INP7079233 - BIG DATA COMPUTING 2023-2024 (prof. Pietracaprina and Silvestri)

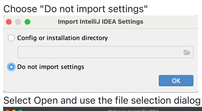
[redacted]

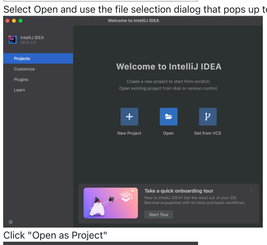
Machine setup (updated 18/03/2024)

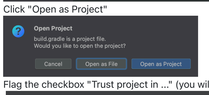
Quick links:

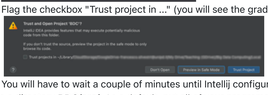
- Instructions for Java users
- Instructions for Python users

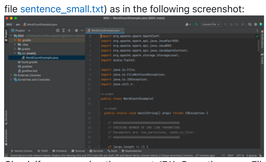
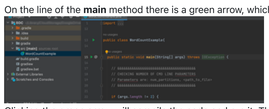
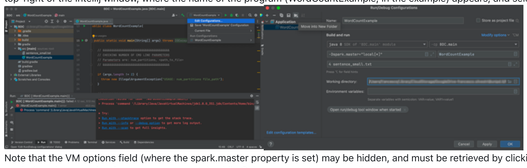
Instructions for Java users

1. Create a directory **BDC** on your computer which will be the main directory for your homeworks. In this directory, put the file build.gradle which you must download here.
2. Install Intellij Idea (Community edition), version 2022.3 on your system from this download page. (For an installation guide you can look at the official install and set-up page.)
3. After installation is completed, you must configure Intellij for a first run. Launch Intellij. After the initial steps for accepting their User Agreement and Data Sharing options, follow these instructions:
   1. Choose "Do not import settings"
      

   2. Select Open and use the file selection dialog that pops up to select the build.gradle file contained in the directory you created in Step 1
      

   3. Click "Open as Project"
      

   4. Flag the checkbox "Trust project in ..." (you will see the gradle URL) and then the "Trust Project" button.
      

   5. You will have to wait a couple of minutes until Intellij configures itself.
4. Create a directory **BDC/src/**. As a default, put all of your programs in **BDC/src/**, and all datasets that you want to provide as inputs to your programs, in the root directory **BDC/**.
5. Use the project navigation panel on the left to open the files (e.g., programs, datasets, etc.). To test, open a java program (for instance copy in BDC/src the file WordCountExample.java and in BDC/ the input file sentence_small.txt) as in the following screenshot:
   
6. Check if you are using the correct JDK. Open the menu File/Project Structure and select Project from the left panel. Then, open the drop-down menu and select JDK version 8, as in the figure above (your list might differ).
   
7. On the line of the **main** method there is a green arrow, which allows you to run your code.
   
8. Clicking the green arrow will compile the code and run it. The run will not succeed (exit code 1 at the bottom of the screen) since you must configure a set of execution parameters. To do so, use the drop-down menu on the top-right of the Intellij window, where the name of the program (WordCountExample, in the example) appears, and select **Edit Configurations** to get a dialog window which you must fill as indicated in the following images.
   
9. Note that the VM options field (where the spark.master property is set) may be hidden, and must be retrieved by clicking on the Modify options blue text. Specifically, VM options specify that you want to run Spark in local mode (spark.master is a java system property), while CLI arguments to your application are the arguments passed to the main method (in this example, the line with the string "4 sentence_small.txt" encodes the arguments for the main in WordCountExample.java).
10. Run again the program by clicking on the green arrow. After a (long) sequence of Spark messages, you will see the output contained in this file.

Instructions for Python users

Step 1: Download and install a JDK

Since Pyspark will translate the Python instructions into Java bytecodes that will be executed in the JVM of each node of the cluster, a Java Runtime Environment must be available throughout the system with its compiler and its APIs. Head to Oracle download page and download the **Java Development Kit version 8**. Other versions might give problems, so we should avoid them for the time being.

Step 2: Install Pyspark

The easiest thing is to do the installation using the package manager conda from the Anaconda distribution. It should work on Windows, MAC and Linux systems. The following instructions are for Windows users but we think that they can apply also to MAC and Linux users with minor adaptations. (This installation guide found in Internet is a useful reference.)

Step 2.1: Download & Install Anaconda Distribution (which includes Conda) from https://www.anaconda.com/download/

- Make sure to execute the installation program as an administrator
- If your machine has multiple users (e.g., a root user and yourself) specify, when asked in a suitable dialog box, that the installation must be done for all users.

Step 2.1.1 Optional create a new environment:

If you have conda installed and you already have an environment, we highly suggest to create and activate a new environment typing inside the anaconda prompt program:

conda create --name spark

conda activate spark

You can find more about this on https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html.

If you do not have other conda environments you can skip this steps

Step 2.2: Install PySpark as follows

- Open the Anaconda prompt program (you can find it in the search box) by right-clicking on it and chosing "run as administrator"
- Run command "conda install pyspark"

  If you get these messages:
  Preparing transaction: done
  Verifying transaction: failed
  EnvironmentNotWritableError: The current user does not have write permissions to the target environment. Environment location: C:|ProgramData|anaconda3
  this means that you are not running the Anaconda prompt program as administrator. So open the Anaconda prompt program again making sure to choose "run as administrator", as instructed above.

- Run command "conda install -c conda-forge findspark"

Step 2.3: Validate the Pyspark installation as follows

- Open the Anaconda prompt program and run command "pyspark". If everything works you should get the the welcom message ending with "SparkSession available as 'spark'." and the ">>>" cursor for interactive programming. Do the same with the Windows prompt program. Everything should work there as well. In this case, go directly to Step 3.

Step 2.4: Set environment variables. This step must be executed if the validation of Step 2.3 did non succeed and you received messages about problems finding python. The reason might be that you have preexisting installations of python. In this case, you should operate as follows:

- On the Anaconda prompt program, run command "where python". For instance, I got 2 paths as a result:

  C:|ProgramData|anaconda3|python.exe
  C:|Users|andrea|AppData|Local|Programs|Python|Python311|python.exe

  where the second path is relative to my preexisting installation.

- Open the Windows tool to modify the environment variables (it can be found searching "environment variables" in the search box)
- Create a new environment variable PYSPARK_PYTHON with value equal to the python path of the anaconda installation. In my case, the path is *C:|ProgramData|anaconda3|python.exe*
- Add to variable PATH the path to the directory where the anaconda installation of python is found. In my case, the path is *C:|ProgramData|anaconda3|* and make sure that this path occurs before the path to the preexisting installation so to ensure that the anaconda installation becomes the default.
- At this point close and reopen the Anaconda prompt program and repeat the verification step (Step 2.3).

Step 3: Run test program

- Open the Anaconda or windows command prompt program
- Download program WordCountExample.py and input file sentence_small.txt in the current directory (the one shown in the prompt program)
- Run command "python WordCountExample.py 4 sentence_small.txt" to execute the WordCountExample program on the input file (the parameter 2 is the number of partitions).
- If everything works, you should get the output contained in this file.

Last modified: Monday, 18 March 2024, 8:29 AM