**RESTFUL WEB SERVICE**

Create a web service using the maven project(setting up service and deploying on tomcat)

Eclipse:

Before this install web developer in help—marketplace—search –java web development tool --install

--new – dynamic web project--rightclick on the project –configure---convert to maven project-----finish

-- right click on the project—create package named testmyserver—create---then

--create class named ServiceProvider---check the .xml then finish

--open **pom.xml** in the link([Java RESTful Web Services Tutorial for Beginner with Jersey and Tomcat (codejava.net)](https://www.codejava.net/java-ee/web-services/java-restful-web-services-tutorial-for-beginner-with-jersey-and-tomcat)) go to the copy dependency named 2.29.1(dependency)---paste

<dependency>

    <groupId>org.glassfish.jersey.containers</groupId>

    <artifactId>jersey-container-servlet</artifactId>

    <version>2.29.1</version>

</dependency>

<dependency>

    <groupId>org.glassfish.jersey.inject</groupId>

    <artifactId>jersey-hk2</artifactId>

    <version>2.29.1</version>

</dependency>

Paste

Run---clean project

--right click—JavaEE tools—generte web.xml

**In web.xml file**

--copy and paste the <servelet> code in the link

<servlet>

    <servlet-name>Jersey REST 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>net.codejava</param-value>

    </init-param>

    <load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

    <servlet-name>Jersey REST Service</servlet-name>

    <url-pattern>/**rest**/\*</url-pattern> ///in the place of rest give **test\_service --for accessing the url**

</servlet-mapping>

Add after the </welcome\_file\_list> the <servelet> above

---create class----ServiceProvider(if created above then skip)

Copy and paste the login code in class

package testMyService;

import javax.ws.rs.GET;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.PathParam;

import javax.ws.rs.Produces;

import com.google.gson.Gson;

//import tips.java.api.ResponseMessage;

@Path("login")

public class ServiceProvider {

@GET

@Produces("application/xml")

public String requestLogin()

{

StringBuilder sb = new StringBuilder();

sb.append("<note>");

sb.append("</note>");

return sb.toString();

}

@POST

@Path("/{p1}/{p2}")

@Produces("application/json")

public String login(@PathParam("p1") String username, @PathParam("p2") String password)

{

ResponseMessage msg = new ResponseMessage();

if(username.equals("Admin") && password.equals("Admin1"))

{

msg.setResponseCode("200");

msg.setResponseMsg("welcome");

}

else

{

msg.setResponseCode("-1");

msg.setResponseMsg("Invalid Credentials");

}

return new Gson().toJson(msg);

}

}

Add one more dependency in the classRestDemo

--right click on the projrct—new—html—

<title> My service</title>

<h1>Welcome</h1>

---run---clean

We get an error named –unstopped class file major version

Bcoz dependency 2.29.1 supports JDK 17 and based on that

But we are running JDK 20

So error occurs

**Google---jersey dependency jdk 20 (first link)**

----change in dependency 2.39.1

**Add after <packaging>war</packaging> in pom.xml or web.xml (check)**

<dependencies>

<dependency>

<groupId>org.glassfish.jersey.containers</groupId>

<artifactId>jersey-container-servlet</artifactId>

<version>2.39.1</version>

</dependency>

<dependency>

<groupId>org.glassfish.jersey.inject</groupId>

<artifactId>jersey-hk2</artifactId>

<version>2.29.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->

<dependency>

<groupId>com.google.code.gson</groupId>

<artifactId>gson</artifactId>

<version>2.10</version>

</dependency>

</dependencies>

--then---run—clean—check the clean

Under the same package named **testMyService** create a class named--- **ResponseMessage**

**Paste the below code**

package testMyService;

public class ResponseMessage {

private String responseCode;

private String responseMsg;

private Object data;

public String getResponseCode() {

return responseCode;

}

public void setResponseCode(String responseCode) {

this.responseCode = responseCode;

}

public String getResponseMsg() {

return responseMsg;

}

public void setResponseMsg(String responseMsg) {

this.responseMsg = responseMsg;

}

public Object getData() {

return data;

}

public void setData(Object data) {

this.data = data;

}

}

Then in the browser url occurs then for url \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**/test\_service/login**(add this)

---right click on project classdemo---export---WAR(choose the destination)

--copy the war file ---go to the webapps----paste

Then open the command prompt--- (in the location)

Startup //command (to start the tomcat)

Then postman [Postman API Platform | Sign Up for Free](https://www.postman.com/) via this link or else go to the above given link i.e., [Java RESTful Web Services Tutorial for Beginner with Jersey and Tomcat (codejava.net)](https://www.codejava.net/java-ee/web-services/java-restful-web-services-tutorial-for-beginner-with-jersey-and-tomcat) at the bottom there is the link to postman

---Then paste the url\_\_\_\_\_\_\_\_\_**/login/a/b**

(we get (p1) (p2) and username and password

--- Then for the same url\_\_\_\_\_\_**/login/Admin/Admin1** -----we get according to the code