1. **Install Eclipse**

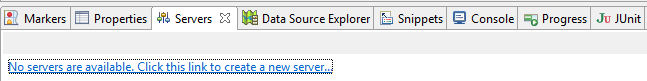
Download Luna Version of Eclipse from:

<https://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/lunasr2>

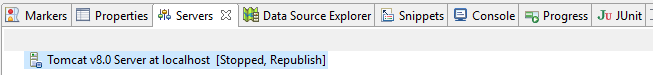
1. **Tomcat Server setup**
2. Download Tomcat Server 8 from:

<http://download.nextag.com/apache/tomcat/tomcat-8/v8.0.38/bin/apache-tomcat-8.0.38-windows-x64.zip>

1. Unzip the source
2. Setup the server in Eclipse:
3. First, there will be No servers as shown in Servers tab:

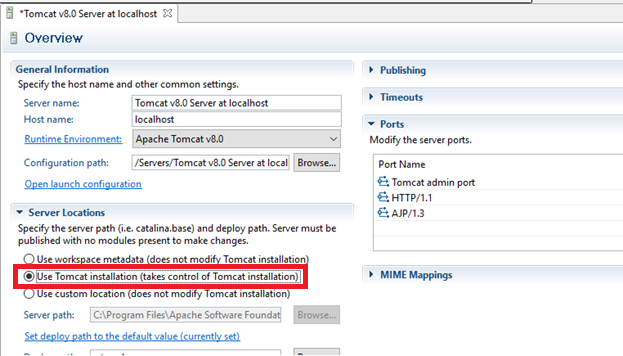


1. Click that link and select **Apache >> Tomcat v8.0 Server** and choose the downloaded folder path for Tomcat server.
2. Then we will see a new instance of Tomcat Server as shown below:

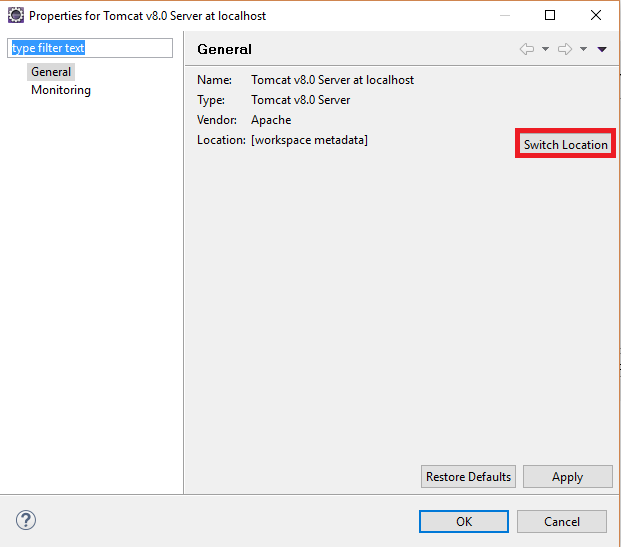


1. Double click that link and change the **Server Locations** option to:

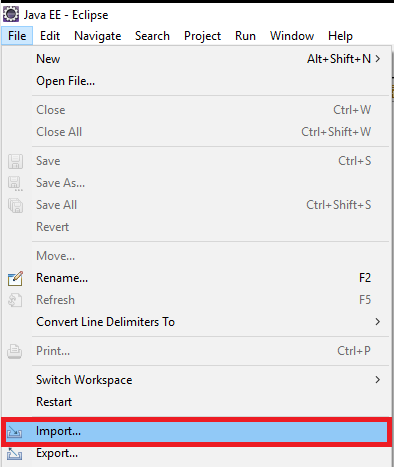
Use Tomcat installation as shown below:

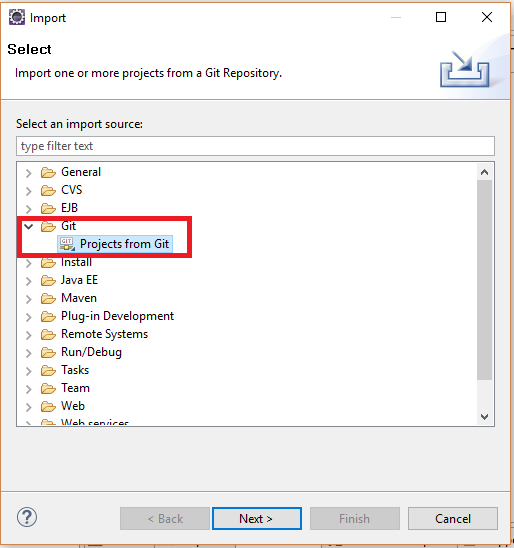


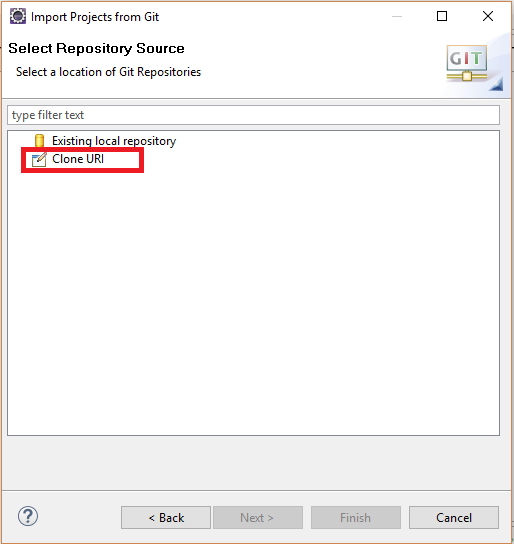
1. Also, we need to Right click the Server link as shown in step c and select the Switch Location button as shown below:

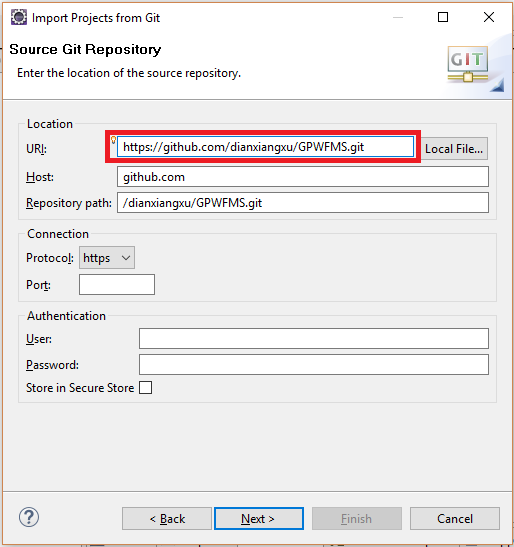


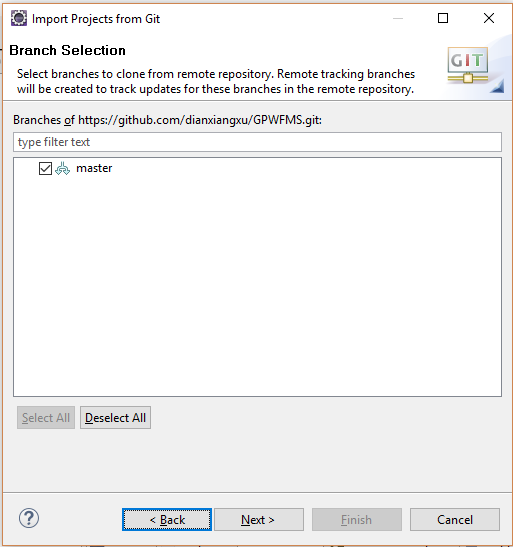
1. **Import GPMS folder from Github repository:**

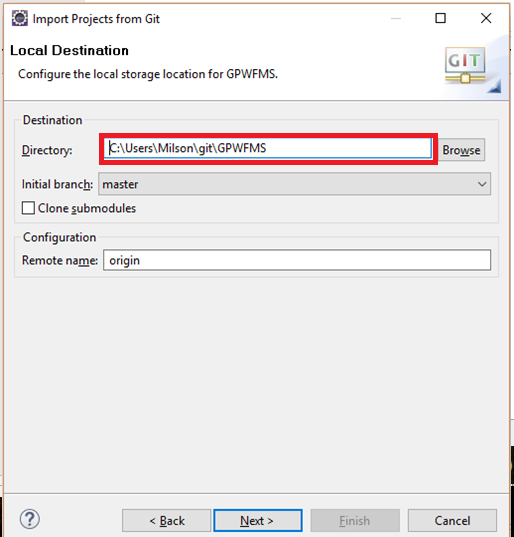




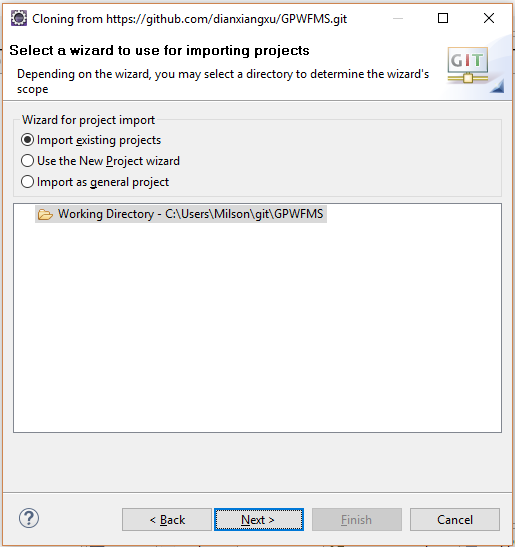


1. Make sure the URI is correct: [***https://github.com/dianxiangxu/GPWFMS.git***](https://github.com/dianxiangxu/GPWFMS.git)

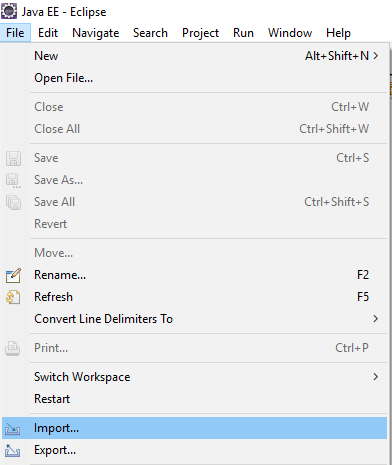


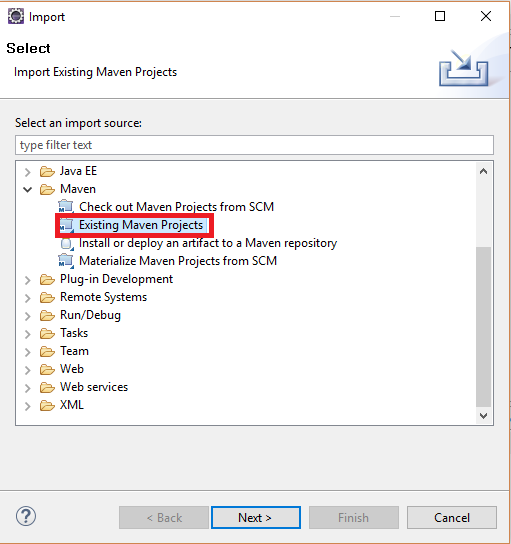


1. Click Next

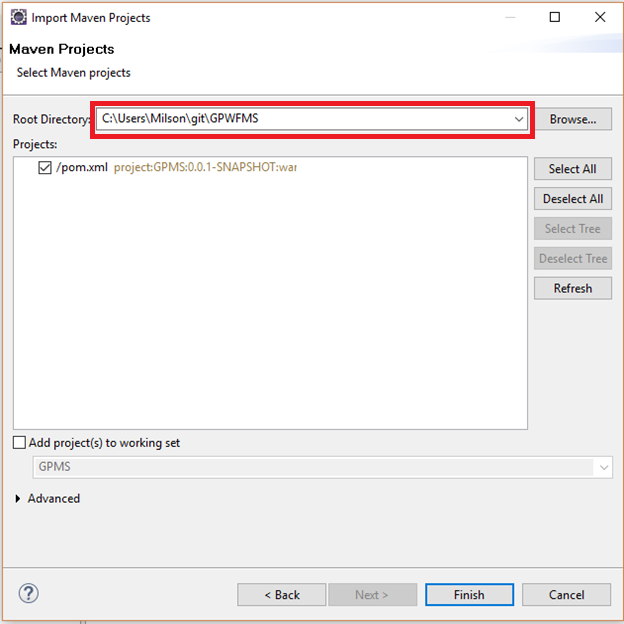


1. Click on Cancel once it successfully done.
2. After the git repo is successfully created on the specified folder, we need to import this project as an existing Maven Project as follows:

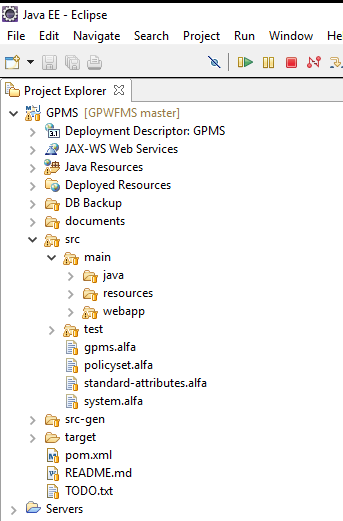




1. Browse and choose the recently created git repo path:

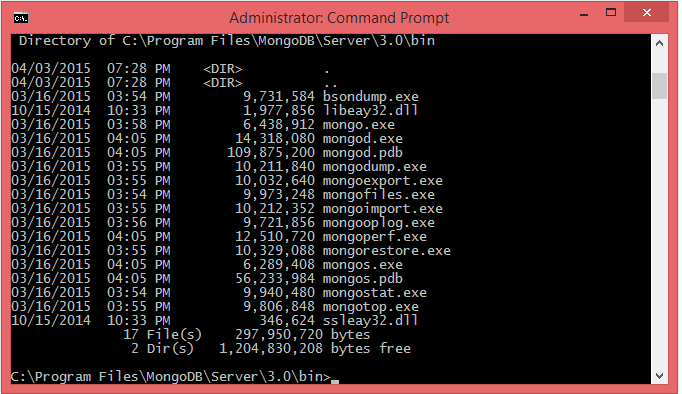


1. Click Finish and the folder structure looks like this:

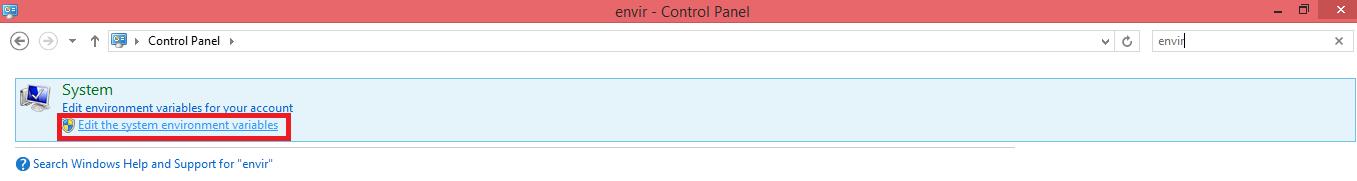


1. **MongoDB Database Setup**
2. Download the Current Stable Release of MongoDB i.e. .msi installer from: <https://www.mongodb.org/downloads> for windows OS.
3. Install the **msi** intaller and installation folder will be like this:

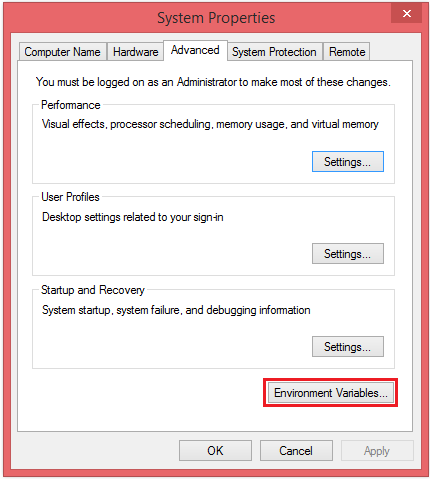
***C:\Program Files\MongoDB\Server\3.0\bin***

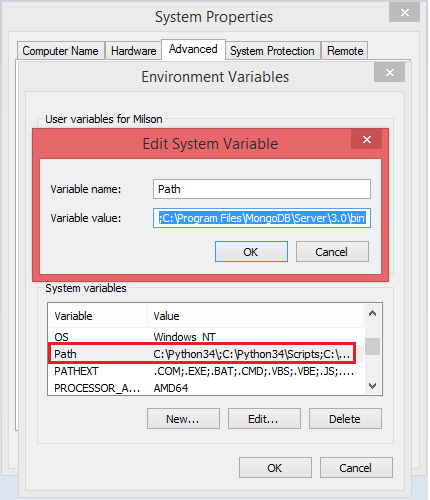


1. Set up the MongoDB Environment:
2. Go to the ***Control Panel*** and “***Edit the System environment variables”***:

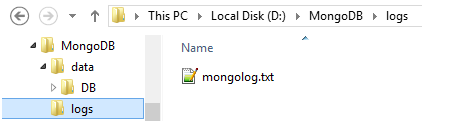


1. Edit **PATH** variable and append “***C:\Program Files\MongoDB\Server\3.0\bin***” as shown below:





1. MongoDB requires a data directory to store all data. We can customize this and define our own folder for convenience. For our purpose, we need to create "**D:\MongoDB**". And create “**D:\MongoDB\data\DB”** and **“D:\MongoDB\logs”** folders and create an Empty **mongolog.txt** file in **“D:\MongoDB\logs”** folder as shown below:



1. Run "**Command Prompt as Administrator**" and go to your MongoDB installation directory path upto **bin** folder using following command:

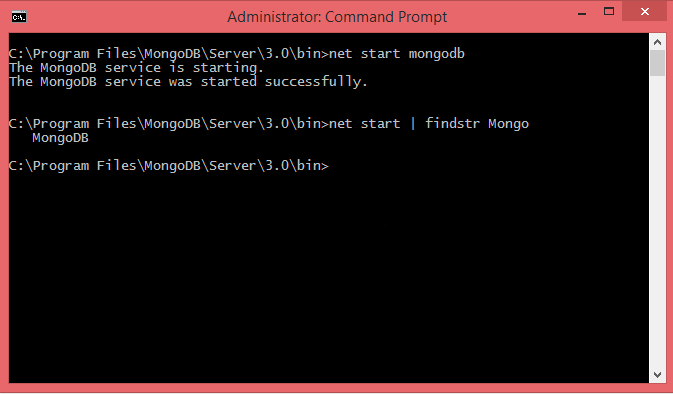
* **cd C:\Program Files\MongoDB\Server\3.0\bin**

1. To install MongoDB as background Windows Service, run this command in terminal:

***mongod --dbpath "D:\MongoDB\data\DB" --logpath "D:\MongoDB\logs\mongolog.txt" --install --serviceName "MongoDB"***

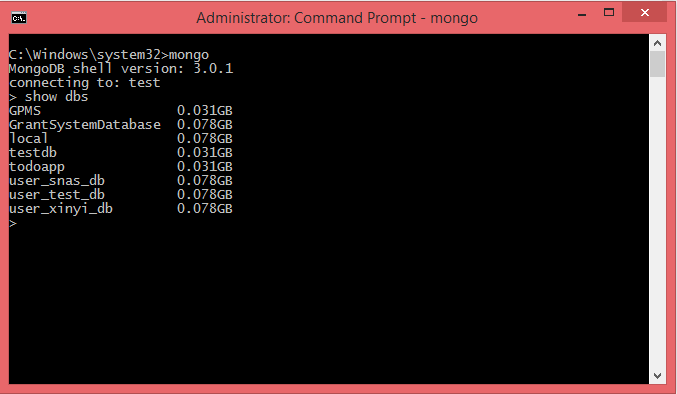
1. To start the registered “**MongoDB**” service execute following command in terminal from the MongoDB installation directory i.e. " **C:\Program Files\MongoDB\Server\3.0\bin** ":  
           -  ***net start mongodb***
2. To check and verify that the MongoDB service is running use the following command:

 -  ***net start | findstr Mongo***



1. To connect with mongo.exe shell, we can just start just typing:

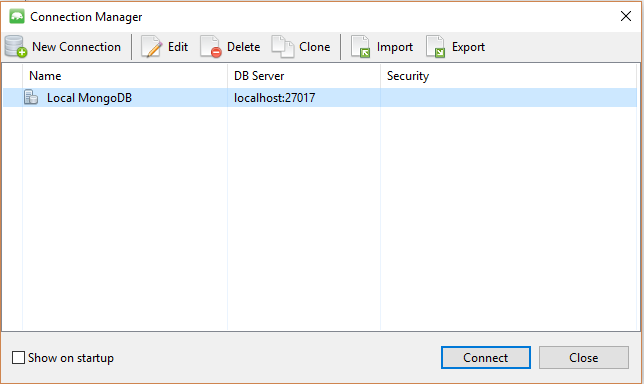
* ***mongo***



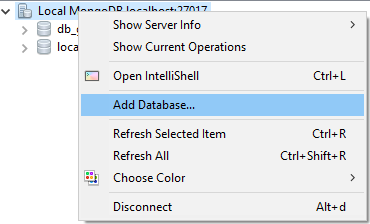
1. Install the Client GUI for MongoDB called **3T MongoChef** from:

<http://3t.io/mongochef/>

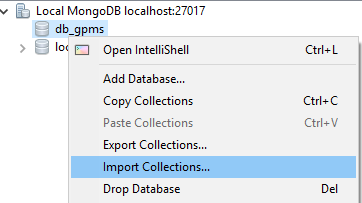
Connect to the Local Connection as a New Conncetion:



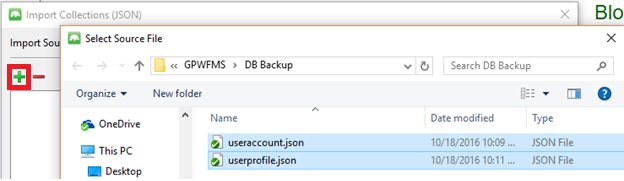
1. Add new database:



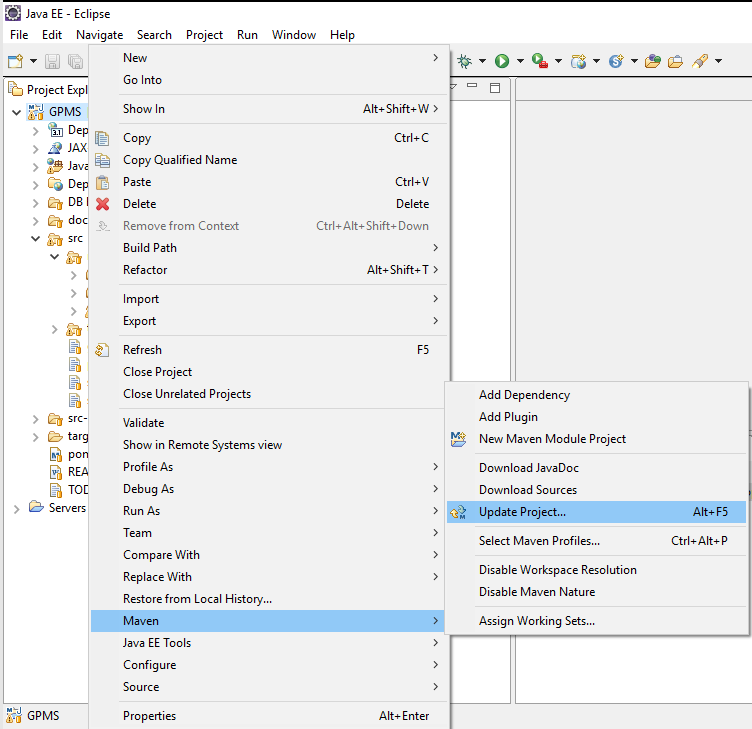
1. Name the new database as “**db\_gpms**” and import the default users using Import option:



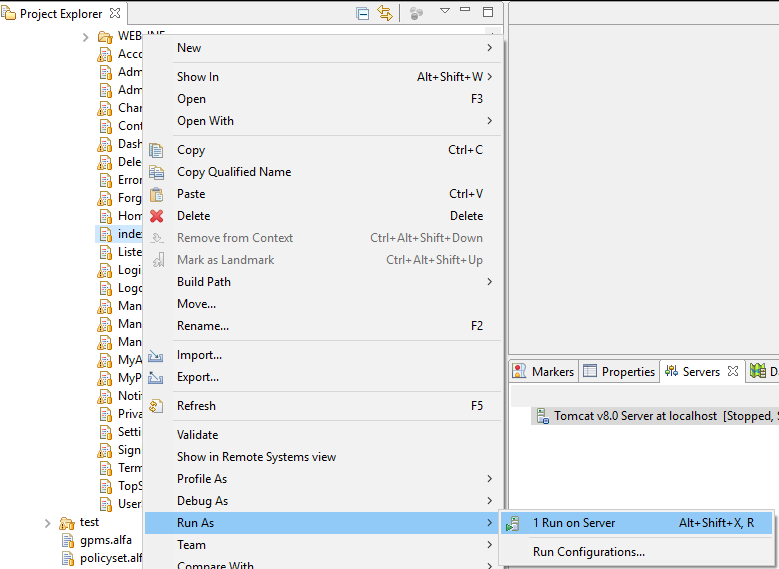
1. Import the default databases json format provided at : **<GPMS Source folder>\DB Backup**



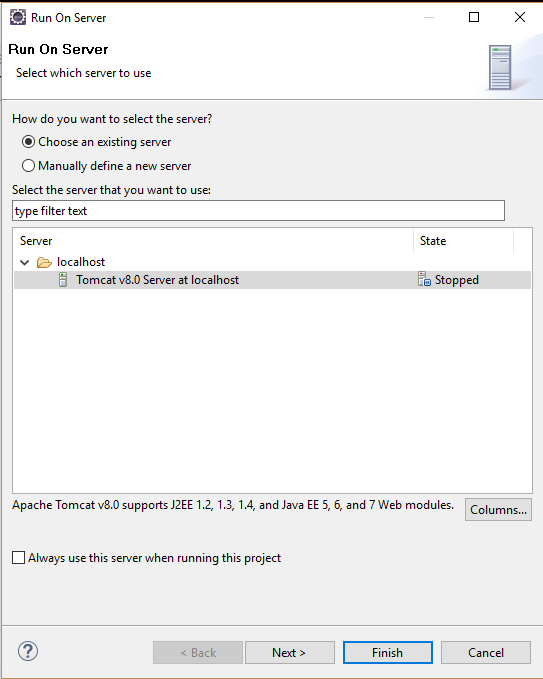
1. **Running GPMS Web Application using Eclipse:**
2. Clean Up the project first:



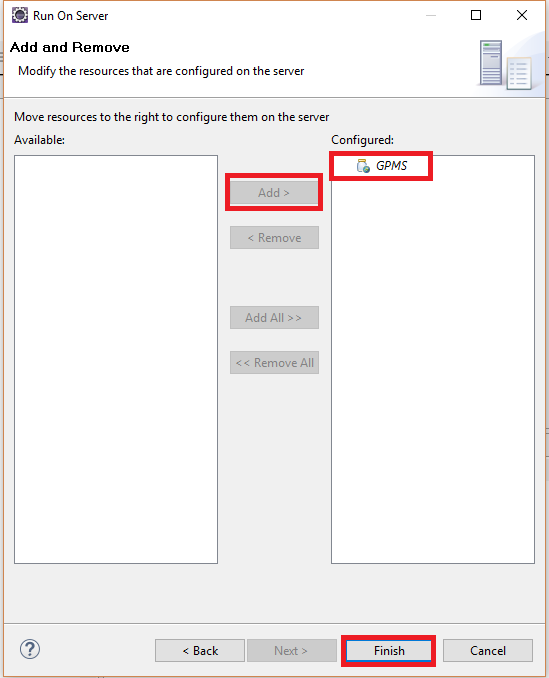
1. Select the **index.jsp** page and right click to open and run it on Server:



1. Choose the previously setup Tomcat Server as shown below:



1. Choose the **GPMS** project and add it to Configured side:

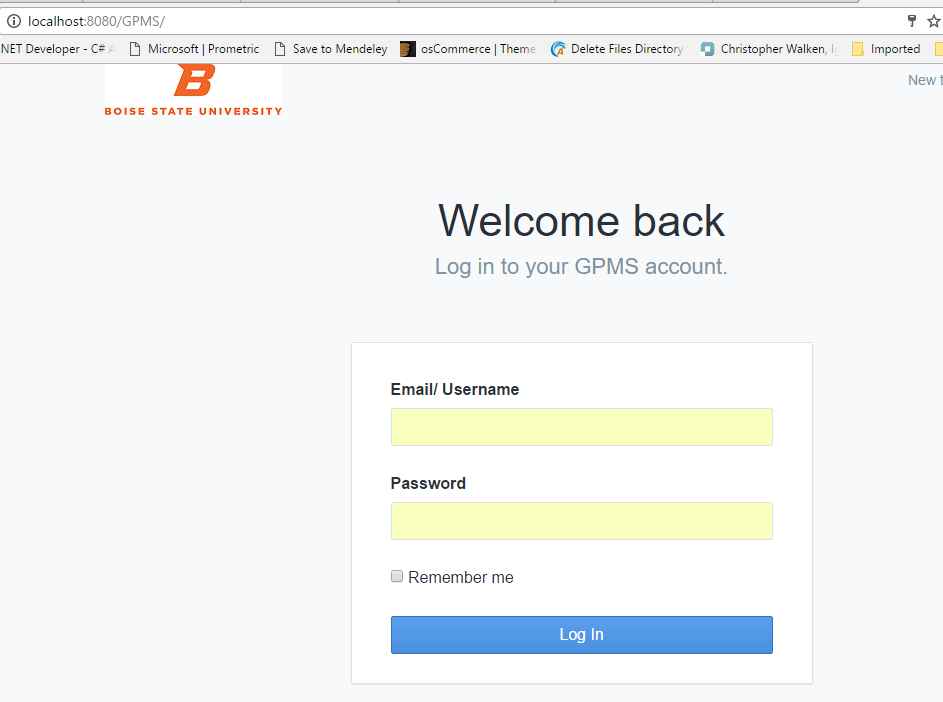


Now after successful run we can see the browser can load the test **index.jsp** page:

<http://localhost:8080/GPMS/index.jsp>

Now, the default GPMS **home** page can be opened browsing this url in web browser of your choice:

<http://localhost:8080/GPMS/>



As we already had imported the default **GPMS users** in our **MongoDB database** i.e. db\_gpms we can do login into the system. For simplicity, we have make all users’ password as: gpmspassword and username is first part of their corresponding email addresses.

For example, [irbcomputerscience@gmail.com](mailto:irbcomputerscience@gmail.com) can be logged into the system as entering whole email address or username: **irbcomputerscience** and password: **gpmspassword**

Some of the default existing User **Logins** are listed below:

**Admin Users:**

Username: admin

Email: [noreplygpms@gmail.com](mailto:noreplygpms@gmail.com)

Password: gpmspassword

**Position Type: University administrator**

**Position Title: IRB, University Research Administrator, University Research Director**

**IRBs:**

[irbcomputerscience@gmail.com](mailto:irbcomputerscience@gmail.com), [irbelectricalengineering@gmail.com](mailto:irbelectricalengineering@gmail.com)

**University Research Administrator:**

[racomputerscience@gmail.com](mailto:racomputerscience@gmail.com)

**University Research Director:**

[directorcomputerscience@gmail.com](mailto:directorcomputerscience@gmail.com)

1. **College: Engineering**

**Department:** Computer Engineering, Computer Science, Electrical Engineering

**Position Type:** Professional staff

**Position Title: Business Manager,** Department Administrative Assistant

**Business Managers:**

[bmcomputerengineering1@gmail.com](mailto:bmcomputerengineering1@gmail.com), [bmcomputerscience@gmail.com](mailto:bmcomputerscience@gmail.com), [bmelectricalengineering@gmail.com](mailto:bmelectricalengineering@gmail.com)

**Position Type:** Administrator

**Position Title:** **Department Chair**, Associate Chair, **Dean**, Associate Dean

**Department Chairs:**

[chaircomputerengineering@gmail.com](mailto:chaircomputerengineering@gmail.com), chaircomputerscience@gmail.com, [chairelectricalengineering@gmail.com](mailto:chairelectricalengineering@gmail.com)

For Delegation Test we have Associate Chair from **Computer Science** Department:

|  |  |
| --- | --- |
| Username | Password |
| edmund | gpmspassword |

**Deans:**

[deancomputerengineering@gmail.com](mailto:deancomputerengineering@gmail.com), [deancomputerscience@gmail.com](mailto:deancomputerscience@gmail.com), deanelectricalengineering@gmail.com

1. **College:** **Science**

**Department:** Physics, Chemistry

**Chairs:**

[chairphysics1@gmail.com](mailto:chairphysics1@gmail.com), chairchemistry@gmail.com

**Business Managers:**

[bmphysics1@gmail.com](mailto:bmphysics1@gmail.com), bmchemistry1@gmail.com

**Deans:**

[deanphysics1@gmail.com](mailto:deanphysics1@gmail.com), deanchemistry1@gmail.com

**Some General Users:**

**Position Type:** Non-tenure-track research faculty

**Position Title:** Research Professor, Associate Research Professor, Assistant Research Professor, Clinical Professor, Clinical Associate Professor, Clinical Assistant Professor, Visiting Professor, Visiting Associate Professor, Visiting Assistant Professor

**Position Type:** Research staff

**Position Title:** Research Associate, Research Scientist, Senior Research Scientist

**Position Type:** Teaching faculty

**Position Title:** Lecturer, Senior Lecturer, Adjunct Professor

**Position Type:** Tenured/tenure-track faculty

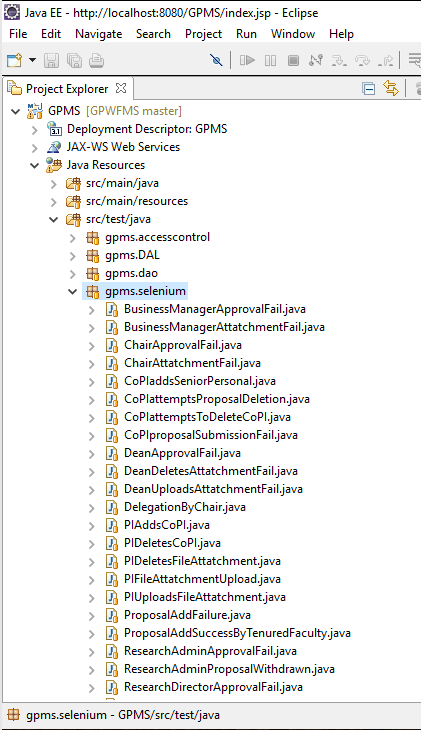
**Position Title:** Distinguished Professor, Professor, Associate Professor, Assistant Professor

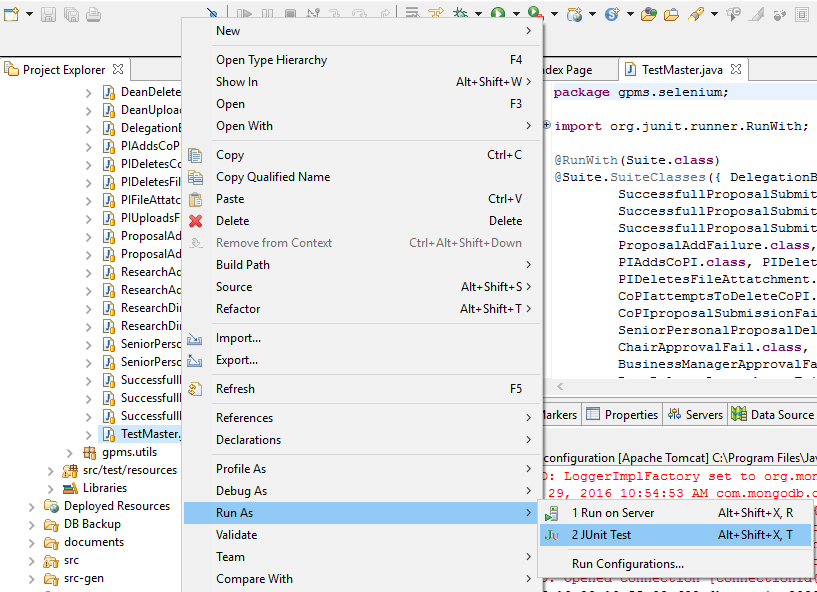
We have listed default users that possess different position titles, departments and colleges as defined above:

|  |  |
| --- | --- |
| Username | Password |
| milson | gpmspassword |
| NickC | gpmspassword |
| liliana | gpmspassword |
| samer | gpmspassword |
| tamanna | gpmspassword |

1. **Selenium Test Cases:**

We can run all at once the all test cases as shown below located in **“src/test/java/gpms.selenium”** just running **TestMaster.java** file as **jUnit** Test.





1. **Conformance Policy Test Cases can be done using:**

We have a template to do Conformance Policy **request/response** testing. For that we need to update and run **AdvanceTestV3.java** jUnit Test Class located at **“src/test/java/gpms.accesscontrol”**. We can add any number of test policies, their corresponding requests and responses in “**src/test/resources/advance/3”** folder as shown below:

