How to setup environment for Deep learning with Deeplearning

Setup environment for Deep learning with Deeplearning4j

In this tutorial, we shall see how to Setup environment for D Deeplearning4j and import dl4j-examples from Git to Inte working on Deep Learning.

Following are the steps :

1. Install Java.
2. Install Apache Maven.
3. Download IntelliJ IDEA Community Edition.
4. Install Git.
5. Start hands-on with dl4j-examplesClone – Git project “dl4j-examples”. – Build the project using Maven. – Imp to IntelliJ IDEA as Maven.

Skip any of the above steps if you have already installed. Also, it is recommended to update the soft available versions.

Install Java

A step-by-step guide to install java.

Install Apache Maven

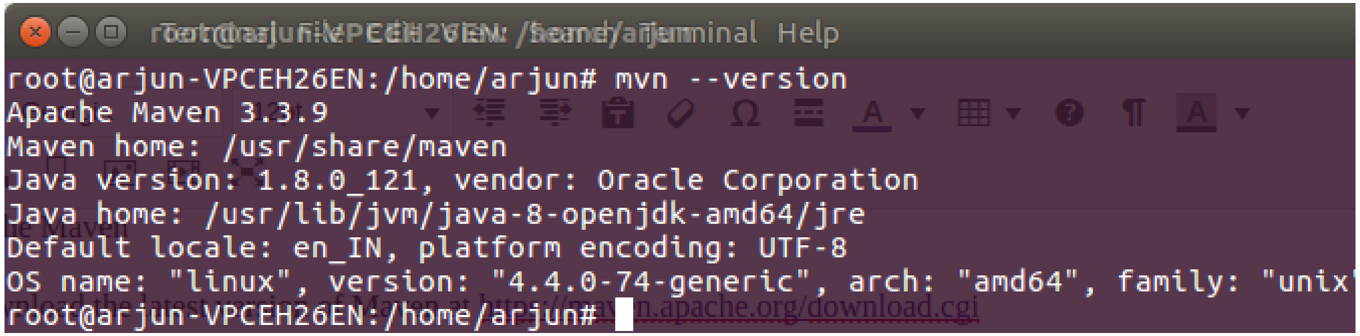
Step 1 : Download the latest version of Maven at https://maven.apache.org/download.cgi

Step 2 : Install Maven following the instructions at https://maven.apache.org/install.html

Step 3 : Verify if maven is installed using the command

```
mvn –version
```

The result would be as shown in the below screenshot :



```
root@arjun-VPCEH26EN:/home/arjun# mvn --version
Apache Maven 3.3.9
Maven home: /usr/share/maven
Java version: 1.8.0_121, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-8-openjdk-amd64/jre
Default locale: en_IN, platform encoding: UTF-8
OS name: "linux", version: "4.4.0-74-generic", arch: "amd64", family: "unix'
root@arjun-VPCEH26EN:/home/arjun#
```

Check Maven installation using the command “mvn –version”

Download IntelliJ IDEA Community Edition

1. Download the latest version of IntelliJ IDEA from https://www.jetbrains.com/idea/download/
2. Extract the downloaded package and go into the folder where bin, lib, plugin, etc., folders are present. An inst "Install-Linux-tar.txt" (for Linux installation), is present in the location. [A similar name would be present for oth Systems].
3. Follow the instructions present in the setup guide.

Install Git

Instructions for installing Git are available at https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

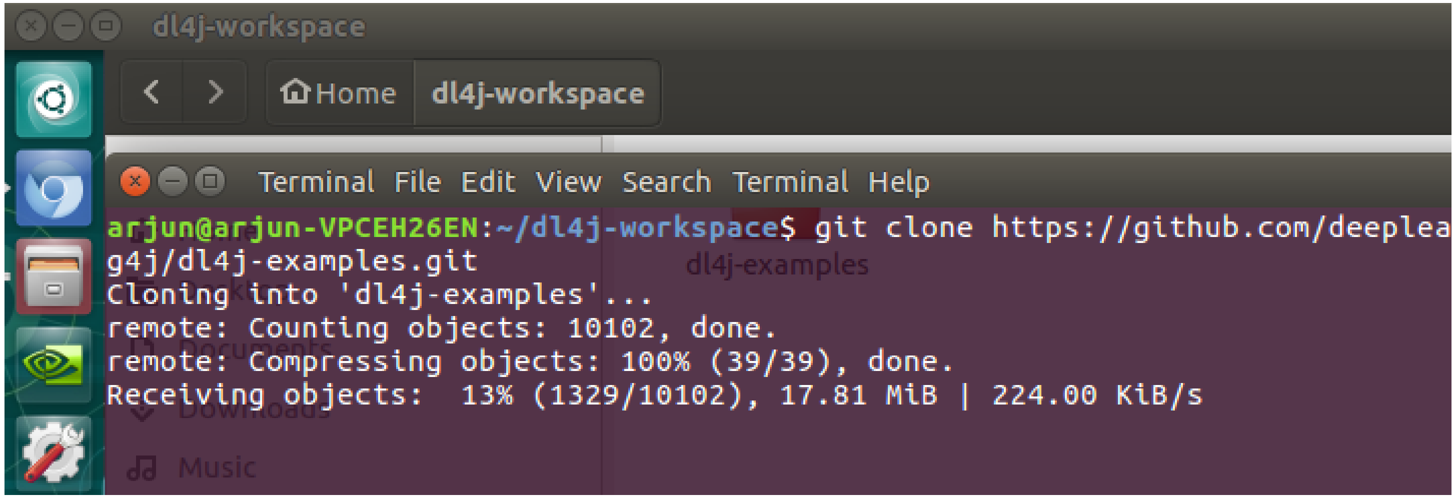
Working with dl4j-examples

Step 1

Create a new directory for working with dl4j projects. Say "dl4j-workspace". And execute the following download the dl4j-examples from the git repository.

```
$ git clone https://github.com/deeplearning4j/dl4j-examples.git
```

Following screenshot shows download in progress for dl4j-examples :



```
arjun@arjun-VPCEH26EN:~/dl4j-workspace$ git clone https://github.com/deeplea
g4j/dl4j-examples.git
Cloning into 'dl4j-examples'...
remote: Counting objects: 10102, done.
remote: Compressing objects: 100% (39/39), done.
Receiving objects:  13% (1329/10102), 17.81 MiB | 224.00 KiB/s
```

Clone dl4j-examples from Git

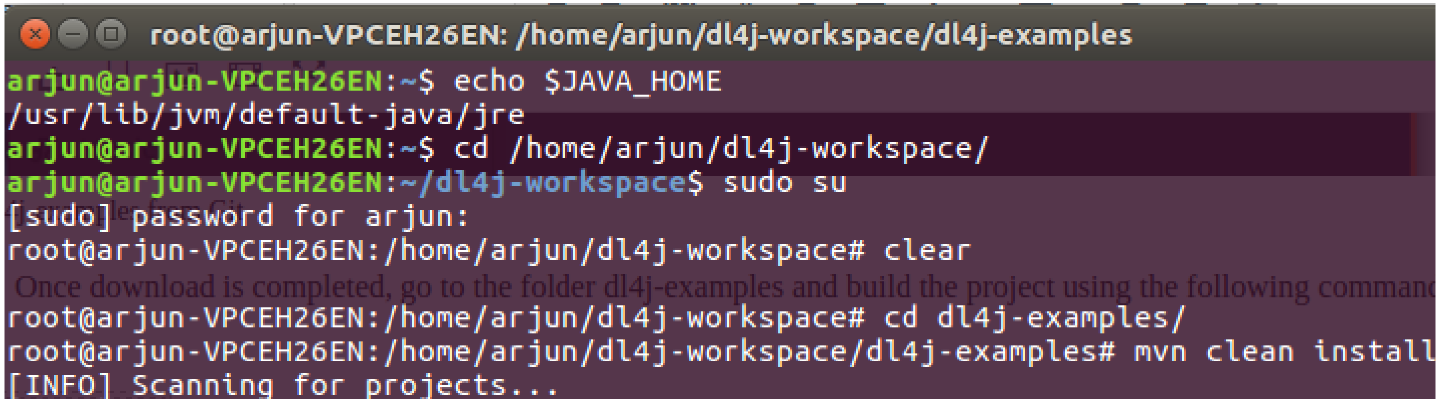
Step 2

Once the download is completed, go to the folder dl4j-examples and build the project using the following c

```
$ mvn clean install
```

Note : Make sure the JAVA_HOME path variable is pointing to the JRE of the latest JDK available in your

When the command "*mvn clean insall*" is run, the result is as shown in the below screenshot :



```
arjun@arjun-VPCEH26EN:~$ echo $JAVA_HOME
/usr/lib/jvm/default-java/jre
arjun@arjun-VPCEH26EN:~$ cd /home/arjun/dl4j-workspace/
arjun@arjun-VPCEH26EN:~/dl4j-workspace$ sudo su
[sudo] password for arjun:
root@arjun-VPCEH26EN:/home/arjun/dl4j-workspace# clear
Once download is completed, go to the folder dl4j-examples and build the project using the following comman
root@arjun-VPCEH26EN:/home/arjun/dl4j-workspace# cd dl4j-examples/
root@arjun-VPCEH26EN:/home/arjun/dl4j-workspace/dl4j-examples# mvn clean install
[INFO] Scanning for projects...
```

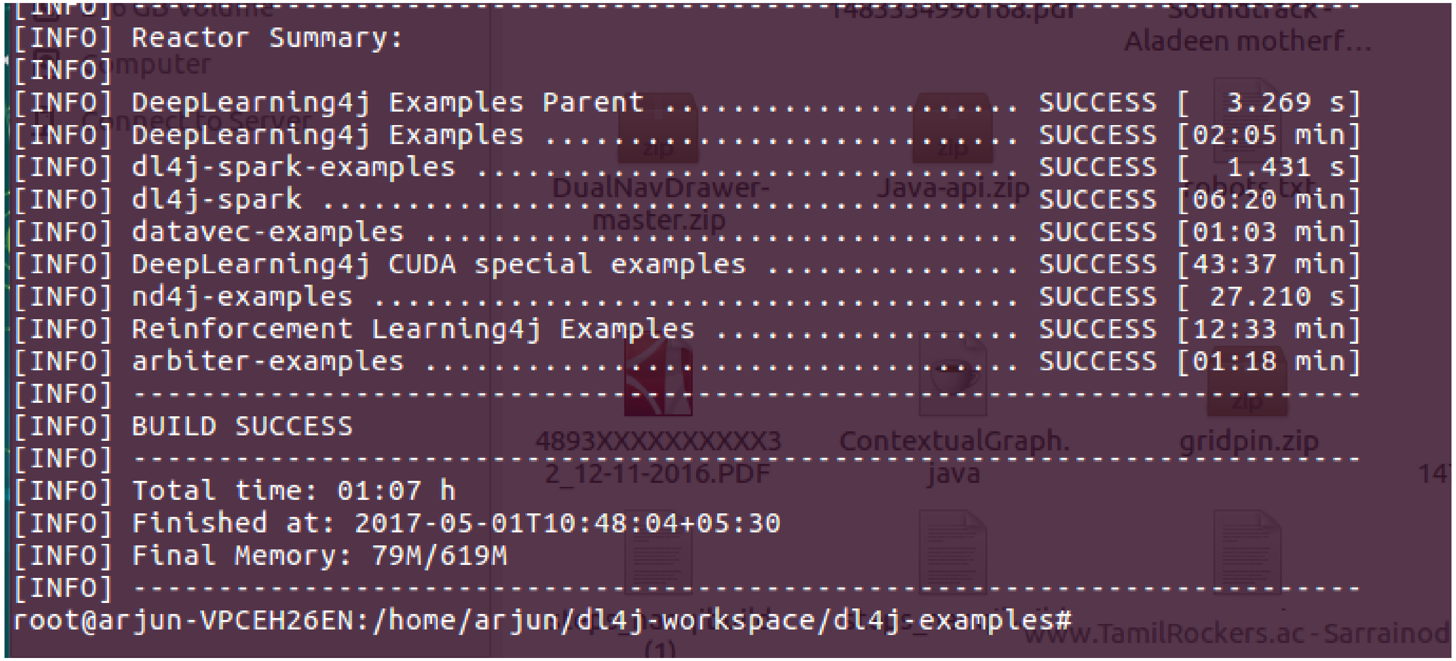
mvn clean install

It takes some time for Maven to download all the artifacts mentioned in the pom file and build the project.

Note : Make sure your Java installation has got JavaFX. In case not, you would get a COMPILE ERR JavaFX by executing the following command and try "*mvn clean install*" again.

```
sudo apt-get install openjfx
```

Once the "mvn clean install" command is successful, you should see the BUILD SUCCESS as se screenshot :



```
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] DeepLearning4j Examples Parent ..................... SUCCESS [  3.269 s]
[INFO] DeepLearning4j Examples ............................ SUCCESS [02:05 min]
[INFO] dl4j-spark-examples ................................ SUCCESS [  1.431 s]
[INFO] dl4j-spark ......................................... SUCCESS [06:20 min]
[INFO] datavec-examples ................................... SUCCESS [01:03 min]
[INFO] DeepLearning4j CUDA special examples ............... SUCCESS [43:37 min]
[INFO] nd4j-examples ...................................... SUCCESS [ 27.210 s]
[INFO] Reinforcement Learning4j Examples .................. SUCCESS [12:33 min]
[INFO] arbiter-examples ................................... SUCCESS [01:18 min]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 01:07 h
[INFO] Finished at: 2017-05-01T10:48:04+05:30
[INFO] Final Memory: 79M/619M
[INFO] ------------------------------------------------------------------------
root@arjun-VPCEH26EN:/home/arjun/dl4j-workspace/dl4j-examples#
```

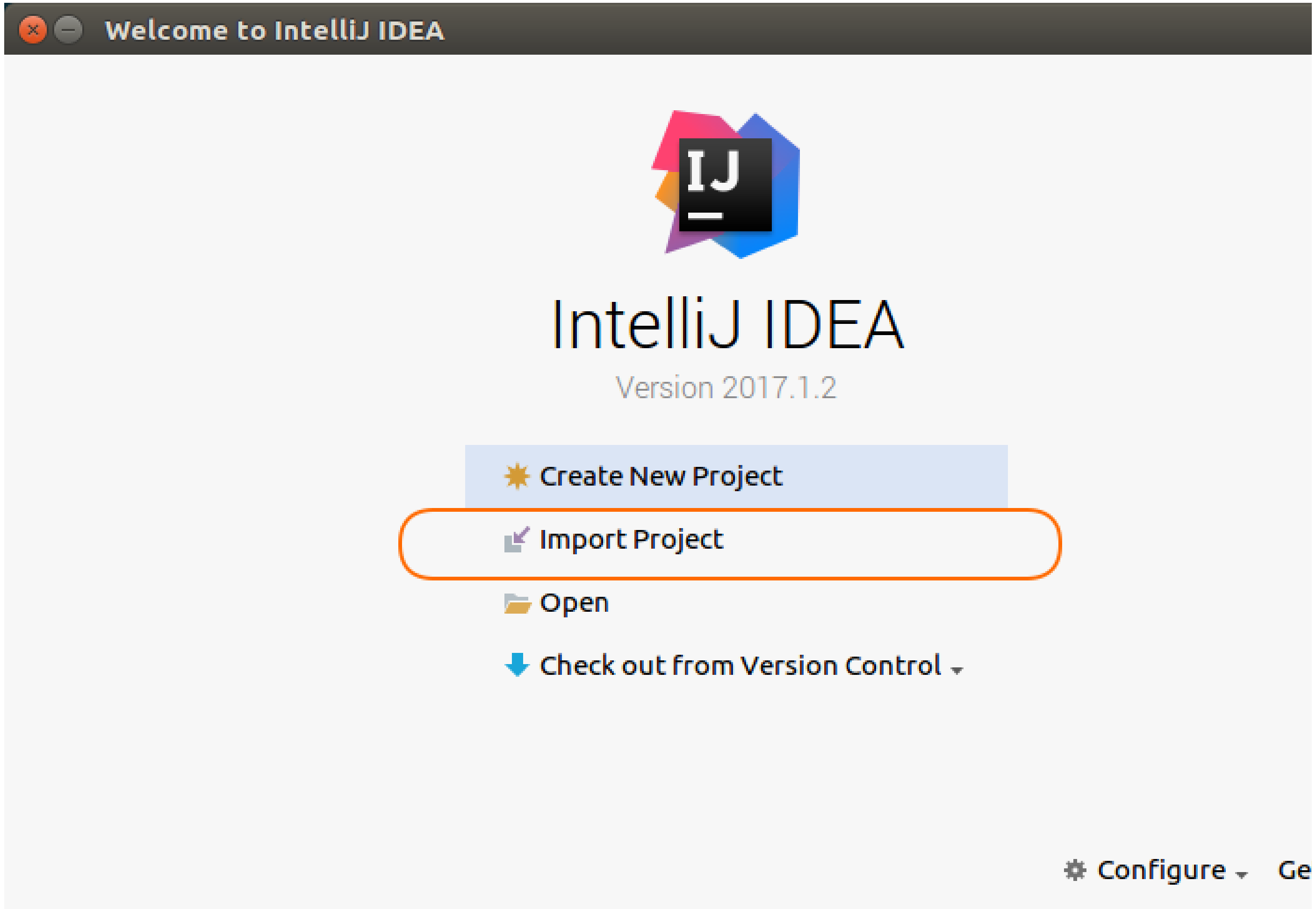
dl4j-examples BUILD SUCCESS

Step 3

Start IntelliJ IDEA.

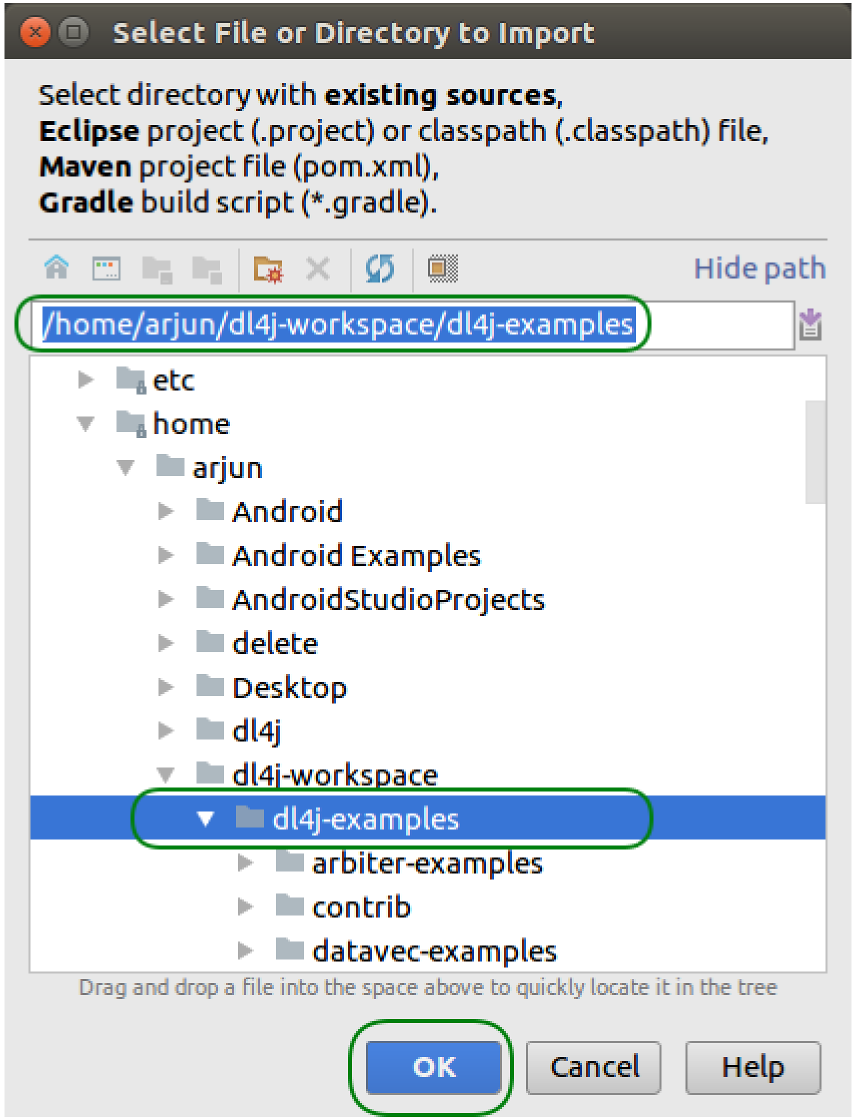
Start intelliJ IDEA

- Click on Import Project



Import Project

- Choose the project folder. Click OK.



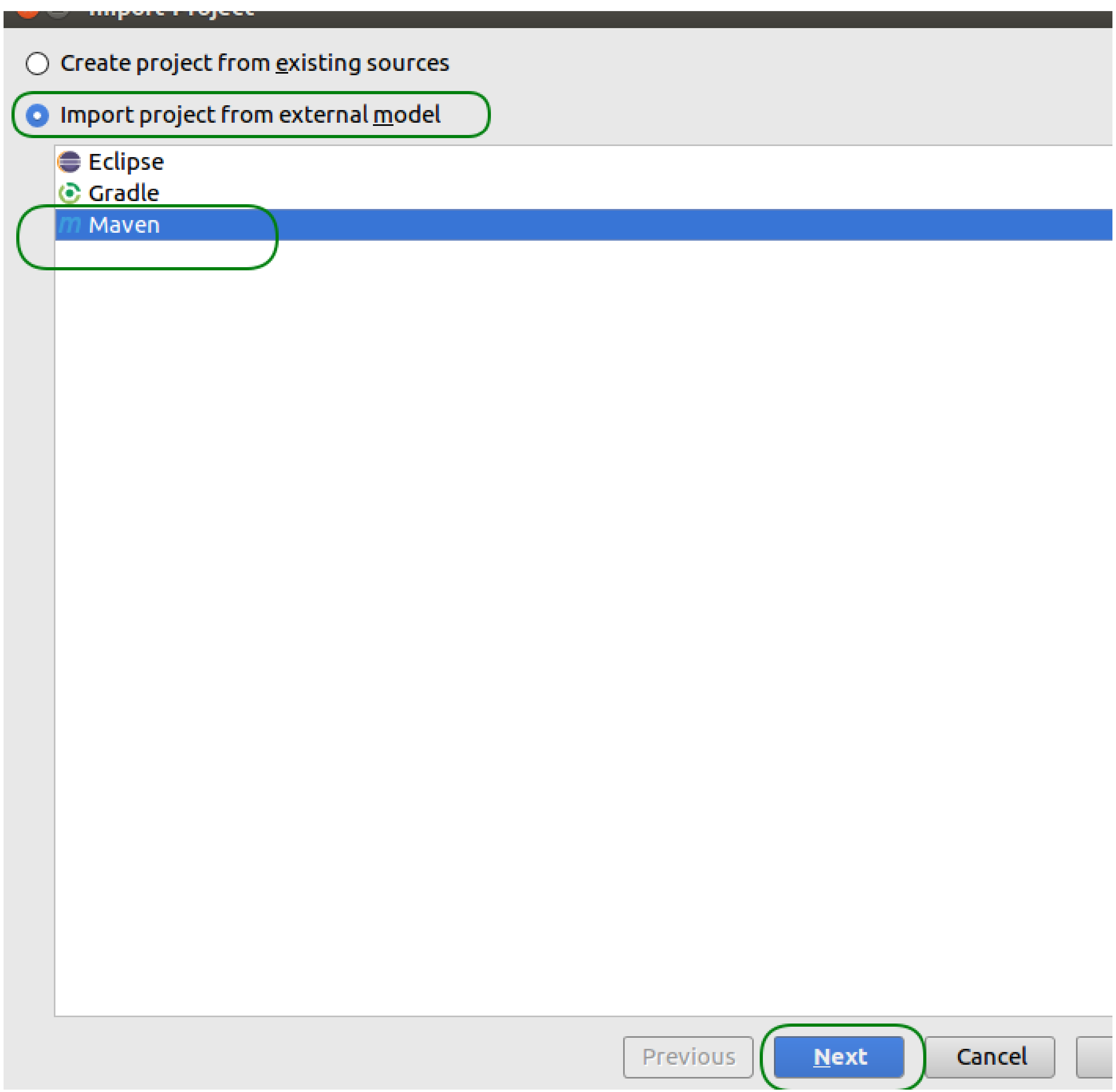
Select directory with **existing sources**,
Eclipse project (.project) or classpath (.classpath) file,
Maven project file (pom.xml),
Gradle build script (*.gradle).

/home/arjun/dl4j-workspace/dl4j-examples

Drag and drop a file into the space above to quickly locate it in the tree

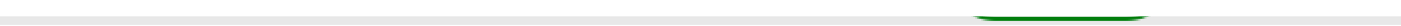
Choose Project Folder

- Import Project as Maven model. Click Next.



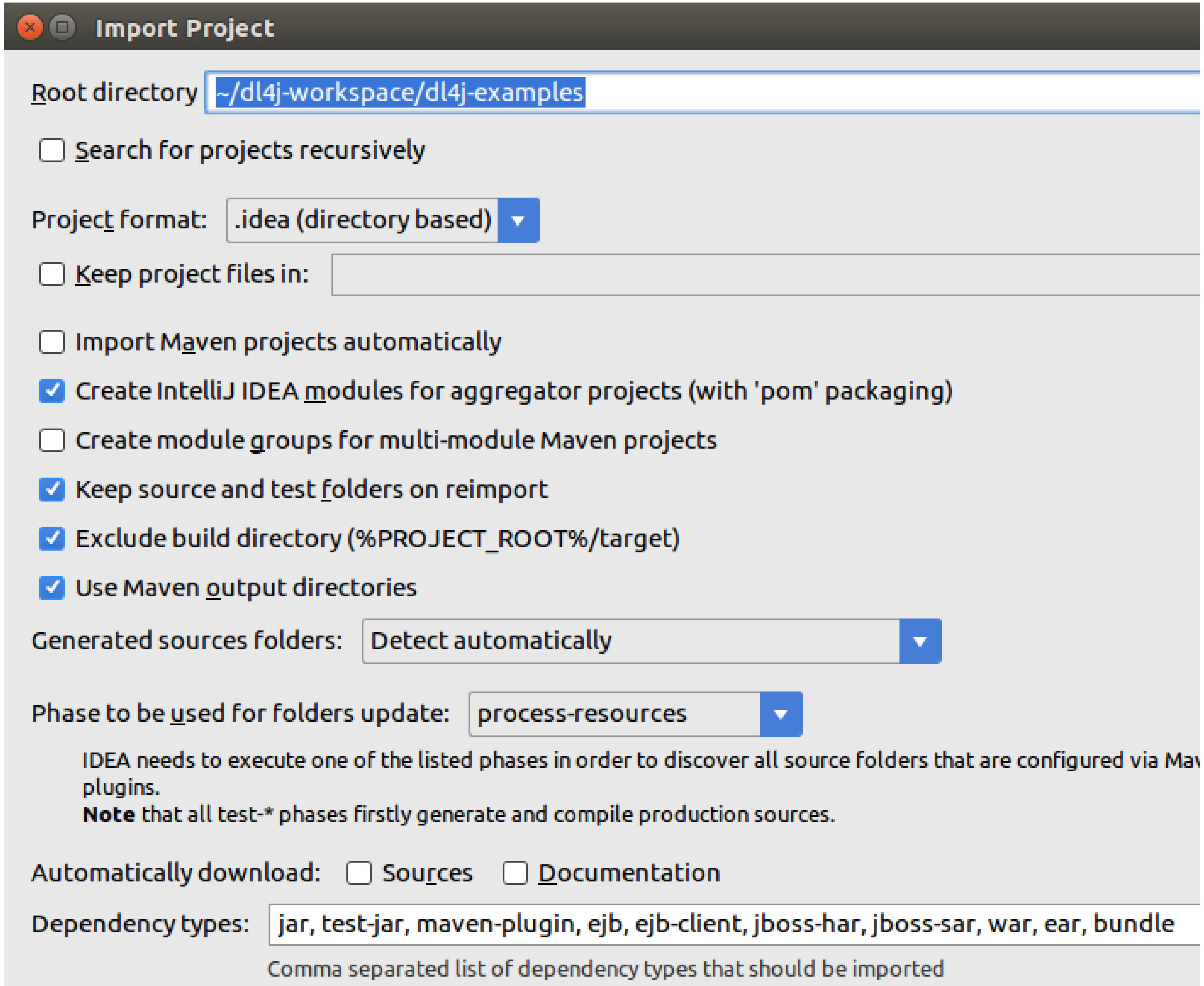
○ Create project from existing sources

● Import project from external model



Import project as Maven

- Project configuration. Keep defaults and Click on Next.



Root directory ~/dl4j-workspace/dl4j-examples

☐ Search for projects recursively

Project format: .idea (directory based)

☐ Keep project files in:

☐ Import Maven projects automatically

☑ Create IntelliJ IDEA modules for aggregator projects (with 'pom' packaging)

☐ Create module groups for multi-module Maven projects

☑ Keep source and test folders on reimport

☑ Exclude build directory (%PROJECT_ROOT%/target)

☑ Use Maven output directories

Generated sources folders: Detect automatically

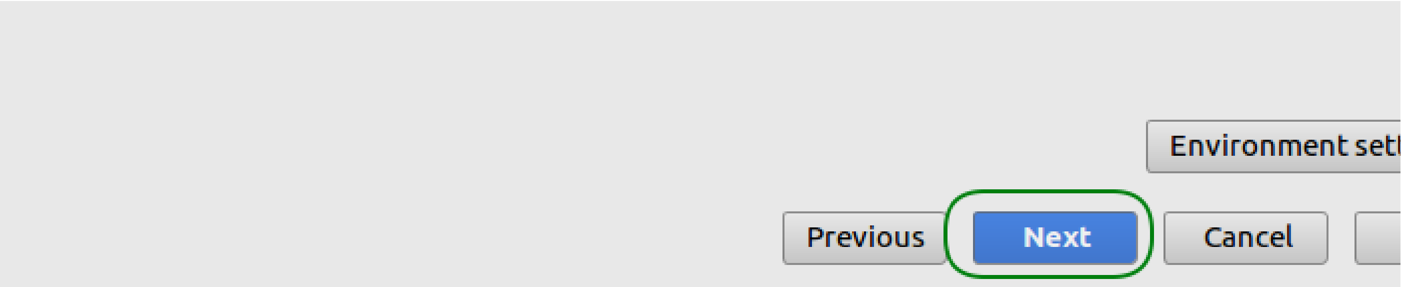
Phase to be used for folders update: process-resources

IDEA needs to execute one of the listed phases in order to discover all source folders that are configured via Mav
plugins.
Note that all test-* phases firstly generate and compile production sources.

Automatically download: ☐ Sources ☐ Documentation

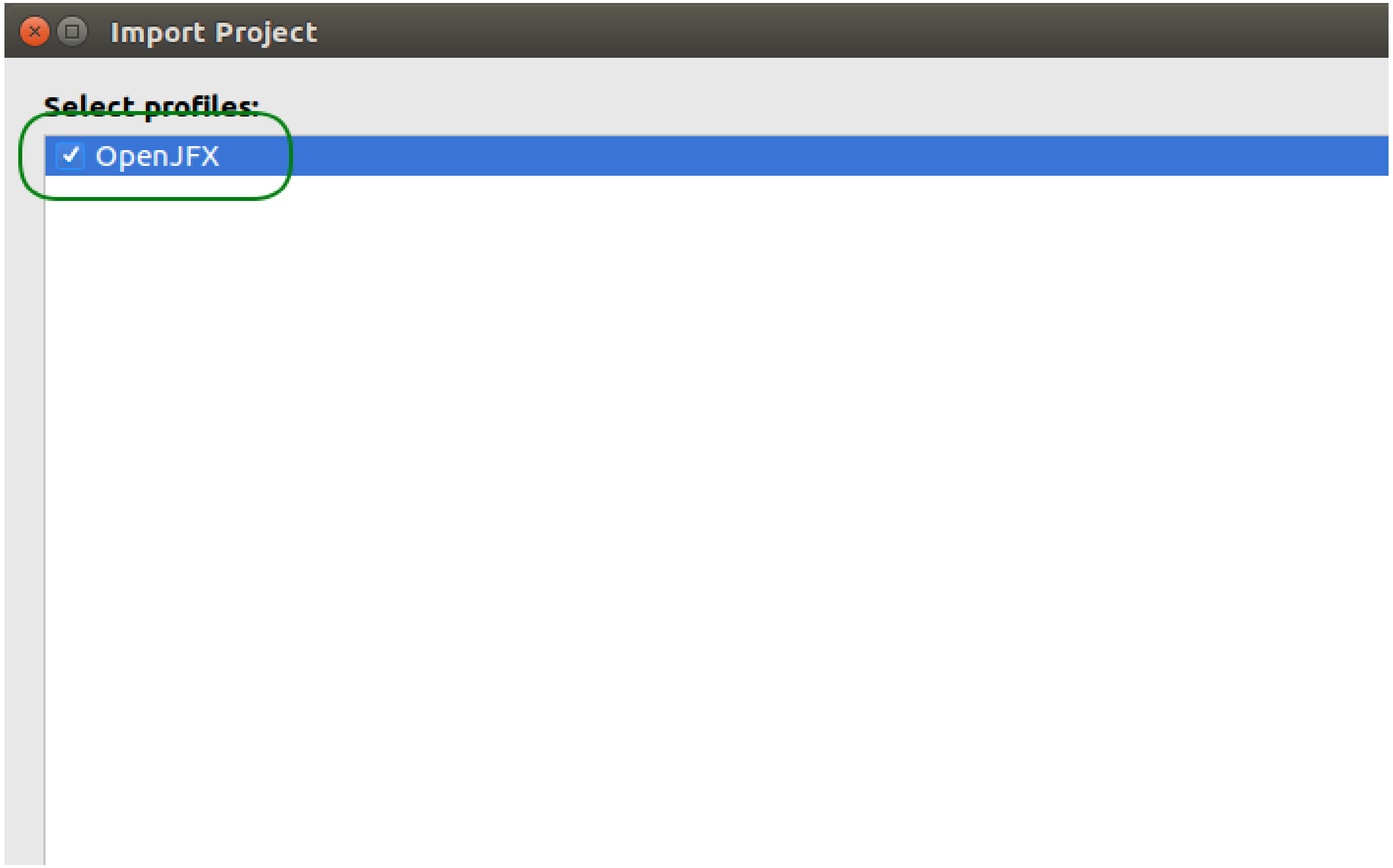
Dependency types: jar, test-jar, maven-plugin, ejb, ejb-client, jboss-har, jboss-sar, war, ear, bundle

Comma separated list of dependency types that should be imported



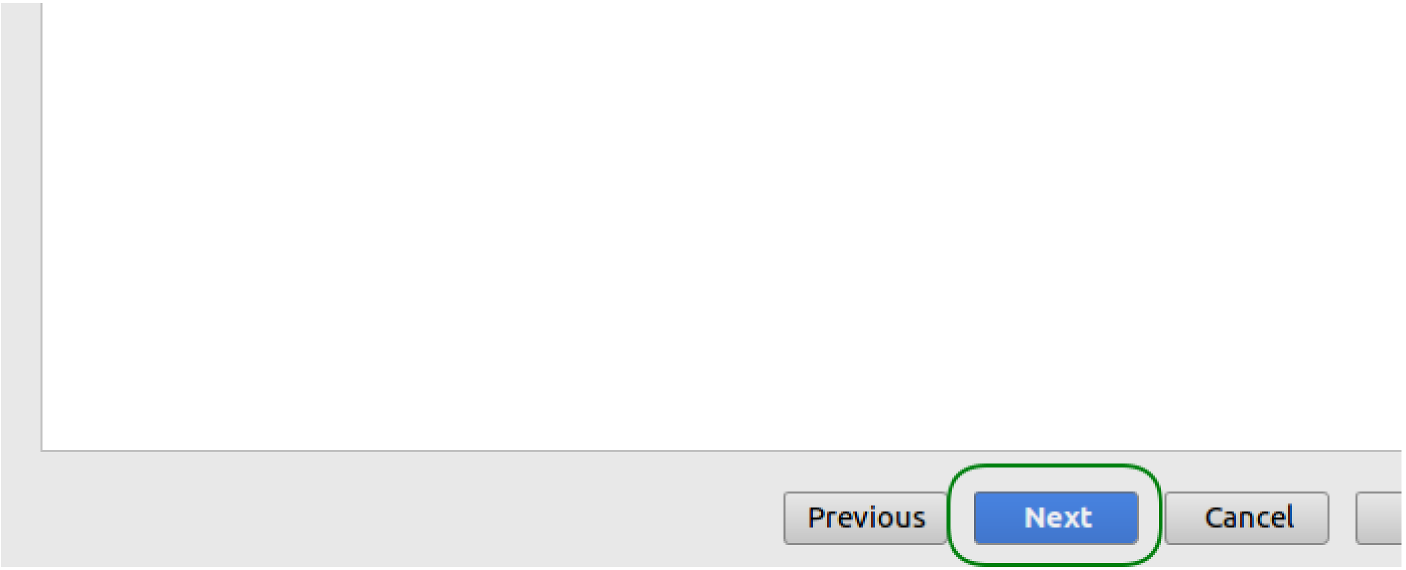
Project Configuration

- Select Profile – OpenJFX. Click Next.



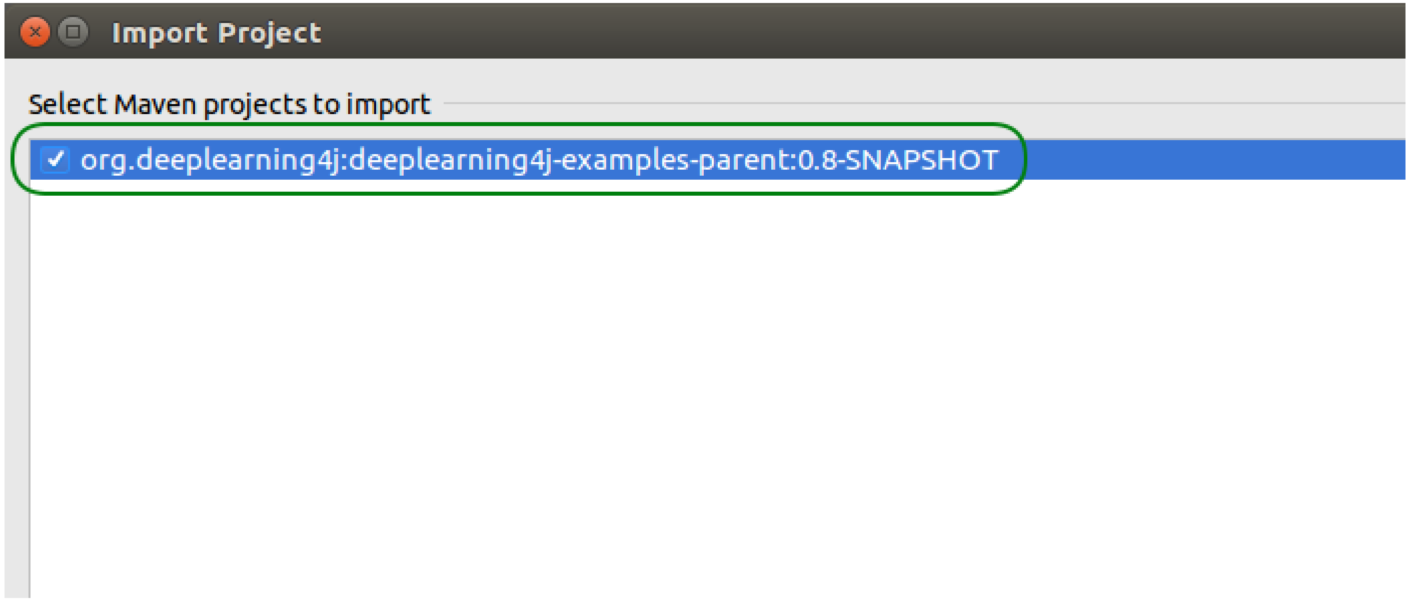
Select profiles:

☑ OpenJFX



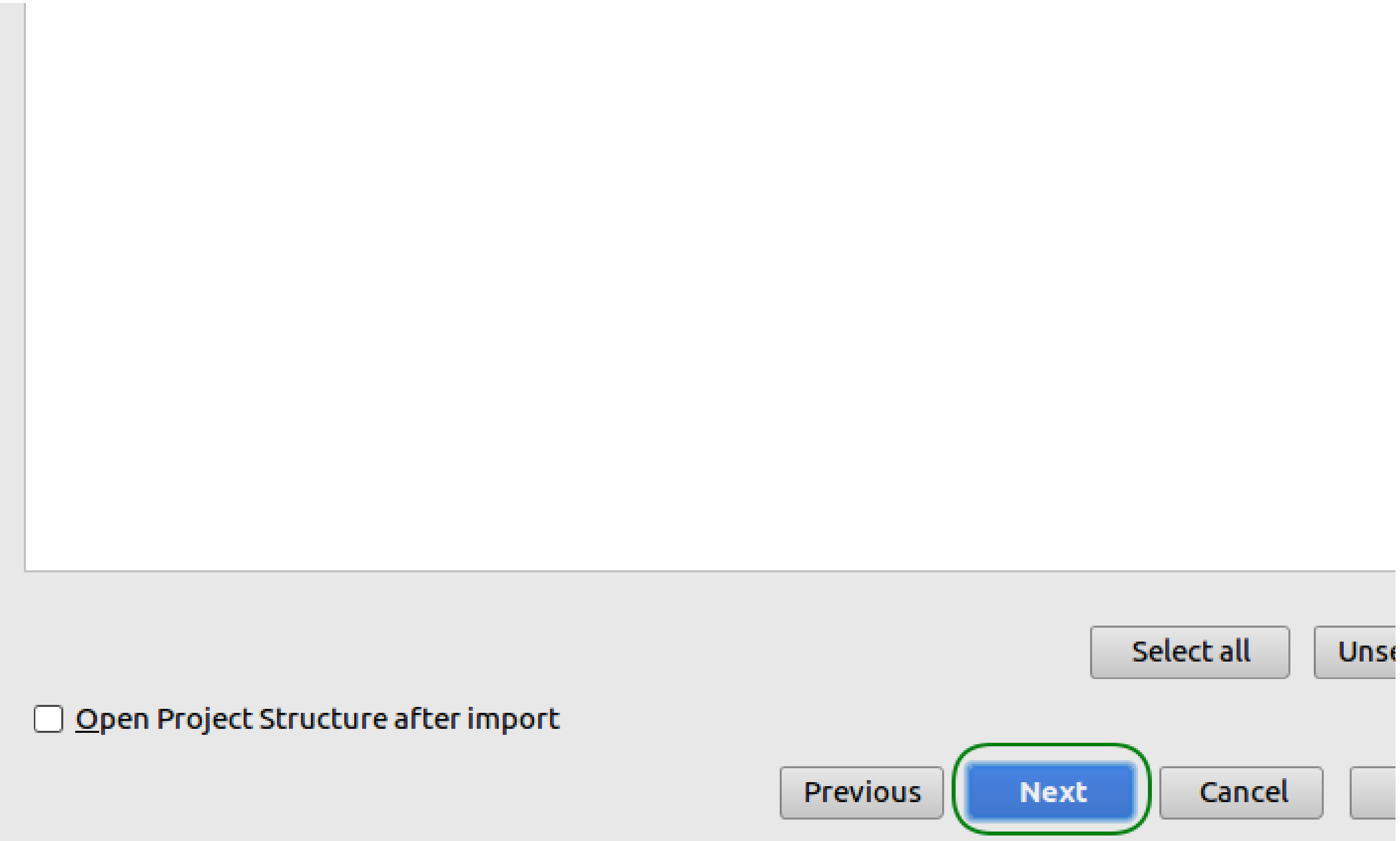
OpenJFX Profile

- Select the Maven projects in the project folder. Click Next.



Select Maven projects to import

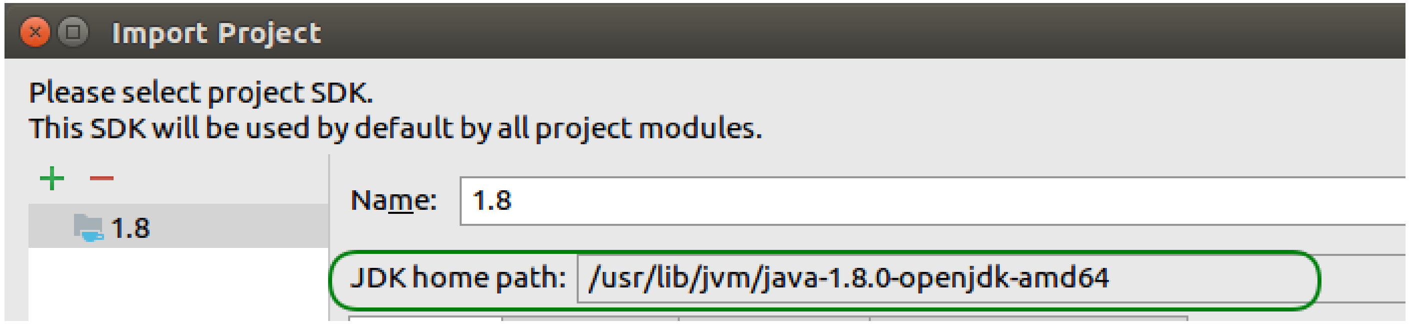
☑ org.deeplearning4j:deeplearning4j-examples-parent:0.8-SNAPSHOT



☐ Open Project Structure after import

Select Maven Projects from the project folder

- Select project SDK. Click Next.

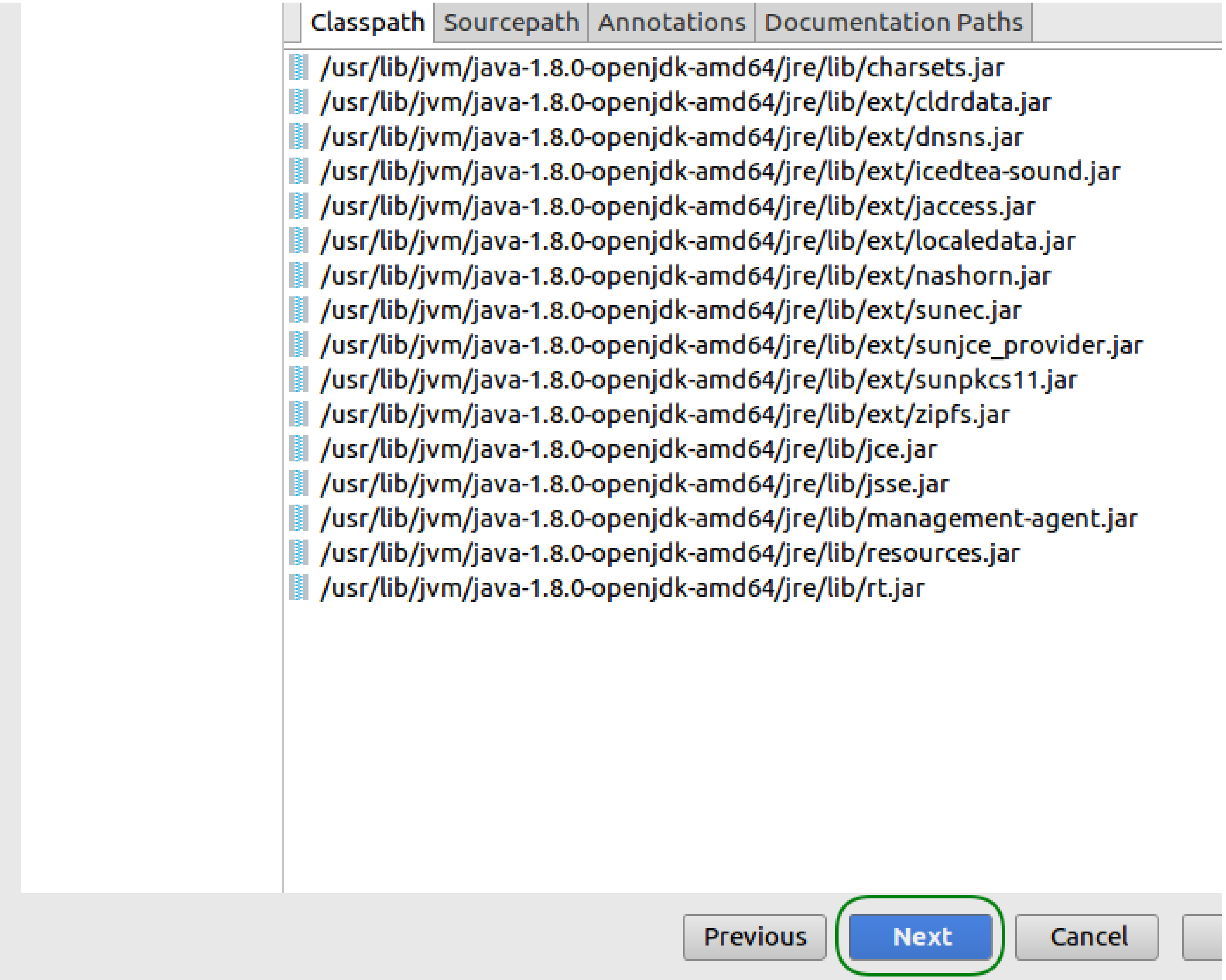


Import Project

Please select project SDK.
This SDK will be used by default by all project modules.

Name: 1.8

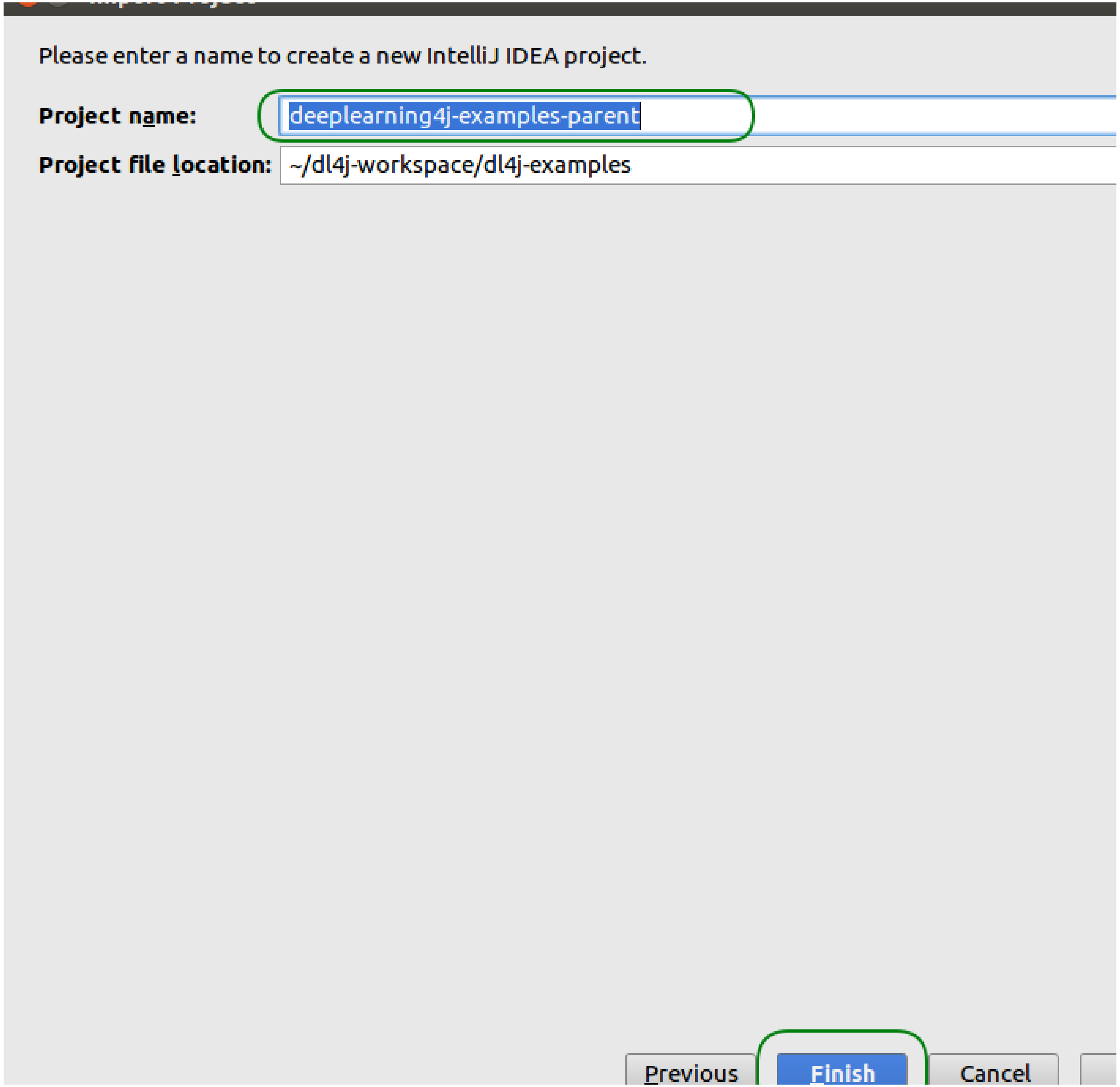
JDK home path: /usr/lib/jvm/java-1.8.0-openjdk-amd64



```
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/charsets.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/ext/cldrdata.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/ext/dnsns.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/ext/icedtea-sound.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/ext/jaccess.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/ext/localedata.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/ext/nashorn.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/ext/sunec.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/ext/sunjce_provider.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/ext/sunpkcs11.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/ext/zipfs.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/jce.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/jsse.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/management-agent.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/resources.jar
/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/rt.jar
```

Select SDK for Project

- The final step, give a name to the project. Click Finish.



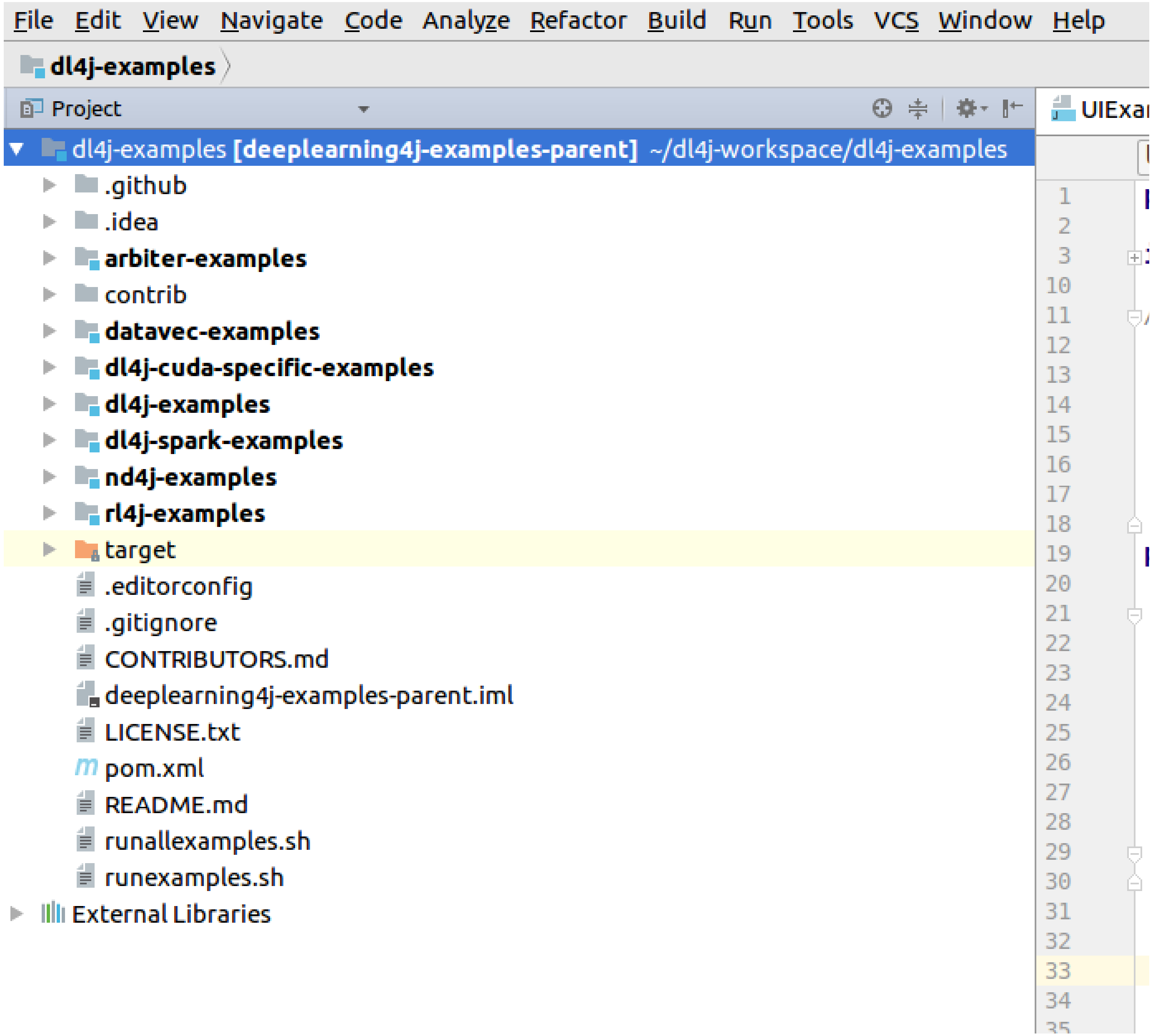
Please enter a name to create a new IntelliJ IDEA project.

Project name: deeplearning4j-examples-parent

Project file location: ~/dl4j-workspace/dl4j-examples

Give Project Name

- It takes some time for IntelliJ to index the resources in the project. And the project structure would look somet following screenshot.



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
dl4j-examples
Project
UIExa

```
dl4j-examples [deeplearning4j-examples-parent] ~/dl4j-workspace/dl4j-examples
  .github
  .idea
  arbiter-examples
  contrib
  datavec-examples
  dl4j-cuda-specific-examples
  dl4j-examples
  dl4j-spark-examples
  nd4j-examples
  rl4j-examples
  target
  .editorconfig
  .gitignore
  CONTRIBUTORS.md
  deeplearning4j-examples-parent.iml
  LICENSE.txt
  pom.xml
  README.md
  runallexamples.sh
  runexamples.sh
External Libraries
```

dl4j examples Project Structure

The project “dl4j-examples” is successfully setup.

Reference

Following is the official link of deeplearning4j quick start.

https://deeplearning4j.org/quickstart

Conclusion

In this Deep Learning tutorial, we have seen how to setup environment for Deep learning with De import the Git project “dl4j-examples” to IntelliJ IDEA.

❮ Previous

Download How to setup environment for Deep learning with Deeplearning4j ? ir