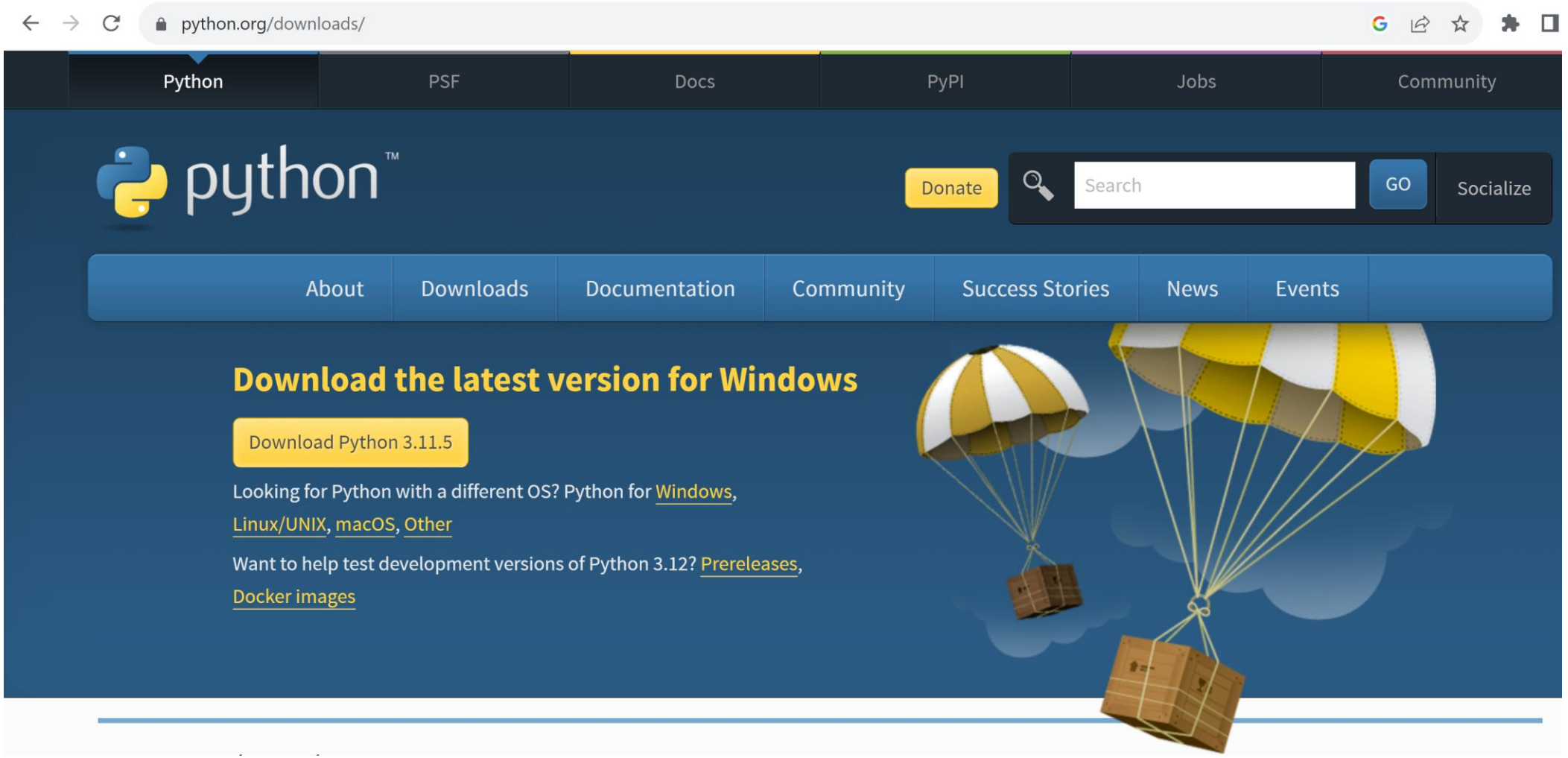
# Installing Jupyter Notebook for Spark

Arnaud.nauwynck@gmail.com

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



# Step 2 : install jupyter

Choose class Jupyter Notebook… OK

## 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
```

## Jupyter Notebook

Install the classic Jupyter Notebook with:

```
pip install notebook
```

To run the notebook:

```
jupyter notebook
```

# Star Jupyter Notebook

**jupyter notebook**

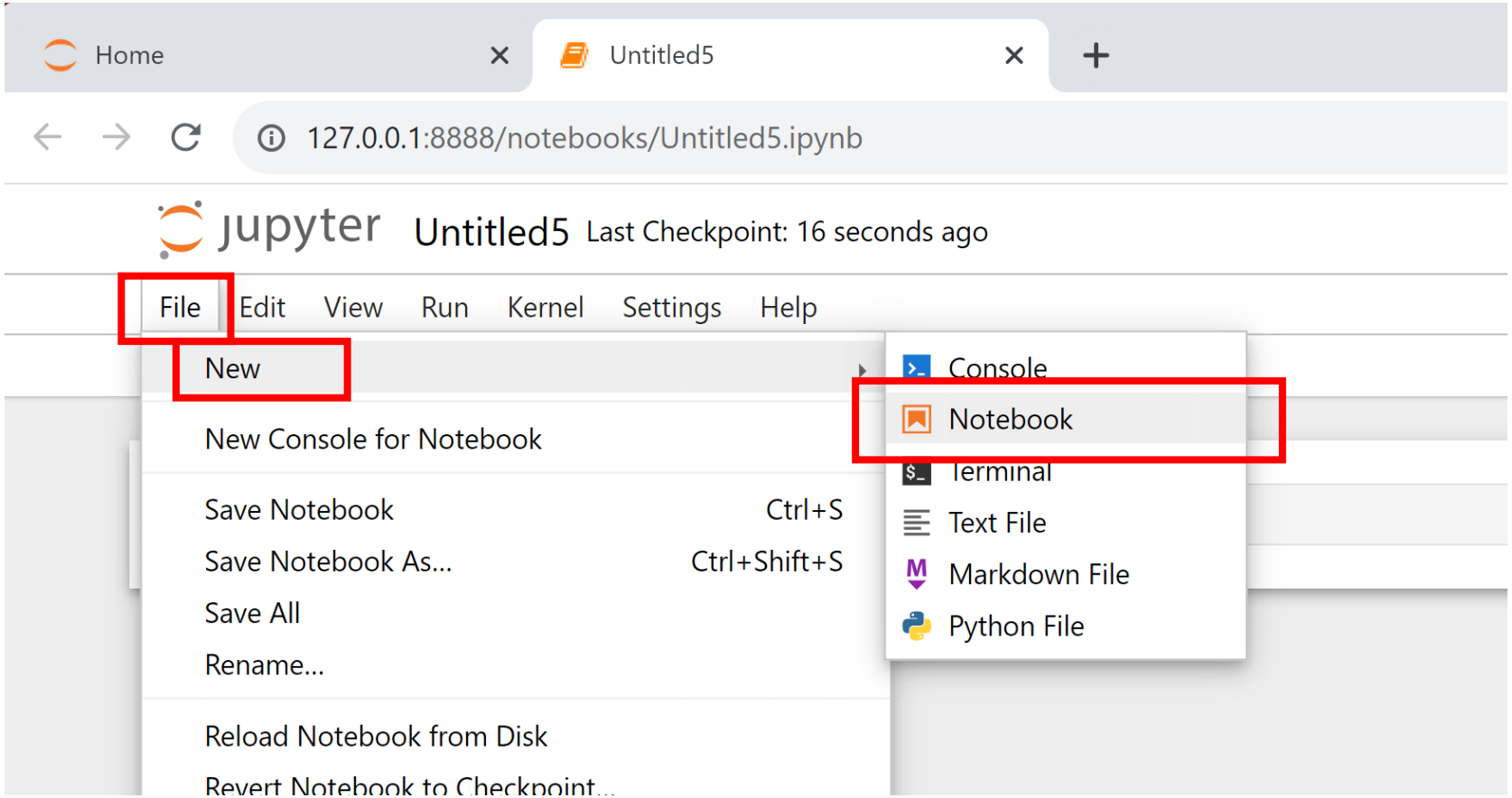
```
C:\Users\arnaud>jupyter notebook
[I 2023-09-06 23:09:00.357 ServerApp] Package notebook took 0.0000s to import
[I 2023-09-06 23:09:00.482 ServerApp] Package jupyter_lsp took 0.1187s to import
[W 2023-09-06 23:09:00.482 ServerApp] A `_jupyter_server_extension_points` function was not found in jupyter_lsp.
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?? ]]

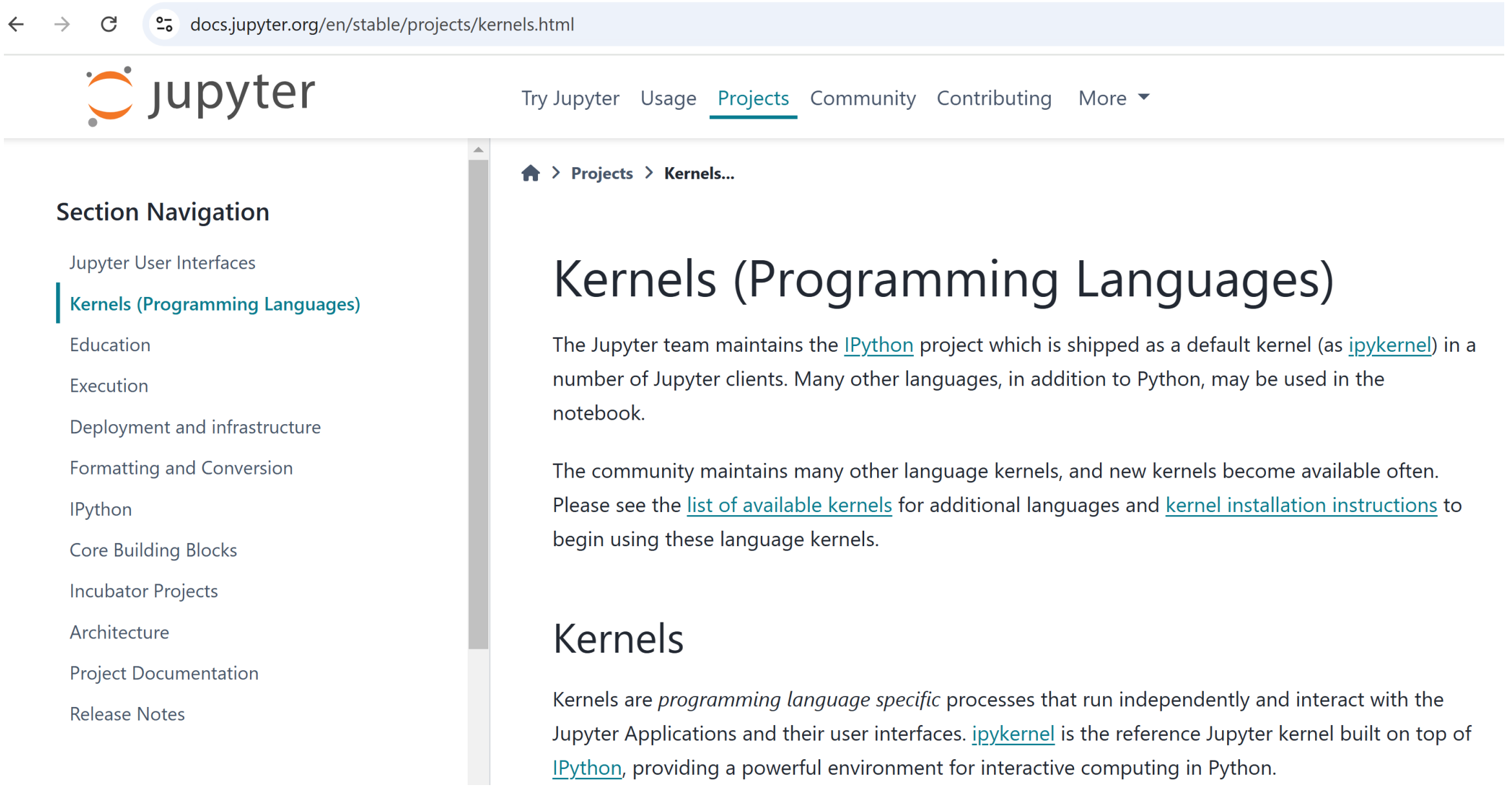
# Jupyter Notebook ..

```
[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=a9195d7e950f42e4e26a7b93bbebf3664d1794b9495
9c5e3
[I 2023-09-06 23:09:01.911 ServerApp]     http://127.0.0.1:8888/tree?token=a9195d7e950f42e4e26a7b93bbebf3664d1794b
94959c5e3
[I 2023-09-06 23:09:01.911 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip c
onfirmation).
[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
```

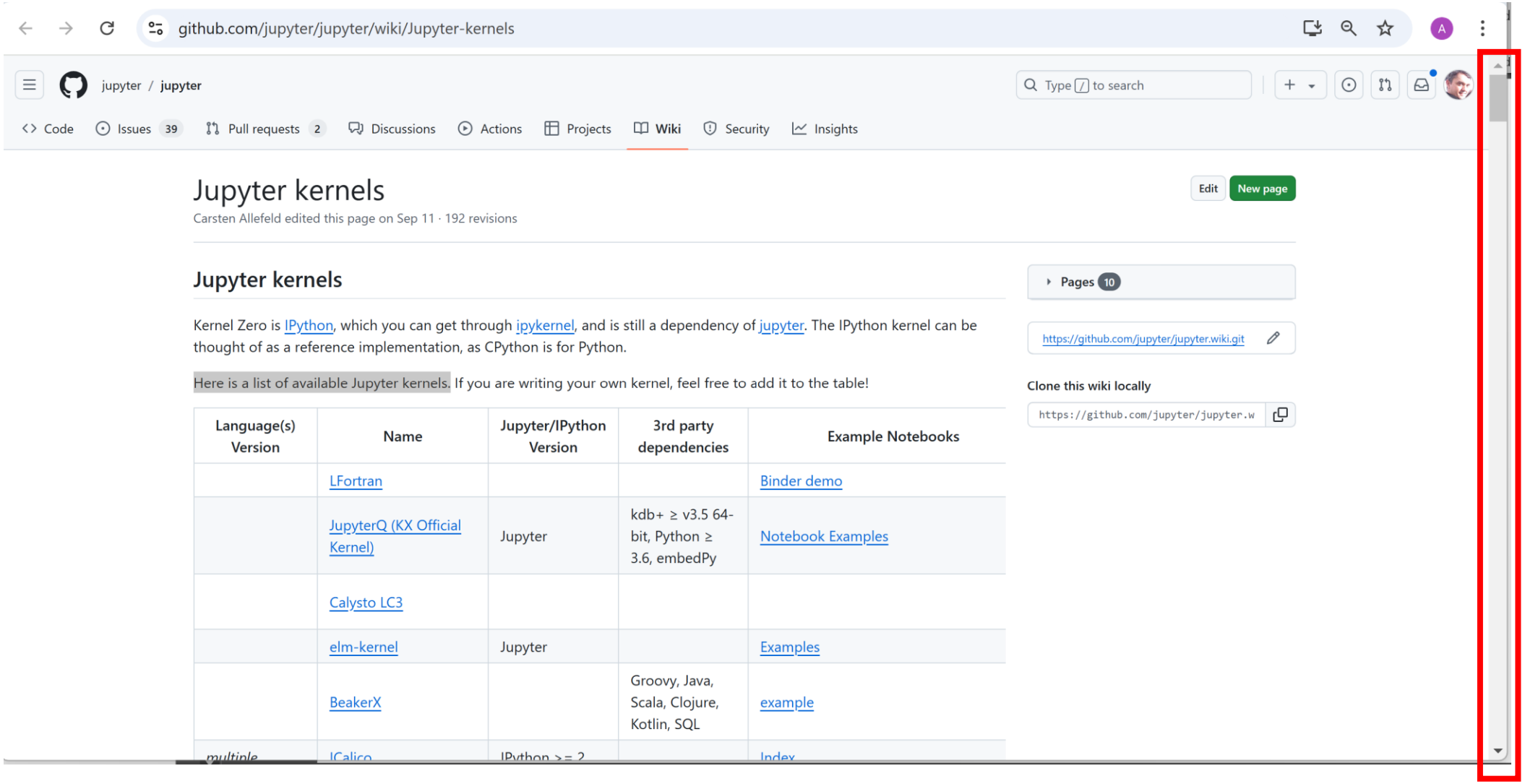
# Creating a new notebook



# Step 3: jupyter needs kernels



# Step 3: list of community jupyter-kernels



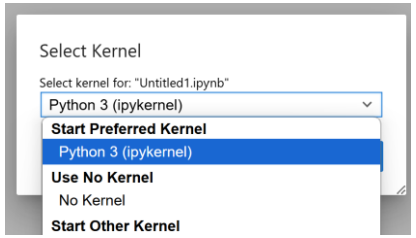
# Jupyter Kernels "*spark*"

| | | | | |
|---|---|---|---|---|
| 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?)

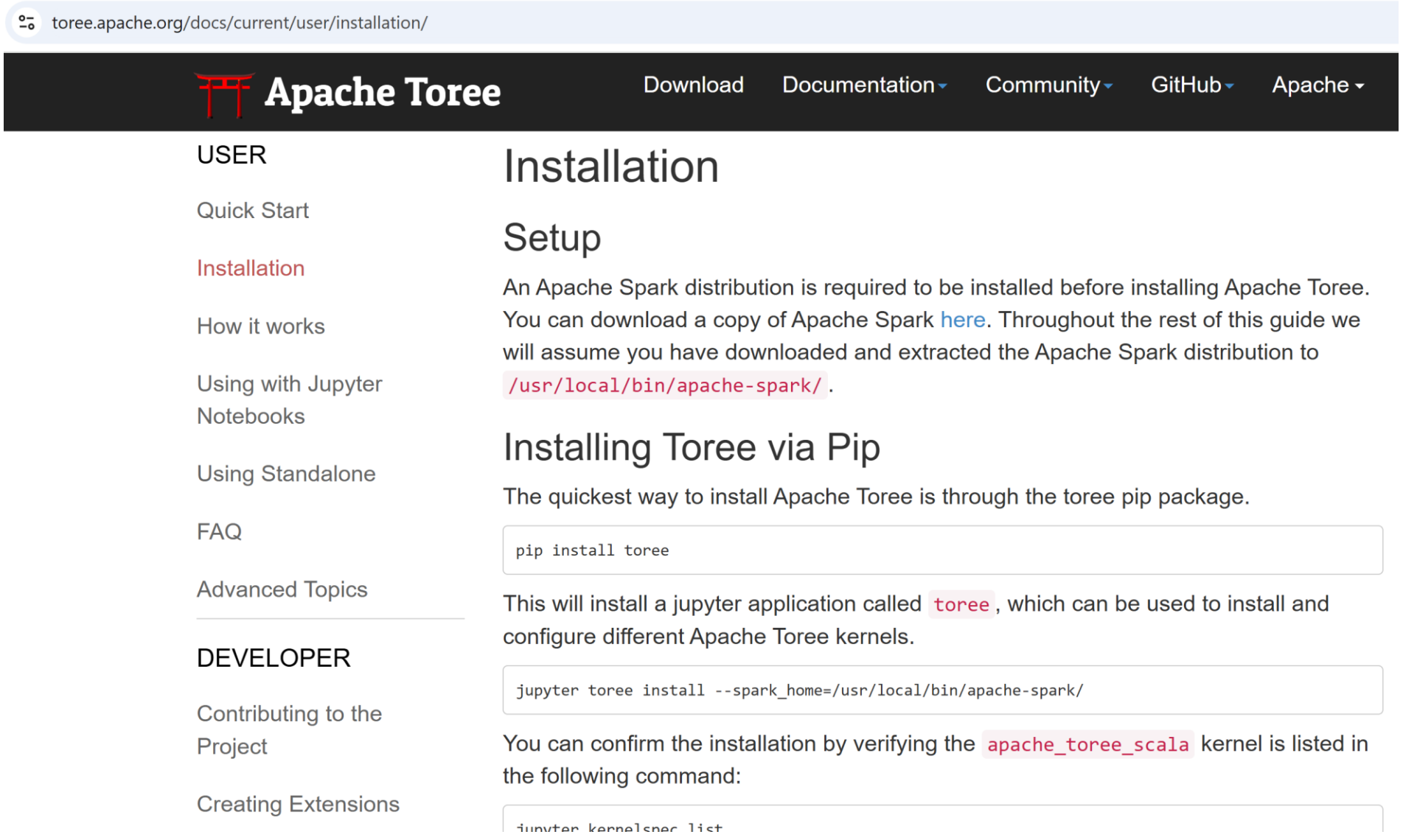
to execute
**scala** (no spark)
scala> 1+1

**almond** kernel

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

**almond** kernel
+ module "**almound-spark**"

# Step 3 alternative : Apache Toree



# Install Toree

c:> **pip install toree**

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

# Toree

```
C:\apps\cmder
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\loca
on\python311\lib\site-packages (from toree) (5.3.1)
Requirement already satisfied: jupyter-client>=4.0 in c:\users\arnaud\appdata\lo
thon\python311\lib\site-pac
Requirement already satisfi
```

```
C:\apps\cmder
λ jupyter toree install --spark_home=%SPARK_HOME%
[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
```

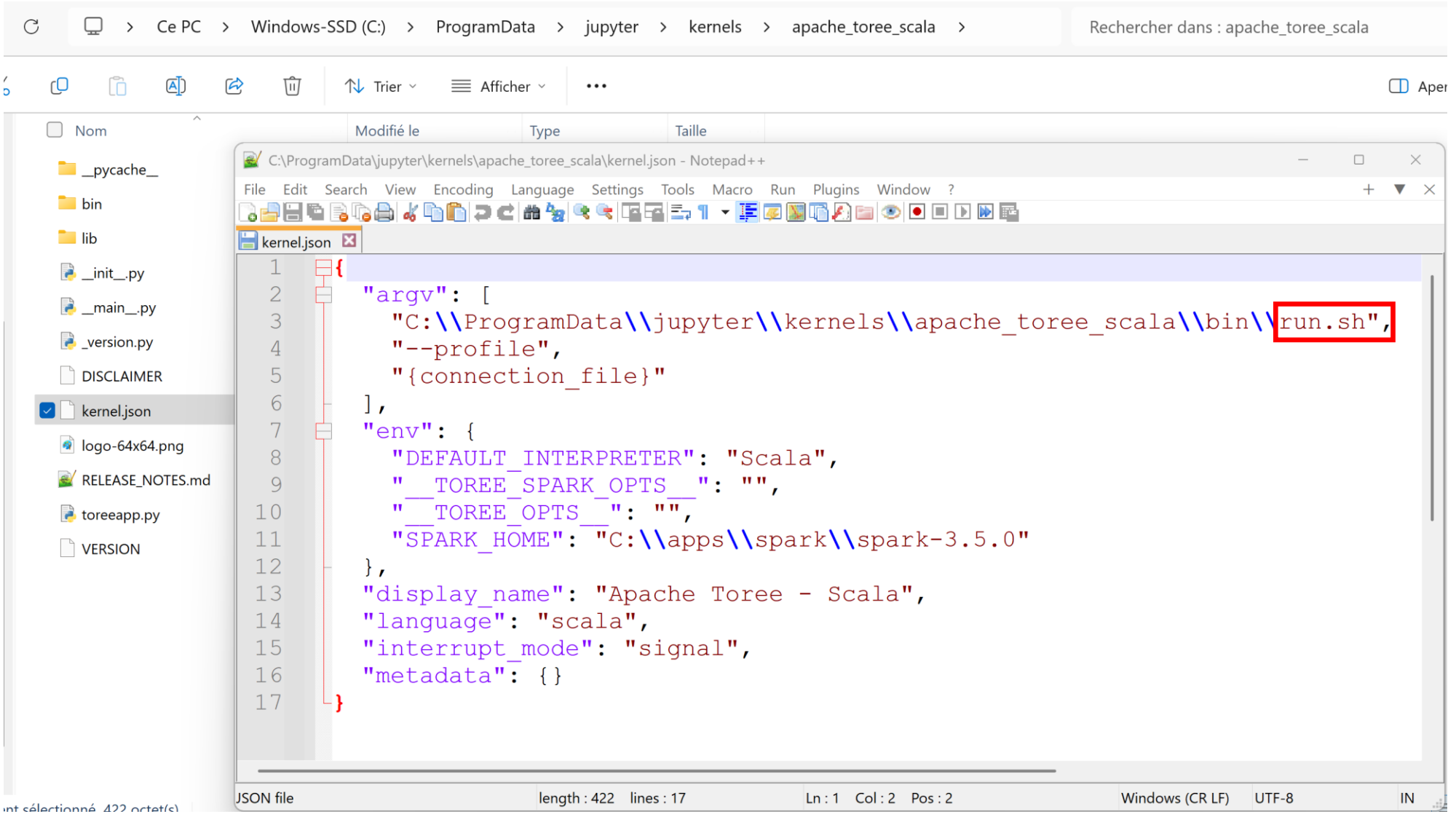
# Check Toree install

to check:

c:> **jupyter kernelspec list**

```
C:\apps\cmder
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 vali
dation.
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
ernels\python3
  apache_toree_scala     C:\ProgramData\jupyter\kernels\apache_toree_scala
```

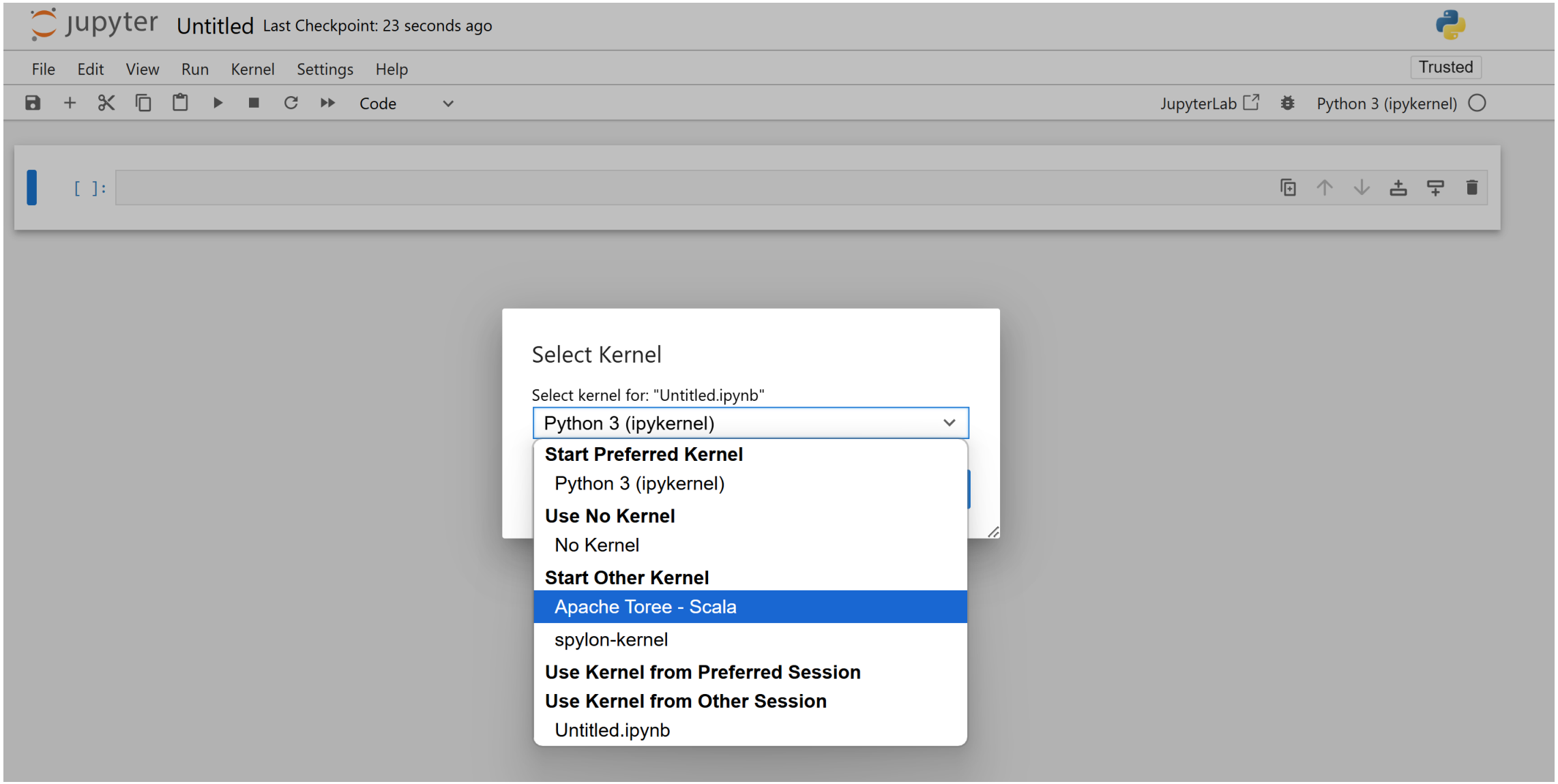
# Toree for windows??  "run.sh" => "run.cmd"



# Toree run.cmd

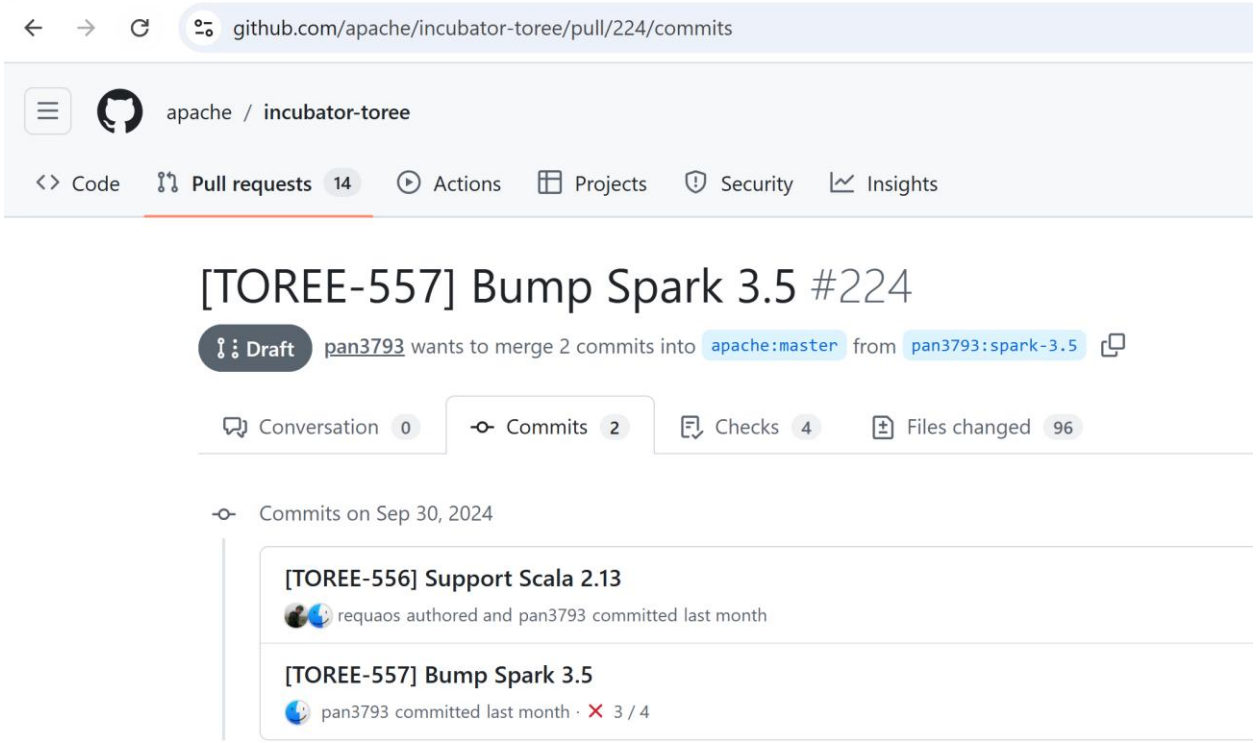
```
echo "... executing Toree/run.cmd"
echo

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 !!

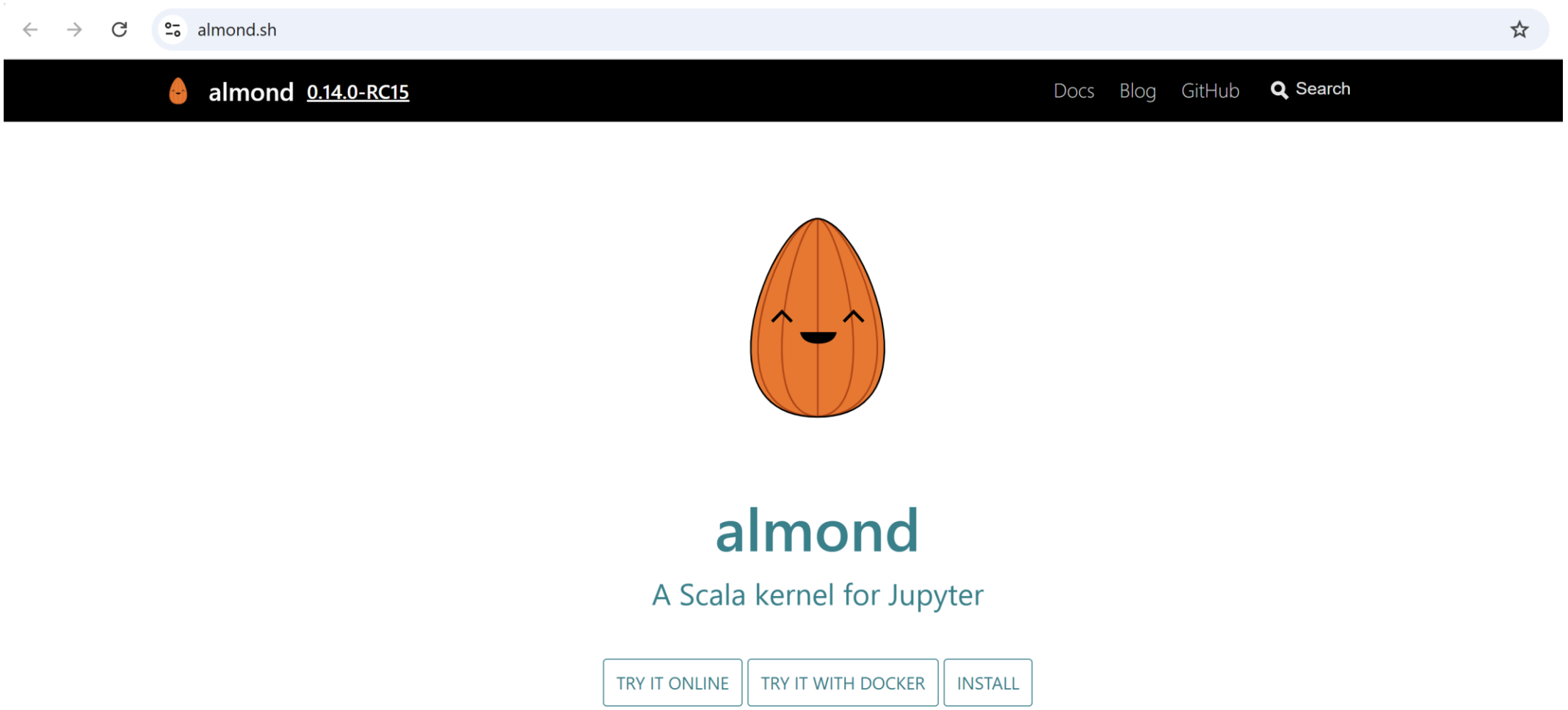
have to wait (or contribute more)
to this PullRequest



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

for using spark with scala langage

# http://almond.sh



# Install Almond

# TODO ... WORK IN PROGRESS

# Step 3 alternative : spylon kernel for spark + scala

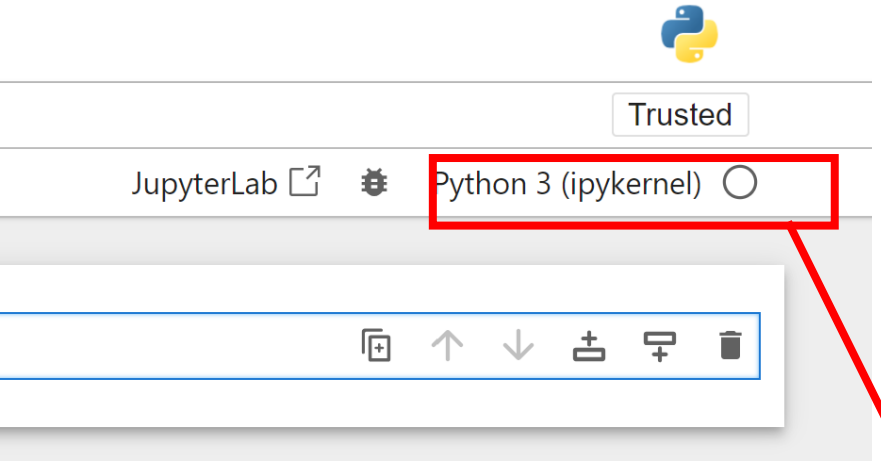
# Step 3 alternative : install jupyter spylon_kernel (for using Spark with Scala langage)

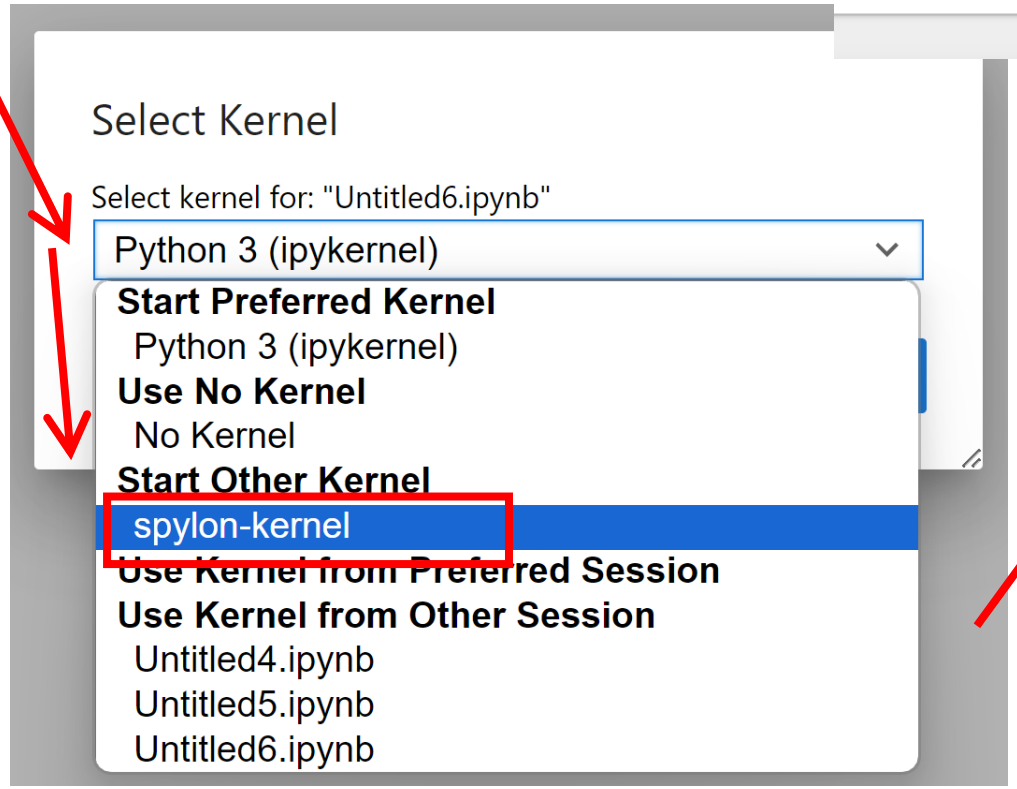
**python –m spylon_kernel install --user**

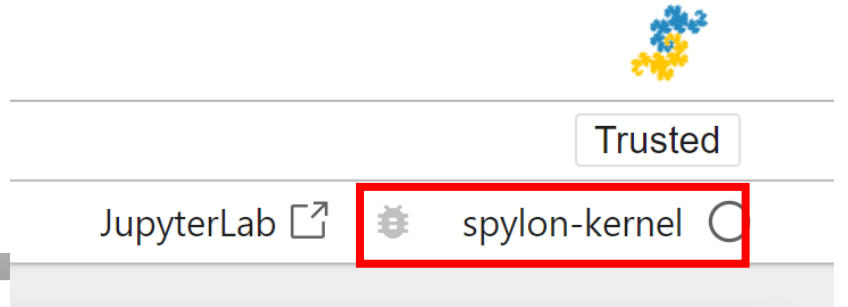
(or  pip install spylon_kernel –user  )

```
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-k
ernel
```

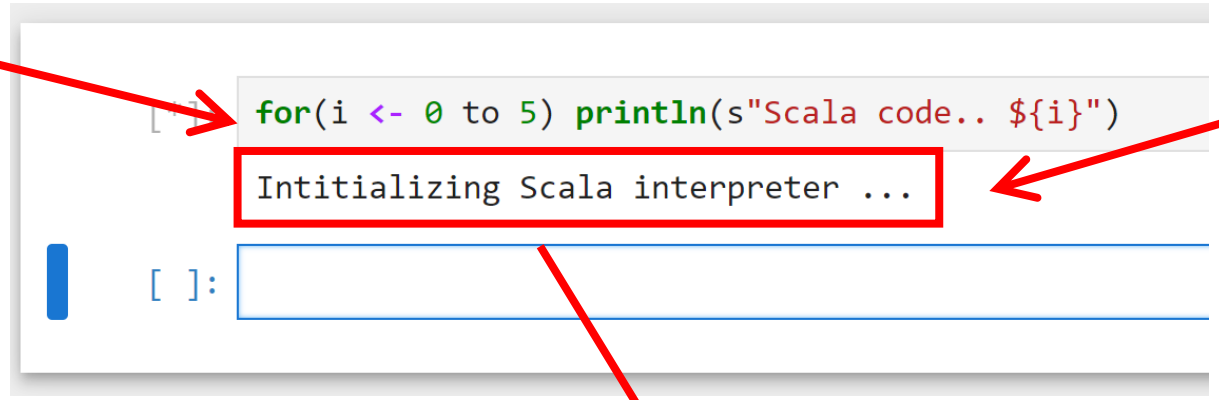
# Step 3 … testing spylon

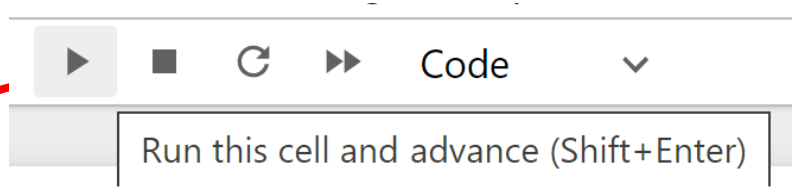


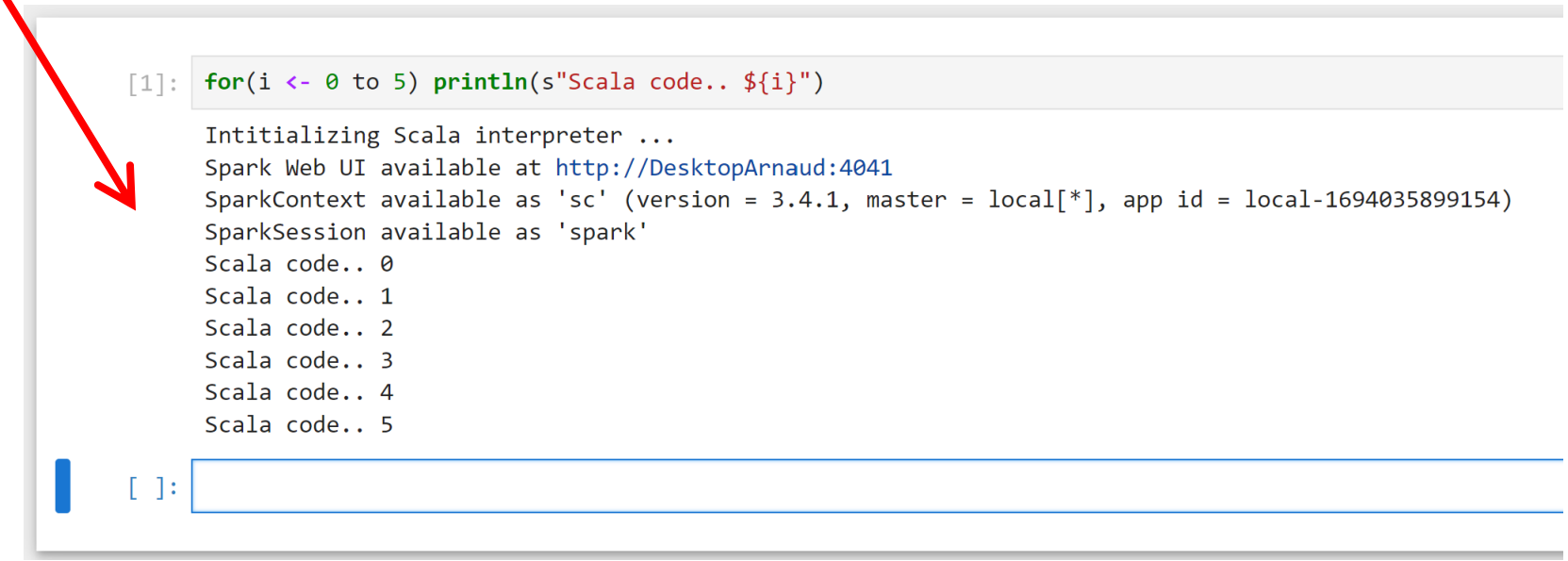
# Step 3 … Testing Spylon .. Write Scala



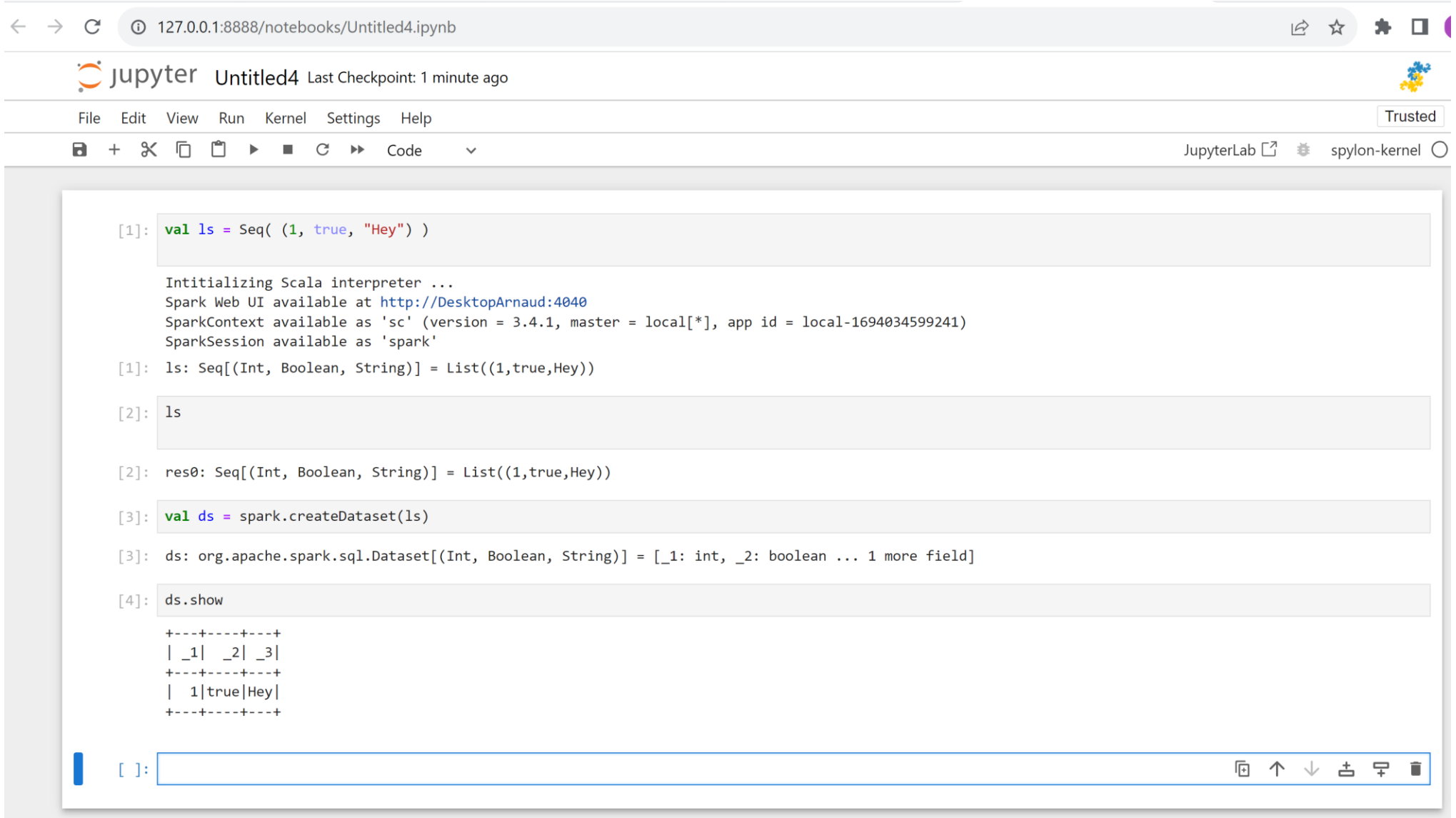




Type SCALA code
… Shift+Enter

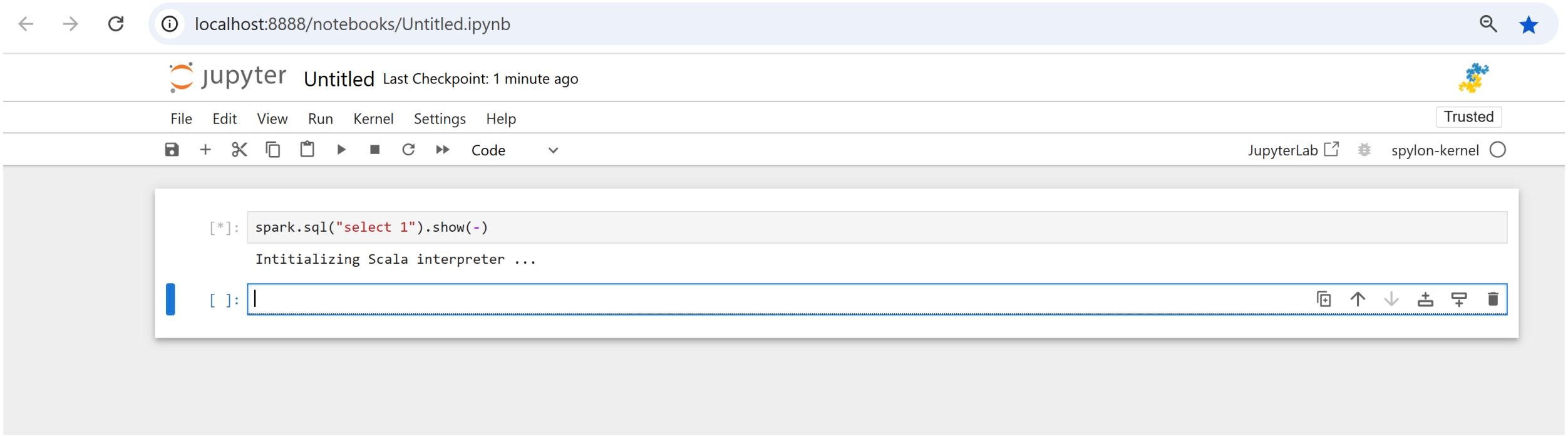
Get Spark process
(Scala interpreter)



# Step 3 … Testing Spylon .. Write Scala

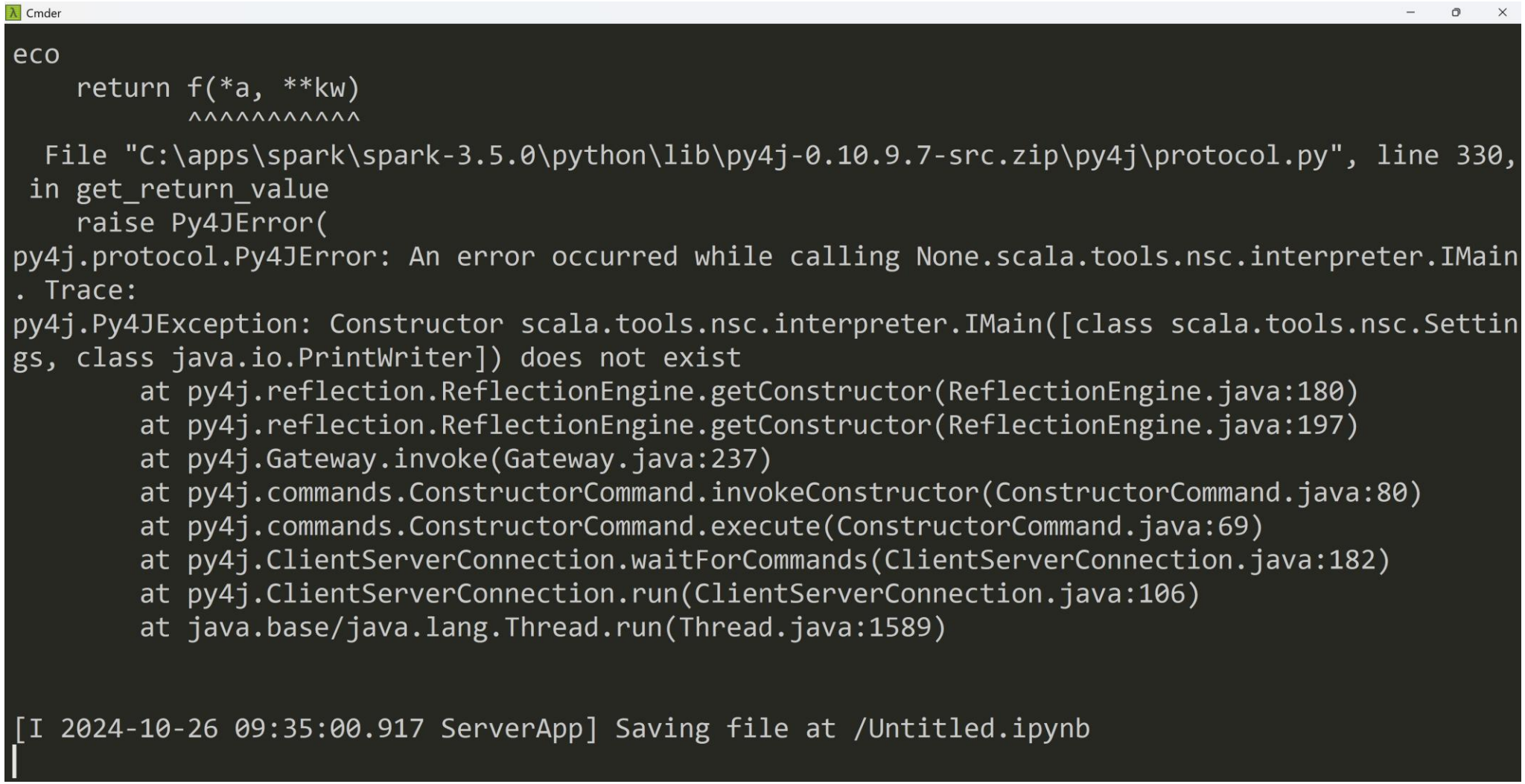


# Spark SQL "Hello World"

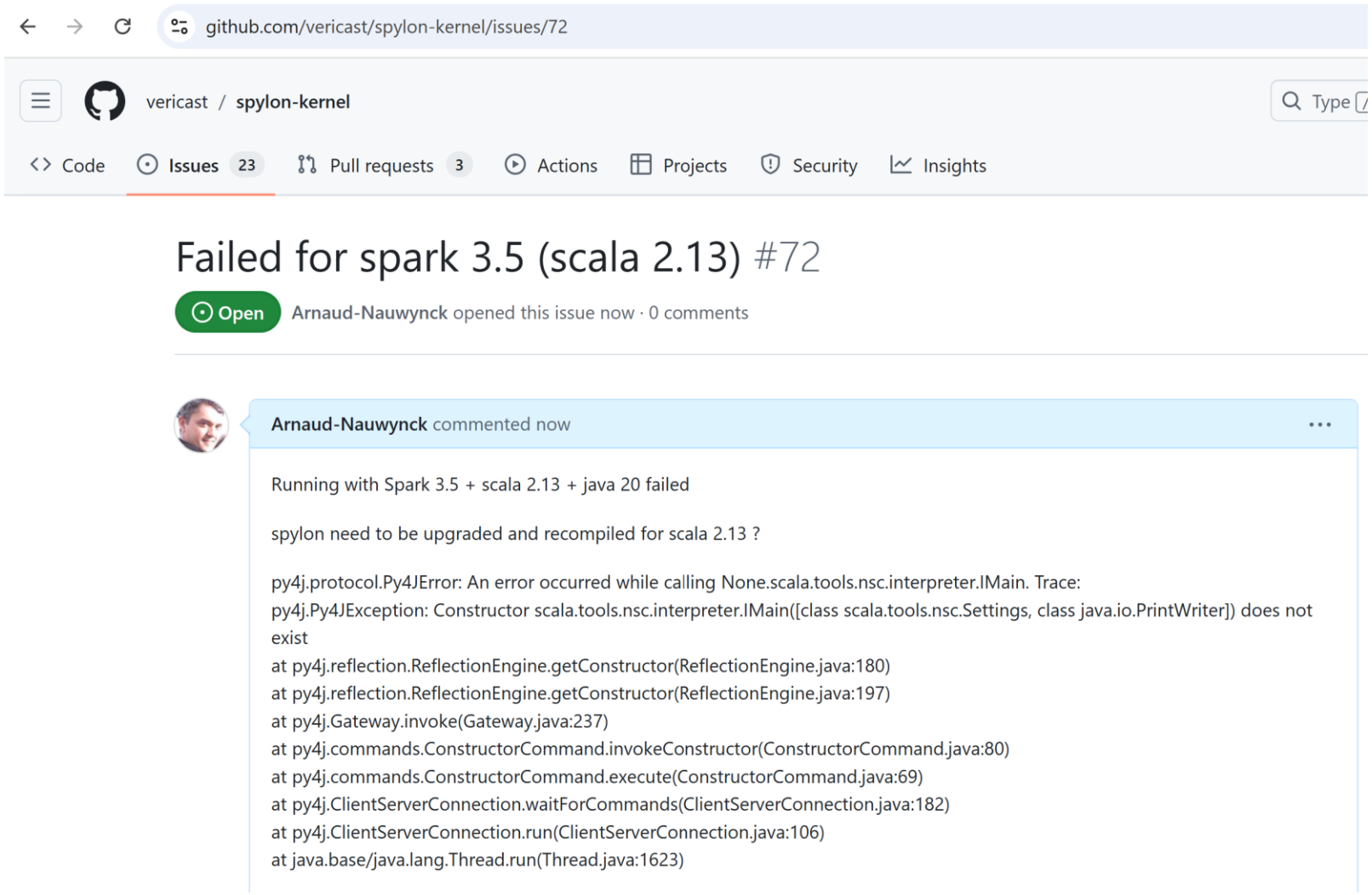



may take 1 minute to start ... spark is starting

# TroubleShooting ???



# Spylon ERROR for spark 3.5



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

# TODO ... WORK IN PROGRESS