

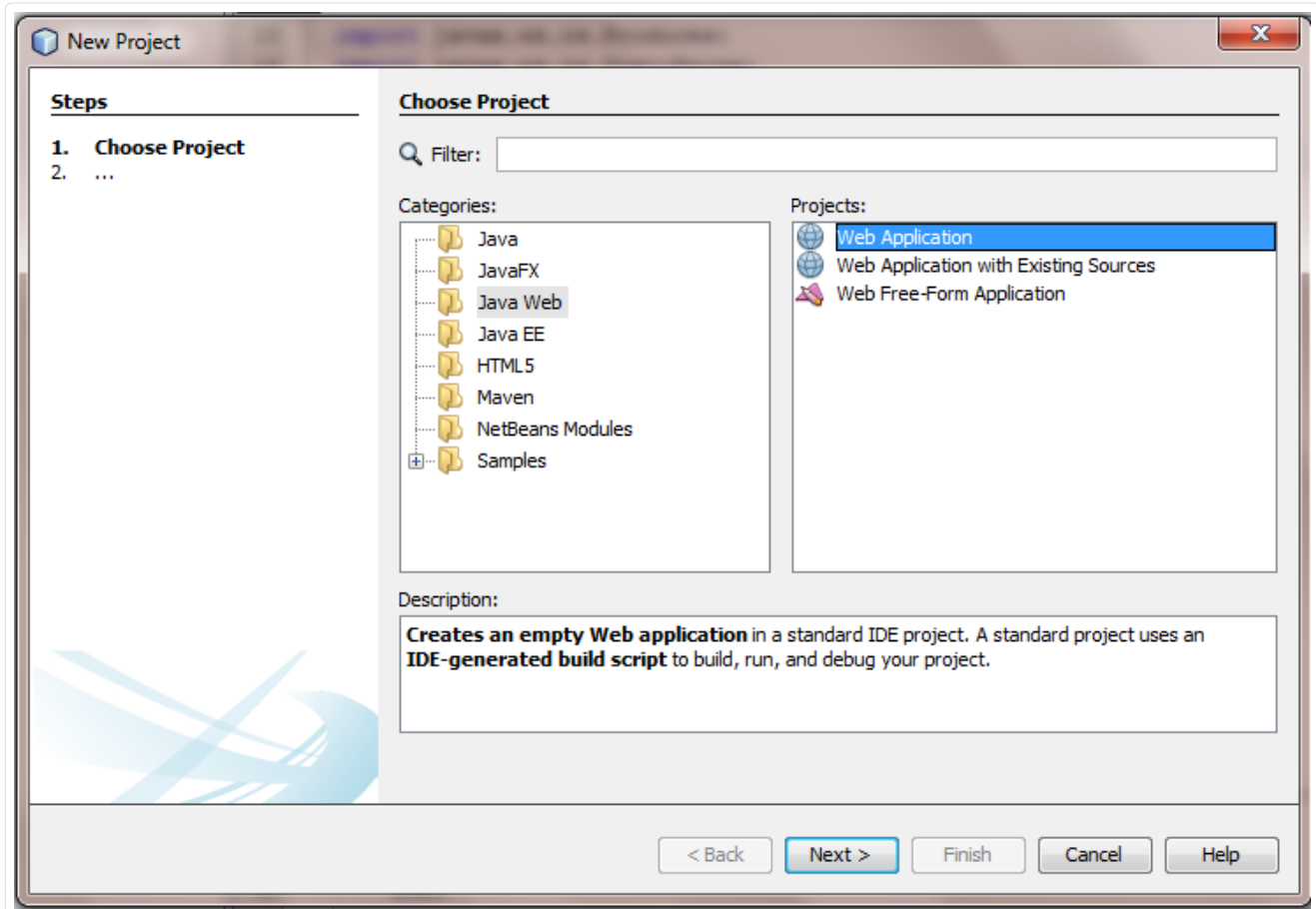
# Creating a Simple Restful Web Service

## Creating a Simple Restful Web Service

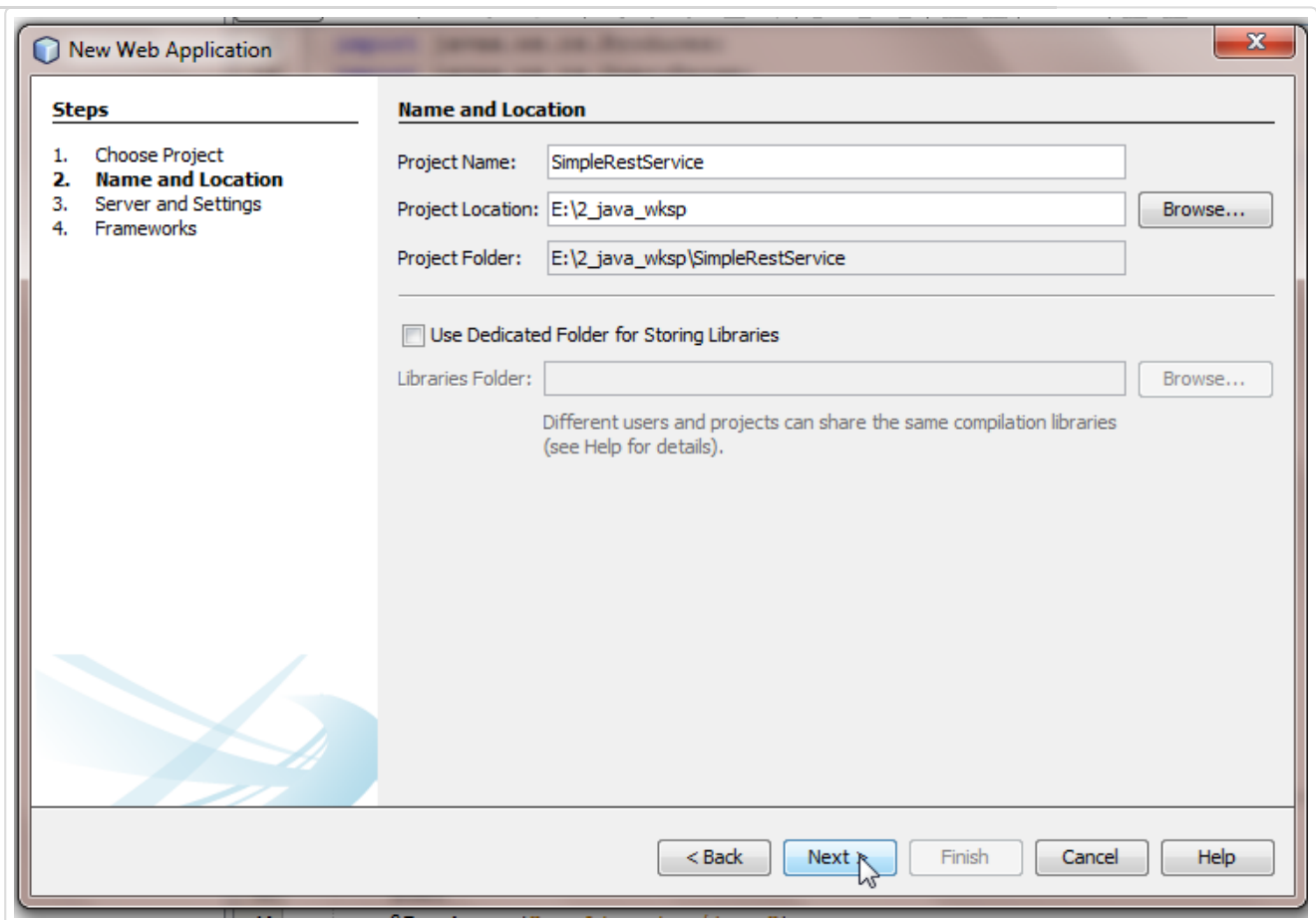
I was creating a restfull webservice in java JDK1.7 using NetBeans 8 IDE and GlassFish Server Open Source Edition 4.0

This webservice is implementation of GET and returns a JSON response.

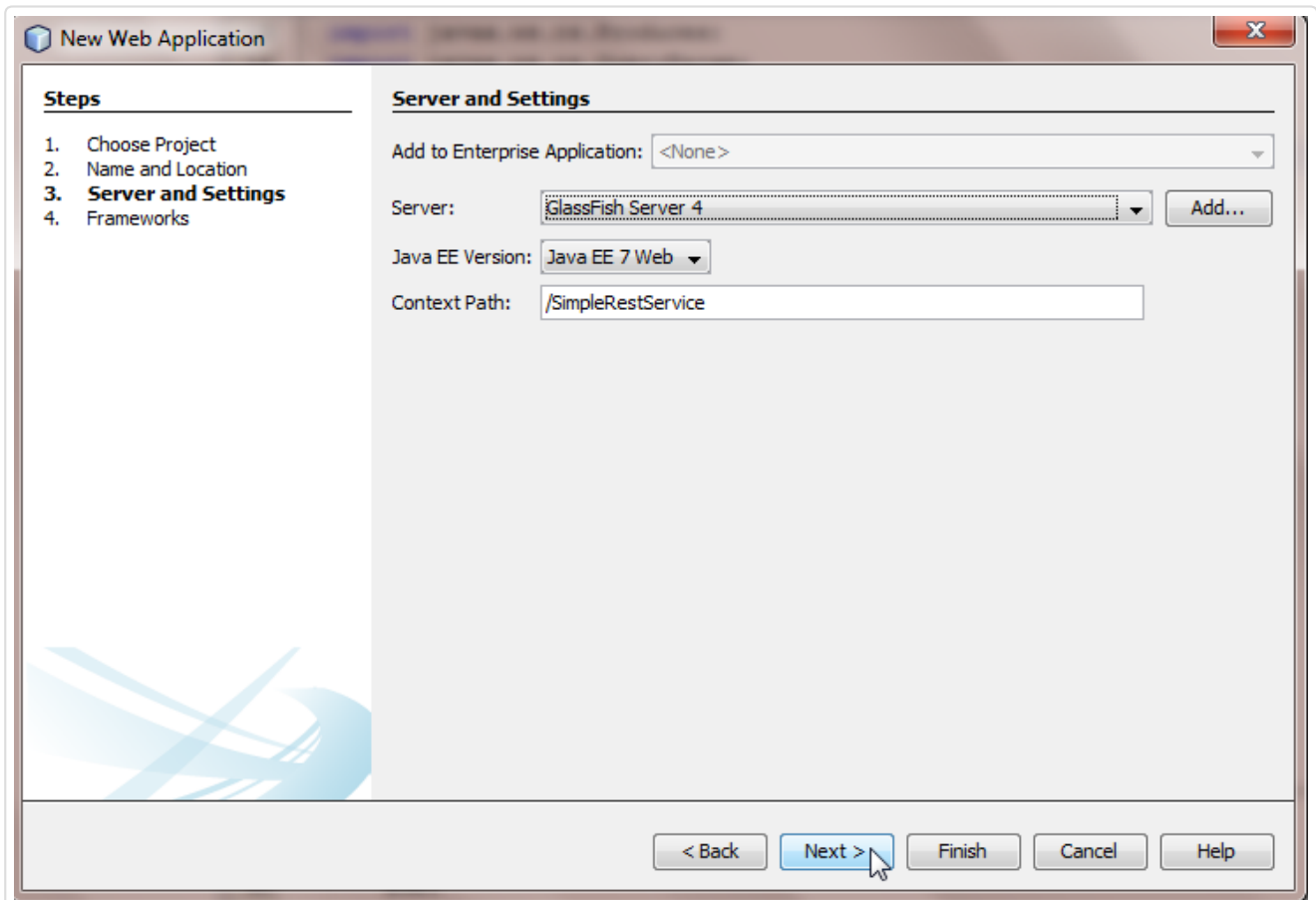
Create a simple dynamic webproject using NetBeans IDE



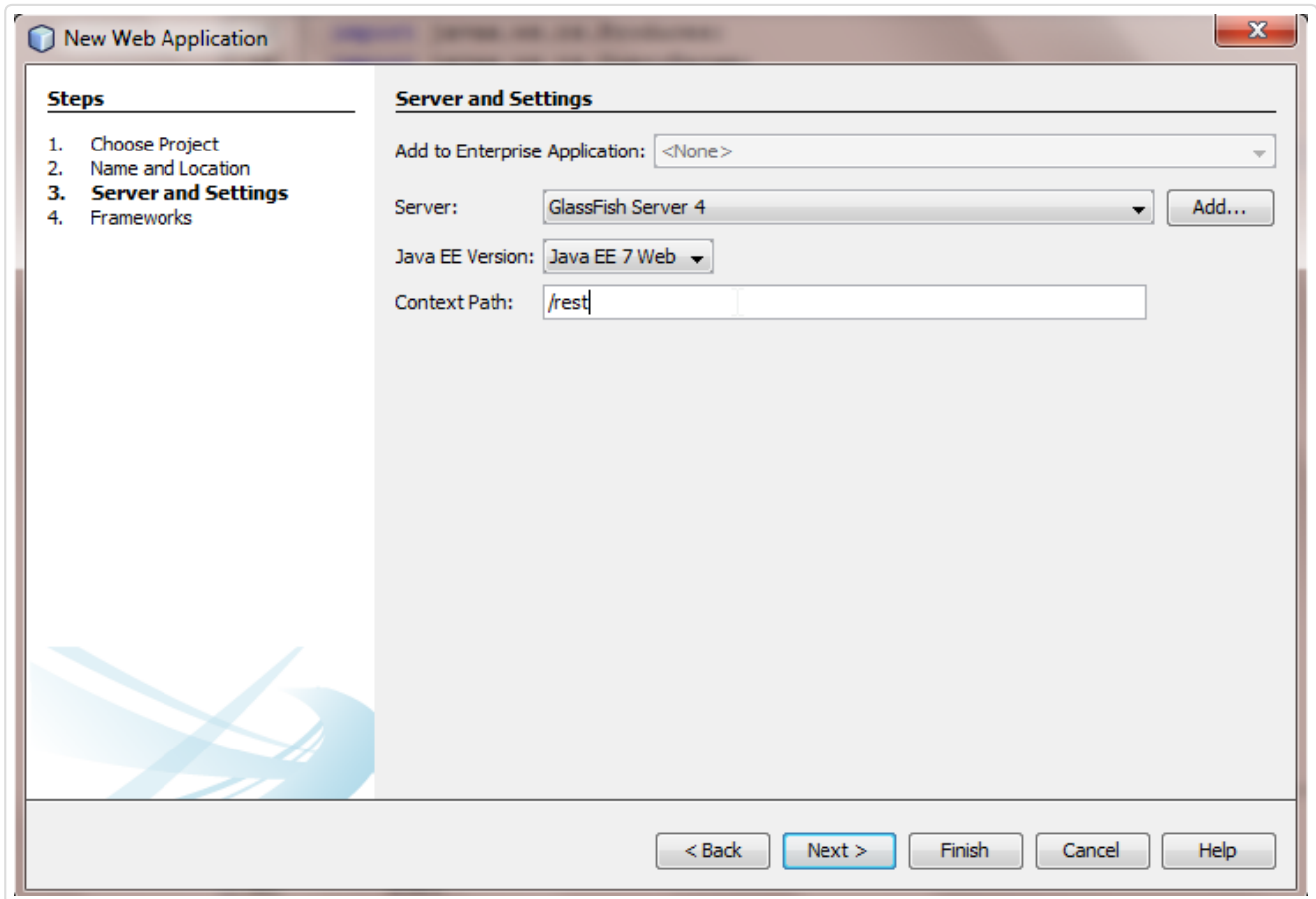
Give a name for the project as say "SimpleRestService"



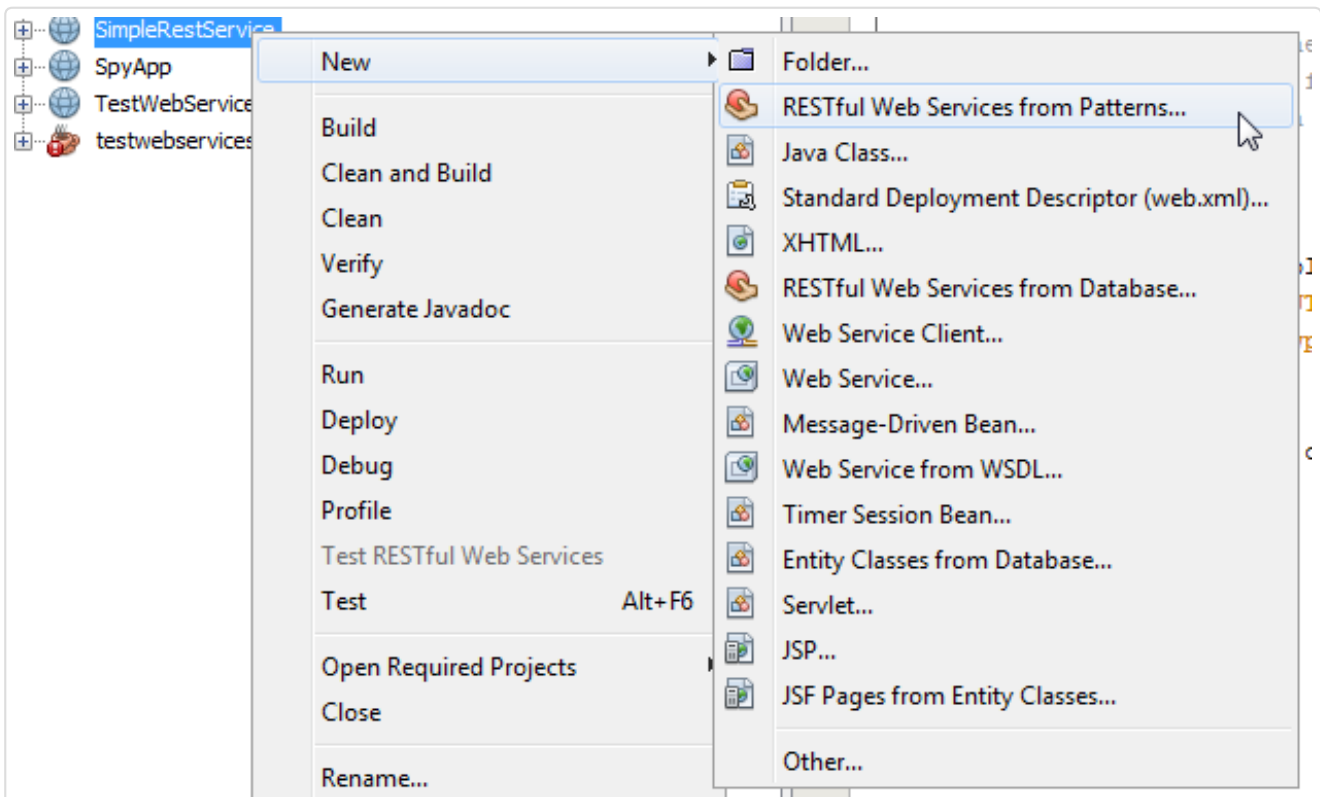
Select the server. In this i have selected GlassFish 4 opensource server provided with the NetBeans IDE. You can select Tomcat 6.xx or Tomcat 7 servers.



You can configure the context path. I have changed it to simpler name "rest". Again this is a personal choice.

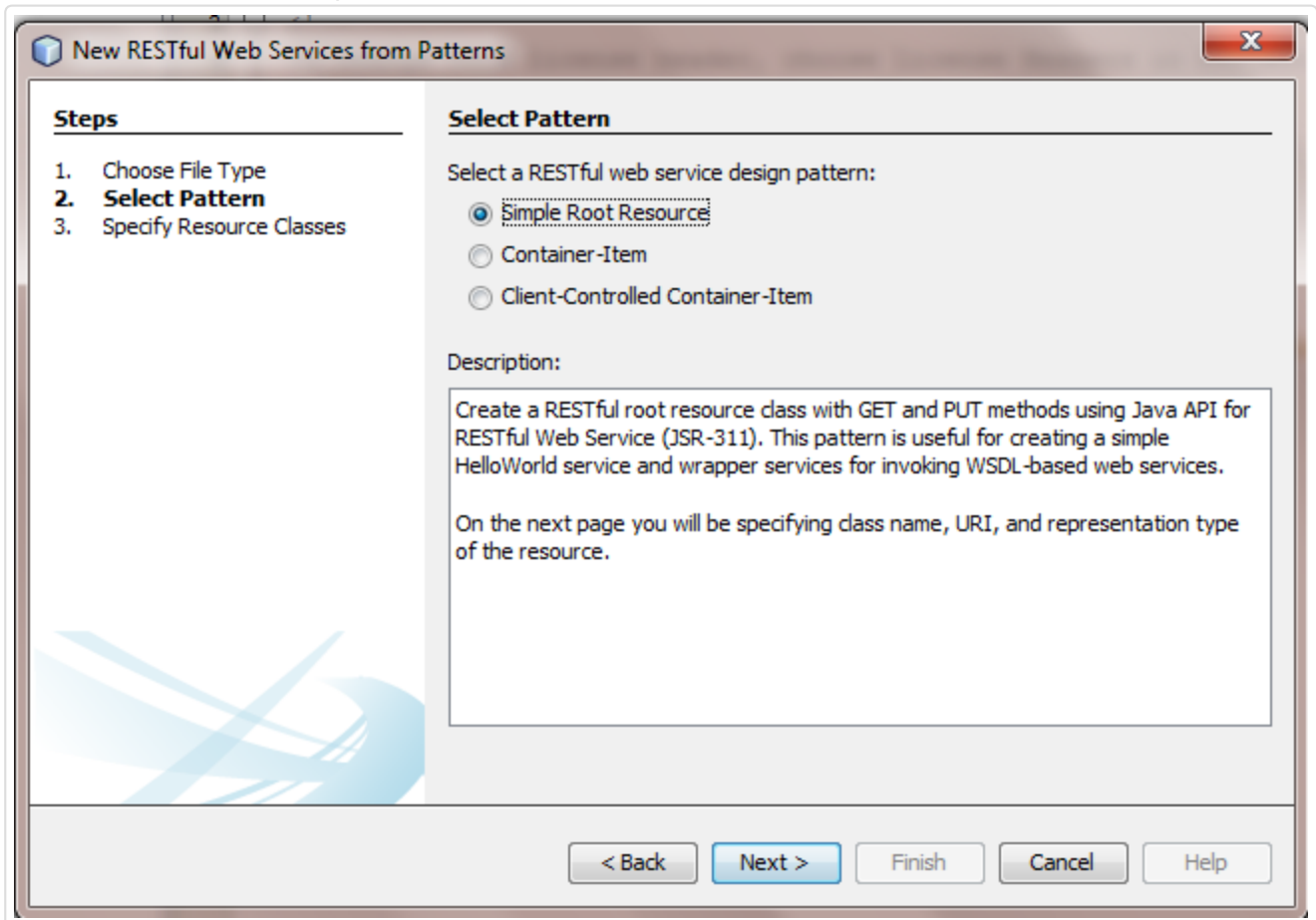


Once done and the project created, now rightclick and using netbeans context wizard, you can create a new restfull webservice as shown below:

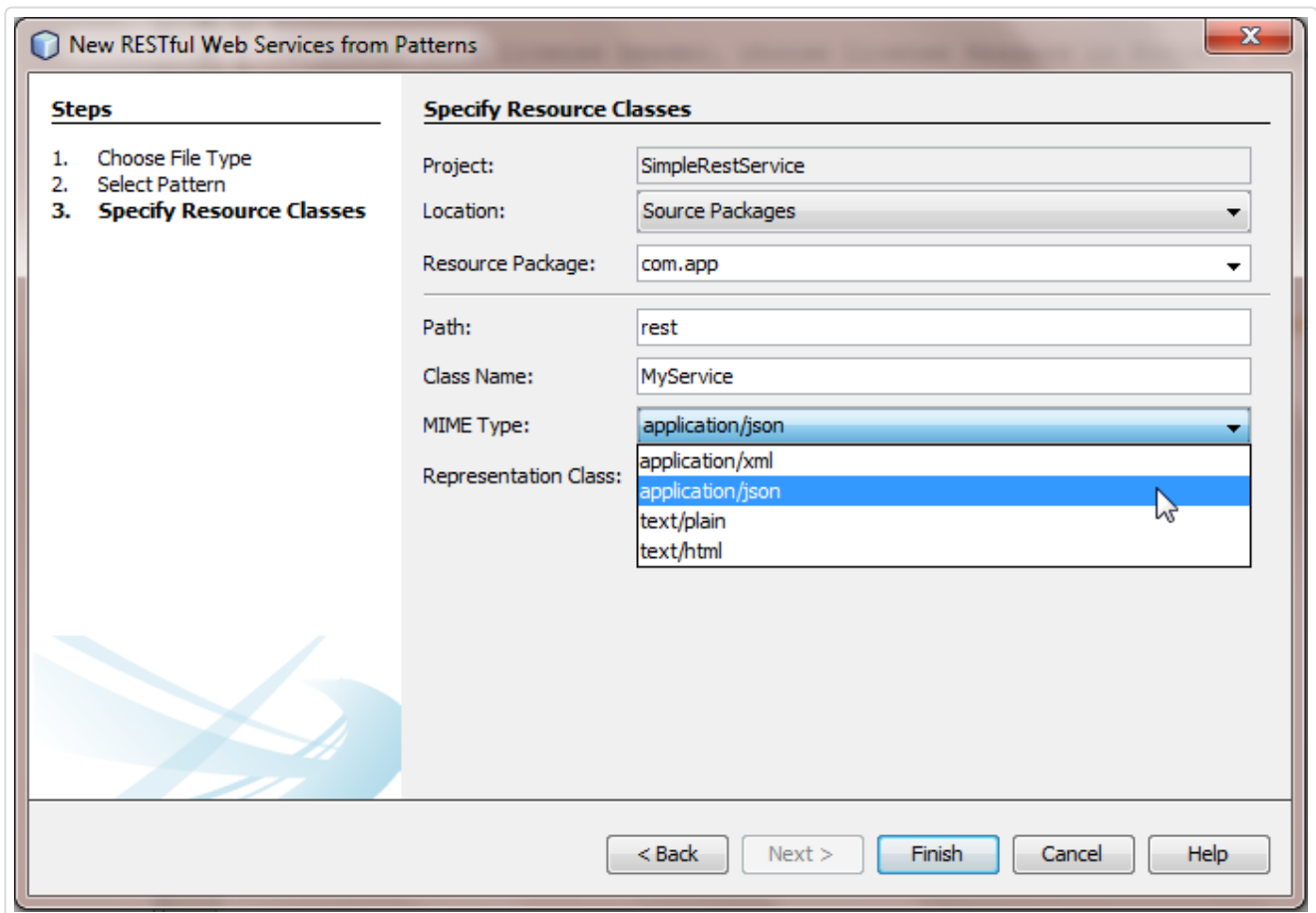


Note: We can create Restful webservice using Patterns or from Database. I am reserving the Database part to my next blog.

In the wizard select "Simple Root Resource"



In the next step select the Mime Type as application/json and give a class name with its package.



After clicking on "Finish", you will get the class built for you. Open the class name provided by you and go to the following method and do modifications as below

```
@GET
@Produces("application/json")
public String getJson(@QueryParam("name") String name) {
    switch(name) {
        case "1" :
            return "{\"name':'ABCD', 'email':abcd@gmail.com}";
        case "2":
            return "{\"name':'PQRS', 'age':pqrs@gmail.com}";
        default:
            return "{\"name':'unknown', 'age':-1}";
    }
}
```

Note: @QueryParam is used to read the parameters which you give to the webservice through url say `http://<>:<>/Projectname/servicename?parameter1=value`

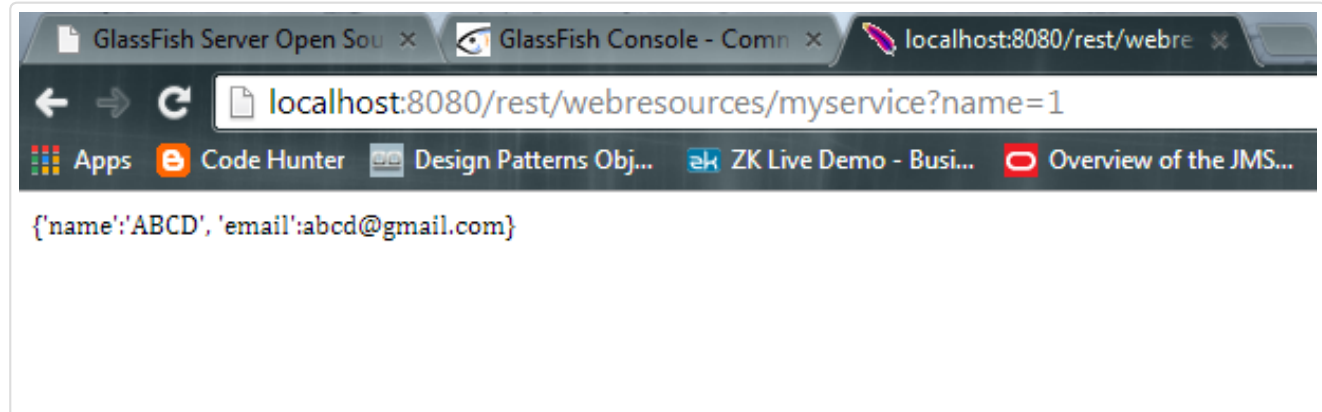
A more simpler example could be `http://xyz:8080/SimpleRestService/myservice?name=1`

Also in the method currently i have hard coded JSON response based on simple switch based on name parameter. You can do further complex business rule execution, like fetching data from database based on some conditions etc and accordingly populate the JSON response.

Since i am using Glassfish Server, the following modifications have to be done in the web.xml of the project

```
<servlet>
  <servlet-name>ServletAdaptor</servlet-name>
  <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>ServletAdaptor</servlet-name>
  <url-pattern>/resources/*</url-pattern>
</servlet-mapping>
```

Once done, clean and build the project and deploy it on the server (in my case Glassfish)  
Now we are ready to access the webservice through browser



As mentioned in the above, you can see the response in the browser.  
This is a simple 11-step process for creating a simple Restful webservice.  
Happy reading!!!