Design Specification

Task Assignment System

Develop By:

Mr.Kitti Wugprakhon [IN]  
  
087 247 4891

[anonwug@hotmail.com](mailto:anonwug@hotmail.com)

Table of Contents

[1. Environment Setup 2](#_Toc477032166)

[2. Project Setup 4](#_Toc477032167)

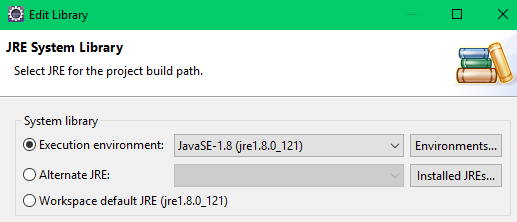
[3. Project Structure Description 8](#_Toc477032168)

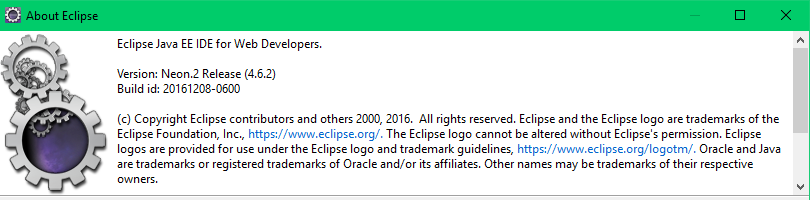
[4. Library Jar 9](#_Toc477032169)

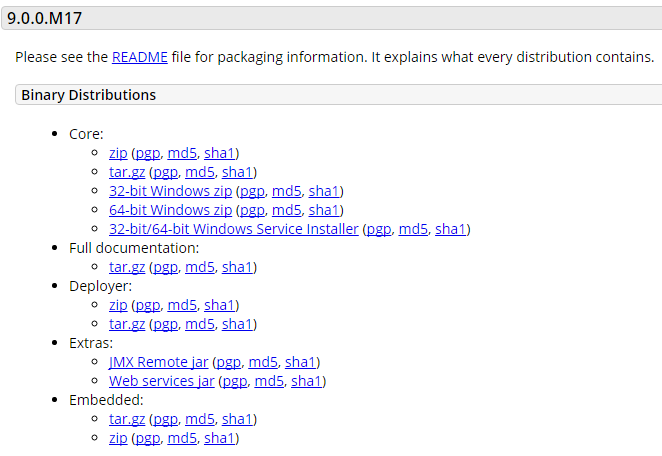
[5. API RESTFul Description 10](#_Toc477032170)

# 

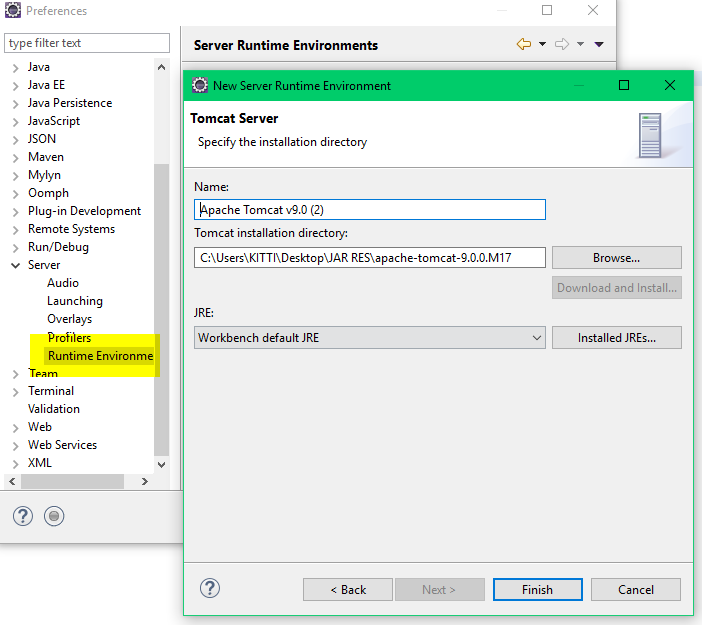
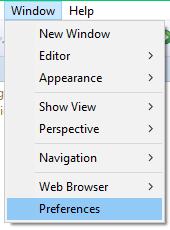
# 1. Environment Setup

 1.1 Java Development Kit (JDK)

 1.2 Eclipse IDE

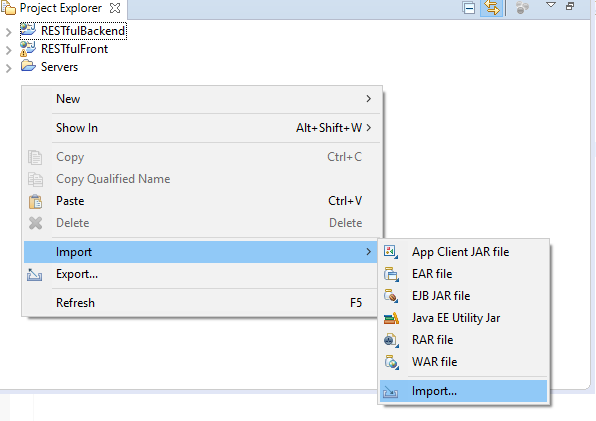
 1.3 Apache Tomcat  
 You can download from http://tomcat.apache.org/download-90.cgi

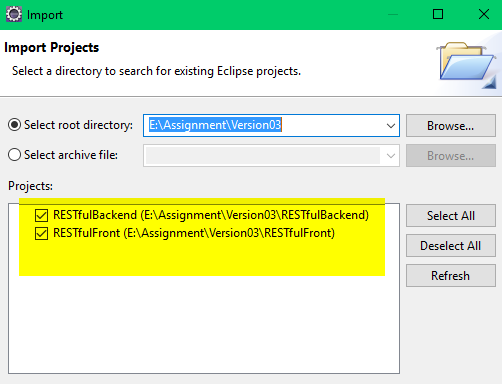
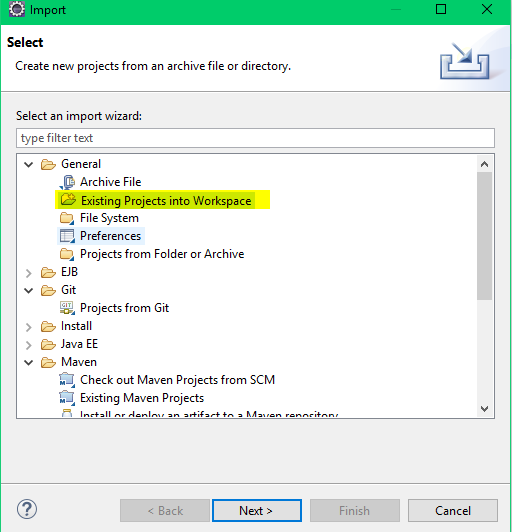
Config Apache Tomcat Server in Eclipse.

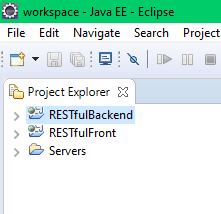


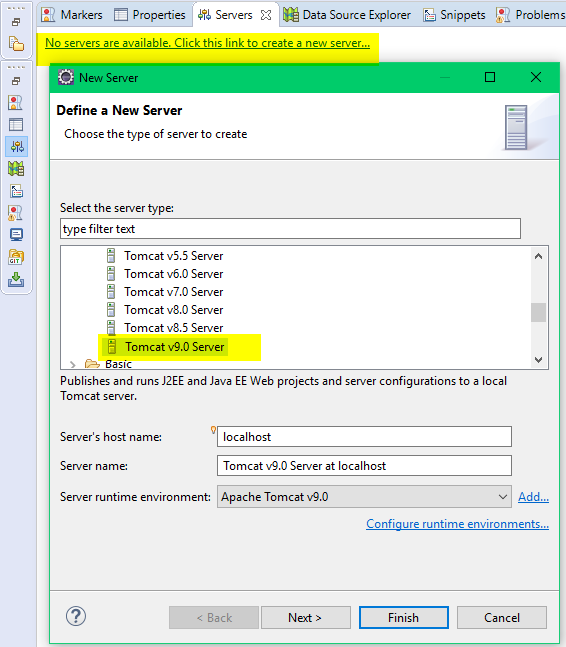
1.4 Git Repository  
 https://github.com/kittiwug/TaskAssignmentProject.git

# 2. Project Setup

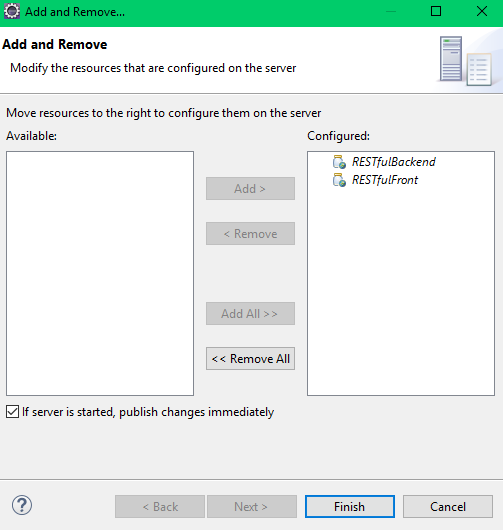
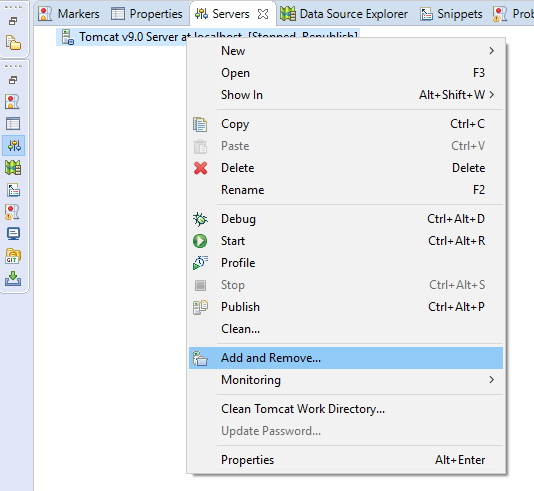
 Import Project.

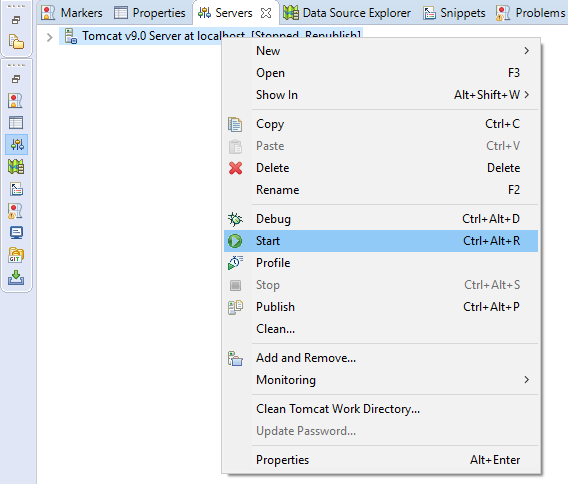
 Choose : General > Existing Project into Workspace



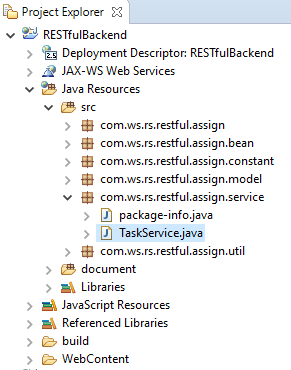
 Create new Server.

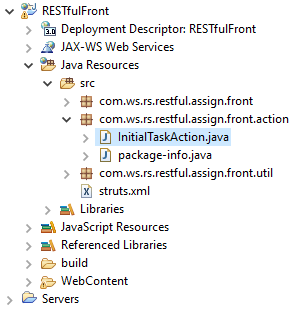
Add project to Tomcat Server.



 Start Server

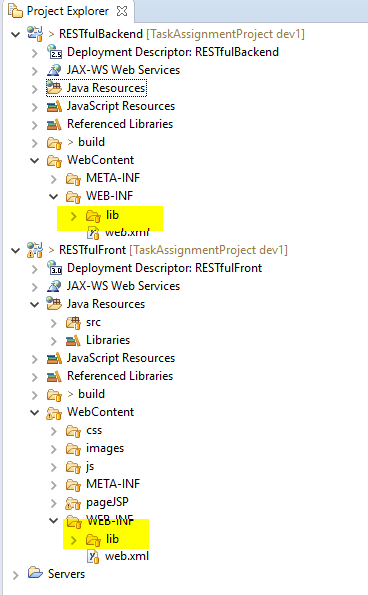
# 3. Project Structure

 3.1 RESTfulBackend  
 Develop by JAVA.

 3.2 RESTfulFront  
 Develop by JAVA , Strut2 framework, AngularJS framework. There is **TaskService.java** for control servlet request from client.

# 4. Library Jar

Both project collect library jar file in folder "WebContent/WEB-INF/lib".



# 5. API RESTFul Description

1. Service back-end API for get Task Assign total in Database.

|  |  |
| --- | --- |
| Request URL | http://localhost:8080/RESTfulBackend/rest/TaskService/AllTask/ |
| Request Method | POST |
| Request | - |
| Response | C:\Users\KITTI\Desktop\1.jpg |
| Description | statusCode ‘0’ = Success  statusCode ‘1’ = Not Success, and will return statusMessage |

2. Service back-end API for create Task Assign.

|  |  |
| --- | --- |
| Request URL | http://localhost:8080/RESTfulBackend/rest/TaskService/CreateTask/ |
| Request Method | POST |
| Request |  |
| Response |  |
| Description | statusCode ‘0’ = Success  statusCode ‘1’ = Not Success, and will return statusMessage |

3. Service back-end API for update status [ Pending, Done ] Task Assign in Database.

|  |  |
| --- | --- |
| Request URL | http://localhost:8080/RESTfulBackend/rest/TaskService/UpdateTaskStatus/ |
| Request Method | POST |
| Request |  |
| Response |  |
| Description | statusCode ‘0’ = Success  statusCode ‘1’ = Not Success, and will return statusMessage |

4. Service back-end API for update Task Assign in Database.

|  |  |
| --- | --- |
| Request URL | http://localhost:8080/RESTfulBackend/rest/TaskService/UpdateTask/ |
| Request Method | POST |
| Request |  |
| Response |  |
| Description | statusCode ‘0’ = Success  statusCode ‘1’ = Not Success, and will return statusMessage |

5. Service back-end API for delete Task Assign in Database.

|  |  |
| --- | --- |
| Request URL | http://localhost:8080/RESTfulBackend/rest/TaskService/DeleteTask/ |
| Request Method | POST |
| Request |  |
| Response |  |

6. Service back-end API for delete all Task Assign in Database.

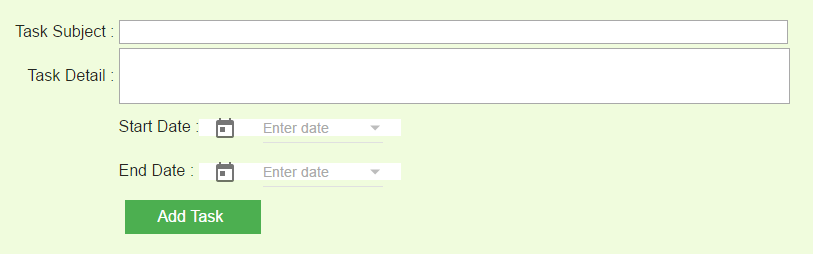
|  |  |
| --- | --- |
| Request URL | http://localhost:8080/RESTfulBackend/rest/TaskService/DeleteAllTask/ |
| Request Method | GET |
| Request | Parameter taskId |
| Response |  |

7. Service back-end API for get Task Assign by task id in Database.

|  |  |
| --- | --- |
| Request URL | http://localhost:8080/RESTfulBackend/rest/TaskService/GetTaskByID |
| Request Method | GET |
| Request | Parameter taskId |
| Response |  |

# 6. User Interface Description

URL <http://localhost:8080/RESTfulFront/initialTask.action>

 Main screen for add Task detail.

