|  |  |
| --- | --- |
| **Roll No.:** A059 | **Name:** Chinmay Parikh |
| **Prg/Yr/Sem:** B.Tech(I.T.)/4th /7 | **Batch:** A3 |
| **Date of Experiment:** 4/10/2014 | **Date of Submission:** 2/11/2014 |

**Aim:** Creating Java Web Application Project and generating Entity Classes and RESTful Web Services from a Database.

**Scenario:** Creating Java Web Application Project and generating Entity Classes and RESTful Web Services from a Database.

**Detailed steps:**

**Creating the Java Web Application Project:**

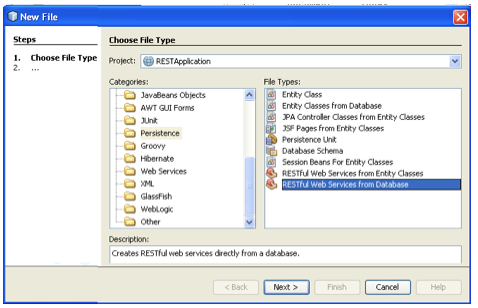
1. Click **File** and then **New Project**.
2. For **New Project**: On the **Categories** side, choose **Java Web** and on the **Projects side**, select **Web Application**.
3. Click **Next**.
4. For **Name and Location**, under Project Name, enter **RESTApplication**.

**Generating Entity Classes and RESTful Services :**

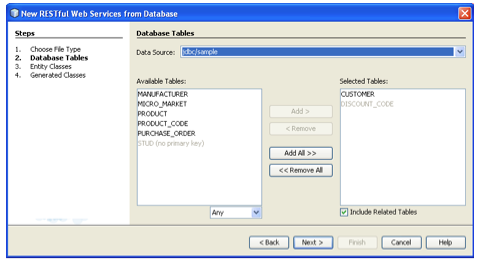
We have a Java web application now, add entity classes and RESTful web services to the project.

**To generate entity classes and RESTful web services :**

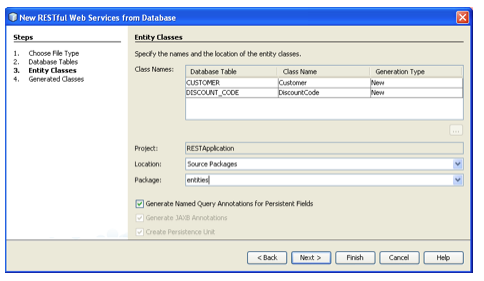
1. Right-click the **RESTApplication** node and choose **New -> Other -> Web Services -> RESTful Web Services from Database**. The New RESTful Web Service wizard opens, on the Database Tables panel.



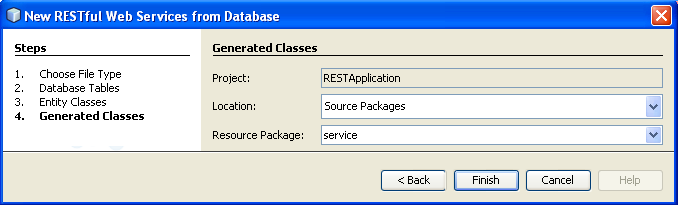
1. In the Database Tables panel, if you are using the GlassFish server, select the jdbc/sample data source from the Data Source drop-down field.
2. Under Available Tables, select CUSTOMER and then click Add. The DISCOUNT\_CODE table, which has a relationship with the CUSTOMER table, is automatically added to the Selected Tables list too.



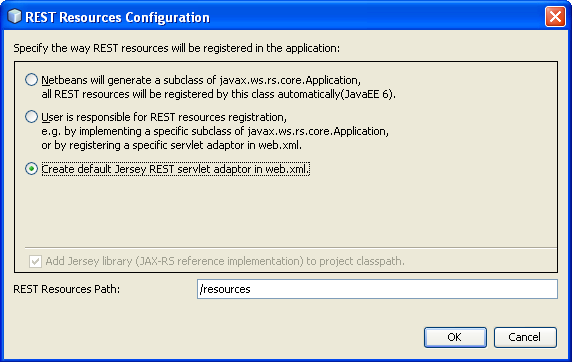
1. Click Next. The Entity Classes page opens. Under Package, type entities.



1. Click Next. The Generated Classes panel opens. In this panel you can set the location of the RESTful web service classes that the IDE generates for your project. The EE6 web services use JAXB annotations in the entity classes.

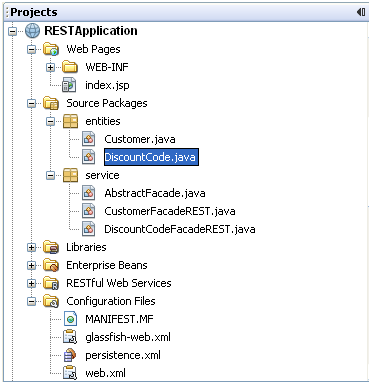


1. Accept the default packages and locations and click Finish. The IDE generates entity classes. Then a dialog opens, asking how you want to register RESTful resources. Choose **create a default Jersey REST servlet adaptor** in web.xml and click OK.



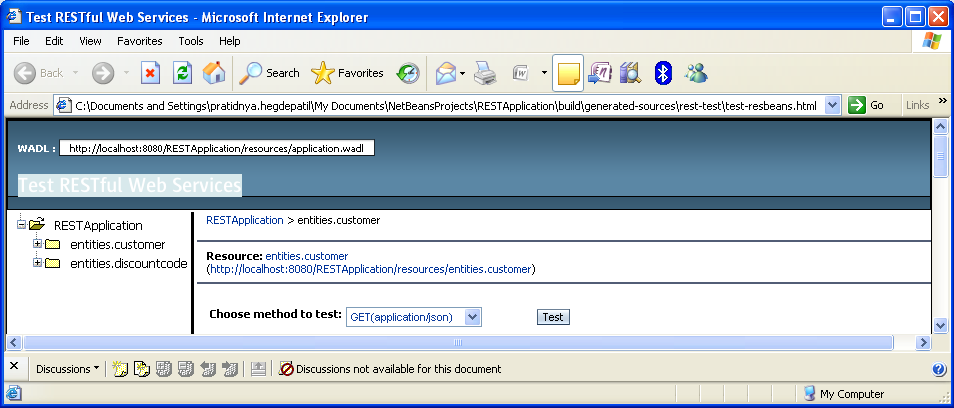
The IDE now generates the RESTful web services. When the IDE is finished, look in the Projects window. The generated entity classes are in the entities package. Services are in the service package.

EE6 RESTful web services from a database, NetBeans IDE uses JAXB annotations in the entity classes and EJB session facades for the service classes. This removes the need for converter classes and generates simpler code.



**Testing the RESTful Web Services :**

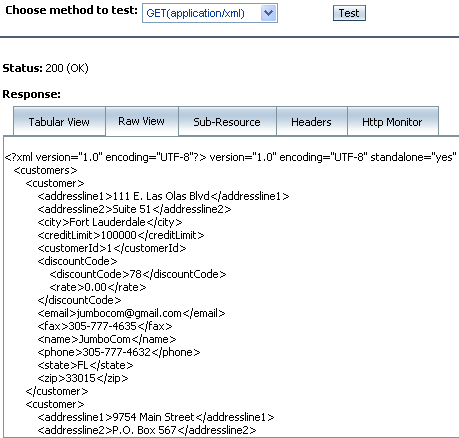
1. Right-click the project node and choose Test RESTful Web Services. The server starts and the application is deployed. When deployment is complete, the browser displays your application, with a link for each of the web services.



1. On the left-hand side is the set of root resources. Here they are named customers and discountCodes.
2. Click the customers node. The browser window shows you a list of parameters for testing the Customers service.

You can set the following parameters:

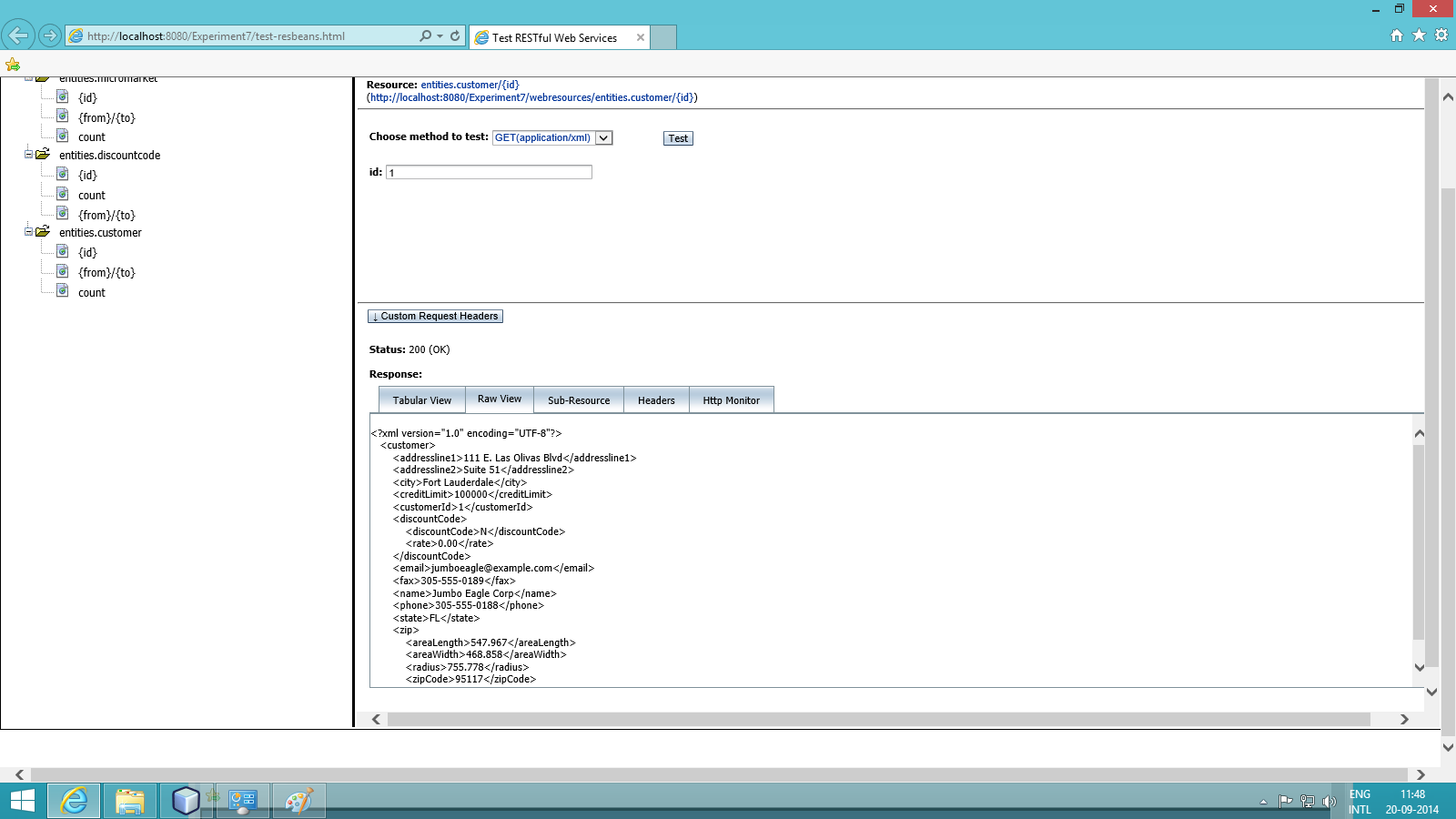
1. **Choose method to test:** Choose the GET or POST method and the MIME type from a drop-down list.
2. Click on the Methods drop-down list to select GET(application/xml). The result is displayed in the Test Output section.

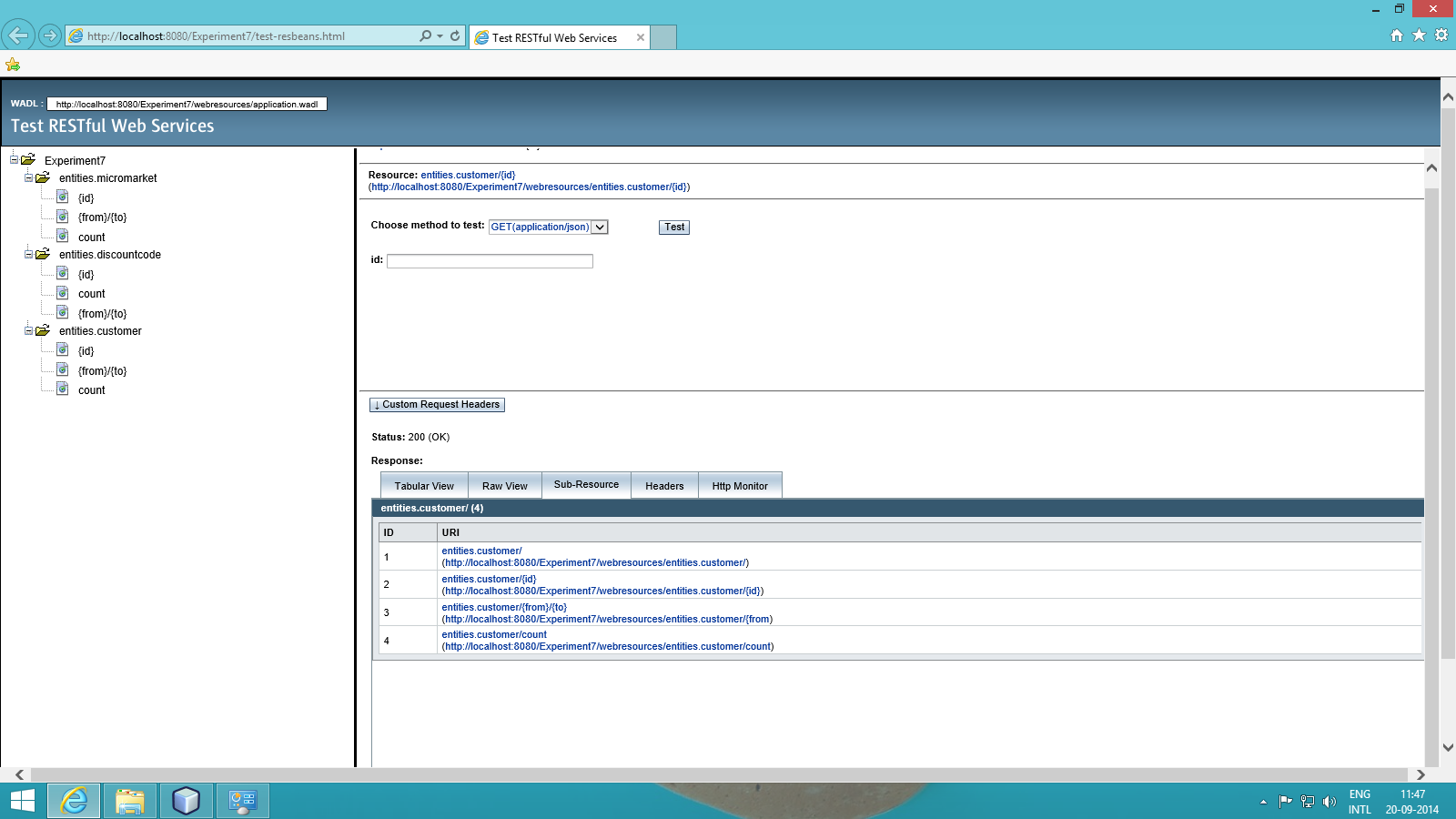


There are 5 tabs in the Test Output section.

* 1. The Tabular View is a flattened view that displays all the URIs in the resulting document, which you can navigate to by clicking on the links.
  2. The Raw View displays the actual data returned. Depending on which mime type you selected (application/xml or application/json), the data displayed will be in either XML or JSON format, respectively.
  3. The Sub Resource tab shows the URLs of the root resource and sub resources. When the RESTful web service is based on database entity classes, the root resource represents the database table, and the sub resources represent the columns.
  4. The Headers tab displays the HTTP header information.
  5. The HTTP Monitor tab displays the actual HTTP requests and responses sent and received.
  6. Exit the browser and return to the IDE.

**Printouts:**





**Conclusion:** RESTful services shows us to identify the web service by creating a client web service and access it like a server to manipulate and store data in a database.