

INSTRUCTIONS

This document consists of three set of instructions,

SET - I - This set of instructions are to be followed and implemented for all the windows users. Also, this set of instructions consists of different versions required to be downloaded and used which can be common to all the users

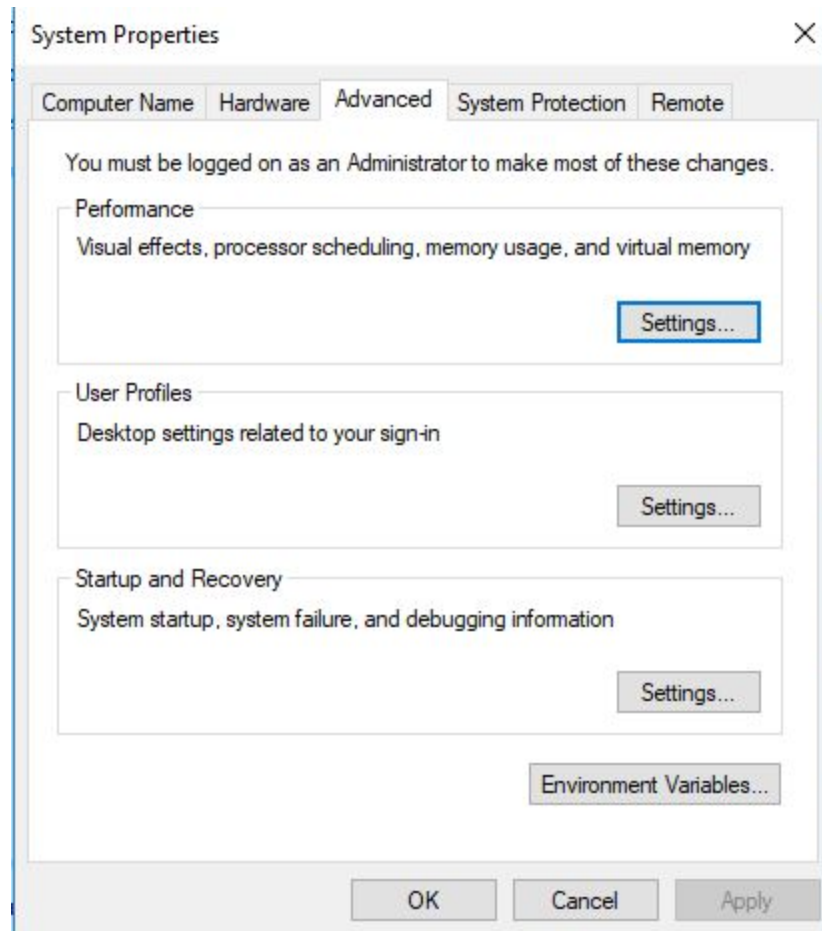
SET - II - This set of instructions are to be followed and implemented by all the users (MAC, Linux and Windows) while importing the project in IntelliJ.

SET III - This set of instructions are common for all the users but not mandatory, as mentioned in details, follow them only if you are facing the bug in observing the results on PGAdmin.

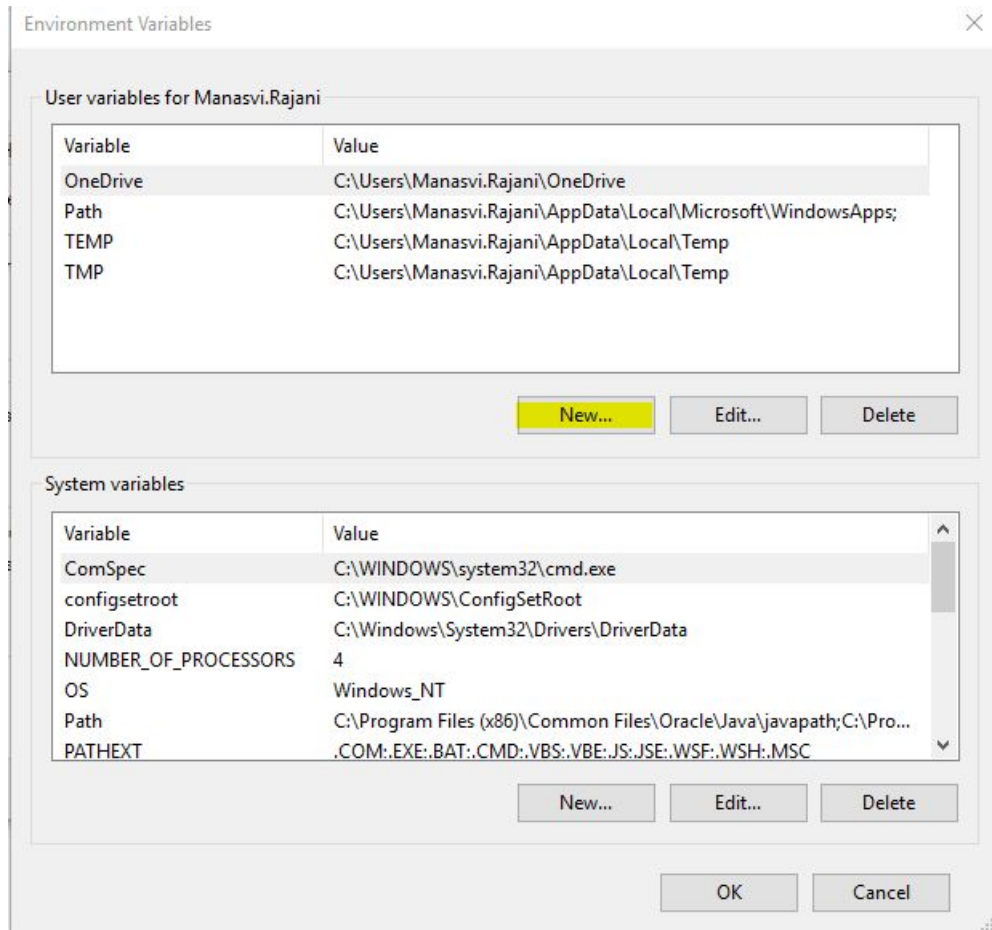
SET - I

1. Install IntelliJ
 - a. Download Ultimate edition of IntelliJ from [here](#) for windows. Check for the links around for other operating systems.
 - b. Install and configure.
2. Install Postgres.
 - a. Download version 9.5.14 from [this](#) page.
 - b. Install and continue.
 - c. Set username and password.
3. Install pgAdmin 4 V3.2 from [here](#).
 - a. Download the .exe file from the page.
 - b. Install which will also install visual studio by itself.
4. Install JDK from Java SE Development for your windows from [here](#).
 - a. Install.
5. Install Maven from [here](#).
 - a. Download it from the “Binary zip archive” link on the page.
6. Set Environment Variables for jdk.

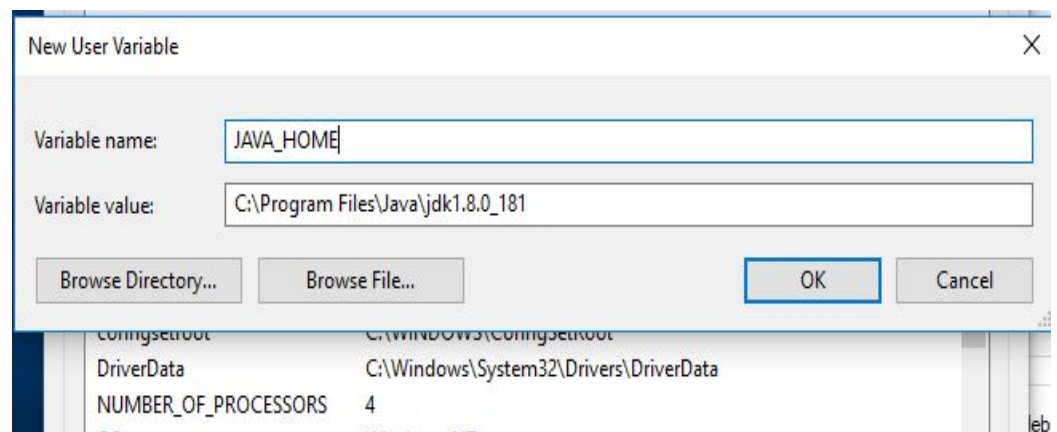
- a. Open Control panel-> System and Security -> System -> Advanced System Settings.
- b. You will observe the following popup window



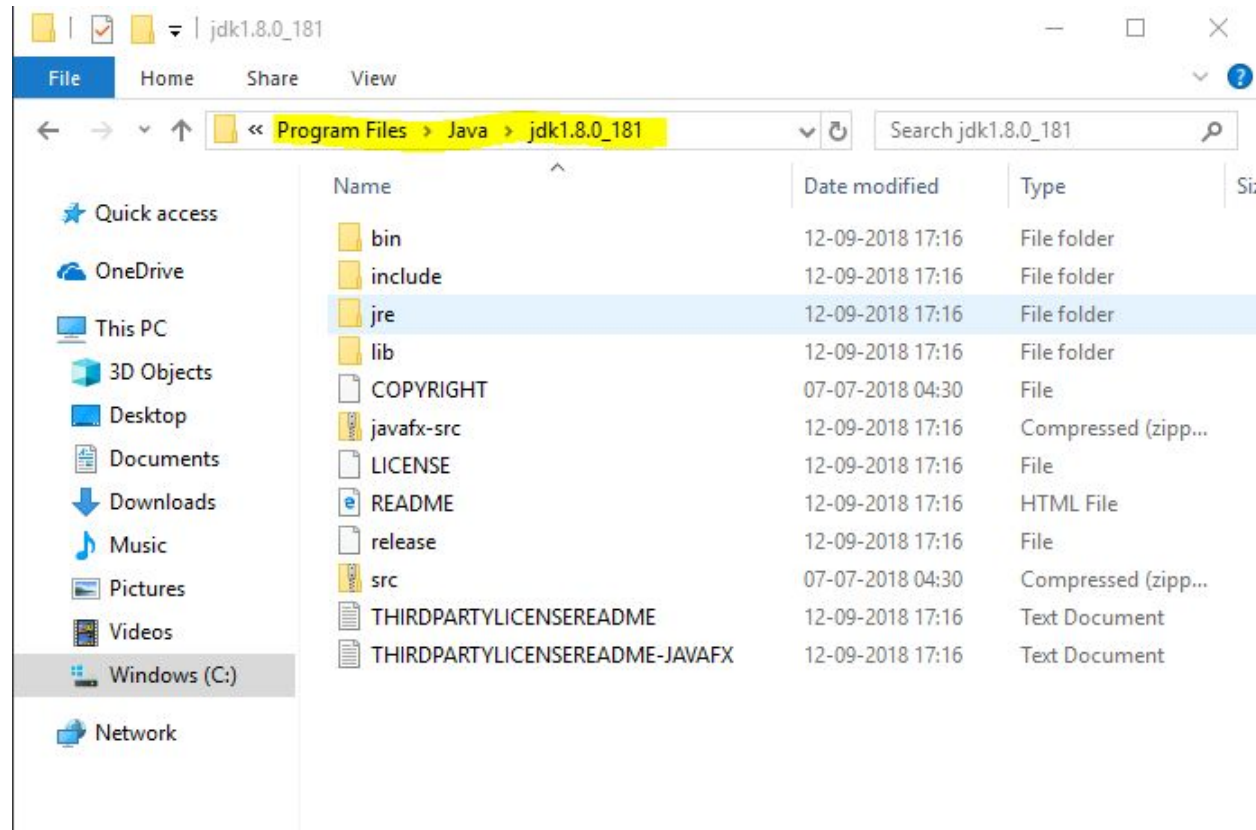
- c. Click on the 'Environment Variables..'
- d. Click on New(Highlighted portion)



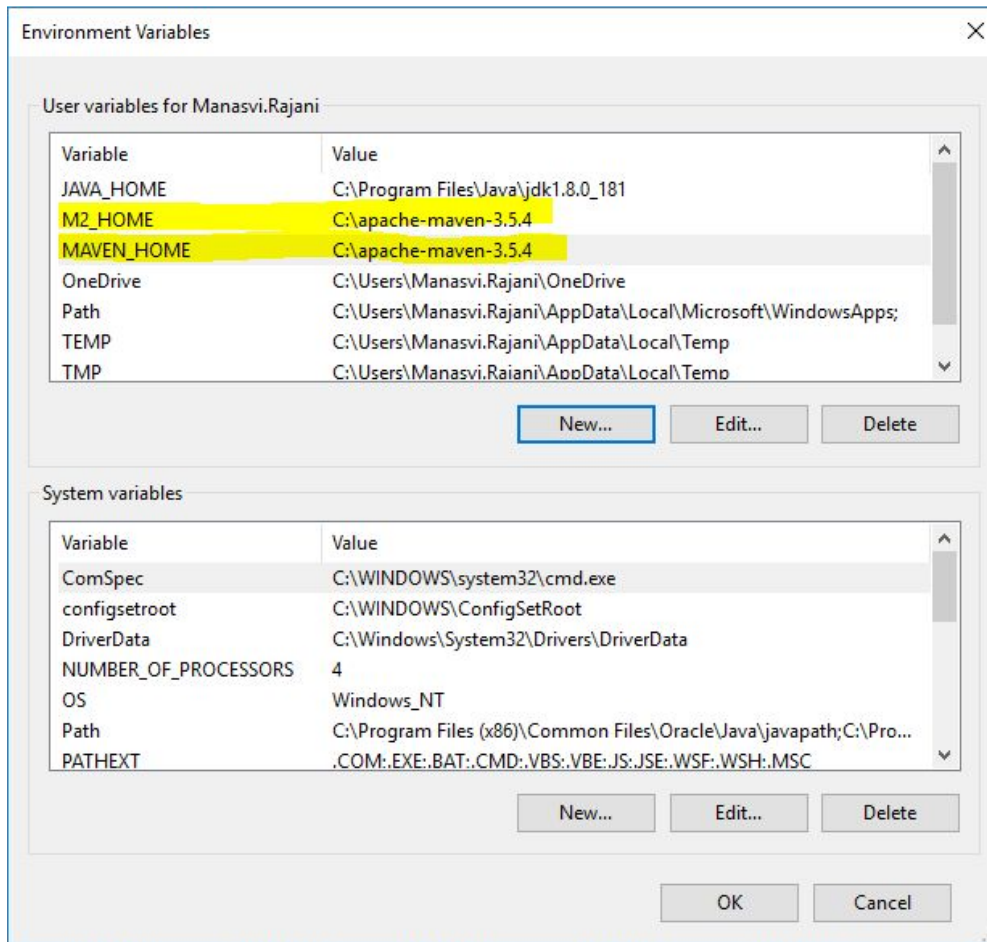
e. You can observe the following window pops up



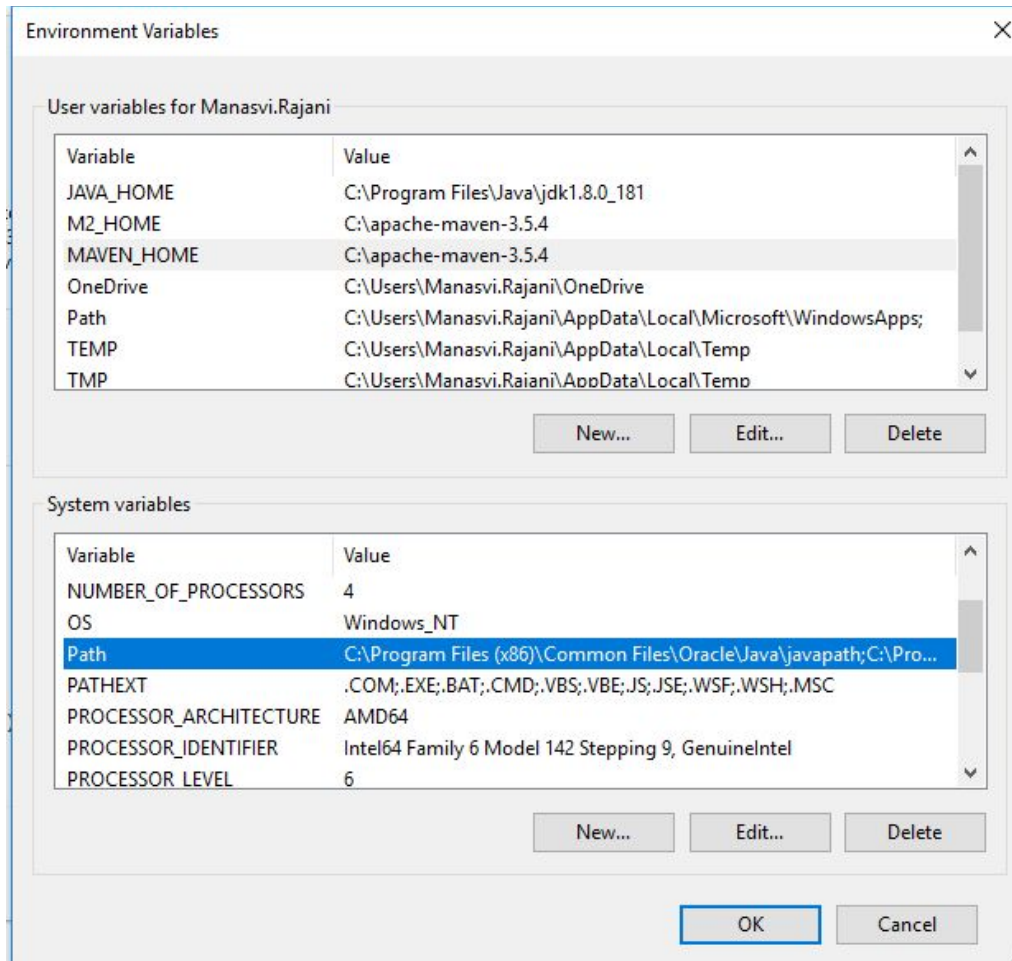
- f. Set the Variable name as JAVA_HOME and variable value as the location of jdk.
- i. Location for jdk can be seen in the image - C:\Program Files\Java\jdk1.8.0_181



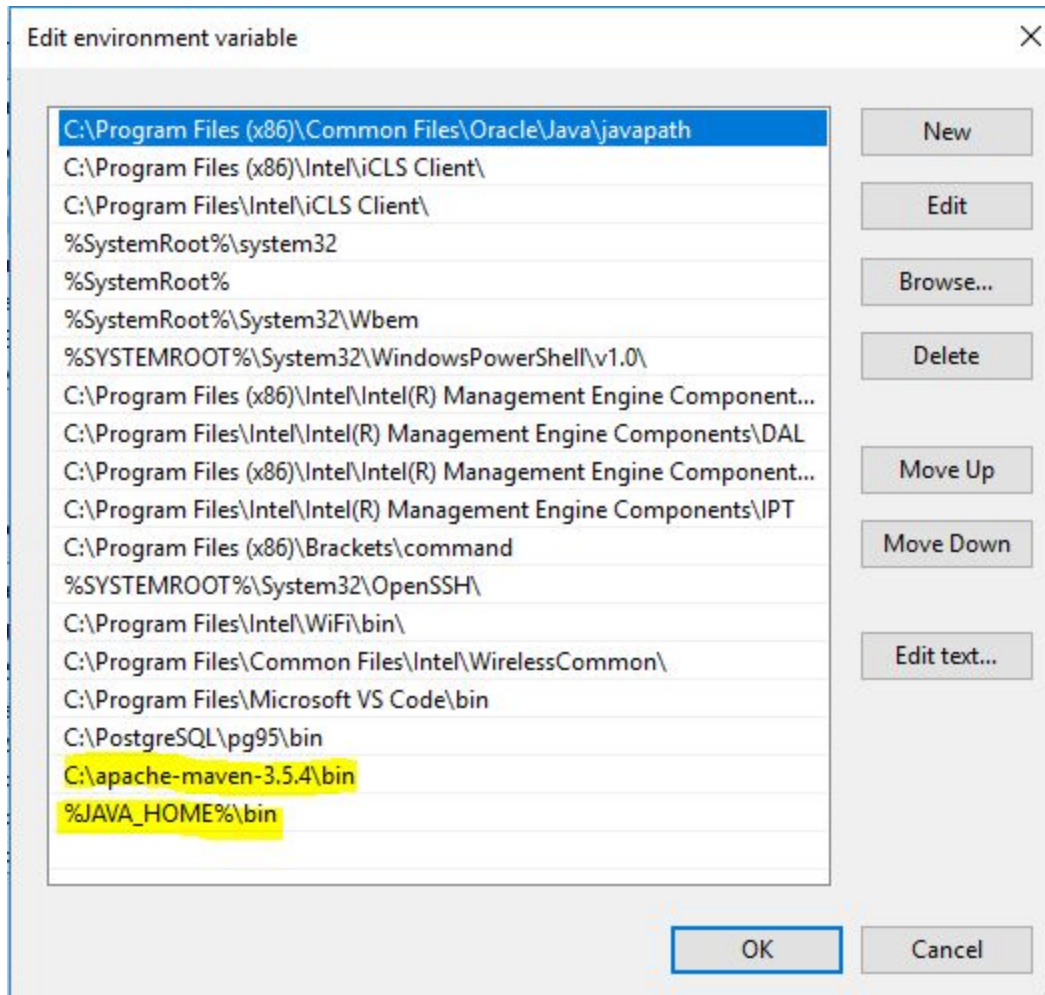
7. Similarly Set Environment Variables for Maven as shown



8. Now click on the path in “System variables” and click edit as shown but please note that the value for your path may be different in the above image.



9. Add the highlighted portion as shown in the below image,



10. In case, if you are facing any issue with the above instructions, please follow [this](#) link if you are facing some issue.
11. Download PostgreSQL JDBC 4.2 Driver, 42.2.5 from [here](#). Note that after you download the JDBC driver, you will also need to add this external JAR file in the IntelliJ IDEA project. How?
 - a. Click File from the toolbar.
 - b. Click on Project Structure.
 - c. Select Modules at the left panel.
 - d. Dependencies tab
 - e. '+' -> JARs or directories
 - f. Add the downloaded JAR file and apply the changes.

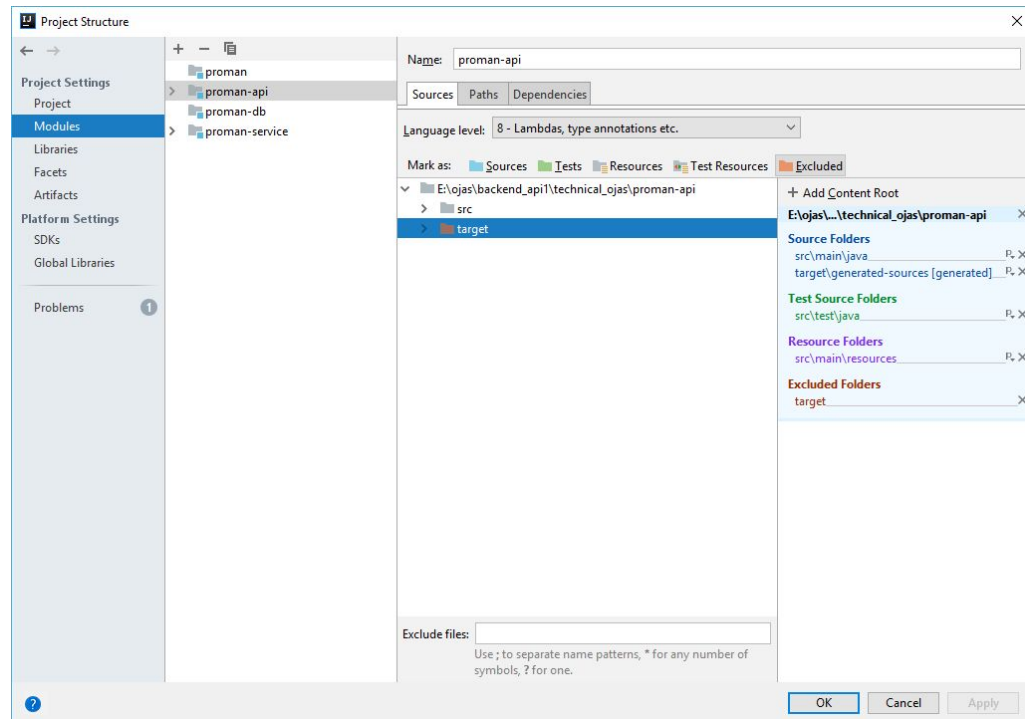
12. Start implementing the Project as shown in the Module.
13. After you complete the implementation of the Project, open the terminal and build the project in IntelliJ with the command 'mvn clean install -DskipTests'.
14. Also, run the command 'mvn clean install -Psetup' to run the given SQL files against the database.
15. Run the project and open it on the localhost.

Special Instructions:

1. Note that when you generate the Request and Response models from the JSON file, they are generated in the 'target' folder and the generated target folder may normally be excluded from being shown in Project Explorer (if it appears orange in colour). Then you need to include this folder as a part of the project.

How?

- a. Click File from the toolbar.
- b. Click on Project Structure.
- c. Select Modules at the left panel.
- d. Click on Sources tab
- e. Select proman-api submodule (Request and Response models are generated in this submodule).
- f. Select the target folder for proman-api.



Note that in the above screenshot, the target folder is Excluded.

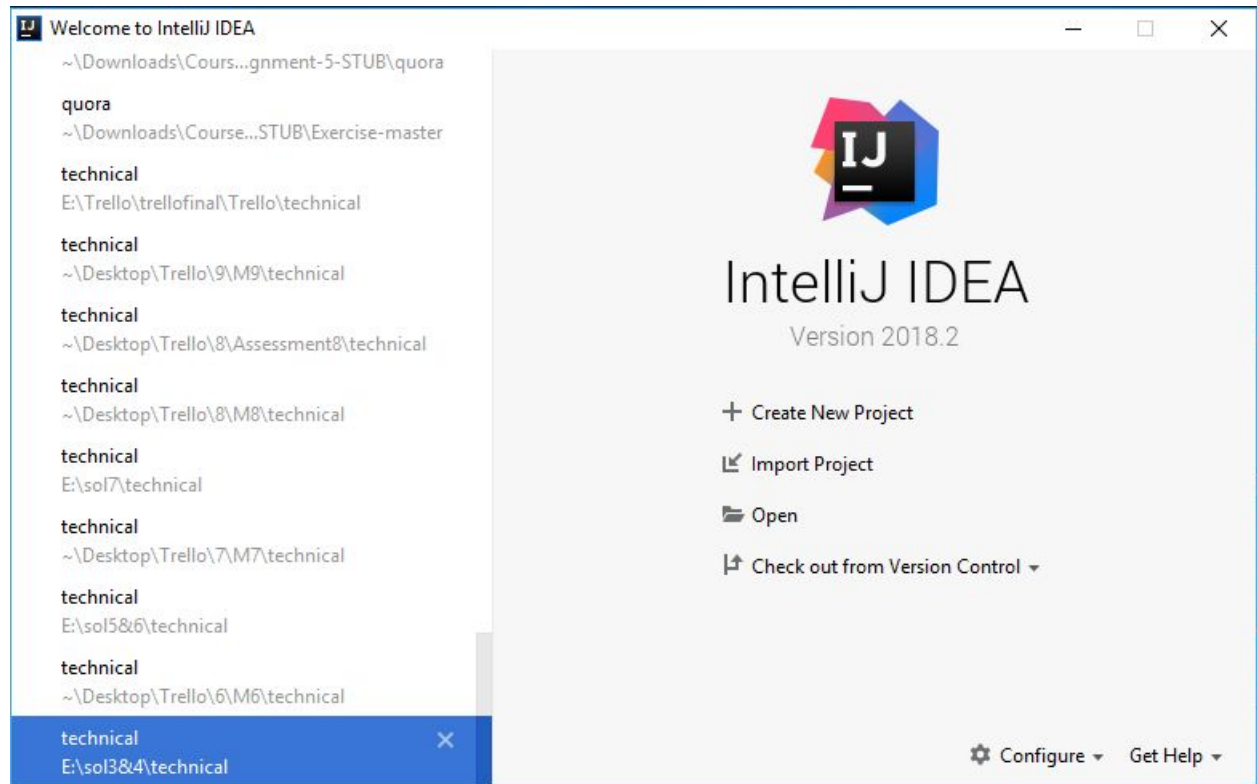
- g. Click on the 'Excluded' tab and unselect it, then click on the Sources to select and apply the changes.
2. It may happen that the library 'postgresql-42.2.5.jar!' is not added to your classpath. You need to open 'application.yaml' file after you implement the project. Press alt+enter on driver class name 'org.postgresql.Driver' and add the library 'postgresql-42.2.5.jar!' to your classpath.

SET - II

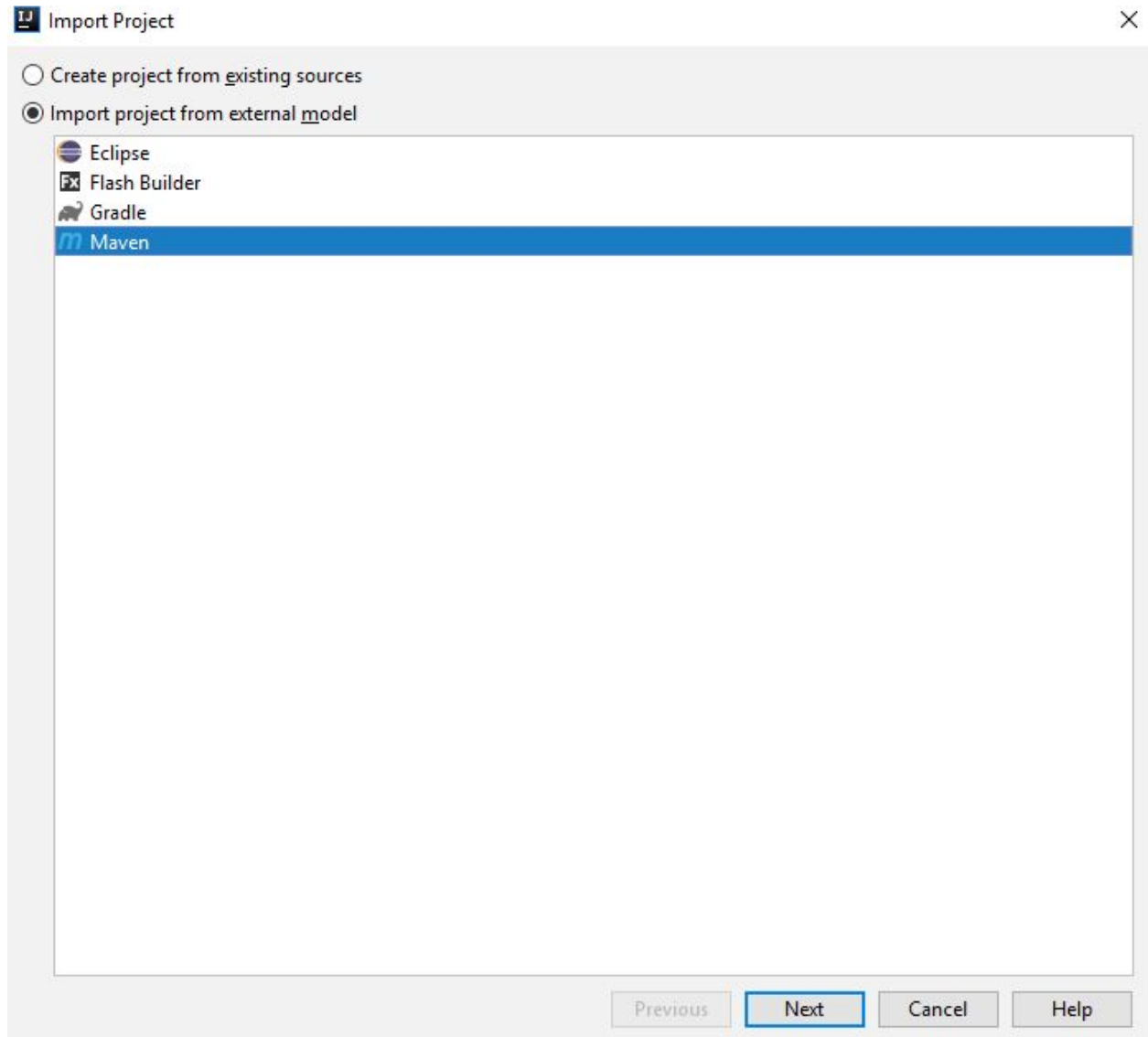
How to import the project given in the stub file?

Do not open the project directly. You need to import the project.


1. Open IntelliJ



2. Import the given Proman project.
3. Import project from external model and select maven.



4. Check that you have the same settings.

 Import Project ✕

Root directory: ...

☐ Search for projects recursively

Project format: ▼

☐ 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: ▼

Phase to be used for folders update: ▼

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

Automatically download: ☒ Sources ☒ Documentation

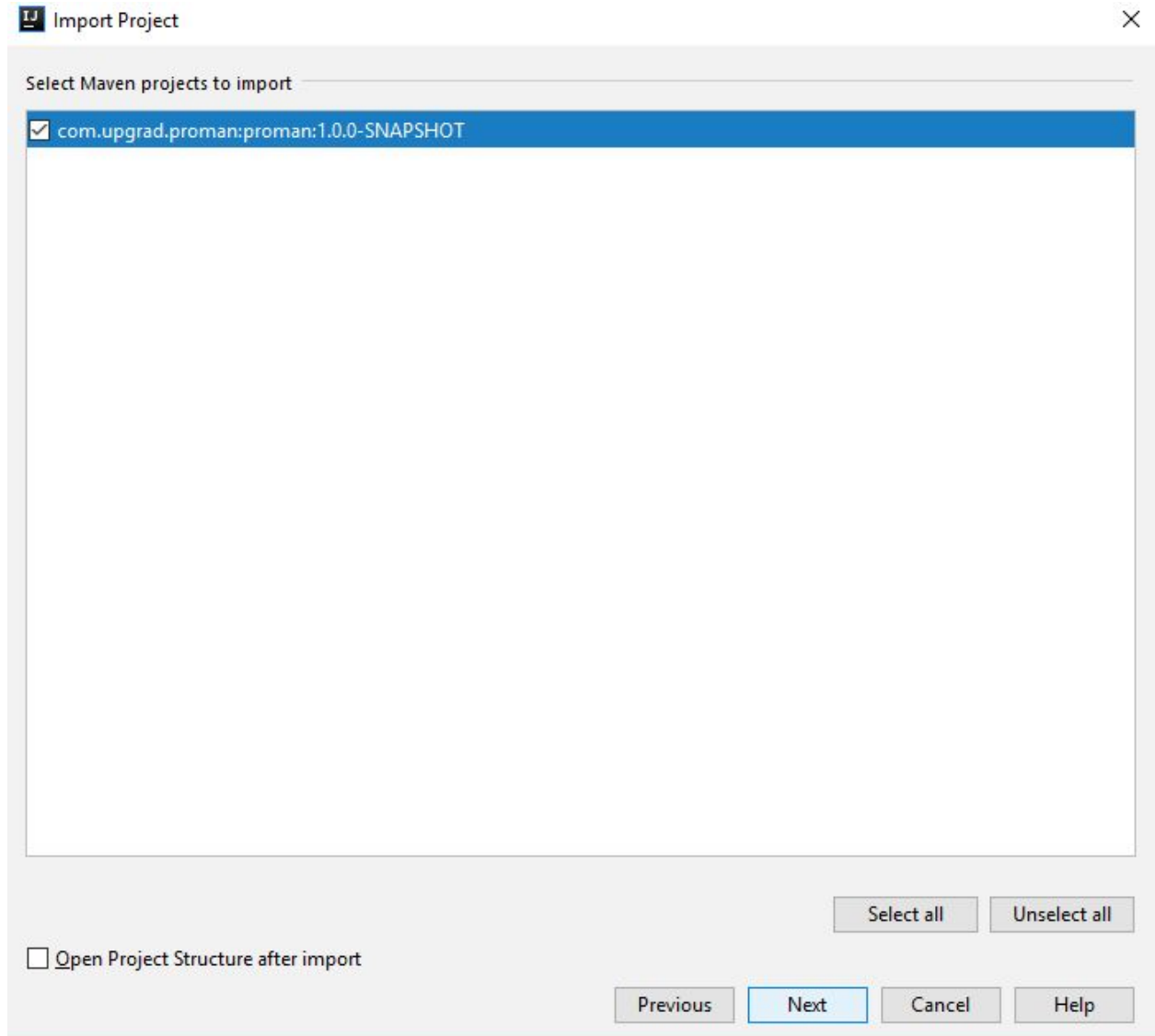
Dependency types:

Comma separated list of dependency types that should be imported

Environment settings...

Previous **Next** Cancel Help

5. Select Maven projects to import.



6. Select the project SDK. Select the home directory of JDK.

Import Project



Please select project SDK.

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

+ -

1.8

Name: 1.8

JDK home path: C:\Program Files\Java\jdk1.8.0_181



Classpath Sourcepath Annotations Documentation Paths

C:\Program Files\Java\jdk1.8.0_181\jre\lib\charsets.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\deploy.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\ext\access-bridge-64.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\ext\clldrdata.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\ext\dnsns.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\ext\jaccess.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\ext\jfxrt.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\ext\localedata.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\ext\nashorn.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\ext\sunec.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\ext\sunec_provider.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\ext\sunmscapi.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\ext\sunpkcs11.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\ext\zipfs.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\javaws.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\jce.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\jfr.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\jfxswt.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\jsse.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\management-agent.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\plugin.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\resources.jar
C:\Program Files\Java\jdk1.8.0_181\jre\lib\rt.jar

+

-

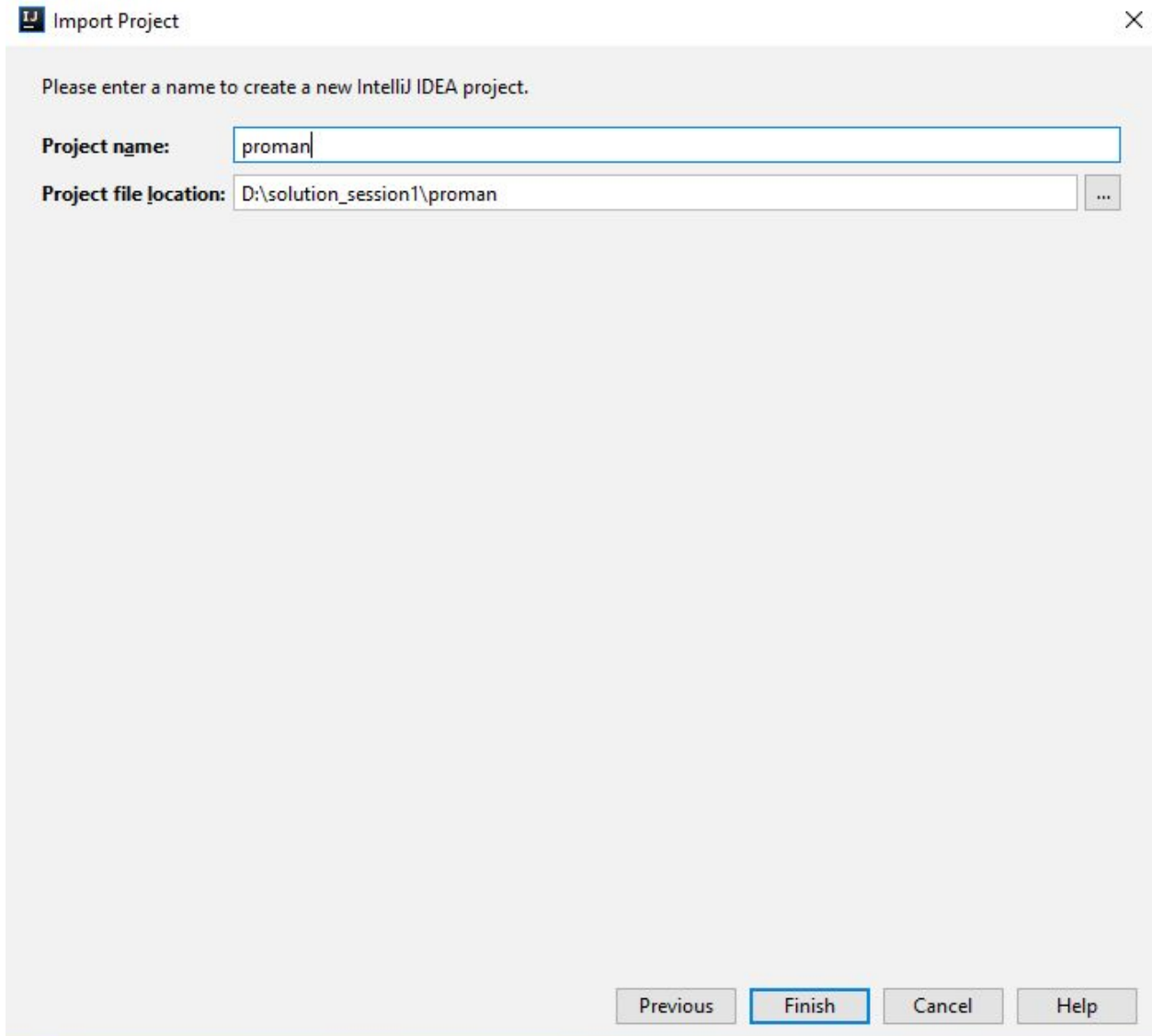
Previous

Next

Cancel

Help

7. Enter the project name as 'proman'. Strictly use the same project name.



The image shows the 'Import Project' dialog box in IntelliJ IDEA. The dialog has a title bar with the IntelliJ logo and the text 'Import Project', and a close button (X) in the top right corner. The main area contains the instruction 'Please enter a name to create a new IntelliJ IDEA project.' Below this, there are two input fields. The first is labeled 'Project name:' and contains the text 'proman'. The second is labeled 'Project file location:' and contains the text 'D:\solution_session1\proman'. To the right of the second field is a button with three dots (...). At the bottom of the dialog, there are four buttons: 'Previous', 'Finish' (which is highlighted with a blue border), 'Cancel', and 'Help'.

Import Project

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

Project name: proman

Project file location: D:\solution_session1\proman ...

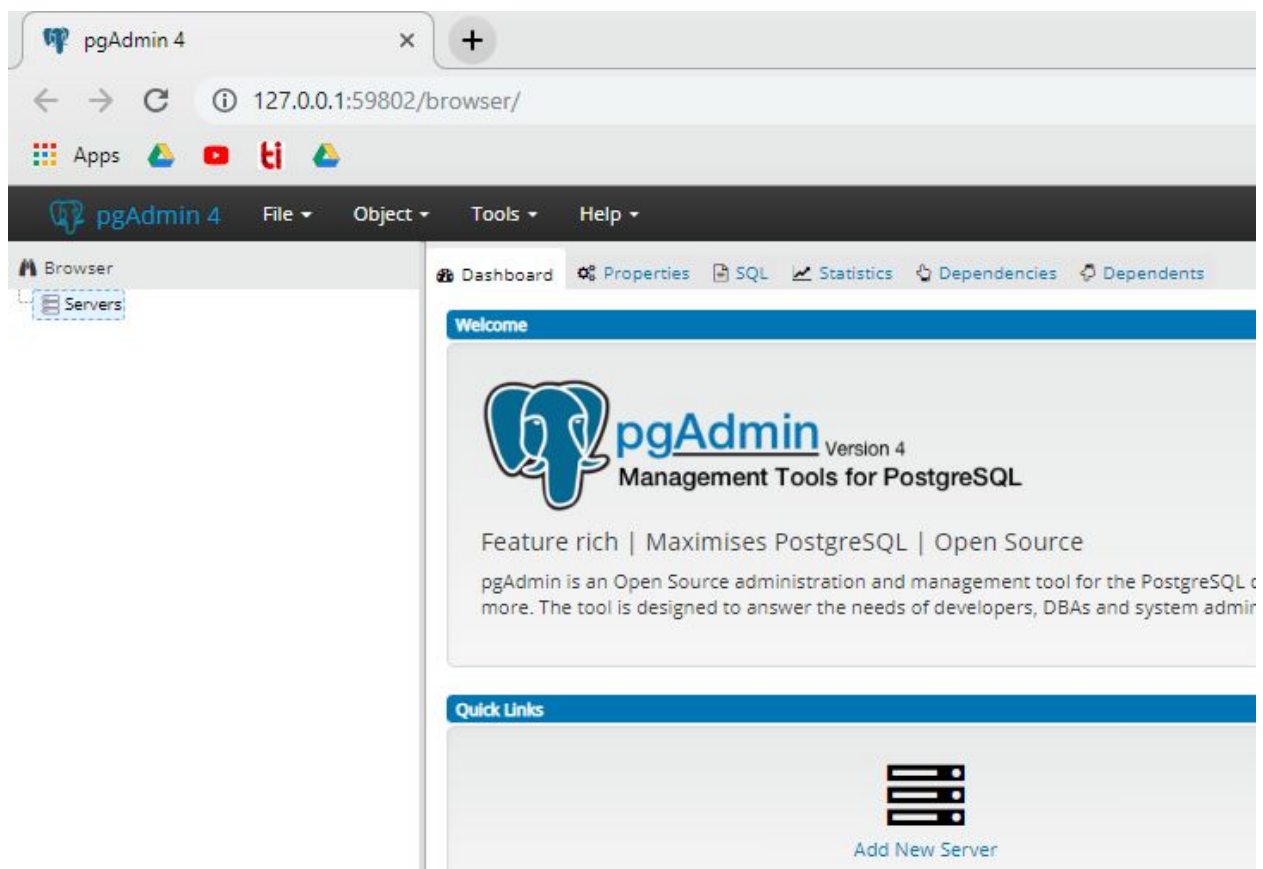
Previous Finish Cancel Help

SET - III

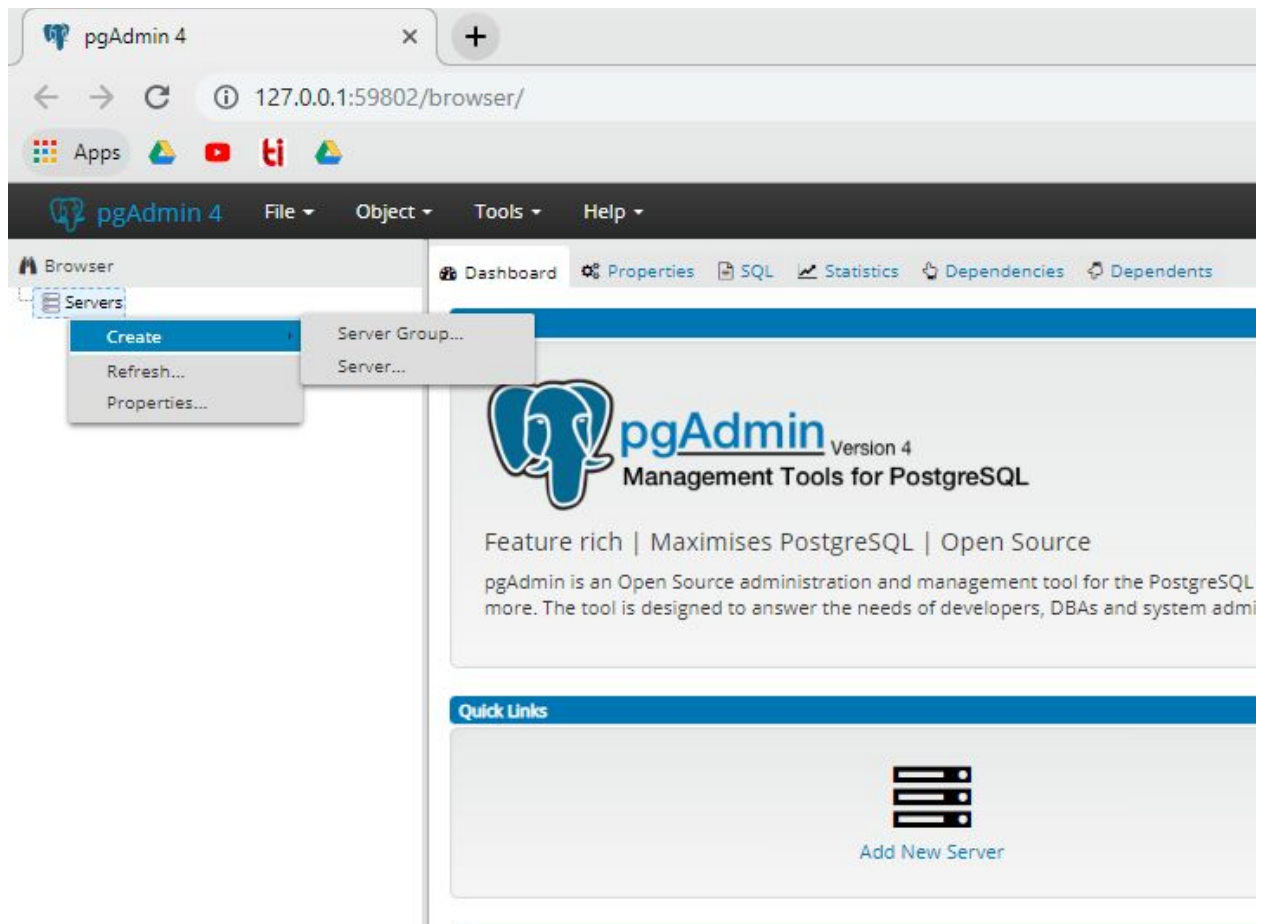
This could be a possible expected bug but not a mandatory change required.

When you create the database, the build is successful but if you do not observe the PostgreSQL9.5 server created in the PGAdmin UI. Please follow the below instructions,

Once after you run the command “mvn clean install -Psetup” and you can observe the SQL commands are run to create the tables. But If you do not observe a server entry for PostgreSQL9.5 in pgAdmin, you need to manually add a server connection. In the below image there is no entry for PostgreSQL9.5 under servers in pgAdmin.



Click Servers -> Create -> Server



Set the name of the server as PostgreSQL9.5 in the General tab.

Create - Server

General | Connection | SSL | SSH Tunnel | Advanced

Name: PostgreSQL9.5

Server group: Servers

Background: ☒

Foreground: ☒

Connect now?: ☒

Comments:

Either Host name, Address or Service must be specified.

Save Cancel Reset

Set the hostname as 'localhost' and provide the details of port, username and password according to your PostgreSQL database. Click save and the server will be added in the pgAdmin.

The image shows a 'Create - Server' dialog box with the 'Connection' tab selected. The fields are filled with the following values:

Field	Value
Host name/address	localhost
Port	5432
Maintenance database	postgres
Username	postgres
Password
Save password?	<input type="checkbox"/>
Role	
Service	

At the bottom, there are buttons for 'Save', 'Cancel', and 'Reset', along with information and help icons.

If you have already run the database command 'mvn clean install -Psetup', once the PostgreSQL9.5 server is created you should be able to observe the information with respect to the three tables i.e., user table, user_auth_token table, role table

