EPI USE Interns Technology Stack Setup Tutorial

Overview:

1.Github

2.Postgresql

3.Wildfly

4.JEE

5. Maven

1.Github

1. go to <https://github.com/> and create a new account.

2. go to <http://gitforwindows.org/> and download git bash

Git commands:

NB!!! ALWAYS PULL BEFORE YOU PUSH. This ensures you have the latest version and you are not working on an older version of a project.

|  |
| --- |
| DOWNLOAD GIT BASH |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| git commands: |
|  |

|  |
| --- |
| make sure pwd is desktop cd ../Desktop |
|  |

|  |
| --- |
| step 1 : git clone https://github.com/DawiePritchard/Brogramming.git |
|  |

|  |
| --- |
| step 2 : cd <repository> brogramming |
|  |

|  |
| --- |
| step 3 : pwd |
|  |

|  |
| --- |
| step 4: git fetch(all remote branches) |
|  |

|  |
| --- |
| step 5: git checkout <branch> (development(Switched to development branch)) |
|  |

|  |
| --- |
| step 6: git branch(list of local branches) |
|  |

|  |
| --- |
| step 7: git branch <branch name> (create branch) |
|  |

|  |
| --- |
| step 8: touch dawie.txt |
|  |

|  |
| --- |
| step 9: ls |
|  |

|  |
| --- |
| step 10: git status |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #Comment on step 11 |
|  |

|  |
| --- |
| #step 11: git add . (green means staging area) always before commit |
|  |

|  |
| --- |
| #git add . is not such a good idea since it adds any and all new files, that may include temporaries and unintended files as well. |
|  |

|  |
| --- |
| #It is better to use 'git add -u', which only looks at existing files that are being tracked, which means only already tracked file's changes will be commited which is a good thing. Now, if you do decide git must track a new file, you can explicitly add it using 'git add thefile'. After running 'git add thefile', whenever you modify or delete it and call 'git add -u', git will stage that as part of the commit for you. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| step 12: git commit -m "<commit message>" (initial commit) |
|  |

|  |
| --- |
| step 13: important note if you changed stuff on a branch then git wont let you switch until you have committed |
|  |

|  |
| --- |
| #comment on step 13: unless you stash the changes, then you can change branches without having to commit anything. |
|  |

|  |
| --- |
| step 14: git push |
|  |

|  |
| --- |
| #comment on step 14: As a matter of taste, one could be more explicit in the way you push changes. For example, if experimental has changed and you want to push changes to the remote master branch you can call 'git push origin experimental', where origin indicates the remote branch. This higher granularity is usefull since you are always aware of what branch you intend to push changes, thereby avoiding accidental changes to master when that was not intended. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| IMPORTANT : ALWAYS PULL BEFORE YOU PUSH TO REMOTE |
|  |

|  |
| --- |
| step 15: git pull |
|  |

|  |
| --- |
| #comment on step 15: 'git pull' has a similar caveat of granularity as step 14. "See https://stackoverflow.com/questions/15797183/differences-between-git-pull-commands-when-pulling-from-origin" for more information on this. |
|  |

|  |
| --- |
|  |
|  |

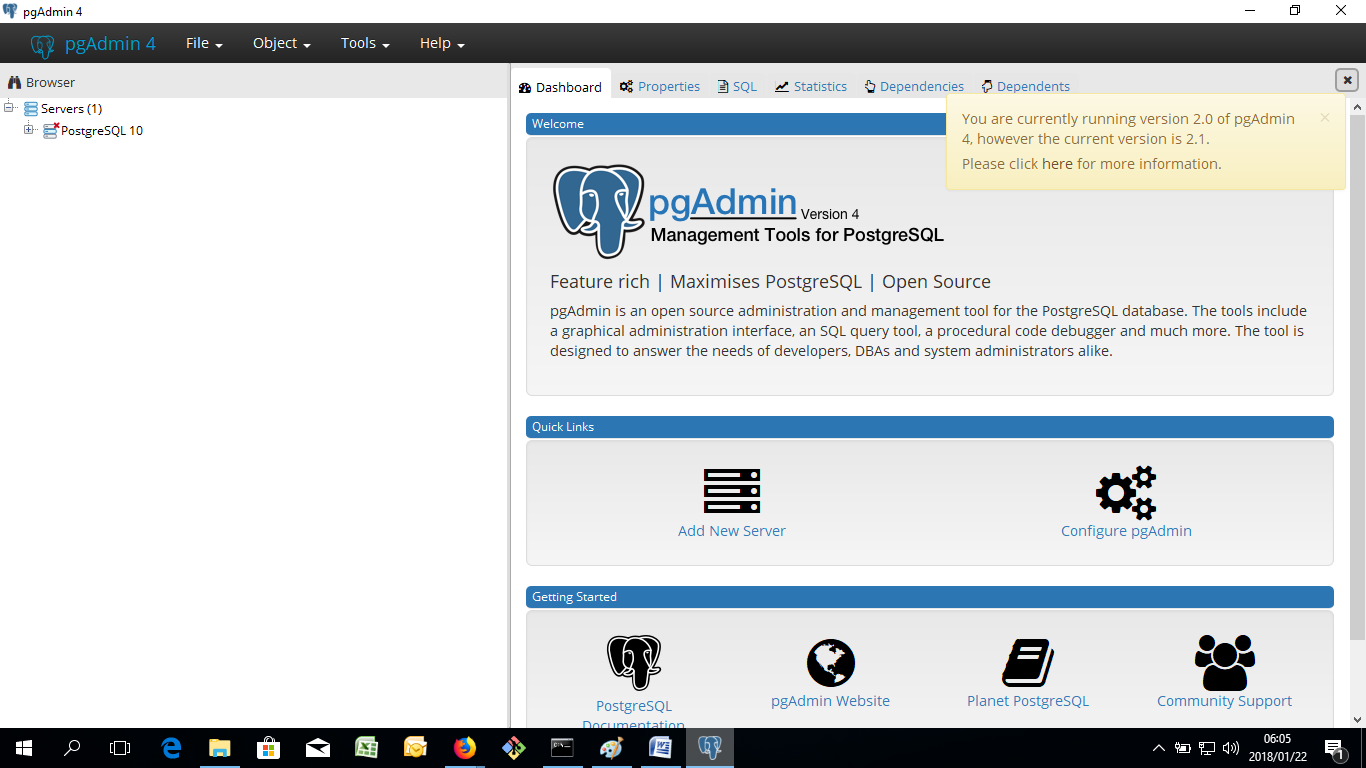
|  |
| --- |
| How merge works: |
|  |

|  |
| --- |
| switch to branch where you want to merge in(test stuff into dev so dev is branch) |
|  |

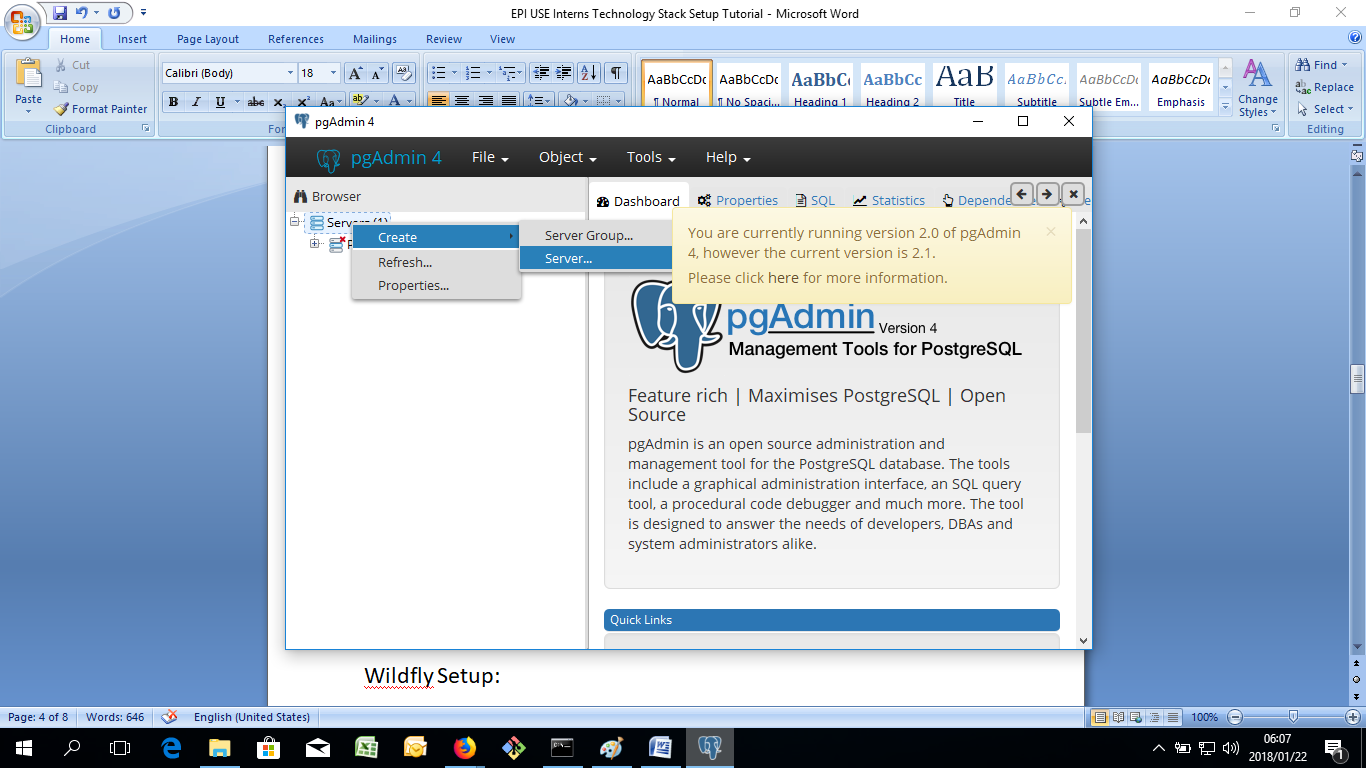
git merge testBranch

2. POSGRESQL

Step 1: Download and install postgresql at <https://www.postgresql.org/download/windows/> note that you do not have to install Stack Builder!

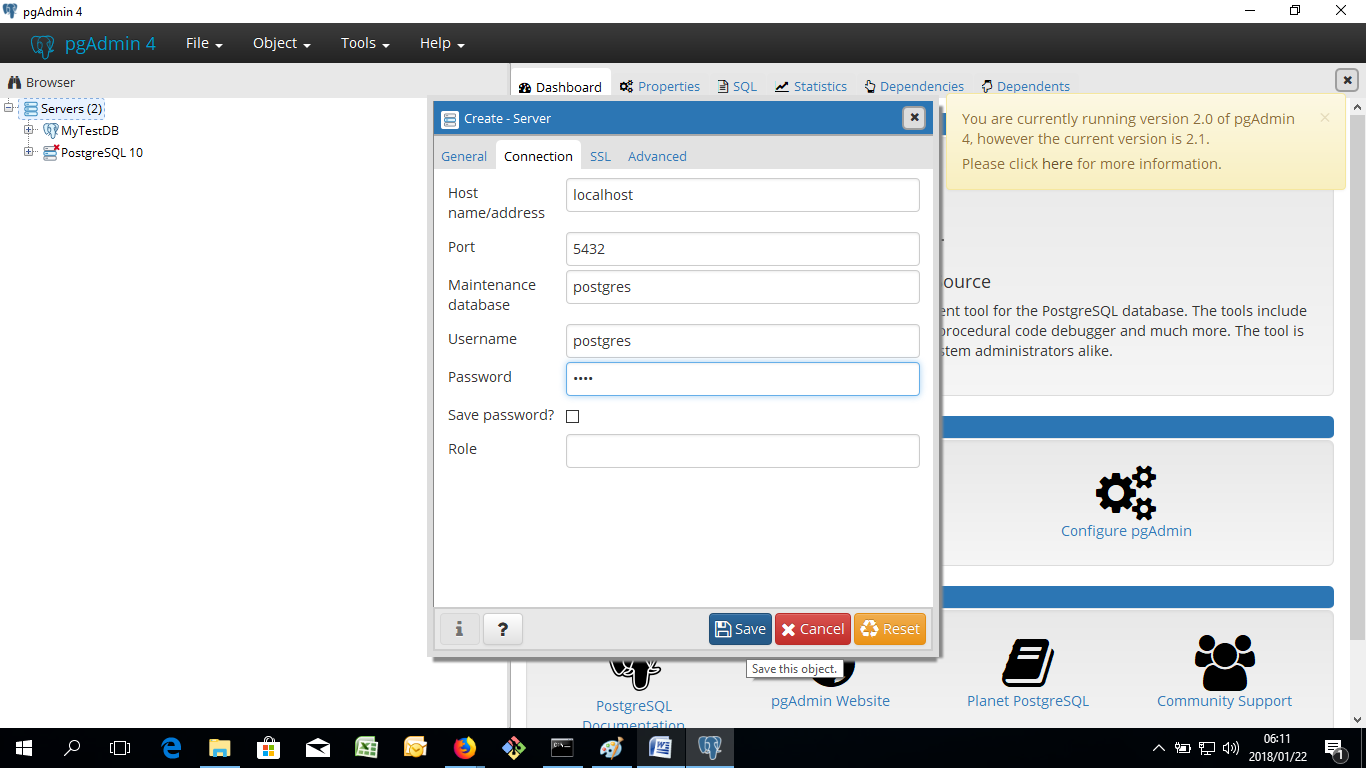
Step 2: After its done installing open up PGADMIN 4 on your desktop 

Step 3: Right click on Servers and select “Create/Server”



Step 4: Underneath the general tab, specify the name as “MyTestDB”

Step 5: Go to the “Connection Tab” and specify the host as “localhost”, the maintenance database as: “postgres” the username as “postgres” and password as “1234” and click on save

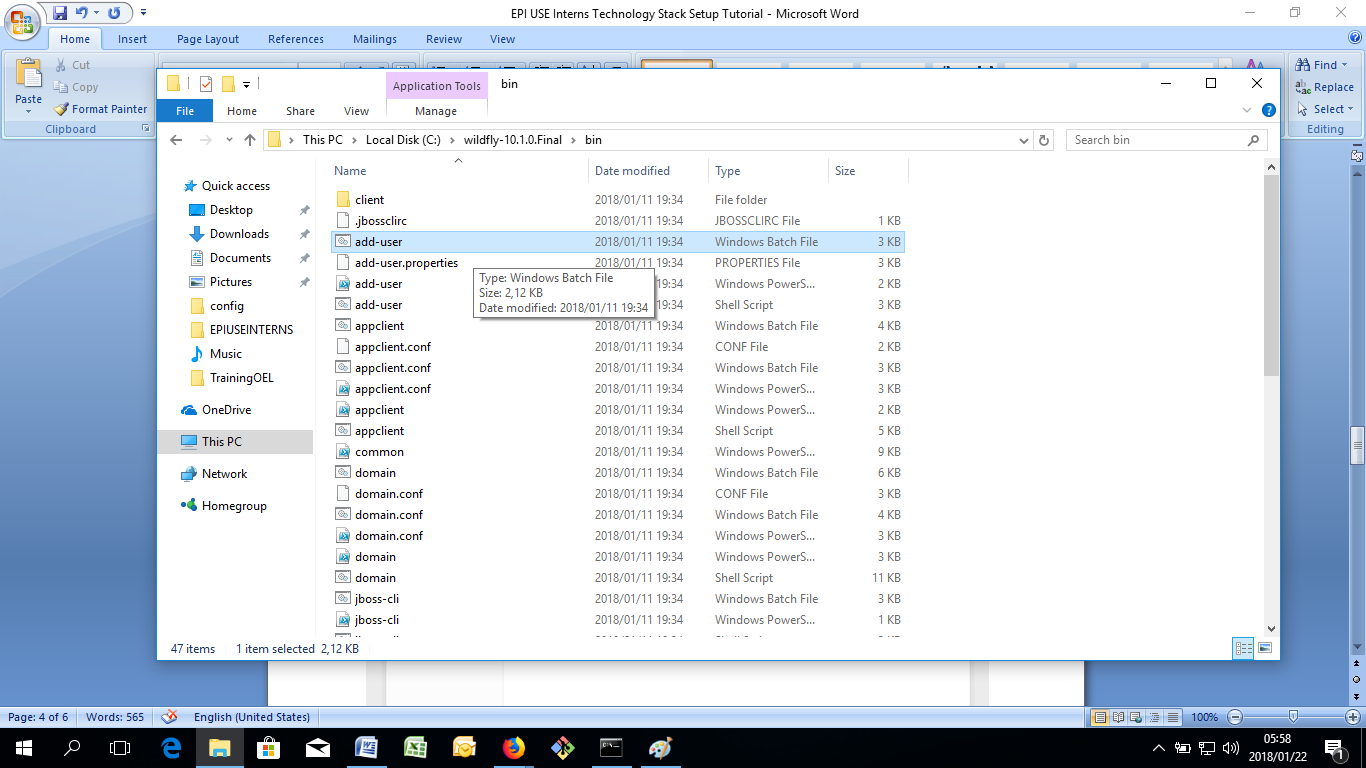


3. Wildfly Setup:

Step 1 : Download and Extract Wildfly

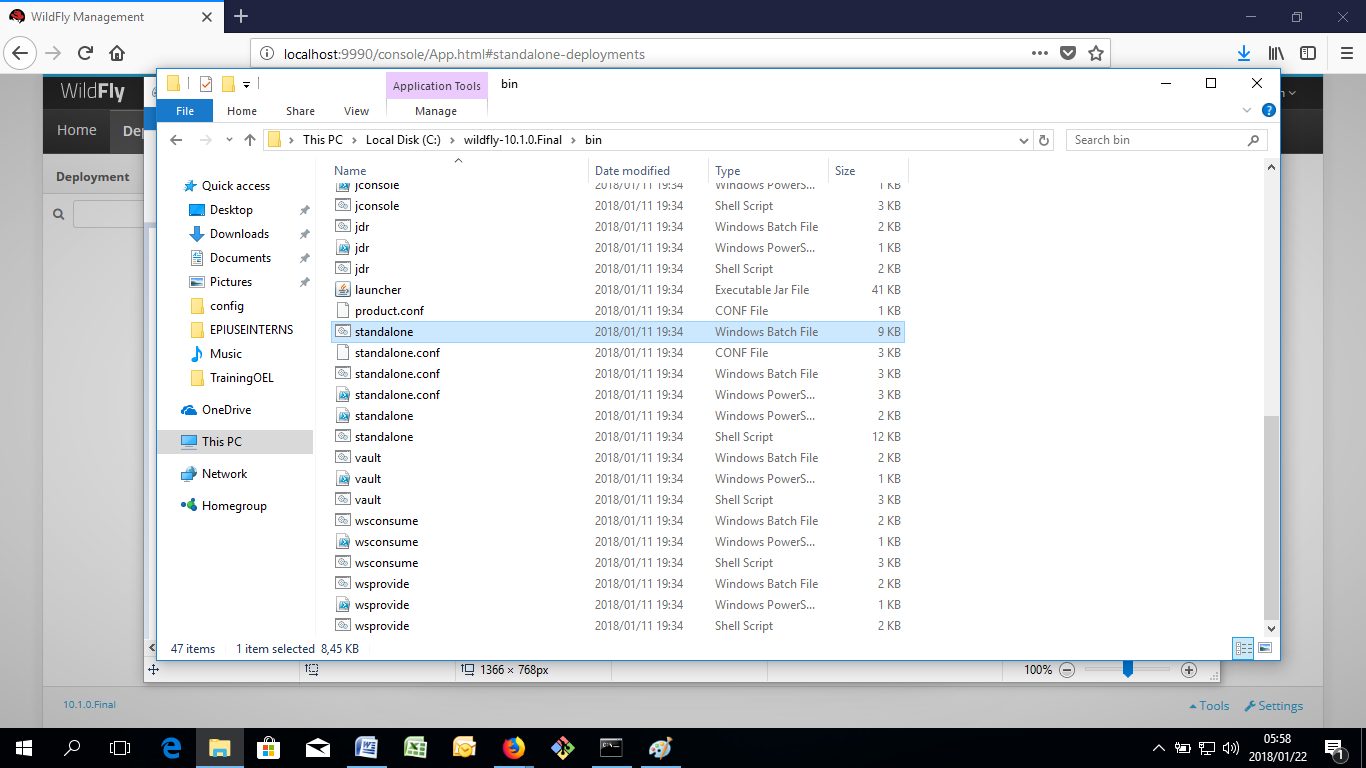
Step 2: Go to <https://jdbc.postgresql.org/download.html> and download the postgresql jdbc driver

Step 2: go to your wildfly directory to the /bin folder and execute add-user.bat



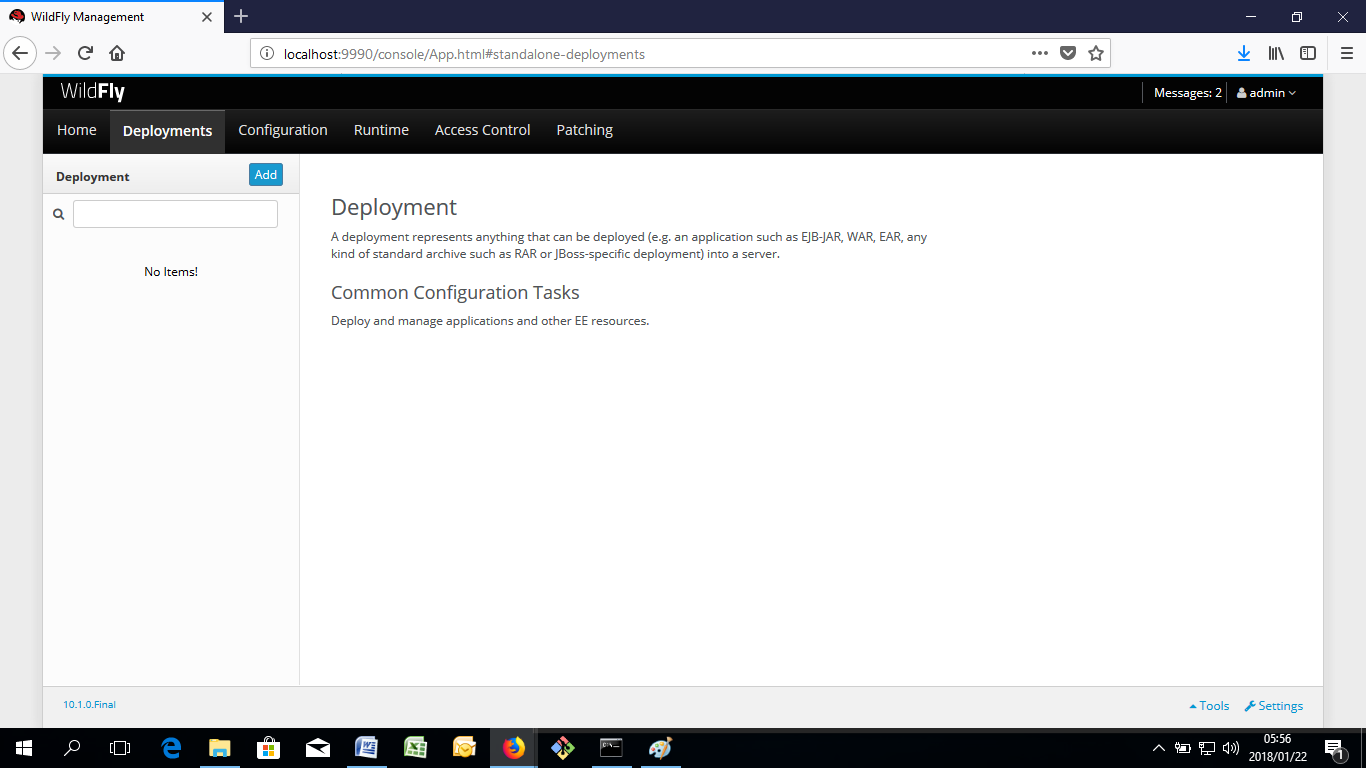
Step 3: follow the steps and add a management user

Step 4: go wildfly directory in the bin folder and execute “standalone.bat” this will start up wildfly for you

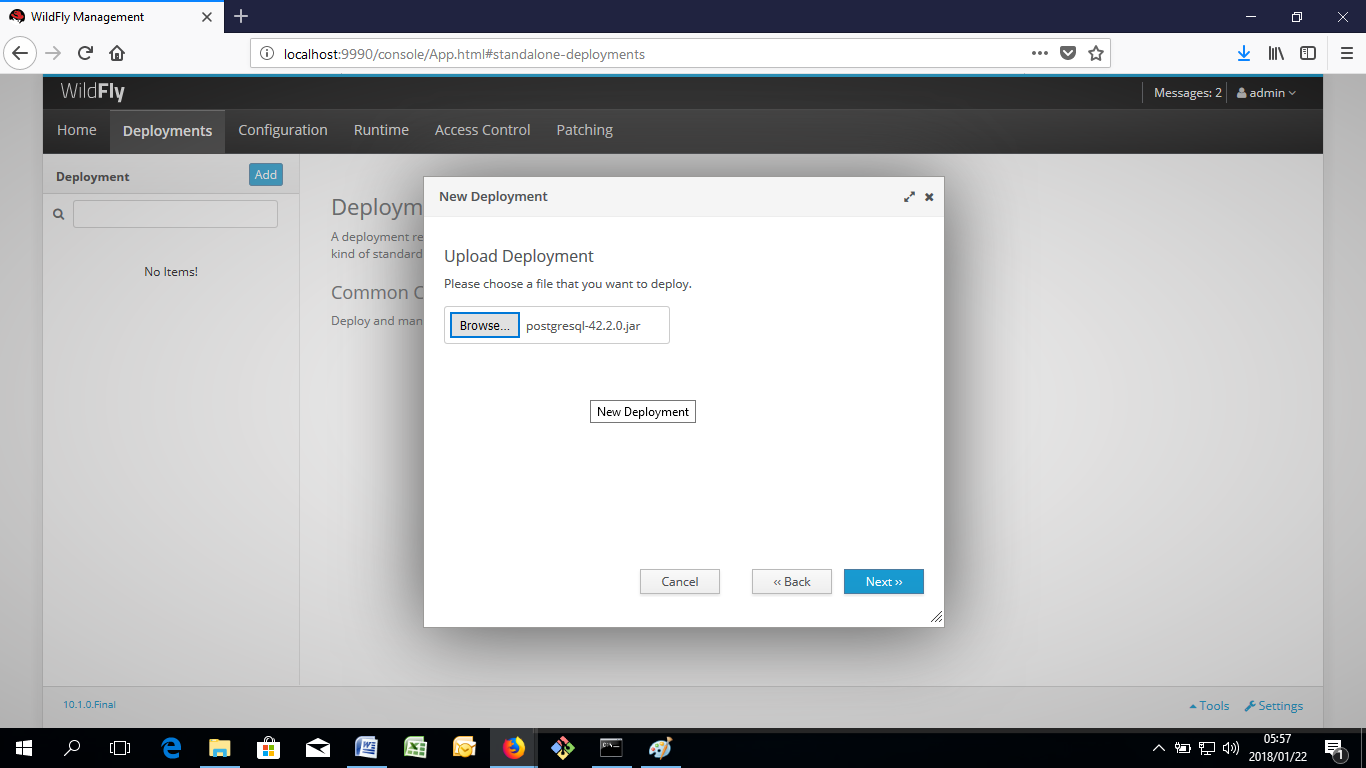


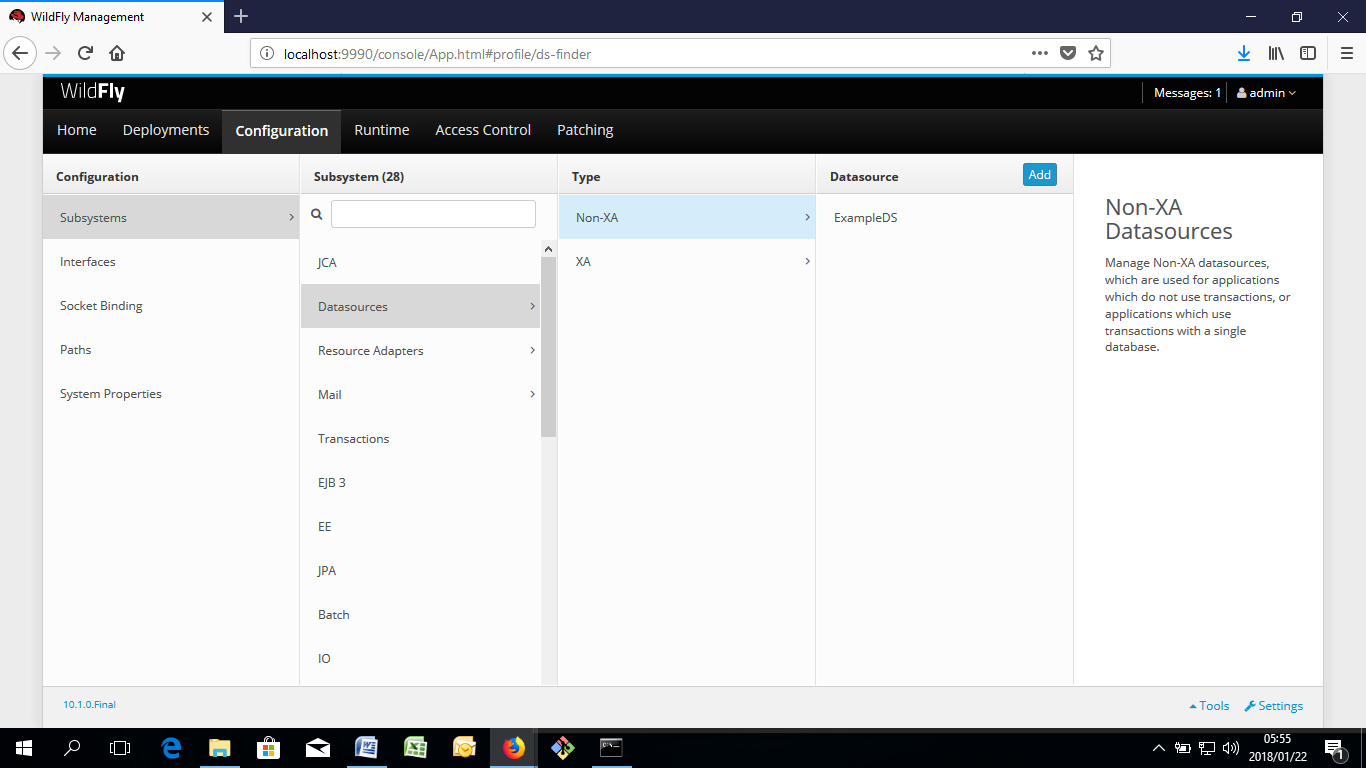
Step 5: Open your web browser and go to localhost:9990, type in the username and password that you assigned from the add-user.bat script

Step 6: Click on the deployments tab.

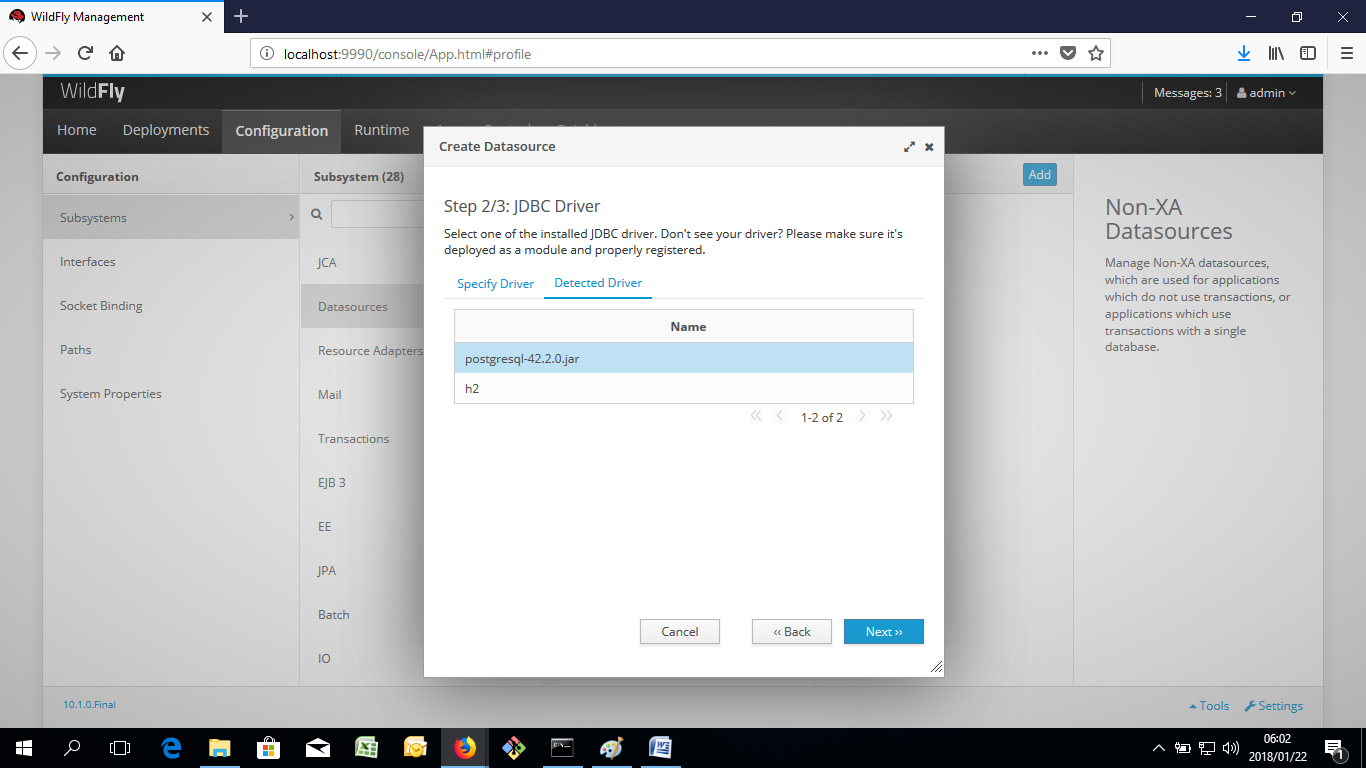


Step 7: click on add, upload a deployment, browse and upload the postgresql .jar file from where you saved the postgresql jdbc jar file and press next until finish

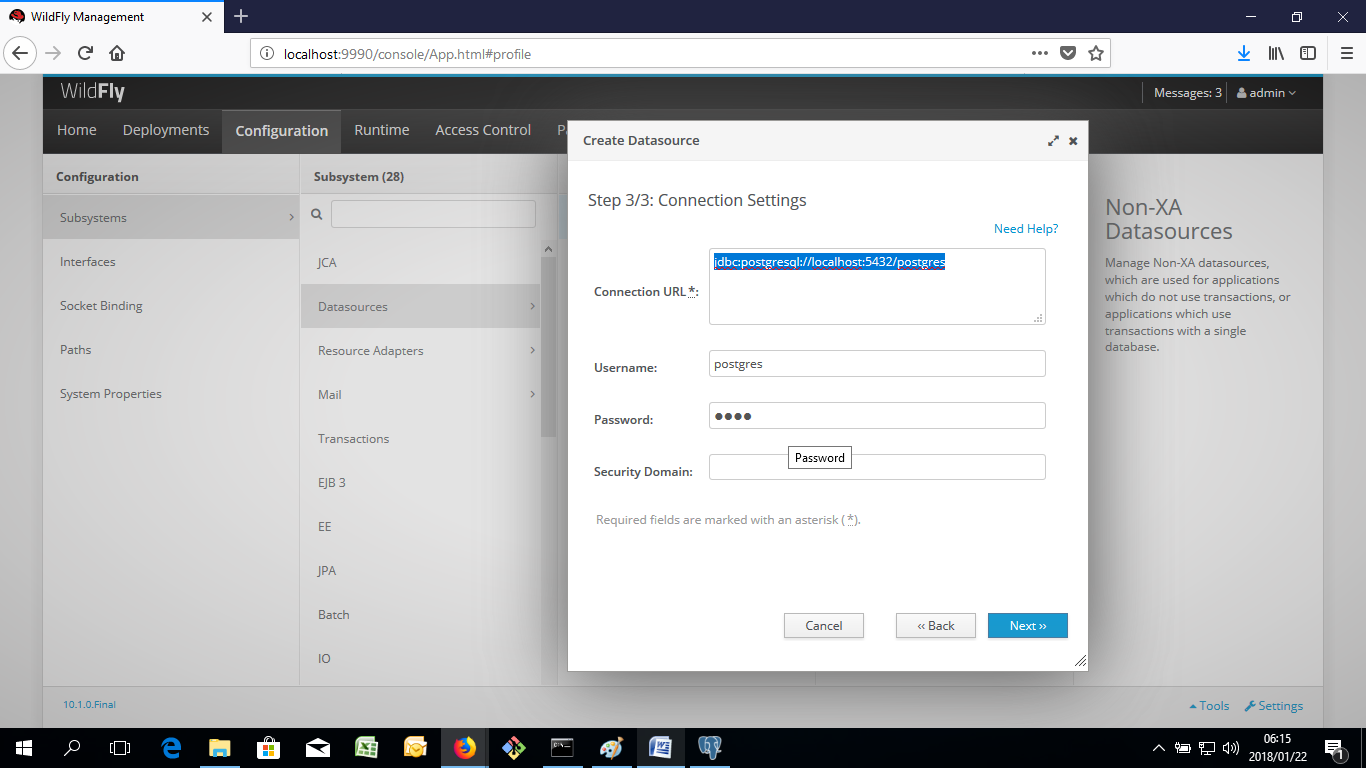


Step 8: click on the configuration tab in wildfly /datasources/Non-XA and click on add a Datasource

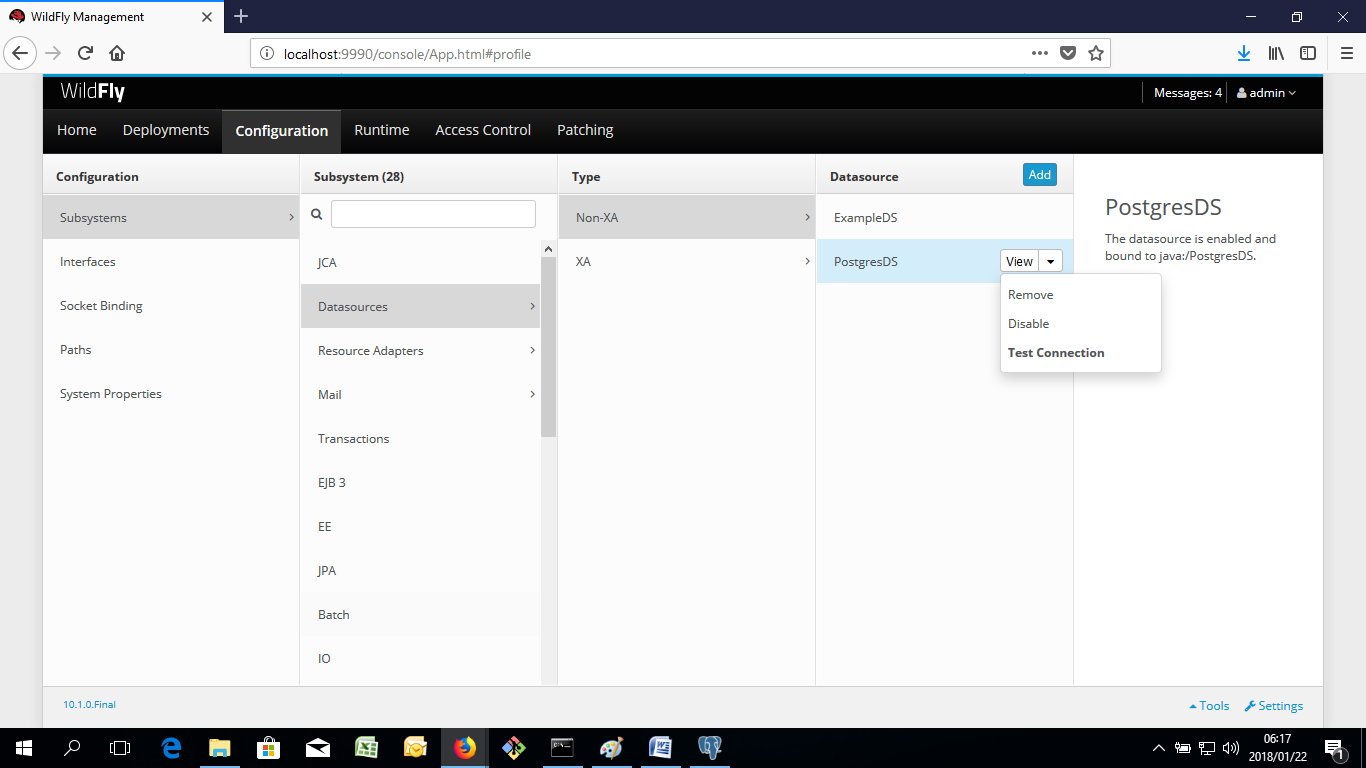
Step 9: Choose PostgreSQL datasource, click next until “step 2/3 jdbc driver” and click on “detected driver” and choose your postgresql driver you uploaded to deployments

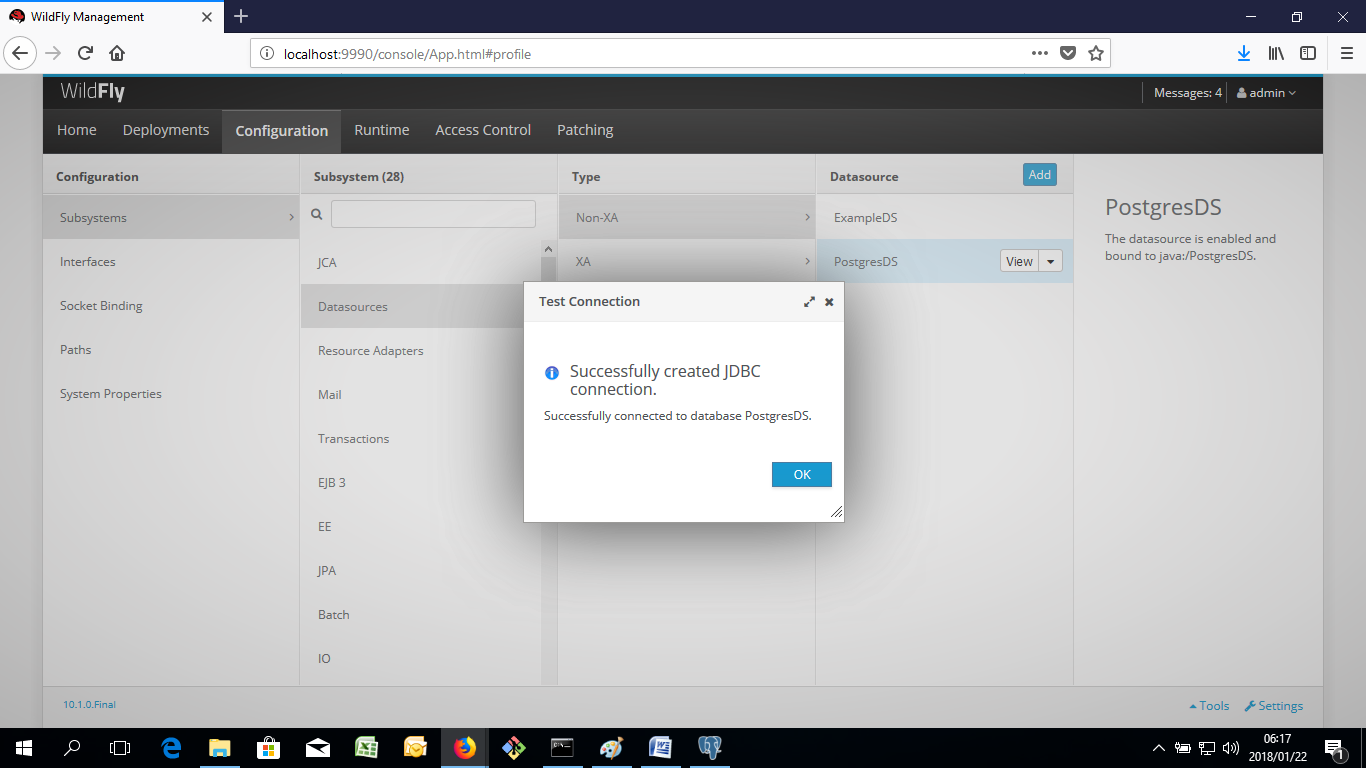


Step 10: At “Step 3/3” update your connection URL string to match your dbname, since we called our db in PGADMIN4 postgres the connection url string should look like this: “jdbc:postgresql://localhost:5432/postgres” and enter your username and password that you set up in PGADMIN 4 for your postgresql server, it should be “username:postgres” “password:1234” , afterwards click next and then finish



Step 11: “Click on your new PostgresDS datasource, select on the dropdown “test connection url” if connected successfully your wildfly can successfully deploy to your postgresql database and your wildfly setup is done.



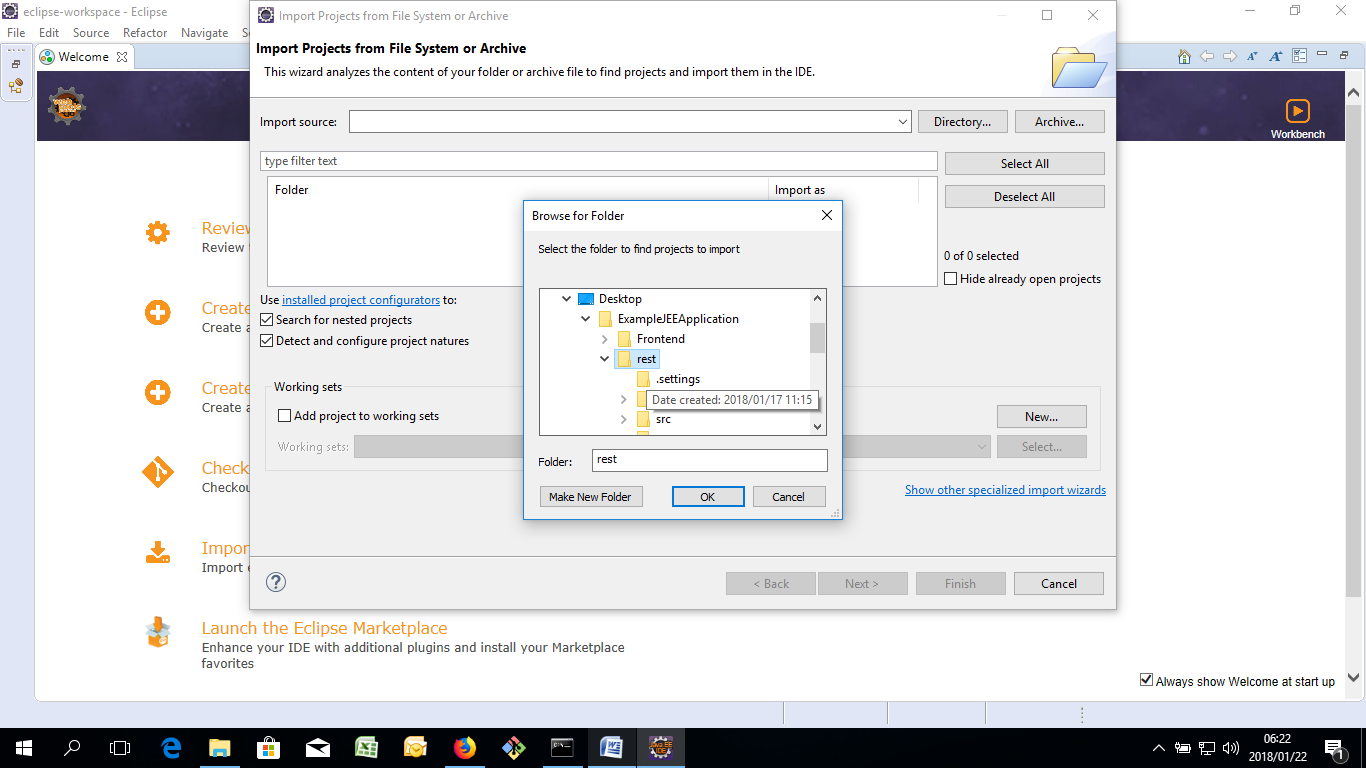


4. JEE:

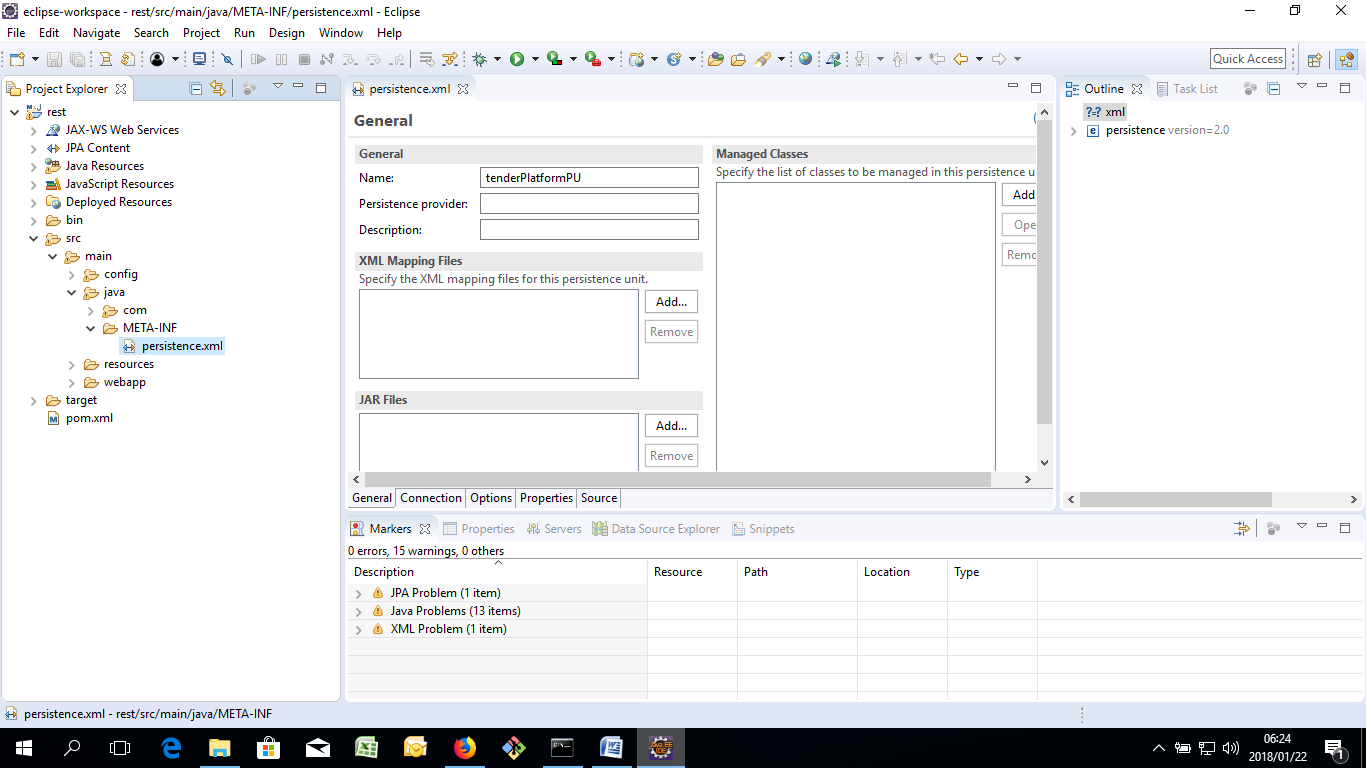
Step 1: Download Eclipse IDE for java ee developers at https://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/neon3

Step 2: Open Eclipse IDE for java ee developers and click on “File/Open Projects from File System

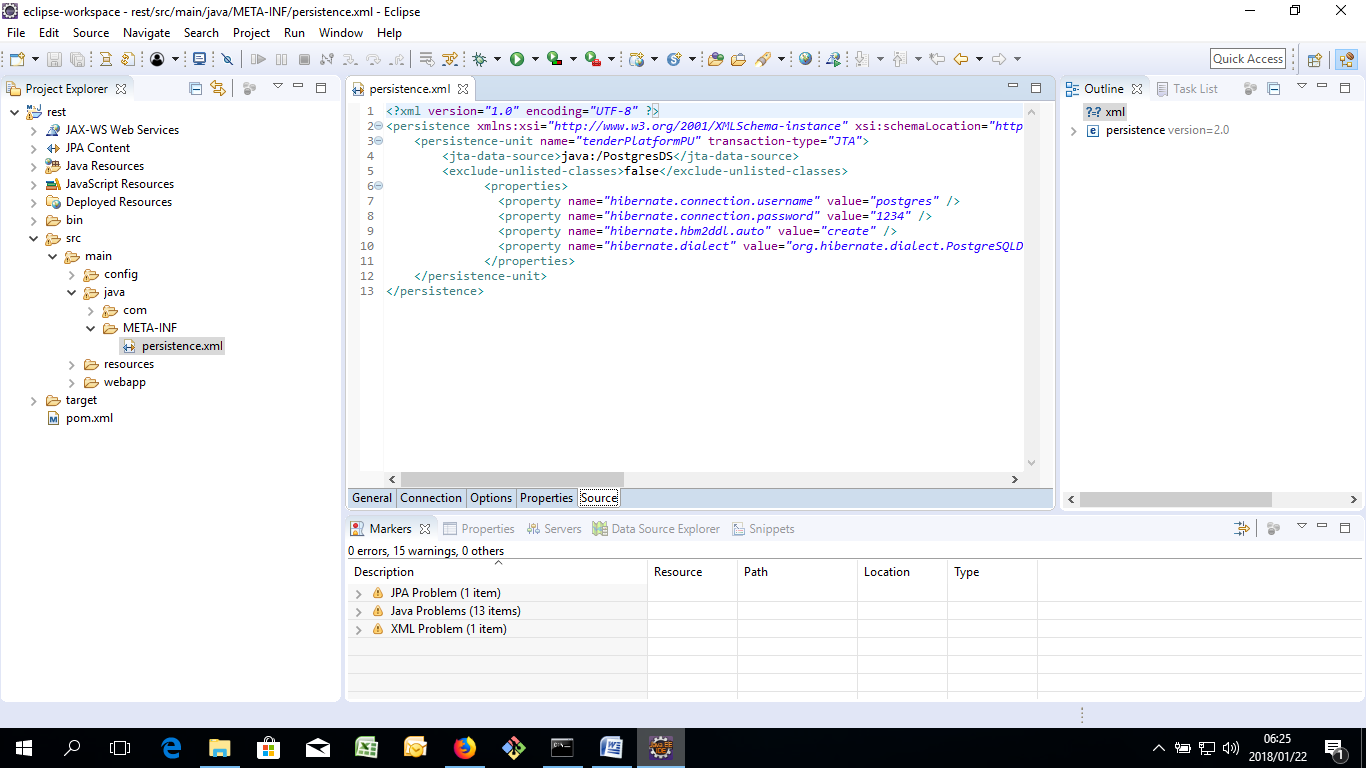
Step 3: click on “Directory button just to the right of “import source” , go to your “ExampleJeeApplication” project I gave you guys and click on the “rest” folder, press “OK” and then “FINISH”



Step 4: Once the “rest” folder is imported, go to “src/main/java/META-INF/persistence.xml



Step 5: Click on the source tab located just below your view, to change to text editor mode, edit your values if they are different to your postgresql server details, only “username and password” should differ



There you go, JEE will then be set up.

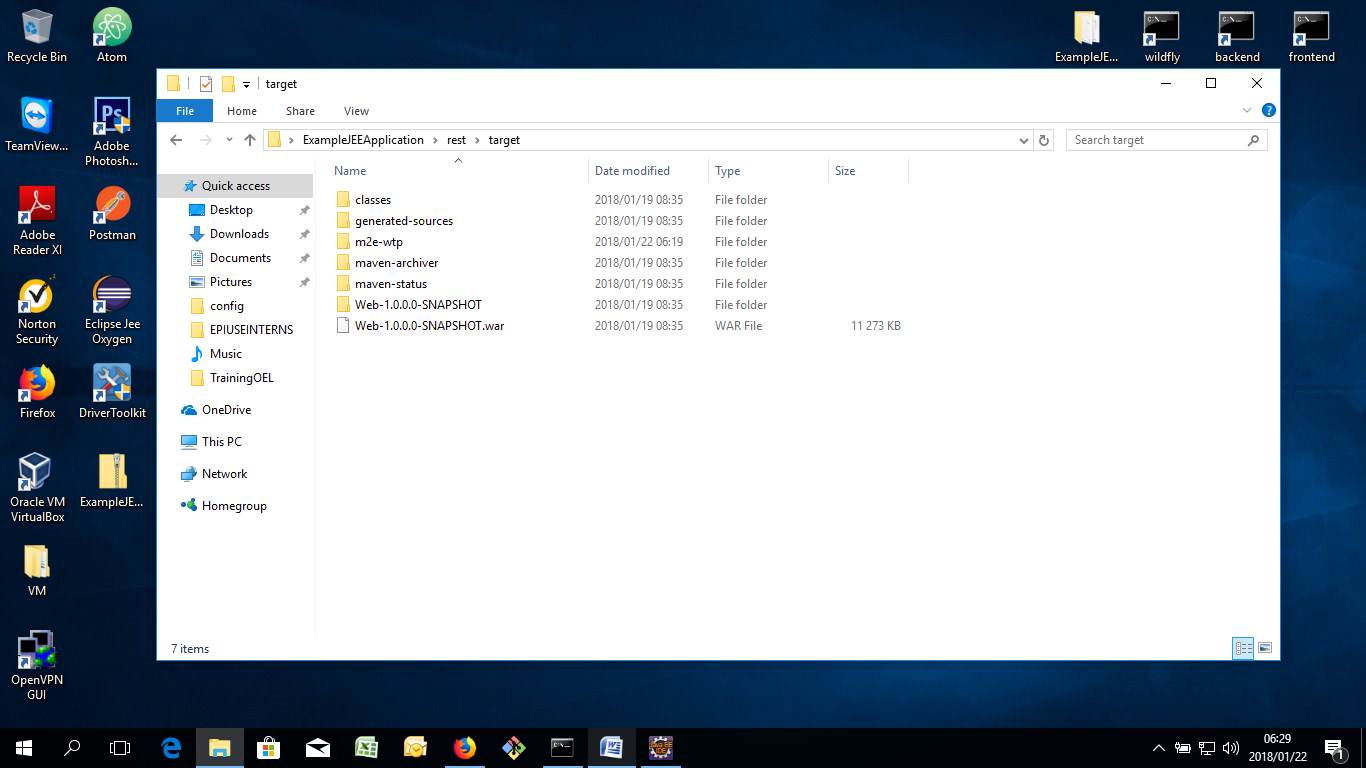
5.Maven

Step 1: Download and install Maven at <https://maven.apache.org/download.cgi>

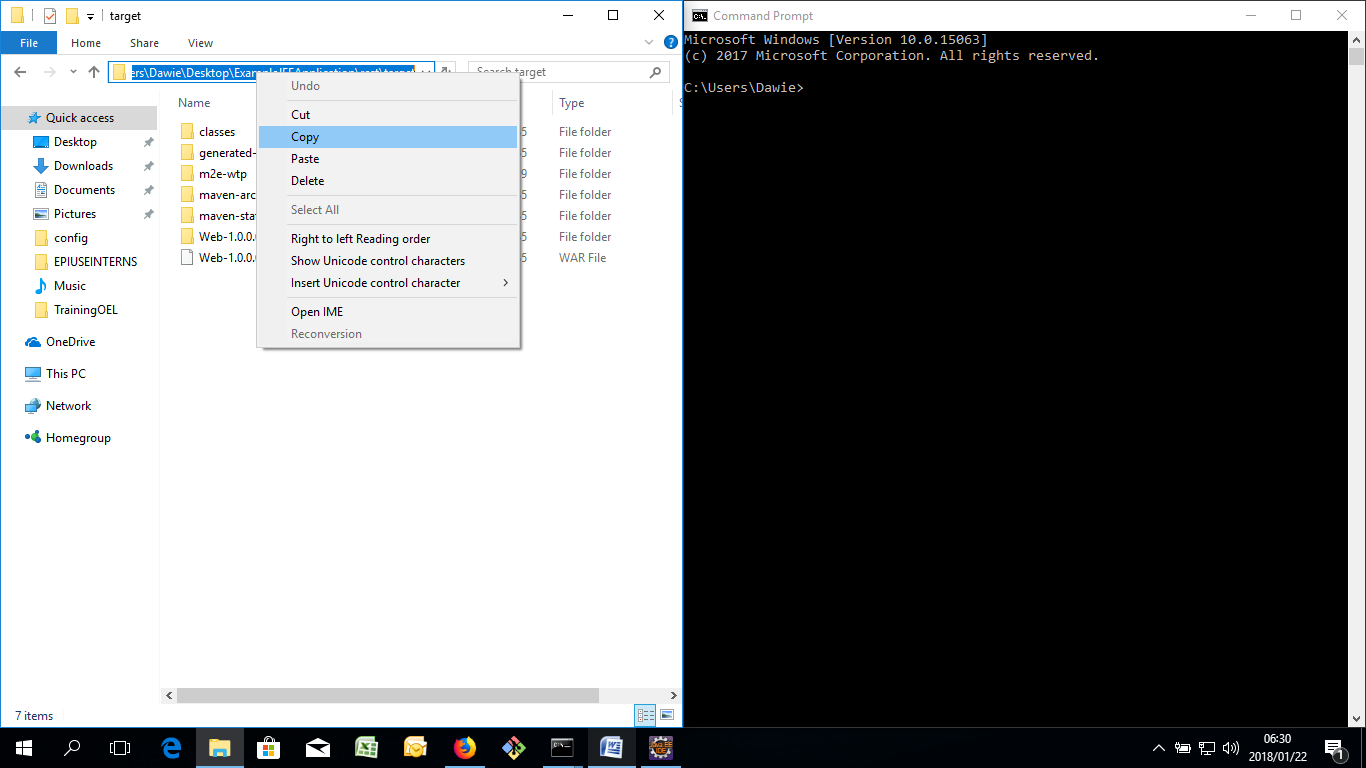
Step 2: Follow this link to setup Maven Path Variables <https://www.mkyong.com/maven/how-to-install-maven-in-windows/>

Step 5: download the maven clean dependency from <https://maven.apache.org/plugins/maven-clean-plugin/dependency-info.html> and copy paste the dependency inside your pom.xml file in your ExampleJeeApplication

Step 4: Go to your “ExampleJeeApplication” folder and go to ExampleJEEApplication\rest\

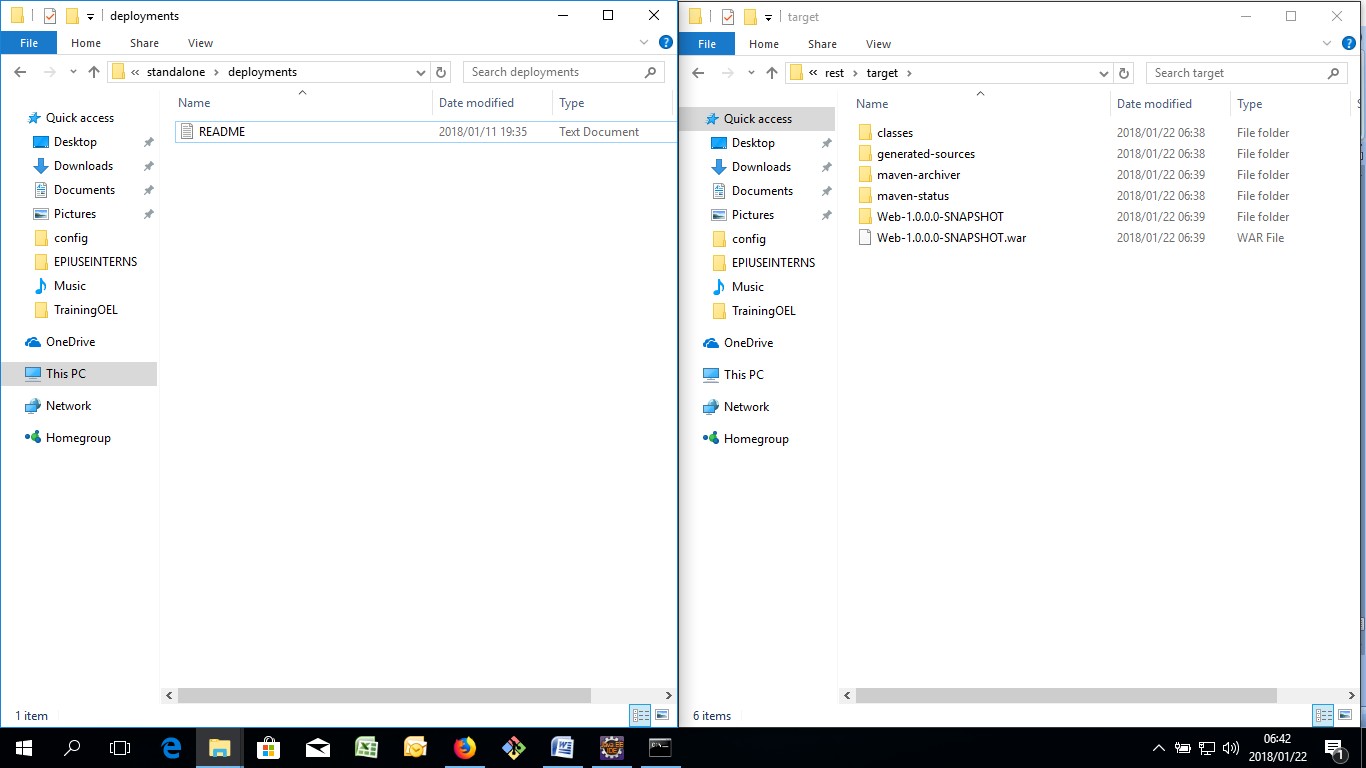


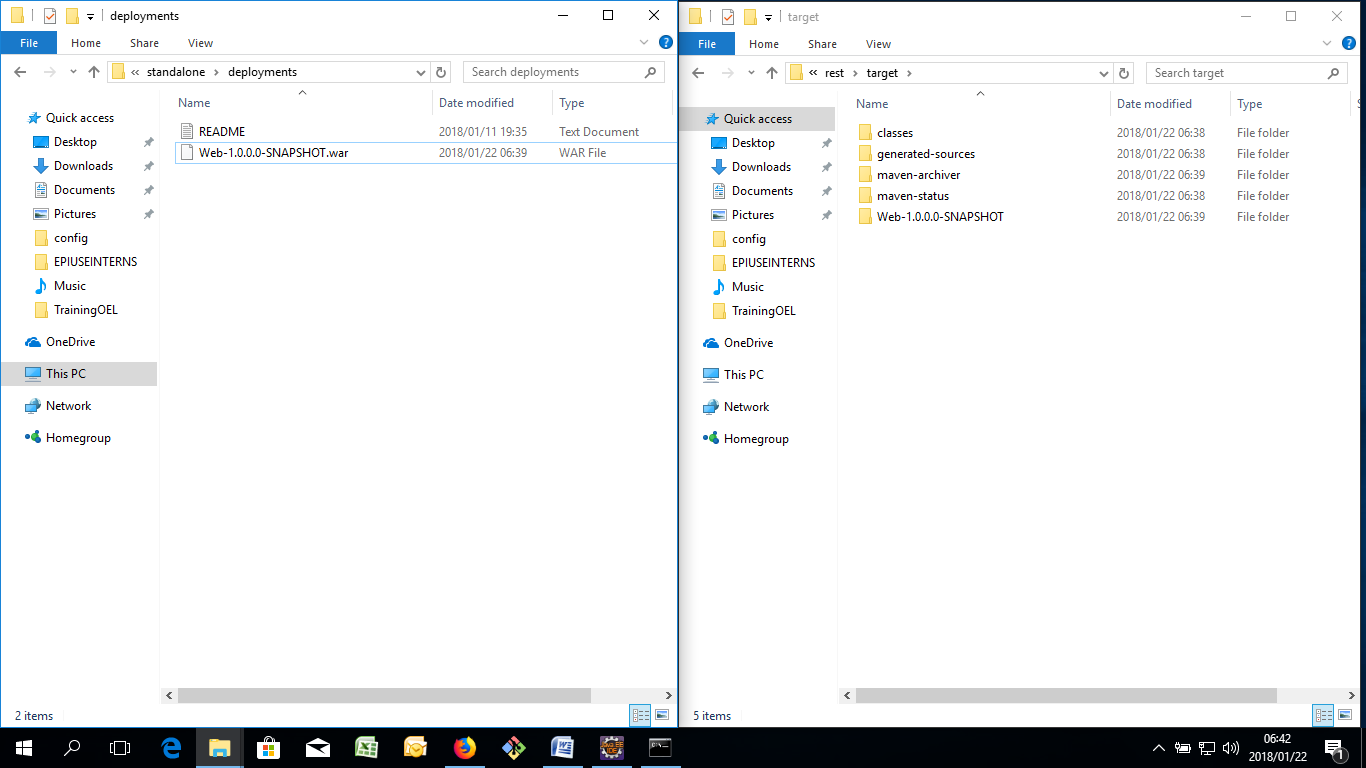
Step 4: Copy the path and open “CMD” and paste the path in the terminal following the “cd” command. Example “cd C:\Users\Dawie\Desktop\ExampleJEEApplication\rest\”



Step 6: Once you are in that directory of yours, run the command “mvn clean install”, this will generate a .WAR file in your rest/target directory in your project. NOTE STEP 6+7 work together

Step 7: Open your wildfly folder and go to /standalone/deployments and copy your .war file to that directory from your ExampleJEEApplication





Step 8: Run wildfly from your standalone.bat file, SEE WILDFLY SECTION UP TOP. If everything is successful, you can look in your /standalone/deployments directory and it should show a .deployed file