Install Apache Spark + PySpark in Jupyter (Windows & Mac)

---

1) install

- **Java JDK** (8 or 11 recommended for Spark 3.5.x)
- **Python 3.8+** with
- **Apache Spark 3.5.x** (prebuilt for Hadoop 3.x)
- **Jupyter Notebook/Lab**
- **Windows only:** Hadoop helper (`pip winutils.exe`)

---

1) Version Matrix

- **Spark:** 3.5.x (e.g., 3.5.1)
- **Hadoop:** 3.3.x (e.g., 3.3.4)
- **Java:** 8 or 11 (recommended)
- **Python:** 3.8–3.12

---

2) Windows Setup (Step-by-Step)

2.1 Install Java (JDK)

1. Download **Temurin (Adoptium) JDK 11** and install.
2. Note install path, e.g. `C:\Program Files\Eclipse Adoptium\jdk-11`.

Set environment variables (System-wide): - Press **Win** → search **"Environment Variables"** → **Edit the system environment variables**. - **System variables → New…** - `JAVA_HOME` = `C:\Program Files\Eclipse Adoptium\jdk-11` - **System variables → Path → Edit → New** - `%JAVA_HOME%\bin`

Verify (new PowerShell):

```
java -version
```

You should see Java 11 output.

2.2 Install Python & pip

- Download Python from python.org and **check** "Add Python to PATH" during install.

Verify:

```
python -V
pip -V
```

2.3 Install Spark (prebuilt)

1. Download **Spark 3.5.x prebuilt for Hadoop 3.x**.
2. Extract to `C:\tools\spark-3.5.1\` (create `C:\tools\` if it doesn't exist).
3. Optionally create a convenience symlink folder `C:\tools\spark` pointing to your versioned folder.

Set environment variables: - **System variables → New…** - `SPARK_HOME` = `C:\tools\spark-3.5.1` - **Path → Edit → New** - `%SPARK_HOME%\bin`

(Optional) If you created `C:\tools\spark`**:** set `SPARK_HOME=C:\tools\spark` and keep paths stable across upgrades.

2.4 Install Hadoop helper (winutils.exe)

Spark on Windows needs `winutils.exe` compatible with your Hadoop version (e.g., 3.3.x).

1. Create directories:
2. `C:\tools\hadoop\bin\`
3. Place `winutils.exe` inside `C:\tools\hadoop\bin\` (matching Hadoop 3.3.x).

Set environment variables: - **System variables → New…** - `HADOOP_HOME` = `C:\tools\hadoop` - **Path → Edit → New** - `%HADOOP_HOME%\bin`

> If you see **"HADOOP_HOME is not set"** or **access denied** errors, this step is missing.

2.5 Install Jupyter, PySpark, findspark

```
pip install --upgrade pip
pip install notebook jupyterlab pyspark findspark
```

Launch Jupyter:

```
jupyter notebook
```

Test inside a new notebook:

```
import findspark
findspark.init()

from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("WindowsTest").getOrCreate()
print("Spark:", spark.version)
print("Java:",
spark.sparkContext._jvm.java.lang.System.getProperty("java.version"))
```

> If it prints versions without error, you're set! ‼️

2.6 Create a dedicated Jupyter Kernel: "PySpark (local)"

Create a kernel so you don't need to import `findspark` every time.

Option A: Minimal kernel (uses environment variables) 1. Create folder (PowerShell):

```
$env:APPDATA + "\\jupyter\\kernels\\pyspark-local" | % { New-Item -ItemType
Directory -Force -Path $_ }
```

2. Create `kernel.json` inside that folder with:

```
{
  "argv": [
    "python", "-m", "ipykernel", "-f", "{connection_file}"
  ],
  "display_name": "PySpark (local)",
  "language": "python",
  "env": {
    "JAVA_HOME": "C:/Program Files/Eclipse Adoptium/jdk-11",
    "SPARK_HOME": "C:/tools/spark-3.5.1",
    "HADOOP_HOME": "C:/tools/hadoop",
    "PYTHONPATH": "${SPARK_HOME}/python;${SPARK_HOME}/python/lib/
py4j-0.10.9.7-src.zip",
    "PATH": "${SPARK_HOME}/bin;${HADOOP_HOME}/bin;${JAVA_HOME}/bin;${PATH}"
  }
}
```

3. Restart Jupyter → choose **PySpark (local)** kernel.

Option B: Use `ipykernel` within a virtualenv

```
python -m venv venv
venv\Scripts\activate
pip install ipykernel pyspark
python -m ipykernel install --user --name=pyspark-local --display-name
"PySpark (local)"
```

Then add env vars to this kernel's `kernel.json` as above.

2.7 Optional: PowerShell profile for auto-env

Add this to your PowerShell profile so env vars are set in every new shell:

```
# Open profile (creates if missing)
notepad $PROFILE
```

Add:

```
$env:JAVA_HOME = "C:\\Program Files\\Eclipse Adoptium\\jdk-11"
$env:SPARK_HOME = "C:\\tools\\spark-3.5.1"
$env:HADOOP_HOME = "C:\\tools\\hadoop"
$env:Path = "$env:SPARK_HOME\\bin;$env:HADOOP_HOME\\bin;$env:JAVA_HOME\\bin;"
+ $env:Path
```

Save → open a new PowerShell and verify with `echo $env:SPARK_HOME`.

---

3) macOS Setup (Intel & Apple Silicon)

3.1 Install Homebrew

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/
install/HEAD/install.sh)"
```

Follow on-screen post-install steps to add brew to your PATH.

3.2 Install Java & Python

```
brew install openjdk@11 python
```

Add to your shell config (`~/.zshrc` on modern macOS):

```
export JAVA_HOME=$(/usr/libexec/java_home -v11)
export PATH="$JAVA_HOME/bin:$PATH"
```

Reload:

```
source ~/.zshrc
java -version
python3 -V
```

3.3 Install Spark

Option A (easy):

```
brew install apache-spark
```

This usually installs under `/opt/homebrew/Cellar/apache-spark/...` (Apple Silicon) or `/usr/local/Cellar/...` (Intel).

Option B (manual): download Spark 3.5.x (prebuilt for Hadoop 3.x), extract to `~/tools/spark-3.5.1` and set:

```
export SPARK_HOME=~/tools/spark-3.5.1
export PATH="$SPARK_HOME/bin:$PATH"
```

Add these lines to `~/.zshrc` to persist.

3.4 Install Jupyter, PySpark, findspark

```
pip3 install --upgrade pip
pip3 install notebook jupyterlab pyspark findspark
```

Launch Jupyter:

```
jupyter notebook
```

Test in notebook:

```
import findspark
findspark.init()
from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder.appName("MacTest").getOrCreate()
print("Spark:", spark.version)
```

3.5 Create a dedicated Jupyter Kernel (macOS)

```
python3 -m venv venv
source venv/bin/activate
pip install ipykernel pyspark
python -m ipykernel install --user --name=pyspark-local --display-name
"PySpark (local)"
```

Find the kernel file (e.g., `~/Library/Jupyter/kernels/pyspark-local/kernel.json`) and add:

```
{
  "argv": ["python", "-m", "ipykernel", "-f", "{connection_file}"],
  "display_name": "PySpark (local)",
  "language": "python",
  "env": {
    "JAVA_HOME": "/Library/Java/JavaVirtualMachines/temurin-11.jdk/Contents/
Home",
    "SPARK_HOME": "/opt/homebrew/Cellar/apache-spark/3.5.1/libexec",
    "PYTHONPATH": "${SPARK_HOME}/python:${SPARK_HOME}/python/lib/
py4j-0.10.9.7-src.zip",
    "PATH": "${SPARK_HOME}/bin:${JAVA_HOME}/bin:${PATH}"
  }
}
```

> Adjust `SPARK_HOME` to match your actual path (`brew --prefix apache-spark` prints it).

---

4) Handy Launch Scripts

4.1 Windows: `launch_pyspark_jupyter.ps1`

Save this as a PowerShell script and run by right-click → **Run with PowerShell**:

```
$env:JAVA_HOME = "C:\\Program Files\\Eclipse Adoptium\\jdk-11"
$env:SPARK_HOME = "C:\\tools\\spark-3.5.1"
$env:HADOOP_HOME = "C:\\tools\\hadoop"
$env:Path = "$env:SPARK_HOME\\bin;$env:HADOOP_HOME\\bin;$env:JAVA_HOME\\bin;"
+ $env:Path
jupyter notebook
```

4.2 macOS: `launch_pyspark_jupyter.sh`

```
#!/usr/bin/env bash
export JAVA_HOME=$(/usr/libexec/java_home -v11)
export SPARK_HOME="$(brew --prefix apache-spark)/libexec"
export PATH="$SPARK_HOME/bin:$JAVA_HOME/bin:$PATH"
jupyter notebook
```

Make executable: `chmod +x launch_pyspark_jupyter.sh`

---

5) First Cells to Try in Notebook

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("HelloSpark").getOrCreate()

# Tiny DataFrame
df = spark.createDataFrame([(1, "A"), (2, "B"), (3, "C")], ["id", "value"])
df.show()
print("Partitions:", df.rdd.getNumPartitions())

# Simple transformation & action
out = df.filter(df.id >= 2)
out.show()

# Stop session when done (optional in notebooks)
spark.stop()
```

---

6) Troubleshooting (Most Common)

- `HADOOP_HOME is not set` **(Windows):** Set `HADOOP_HOME` and ensure `%HADOOP_HOME%\bin` is in PATH and `winutils.exe` exists.
- `Java gateway process exited before sending its port number`: Java not found or mismatched versions. Check `JAVA_HOME` and `PATH`. Prefer Java 11.
- `ModuleNotFoundError: No module named 'pyspark'`: Install `pyspark` in the same Python environment Jupyter is using.
- **Arm/M1 Macs:** Use Homebrew's Spark; ensure `JAVA_HOME` points to an Arm build (Temurin 11) and avoid mixing Intel/Arm binaries.
- **Permission errors on Windows temp dirs:** Run PowerShell as Admin once, or set `HADOOP_HOME` and ensure folder permissions for `%TEMP%`.
- **Kernel not showing:** Reinstall kernel: `python -m ipykernel install --user --name=pyspark-local --display-name "PySpark (local)"`.
- **Weird warnings (`illegal reflective access`):** benign for local dev; safe to ignore.

---

7) Optional: Conda-based Setup

```
conda create -n pyspark-3.5 python=3.11 -y
conda activate pyspark-3.5
pip install pyspark notebook jupyterlab findspark ipykernel
python -m ipykernel install --user --name=pyspark-3.5 --display-name
"PySpark 3.5 (conda)"
```

Then add the `env` block to that kernel's `kernel.json` like earlier.

---

8) Verify Everything

- `java -version` → 11.x
- `python -V` → 3.x
- `pyspark --version` → shows Spark + Hadoop
- In Jupyter, `spark.version` prints 3.5.x

---

9) Upgrades & Uninstall

- **Upgrade Spark:** Download new version to a new folder, point `SPARK_HOME` to it.
- **Remove kernel:** delete its folder under Jupyter kernels dir (`%APPDATA%\jupyter\kernels\...` on Windows, `~/Library/Jupyter/kernels/...` on macOS).
- **Uninstall brew Spark:** `brew uninstall apache-spark`
