providing a powerful environment for interactive computing in Python.

Step 3: list of community jupyter-kernels

The screenshot shows a GitHub wiki page titled "Jupyter kernels". The URL in the address bar is github.com/jupyter/jupyter/wiki/Jupyter-kernels. The page has a sidebar with links for Code, Issues (39), Pull requests (2), Discussions, Actions, Projects, Wiki (which is active), Security, and Insights. On the right side, there are buttons for Edit, New page, Pages (10), and a link to the repository's git page (<https://github.com/jupyter/jupyter.wiki.git>). A red box highlights the vertical scroll bar on the right.

Jupyter kernels

Carsten Allefeld edited this page on Sep 11 · 192 revisions

Jupyter kernels

Kernel Zero is [IPython](#), which you can get through [ipykernel](#), and is still a dependency of [jupyter](#). The IPython kernel can be thought of as a reference implementation, as CPython is for Python.

Here is a list of available Jupyter kernels. If you are writing your own kernel, feel free to add it to the table!

Language(s) Version	Name	Jupyter/IPython Version	3rd party dependencies	Example Notebooks
	LFortran			Binder demo
	JupyterQ (KX Official Kernel)	Jupyter	kdb+ ≥ v3.5 64-bit, Python ≥ 3.6, embedPy	Notebook Examples
	Calysto LC3			
	elm-kernel	Jupyter		Examples
	BeakerX		Groovy, Java, Scala, Clojure, Kotlin, SQL	example
multiple	ICalico	IPython >= 2		Index

Jupyter Kernels "*spark*"

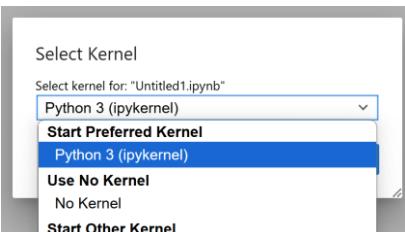
Java 11+, Groovy , Javascript , Kotlin , Scala , Apache Spark , and more	Ganymede	Jupyter >= 4.0	JShell , Apache Maven Resolver	Examples
Pyspark (Python 2 & 3), Spark (Scala), SparkR (R)	sparkmagic	Jupyter >=4.0	Livy	Notebooks , Docker Images
Scala, Python, R	Apache Toree (formerly Spark Kernel)	Jupyter	Spark >= 1.5	Example
Scala>=2.10	almond (old name: Jupyter-scala)	IPython>=3.0		examples
Python >= 3.5, scala >= 2.11	spylon-kernel	ipykernel >=4.5	Apache Spark >=2.0	Example

Jupyter Kernel(s)

to execute
python
`>>> 1+1`



ipykernel (built-in)



to execute
spark python code
`>>> spark.sql("...")`



pyspark ...

to execute
spark scala code
`scala> spark.sql("...")`



toree kernel

does not work on windows
support only spark version 2

to execute
spark scala code
`scala> spark.sql("...")`



spylon kernel
(deprecated?)

to execute
scala (no spark)
`scala> 1+1`



almond kernel

to execute
spark scala code
`scala> spark.sql("...")`



almond kernel
+ module "almond-spark"

Step 3 alternative : Apache Toree

The screenshot shows the Apache Toree website at toree.apache.org/docs/current/user/installation/. The navigation bar includes links for Download, Documentation, Community, GitHub, and Apache. The main content area is titled "Installation" and has a sub-section titled "Setup". It explains that an Apache Spark distribution is required and provides instructions for installing Toree via Pip or Jupyter.

USER

Quick Start

Installation

How it works

Using with Jupyter
Notebooks

Using Standalone

FAQ

Advanced Topics

DEVELOPER

Contributing to the
Project

Creating Extensions

Installation

Setup

An Apache Spark distribution is required to be installed before installing Apache Toree. You can download a copy of Apache Spark [here](#). Throughout the rest of this guide we will assume you have downloaded and extracted the Apache Spark distribution to `/usr/local/bin/apache-spark/`.

Installing Toree via Pip

The quickest way to install Apache Toree is through the `toree` pip package.

```
pip install toree
```

This will install a jupyter application called `toree`, which can be used to install and configure different Apache Toree kernels.

```
jupyter toree install --spark_home=/usr/local/bin/apache-spark/
```

You can confirm the installation by verifying the `apache_toree_scala` kernel is listed in the following command:

```
jupyter kernelspec list
```

Install Toree

```
c:> pip install toree
```

```
c:> jupyter toree install --spark_home=%SPARK_HOME%
```

Toree

```
C:\apps\cmdr
λ pip install toree
Collecting toree
  Downloading toree-0.5.0.tar.gz (24.3 MB)
    ----- 24.3/24.3 MB 6.7 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: jupyter-core>=4.0 in c:\users\arnaud\appdata\local\python311\lib\site-packages (from toree) (5.3.1)
Requirement already satisfied: jupyter-client>=4.0 in c:\users\arnaud\appdata\local\python311\lib\site-pac C:\apps\cmdr
Requirement already satisfied: jupyter toree install --spark_home=%SPARK_HOME%
python311\lib\site packages [ToreeInstall] Installing Apache Toree version 0.5.0
[ToreeInstall]
Apache Toree is an effort undergoing incubation at the Apache Software Foundation (ASF), sponsored by the Apache Incubator PMC.

Incubation is required of all newly accepted projects until a further review indicates that the infrastructure, communications, and decision making process have stabilized in a manner consistent with other successful ASF projects.

While incubation status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF.
[ToreeInstall] Creating kernel Scala
[ToreeInstall] Installed kernelspec apache_toree_scala in C:\ProgramData\jupyter\kernels\apache_toree_scala
```

Check Toree install

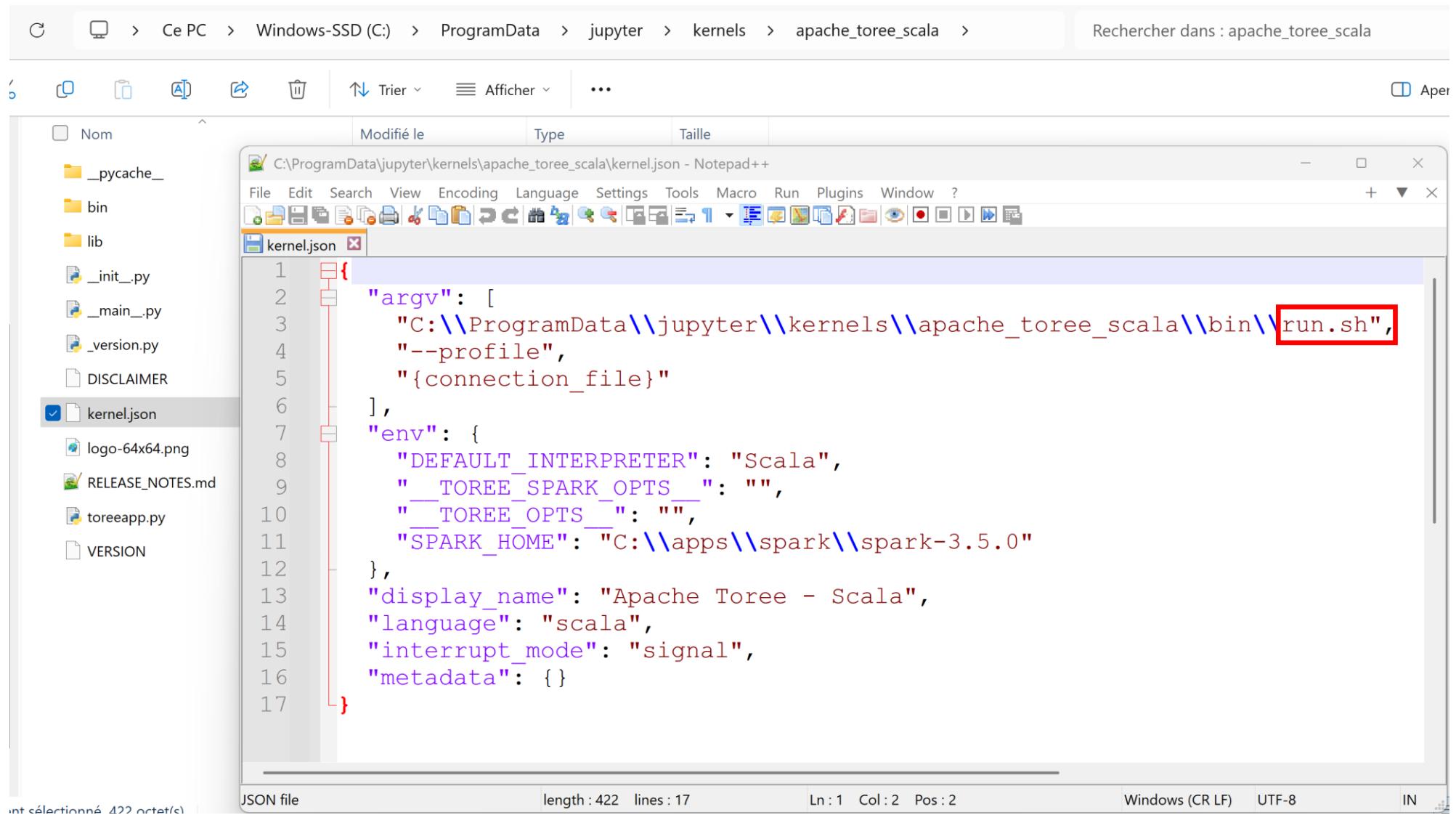
to check:

c:> **jupyter kernelspec list**

```
C:\apps\cmdr
λ jupyter kernelspec list
0.00s - Debugger warning: It seems that frozen modules are being used, which may
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off
0.00s - to python to disable frozen modules.
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation.

Available kernels:
  spylon-kernel          C:\Users\arnaud\AppData\Roaming\jupyter\kernels\spylon-kernel
  python3                 C:\Users\arnaud\AppData\Local\Programs\Python\Python311\share\jupyter\k
  ernel
  python3
  apache_toree_scala      C:\ProgramData\jupyter\kernels\apache_toree_scala
```

Toree for windows?? "run.sh" => "run.cmd"



Toree run.cmd

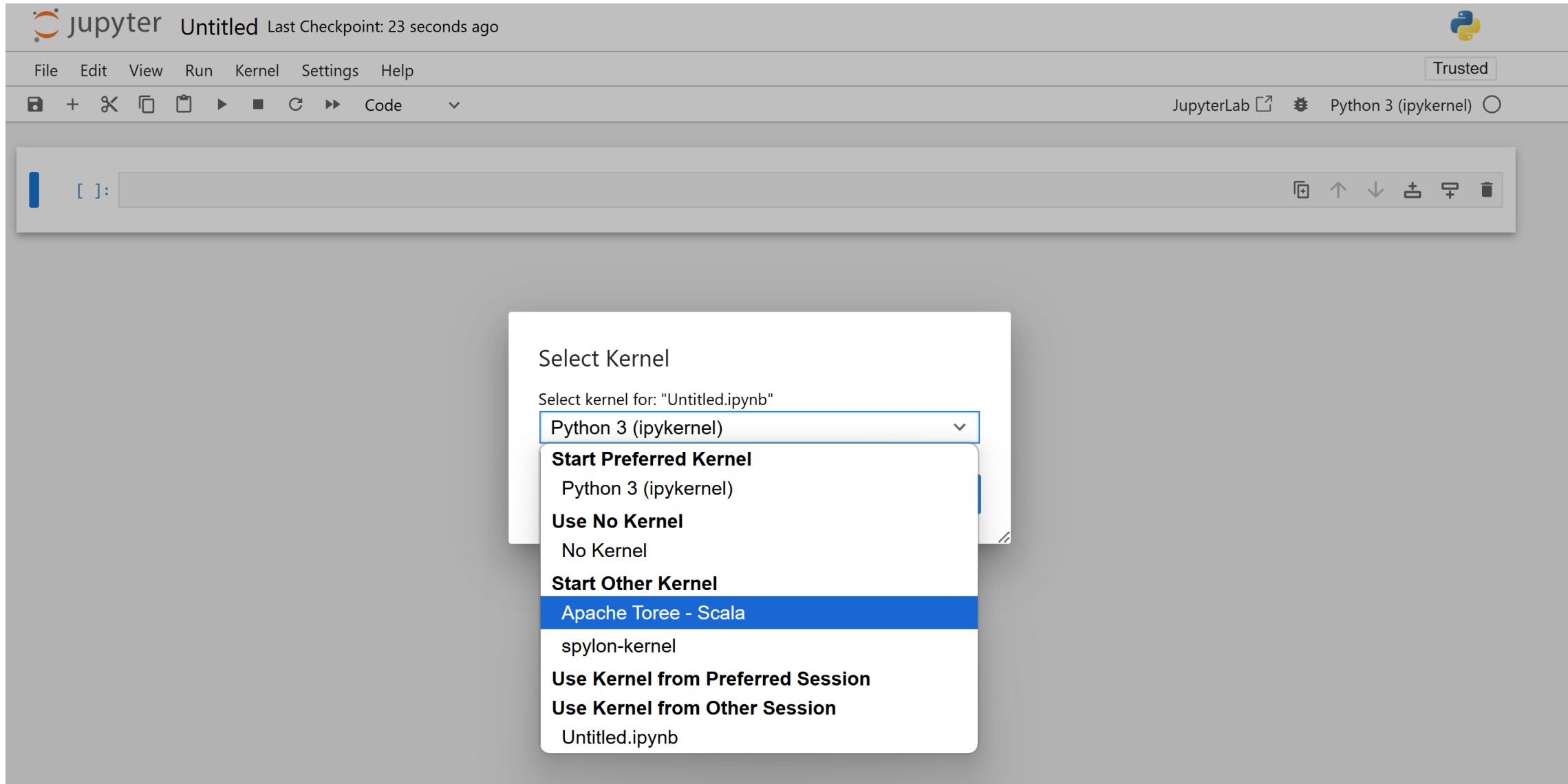
```
echo "... executing Toree/run.cmd"
echo

set PROG_HOME=C:\ProgramData\jupyter\kernels\apache_toree_scala
set TOREE_ASSEMBLY=%PROG_HOME%\lib\toree-assembly-0.5.0-incubating.jar
set PYTHONHASHSEED=0
set _JAVA_OPTIONS=-Dscala.usejavacp=true
if "%SPARK_OPTS%"=="" set SPARK_OPTS=%__TOREE_SPARK_OPTS__%
if "%TOREE_OPTS%"=="" set TOREE_OPTS=%__TOREE_OPTS__%

echo JAVA_HOME: %JAVA_HOME%
echo SPARK_HOME: %SPARK_HOME%
echo HADOOP_HOME: %HADOOP_HOME%
echo TOREE_ASSEMBLY: %TOREE_ASSEMBLY%
echo ... "%SPARK_HOME%\bin\spark-submit" --name "Apache-Toree" ^
    %SPARK_OPTS% --class org.apache.toree.Main "%TOREE_ASSEMBLY%" %TOREE_OPTS% %

"%SPARK_HOME%\bin\spark-submit" --name "Apache-Toree" ^
    %SPARK_OPTS% --class org.apache.toree.Main "%TOREE_ASSEMBLY%" %TOREE_OPTS% 
```

new Notebook > change Kernel



ERROR ... as of version 0.5, Toree support
only Spark 2 !!
for spark 3.5 with scala 2.13
=> need to patch + recompile yourself !!

have to wait (or contribute more)
to this PullRequest

[TOREE-557] Bump Spark 3.5 #224

Draft pan3793 wants to merge 2 commits into apache:master from pan3793:spark-3.5

Conversation 0 Commits 2 Checks 4 Files changed 96

Commits on Sep 30, 2024

[TOREE-556] Support Scala 2.13

[TOREE-557] Bump Spark 3.5

Step 3 alternative : almond kernel
+ "almound-spark" module

for using spark with scala langage

http://almond.sh



A screenshot of a web browser window displaying the almond.sh homepage. The address bar shows the URL. The page has a dark header with the almond logo and version 0.14.0-RC15. Navigation links for Docs, Blog, GitHub, and Search are also present. The main content area features a large, smiling cartoon almond character.



almond

A Scala kernel for Jupyter

[TRY IT ONLINE](#)

[TRY IT WITH DOCKER](#)

[INSTALL](#)

Install Almond

TODO ... WORK IN PROGRESS

Step 3 alternative : spylon kernel
for spark + scala

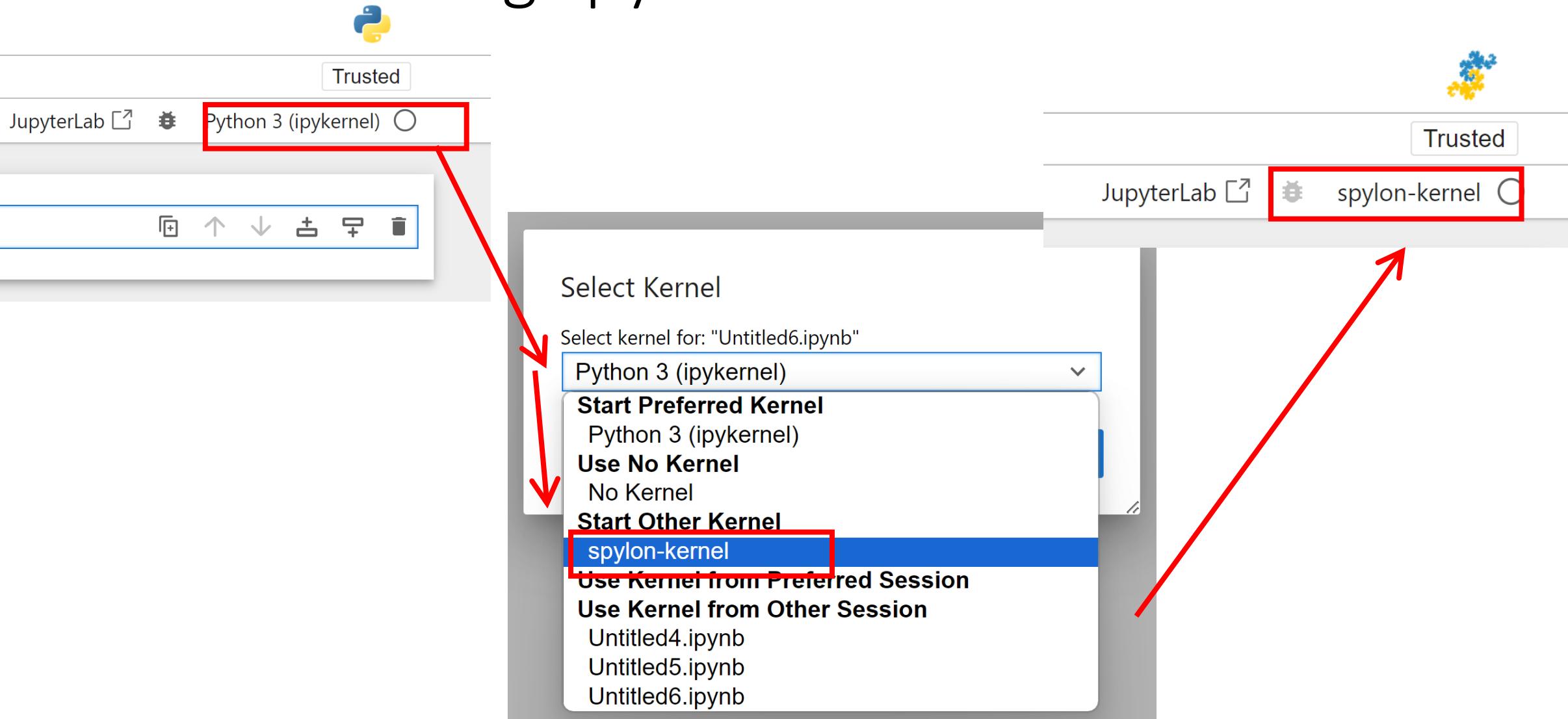
Step 3 alternative : install jupyter spylon_kernel
(for using Spark with Scala langage)

python -m spylon_kernel install --user

(or pip install spylon_kernel --user)

```
C:\Users\arnaud>python -m spylon_kernel install --user
0.00s - Debugger warning: It seems that frozen modules are being used, which may
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off
0.00s - to python to disable frozen modules.
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation.
[InstallKernelSpec] Installed kernelspec spylon-kernel in C:\Users\arnaud\AppData\Roaming\jupyter\kernels\spylon-kernel
```

Step 3 ... testing spylon



Step 3 ... Testing Spylon .. Write Scala

The screenshot shows a Jupyter Notebook interface with two code cells. The top cell contains the Scala code: `for(i <- 0 to 5) println(s"Scala code.. ${i}")`. A red arrow points from the text "Get Spark process (Scala interpreter)" to the output of this cell, which is "Initializing Scala interpreter ...". Another red arrow points from the output of the first cell to the second cell's output. The second cell also contains the same Scala code. Its output includes "Initializing Scala interpreter ...", "Spark Web UI available at <http://DesktopArnaud:4041>", "SparkContext available as 'sc' (version = 3.4.1, master = local[*], app id = local-1694035899154)", "SparkSession available as 'spark'", and a list of numbers from 0 to 5, each preceded by "Scala code..". The interface includes standard Jupyter controls like play/pause, stop, and run, along with a "Code" dropdown menu and a status bar message: "Run this cell and advance (Shift+Enter)".

Get Spark process
(Scala interpreter)

```
[1]: for(i <- 0 to 5) println(s"Scala code.. ${i}")
```

Initializing Scala interpreter ...

```
[1]: for(i <- 0 to 5) println(s"Scala code.. ${i}")
```

Initializing Scala interpreter ...
Spark Web UI available at <http://DesktopArnaud:4041>
SparkContext available as 'sc' (version = 3.4.1, master = local[*], app id = local-1694035899154)
SparkSession available as 'spark'
Scala code.. 0
Scala code.. 1
Scala code.. 2
Scala code.. 3
Scala code.. 4
Scala code.. 5

Run this cell and advance (Shift+Enter)

Type SCALA code
... Shift+Enter

Step 3 ... Testing Spylon .. Write Scala

← → ⌂ ⓘ 127.0.0.1:8888/notebooks/Untitled4.ipynb

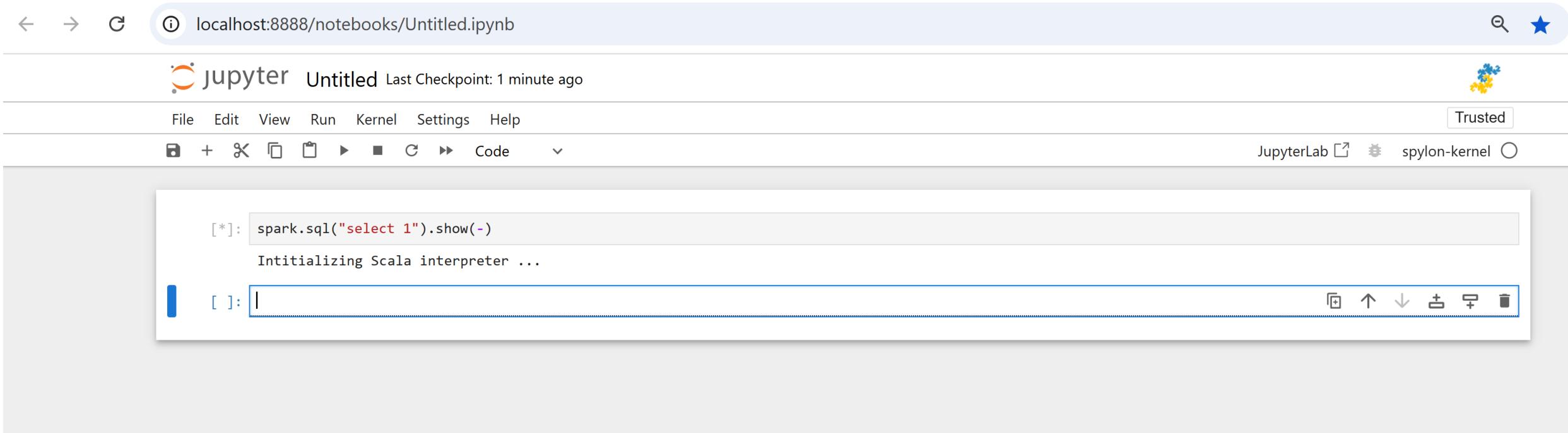
jupyter Untitled4 Last Checkpoint: 1 minute ago

File Edit View Run Kernel Settings Help Trusted JupyterLab JupyterLab spylon-kernel

```
[1]: val ls = Seq( (1, true, "Hey") )  
  
Initializing Scala interpreter ...  
Spark Web UI available at http://DesktopArnaud:4040  
SparkContext available as 'sc' (version = 3.4.1, master = local[*], app id = local-1694034599241)  
SparkSession available as 'spark'  
[1]: ls: Seq[(Int, Boolean, String)] = List((1,true,Hey))  
  
[2]: ls  
  
[2]: res0: Seq[(Int, Boolean, String)] = List((1,true,Hey))  
  
[3]: val ds = spark.createDataset(ls)  
  
[3]: ds: org.apache.spark.sql.Dataset[(Int, Boolean, String)] = [_1: int, _2: boolean ... 1 more field]  
  
[4]: ds.show  
+---+---+---+  
| _1| _2| _3|  
+---+---+---+  
| 1|true|Hey|  
+---+---+---+
```

[]:

Spark SQL "Hello World"



A screenshot of a Jupyter Notebook interface. The title bar shows the URL `localhost:8888/notebooks/Untitled.ipynb`. The header includes the Jupyter logo, the notebook name `Untitled`, a timestamp `Last Checkpoint: 1 minute ago`, a gear icon, and a `Trusted` badge. The toolbar below has buttons for File, Edit, View, Run, Kernel, Settings, Help, and various cell type icons. On the right, there are links to `JupyterLab` and `spylon-kernel`. The main area contains a code cell with the command `spark.sql("select 1").show()` and its output, which includes the message `Intitializing Scala interpreter ...`. A new cell is being created at the bottom with the prompt `[]:`.

may take 1 minute to start ... spark is starting

TroubleShooting ???

```
Cmder
eco
    return f(*a, **kw)
    ^^^^^^^^^^

    File "C:\apps\spark\spark-3.5.0\python\lib\py4j-0.10.9.7-src.zip\py4j\protocol.py", line 330,
in get_return_value
    raise Py4JError(
py4j.protocol.Py4JError: An error occurred while calling None.scala.tools.nsc.interpreter.IMain
. Trace:
py4j.Py4JException: Constructor scala.tools.nsc.interpreter.IMain([class scala.tools.nsc.Settings, class java.io.PrintWriter]) does not exist
    at py4j.reflection.ReflectionEngine.getConstructor(ReflectionEngine.java:180)
    at py4j.reflection.ReflectionEngine.getConstructor(ReflectionEngine.java:197)
    at py4j.Gateway.invoke(Gateway.java:237)
    at py4j.commands.ConstructorCommand.invokeConstructor(ConstructorCommand.java:80)
    at py4j.commands.ConstructorCommand.execute(ConstructorCommand.java:69)
    at py4j.ClientServerConnection.waitForCommands(ClientServerConnection.java:182)
    at py4j.ClientServerConnection.run(ClientServerConnection.java:106)
    at java.base/java.lang.Thread.run(Thread.java:1589)

[I 2024-10-26 09:35:00.917 ServerApp] Saving file at /Untitled.ipynb
```

Spylon ERROR for spark 3.5

The screenshot shows a GitHub issue page for the repository 'vericast/spylon-kernel'. The URL in the address bar is github.com/vericast/spylon-kernel/issues/72. The page title is 'Failed for spark 3.5 (scala 2.13) #72'. The issue is marked as 'Open' by Arnaud-Nauwynck, who also opened it and has 0 comments. A comment from Arnaud-Nauwynck states: 'Running with Spark 3.5 + scala 2.13 + java 20 failed. spylon need to be upgraded and recompiled for scala 2.13 ?' followed by a detailed stack trace:

```
py4j.protocol.Py4JError: An error occurred while calling None.scala.tools.nsc.interpreter.IMain. Trace:  
py4j.Py4JException: Constructor scala.tools.nsc.interpreter.IMain([class scala.tools.nsc.Settings, class java.io.PrintWriter]) does not exist  
at py4j.reflection.ReflectionEngine.getConstructor(ReflectionEngine.java:180)  
at py4j.reflection.ReflectionEngine.getConstructor(ReflectionEngine.java:197)  
at py4j.Gateway.invoke(Gateway.java:237)  
at py4j.commands.ConstructorCommand.invokeConstructor(ConstructorCommand.java:80)  
at py4j.commands.ConstructorCommand.execute(ConstructorCommand.java:69)  
at py4j.ClientServerConnection.waitForCommands(ClientServerConnection.java:182)  
at py4j.ClientServerConnection.run(ClientServerConnection.java:106)  
at java.base/java.lang.Thread.run(Thread.java:1623)
```

Step 3 alternative : kernel for pyspark
(for using PySpark with Python langage)

TODO ... WORK IN PROGRESS