SE3020 – Distributed Computing
Lab 6 – RESTful Services

Creating the REST Service

1. Download Eclipse IDE for Java EE if it's not already installed.

http://www.eclipse.org/downloads/eclipse-packages/

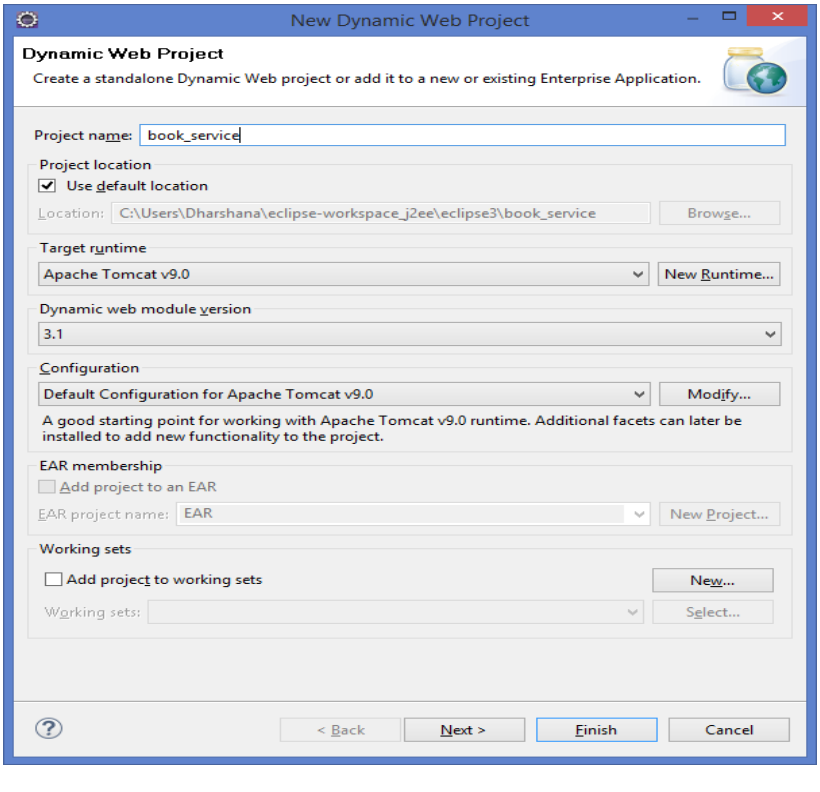
2. Download Jersey libraries from the following location.

https://jersey.github.io/download.html

3. Download and extract the Tomcat web server from the following location.

https://tomcat.apache.org/download-90.cgi

4. Create a new Dynamic web project. Give the project name as 'book_service'. Select 'Apache Tomcat 9.0' as the target runtime.



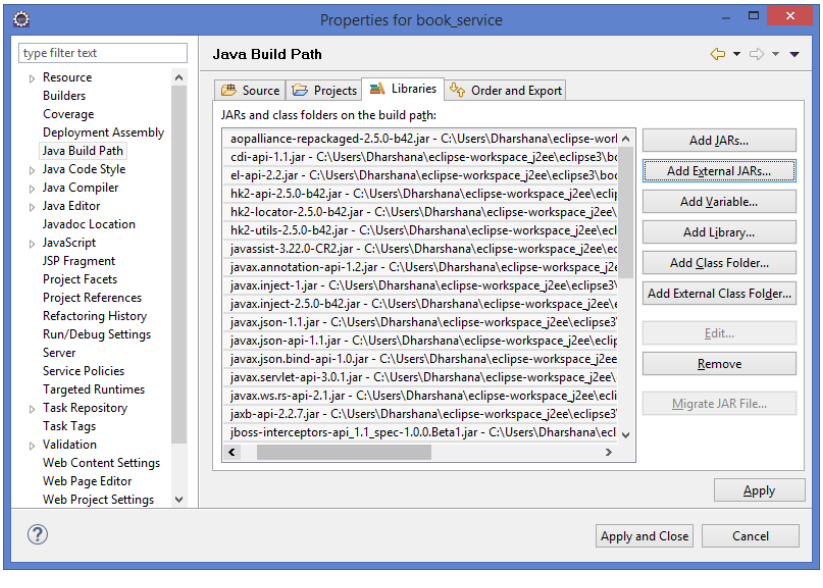
5. Click ‘Next’ until the project is created.

6. Under ‘Java Resources’->’Src’ folder, create the three classes Book, BookService and Books and copy the attached code to the classes. The package name should be ‘bookservice’ (you may get compile errors from the BookService class).

7. Extract the Jersey package to a directory and copy the jar files in **all of its** subfolders to the WebContent/WEB-INF/lib directory of the project folder (you can right click ->Properties of the WebContent to locate where it’s located in the hard drive).

8. Refresh the WEB-INF folder in the eclipse project explorer.

9. Right click the eclipse project and select properties->java build path. Add the copied jar files in the WEB-INF/lib directory to the project using ‘Add external JARs’ option.



10. Right click the WebContent\WEB-INF directory and select Add new item->Other->XML File. Name the xml file as web.xml.

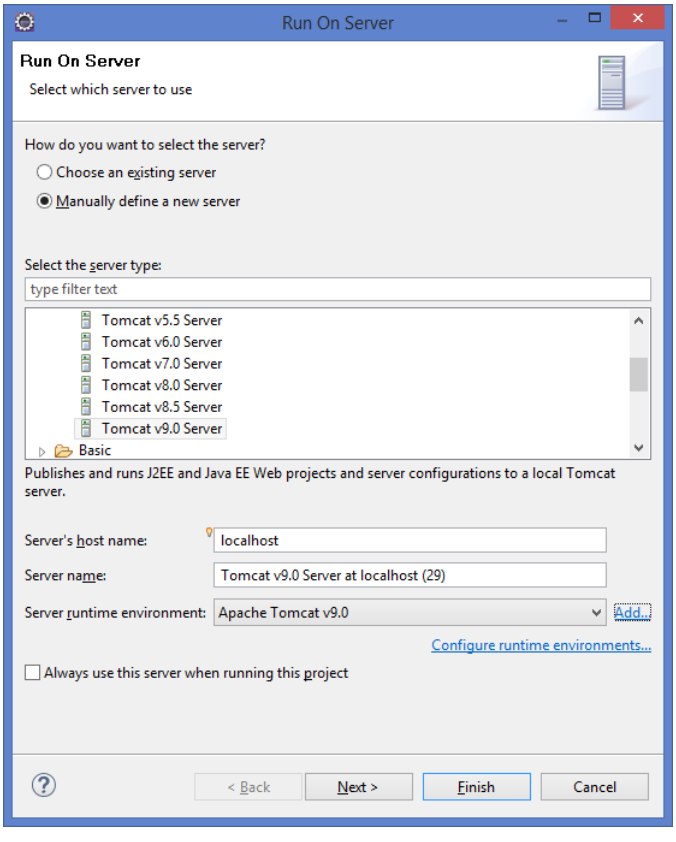
11. Copy the contents of the following web.xml file to the web.xml file in the project.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://xmlns.jcp.org/xml/ns/javaee"
      xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
      version="3.1">
        <display-name>Book Service</display-name>

  <servlet>
    <servlet-name>book_service</servlet-name>
    <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-
class>
    <init-param>
        <param-name>jersey.config.server.provider.packages</param-name>
        <param-value>bookservice</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>book_service</servlet-name>
    <url-pattern>/rest/*</url-pattern>
    </servlet-mapping>

    <welcome-file-list>
    <welcome-file>index.html</welcome-file>
   </welcome-file-list>

</web-app>
```

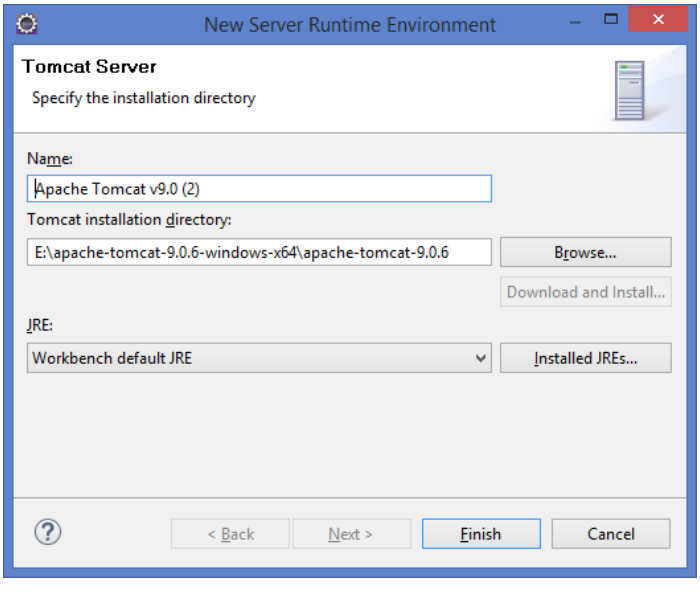
The package name given is bookservice. All urls with the phrase /rest/ will be mapped to the servelet that handles the book REST service.

12. Right click the WebContent directory and select Add new html file. Name the html file as index.html. Leave the html file blank for the moment (this is the welcome file according to the web.xml given above).

13. Right click the project and select Run as ->Run on Server. Select 'Manually define a new Server'.



14. If there's nothing under 'Server runtime environment', click 'Add' link. In the resulting window, give the Tomcat server name and its path.



15. Click next and Finish. The web service will now get deployed. You should see a blank page as the index.html is blank.

16. Open a browser window or open Soap UI (https://www.soapui.org/downloads/latest-release.html) (if it's installed in your pc) and type the following url in the address bar. You should see the book list.

    http://localhost:8080/book_service/rest/books

    If you give the book id as part of the url you should see the relevant book details. If you give an invalid book id, you should get the HTTP 404 error as the book does not exist.

    http://localhost:8080/book_service/rest/books/1

Now the service is deployed. You can stop the Tomcat server from the 'Servers' tab in eclipse.

Creating a JQuery + AJAX client for the REST service

Since the REST service gives a JSON object as the output and since it runs on http, you can write a client using any technology to consume the service. In this example, we'll write a web client that uses AJAX + JQuery javascript.

1. Right click the WebContent directory of the same project and add a new javascript file (Add->Other->Javascript source file). Name it as book.js.

2. Update the index.html file and book.js file with the files attached.

3. Now Run the web project again. This time, you can select the option 'Choose and Existing server' and select the Tomcat server instance that was created earlier.

4. You should be able to fetch the book details by giving the book id in the text box. If an invalid id is given, a custom error message will be given. You can run the client by pasting the url http://localhost:8080/book_service/ in any browser window. This client access the web service that was created and deployed earlier.

5. **Add new functionality to add a new book and to delete a book. You will have to perform the following steps**

i. Modify the Books class to add methods to add a book and to remove a book.
ii. Update the BookService to add a book (you can use the POST http method) and to remove a book (DELETE http method).

    Note that you can choose to do any operation under these HTTP methods. However, the convention is to use GET to retrieve, POST/PUT to create/update and DELETE to delete the remote objects).

    You can test the updated Service with a browser or Soap ui before proceeding to update the client.

iii. Update the index.html and the book.js files accordingly to update the client to access the new operations.

Submission:

Upload the modified code as a single zip file to the courseweb link. The name of the file should be your registration number.