🚀 Apache Spark Setup on Windows (2025 Edition)

🧰 Tools You'll Be Installing

Tool	Why It's Needed
Python 3.10	Required for running PySpark
Java JDK 17	Spark is built on JVM
Apache Spark	The main engine we want to run
Winutils (Hadoop)	Helps Spark interact with Windows FS
Jupyter Notebook	Optional, but great for coding

📁 Recommended Folder Setup

Create a central workspace in `C:\bigdata\`:

```
makefile
CopyEdit
C:\bigdata\
├── spark\
├── winutils\
├── spark_projects\
```

🔹 STEP 1: Install Python 3.10.11

🔗 Download

👉 https://www.python.org/ftp/python/3.10.11/python-3.10.11-amd64.exe

✅ During Installation:

- ✅ Check ✅ **“Add Python to PATH”**
- ✅ Install for **All Users**
- ✅ Let it install at:
  `C:\Users\USER\AppData\Local\Programs\Python\Python310\`

🧪 Verify (CMD):

```
cmd
CopyEdit
python --version
pip --version
```

🔹 STEP 2: Install Java JDK 17 (Temurin)

🔗 Download

👉 https://github.com/adoptium/temurin17-binaries/releases

Choose:

- `OpenJDK17U-jdk_x64_windows_hotspot_17.0.16_8.msi`

✅ Install to:

```
C:\Program Files\Eclipse Adoptium\jdk-17.0.16.8-hotspot\
```

⚙️ Set Environment Variables

System Variables (not User!):

Name	Value
`JAVA_HOME`	`C:\Program Files\Eclipse Adoptium\jdk-17.0.16.8-hotspot`

Add to `Path` `%JAVA_HOME%\bin`

🧪 Verify (CMD):

```
cmd
CopyEdit
java -version
javac -version
```

🔹 STEP 3: Install Apache Spark (3.5.x or 4.0.0)

🔗 Download

👉 https://spark.apache.org/downloads.html

- Spark version: `3.5.1` or `4.0.0-preview`
- Package type: **Pre-built for Apache Hadoop 3**

✅ Unzip to:

```
C:\bigdata\spark\
```

⚙️ Set Environment Variables

Name	Value
`SPARK_HOME`	`C:\bigdata\spark`

Add to `Path` `%SPARK_HOME%\bin`

🧪 Verify (CMD):

```
cmd
CopyEdit
spark-shell --version
pyspark --version
```

🔹 STEP 4: Setup Hadoop Winutils

🔗 Download:

👉 https://github.com/cdarlint/winutils/tree/master/hadoop-3.0.0

(Download `winutils.exe` and `hadoop.dll`)

✅ Folder Setup:

Create this folder:
`C:\bigdata\winutils\hadoop-3.0.0\`
Paste the `winutils.exe` and `hadoop.dll` here.

⚙️ Environment Variables:

Name	Value
`HADOOP_HOME`	`C:\bigdata\winutils\hadoop-3.0.0`
Add to `Path`	`%HADOOP_HOME%\bin`

🔹 STEP 5: Configure Spark Environment (important)

Create or edit this file:

📄 `C:\bigdata\spark\conf\spark-env.cmd`

Add:

```
cmd
CopyEdit
set JAVA_HOME=C:\Program Files\Eclipse Adoptium\jdk-17.0.16.8-hotspot
set HADOOP_HOME=C:\bigdata\winutils\hadoop-3.0.0
set PYSPARK_PYTHON=python
```

🔹 STEP 6: Install Jupyter Notebook (Optional but Recommended)

🧪 In CMD:

```
cmd
CopyEdit
pip install notebook
pip install jupyterlab
```

🧪 Launch:

```
cmd
CopyEdit
python -m notebook
```

🔹 STEP 7: Test a Sample PySpark Script

📄 Save this as `rdd_example.py` inside `C:\bigdata\spark_projects\`

```
python
CopyEdit
```

```
from pyspark import SparkContext

sc = SparkContext("local", "BasicRDDApp")

data = [1, 2, 3, 4, 5, 6]
rdd = sc.parallelize(data)

squared = rdd.map(lambda x: x * x)
filtered = squared.filter(lambda x: x > 10)
result = filtered.collect()

print("Result of RDD operations:", result)

sc.stop()
```

🧪 Run:

```
cmd
CopyEdit
cd C:\bigdata\spark_projects
python rdd_example.py
```

You should see:

```
less
CopyEdit
Result of RDD operations: [16, 25, 36]
```

🔎 Final Checklist: Command List to Verify Everything

```
cmd
CopyEdit
python --version
pip --version
java -version
javac -version
```

```
git --version
node --version
spark-shell --version
pyspark --version
winutils ls /
python -m notebook
```

⚙️ Where to Set Environment Variables

System Environment Variables:

- **JAVA_HOME**
- **SPARK_HOME**
- **HADOOP_HOME**

➡️ Set from:

Control Panel → System → Advanced System Settings → Environment Variables

Section	Use
System Variables	Global for all users (recommended)
User Variables	Affects current user only

Always add `...\bin` folders to **Path** manually.

✅ Tips

- If `spark-shell` or `pyspark` is not recognized, make sure `%SPARK_HOME%\bin` is in your Path.
- If `winutils ls /` doesn't work, double-check `HADOOP_HOME` and the files inside.

- In Jupyter, Spark might look for the **wrong Python** — avoid this by setting `PYSPARK_PYTHON=python` in `spark-env.cmd`.