Arun Gupta is a technology enthusiast, a passionate runner, and a community guy who works for Sun Microsystems.
And this is his blog!

« « TOTD #111: Rails Scaffold for a pre-existing table using Oracle and GlassFish

October 8, 2009

TOTD #112: Exposing Oracle database tables as RESTful entities using JAX-RS, GlassFish, and NetBeans

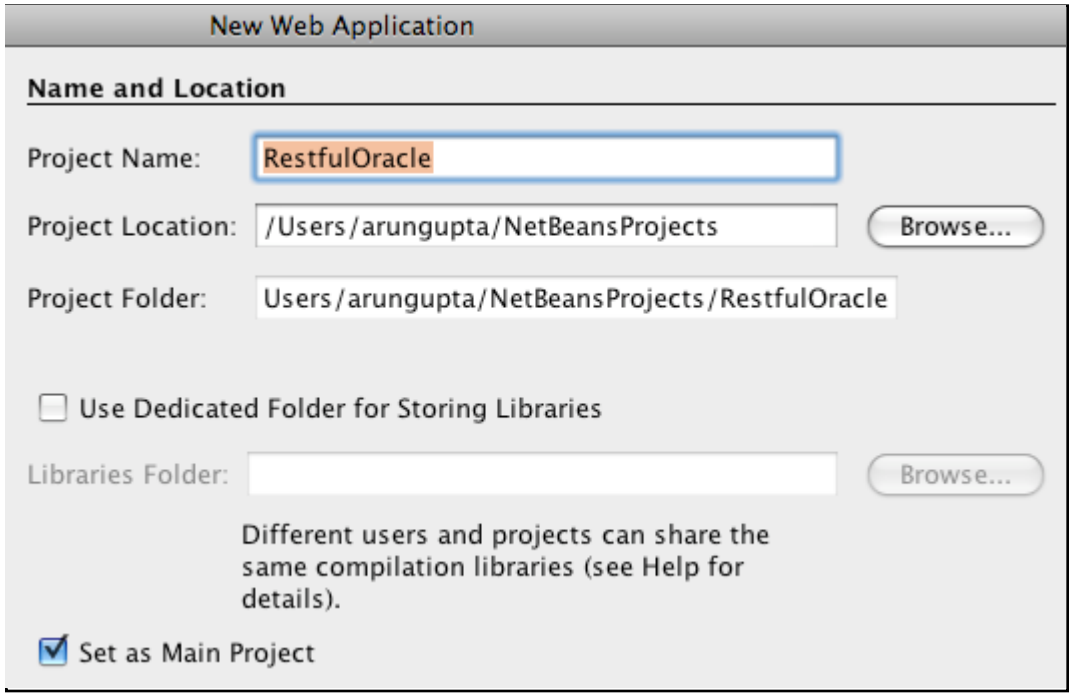
Categories: frameworks, glassfish, javaee, netbeans, totd, webservices

Edit This

This **T**ip **O**f **T**he **D**ay explains how to expose an existing Oracle database table as a RESTful Web service endpoint using NetBeans tooling and deployed on GlassFish.

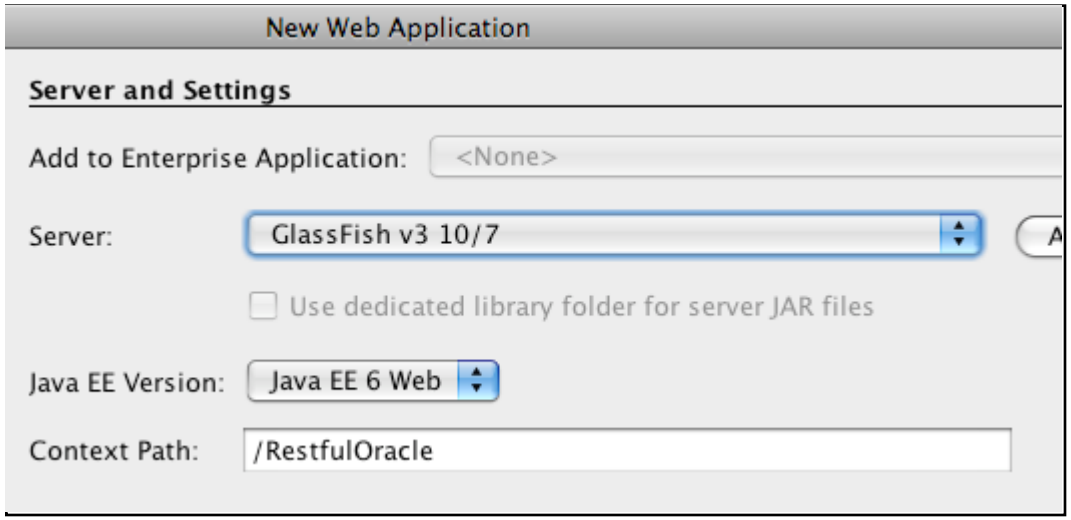
Lets get started!

1. Configure GlassFish v3 10/7 or a later nightly in a recent NetBeans 6.8 build (latest nightly). As issue# 9885 is fixed, so copy ojdbc6.jar in the "domains/domain1/lib/ext" directory.
2. Create a Web application
    1. Create a new "Web application" and name the project "RestfulOracle":



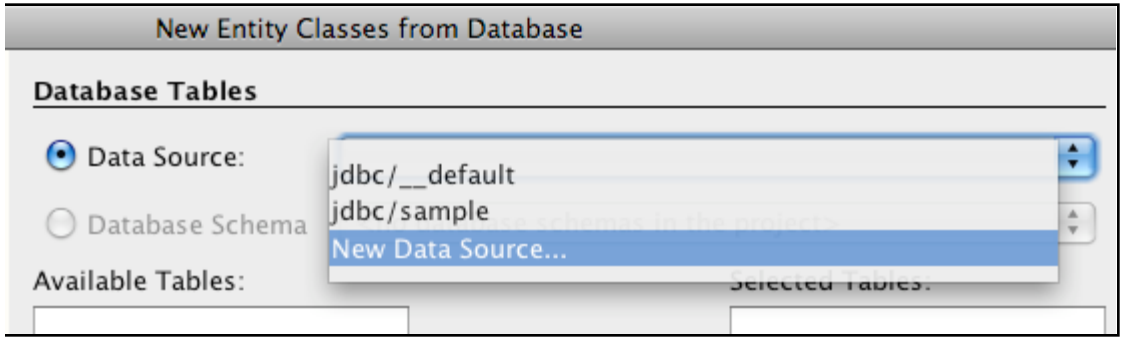
click on "Next >".

2. Choose the newly added server and "Java EE 6 Web" as the Java EE version:

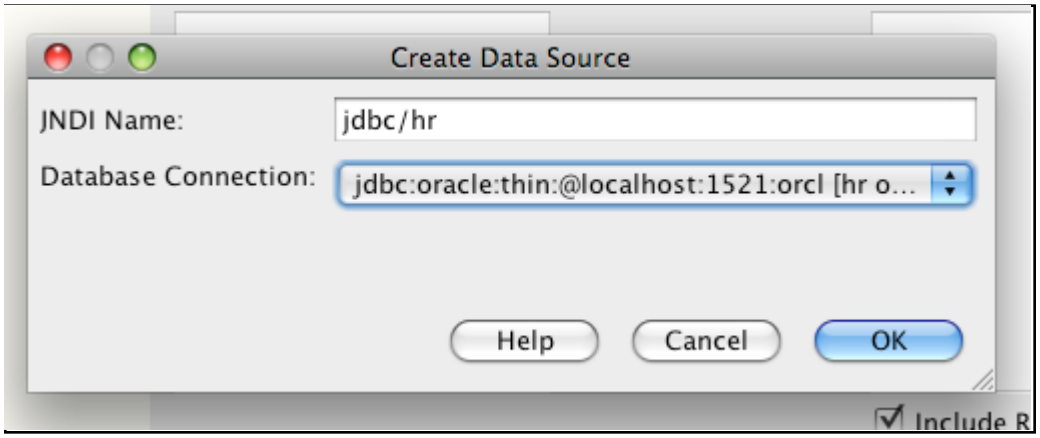


and click on "Finish".

3. Create JPA entities for "HR" schema. The steps outlined below uses NetBeans solely for creating the JPA entities. Alternatively, TOTD #108 explains how to define a JDBC connection pool and JDBC resource using "asadmin" CLI and then use that resource from within NetBeans. Either way, the JDBC resource is stored in the underlying "domain.xml".
    1. Right-click on the project and select "New", "Entity Classes from Database…".
    2. In "Data Source:" select "New Data Source…" as shown below:

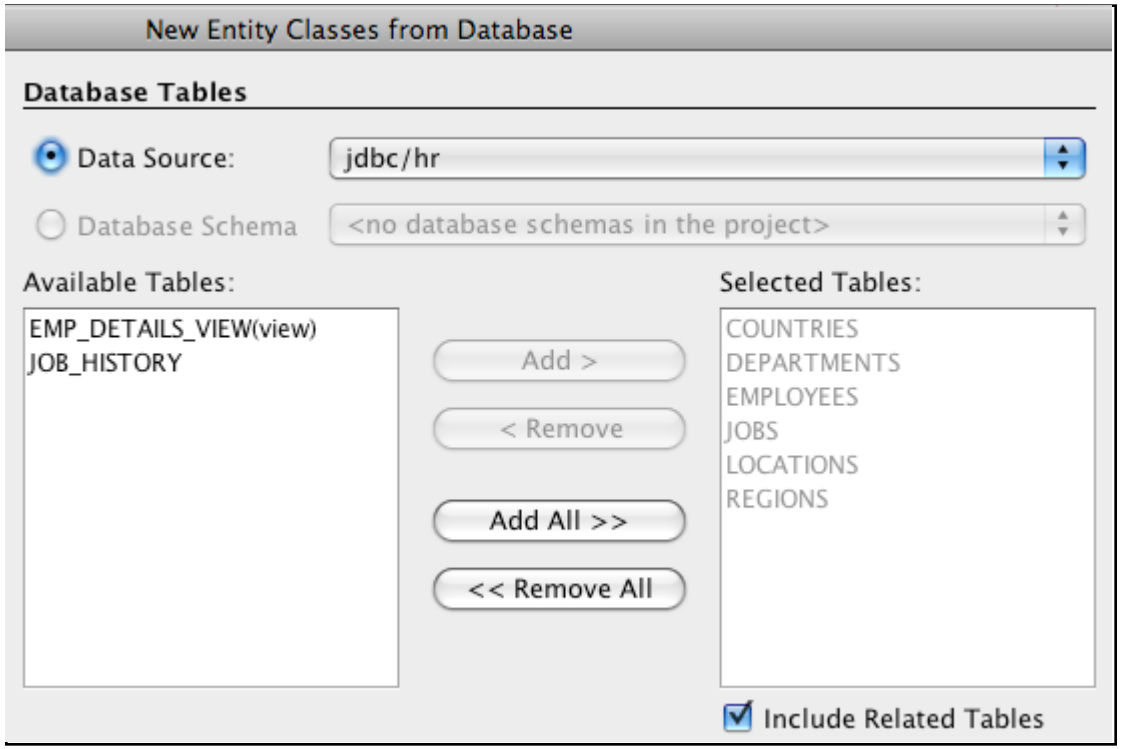


    3. Specify the JNDI name as "jdbc/hr" and choose the pre-configured database connection as shown below:



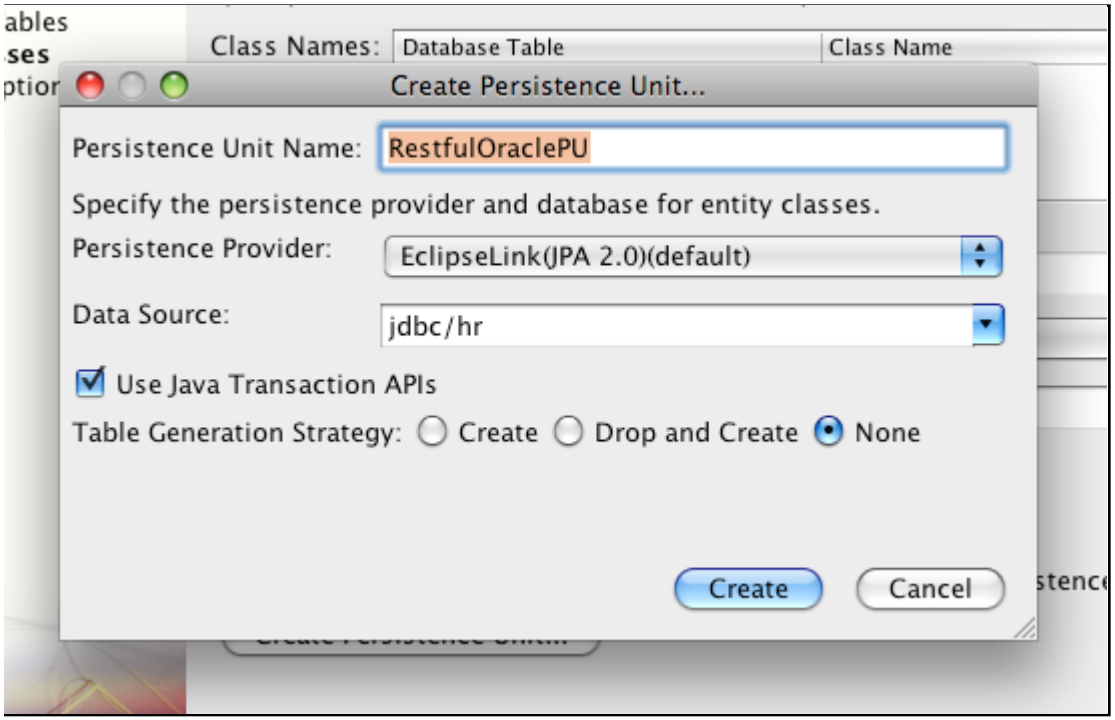
TOTD #107 explains how to configure Oracle database in NetBeans.

4. In the list of "Available Tables:", select "EMPLOYEES" and click on "Add >" to see the following:



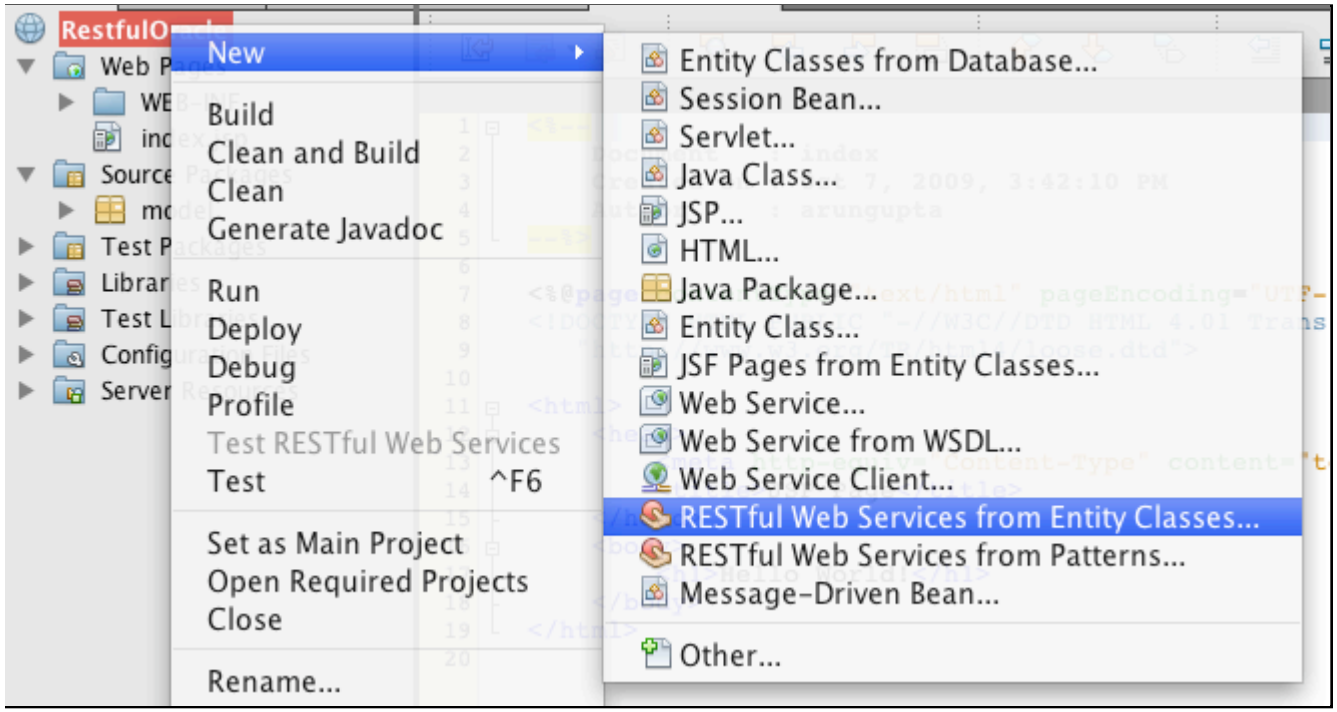
Notice the list of related tables are included as well. Click on "Next >".

5. Specify the package name as "model".
6. Click on "Create Persistence Unit…", take the defaults, and click on "Create":

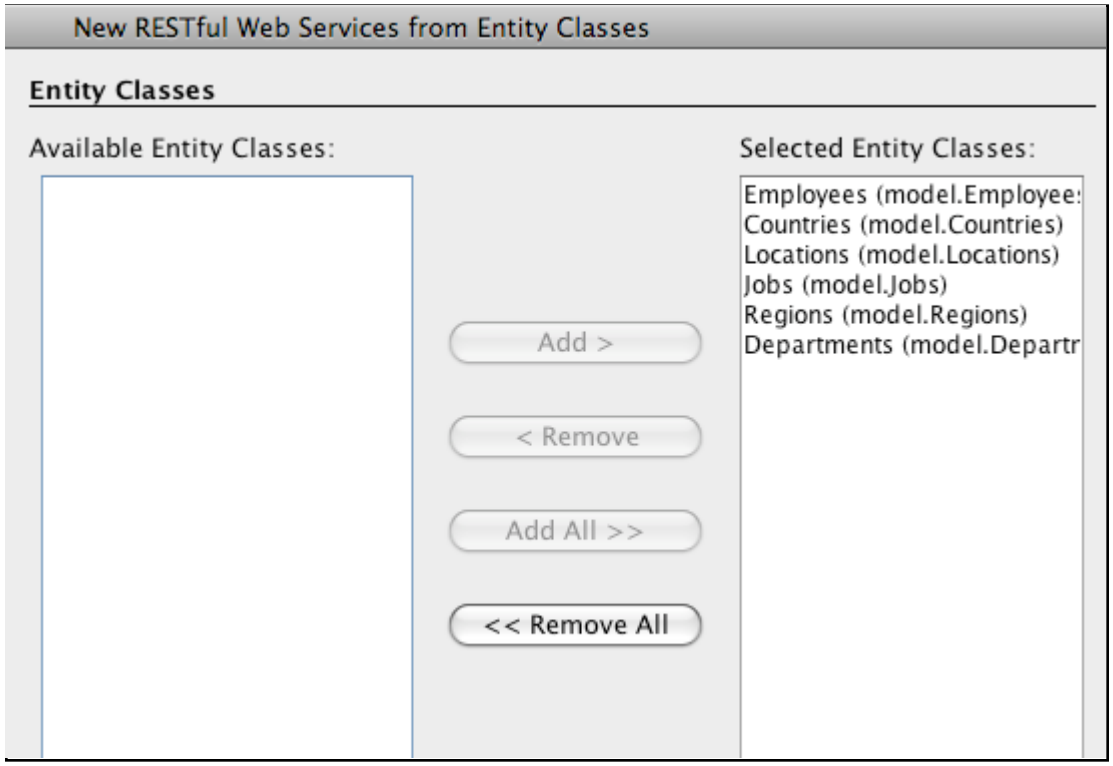


and click on "Finish". Notice EclipseLink, the reference implementation for JPA 2.0, is used as the persistence provider. This generates POJOs that provide database access using JPA 2.0 APIs. These APIs are included as part of the Java EE 6 platform.

4. Create RESTful entities
    1. Right-click on the project and select "RESTful Web Services from Entity Classes…":

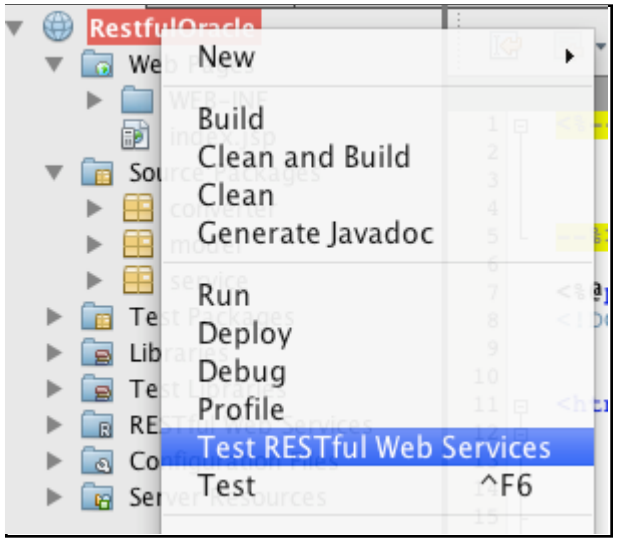


2. Select "Employees (model.Employees)" from "Available Entity Classes:" and click on "Add >" to see the following:



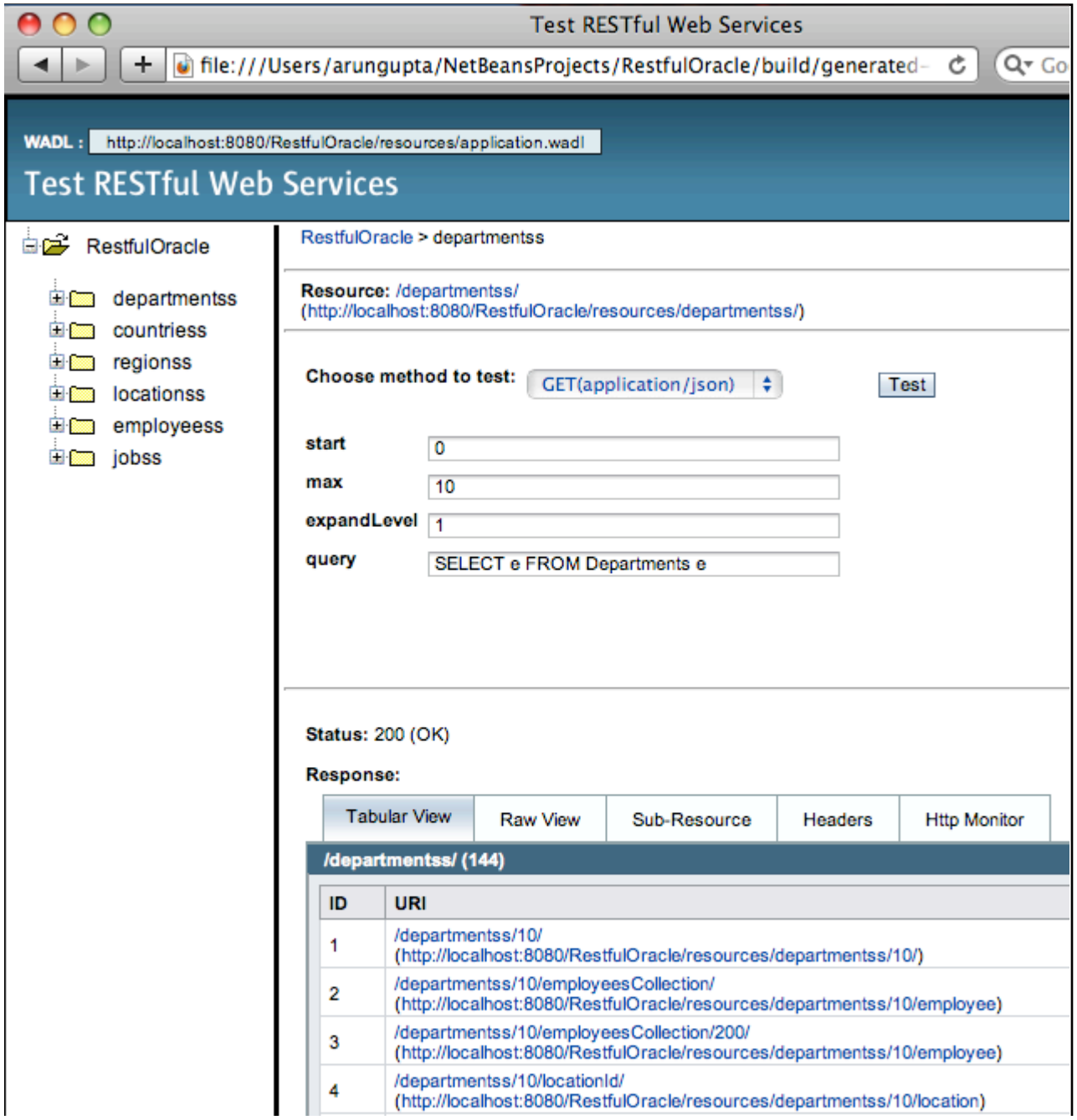
click on "Next >", take the defaults, and click on "Finish". This generates a bunch of wrapper classes using JAX-RS to expose the JPA Entity classes as RESTful Web services. JAX-RS 1.1 is also included as part of the Java EE 6 platform.

5. Run the Web service
    1. Right-click the project and select "Test RESTful Web Services":

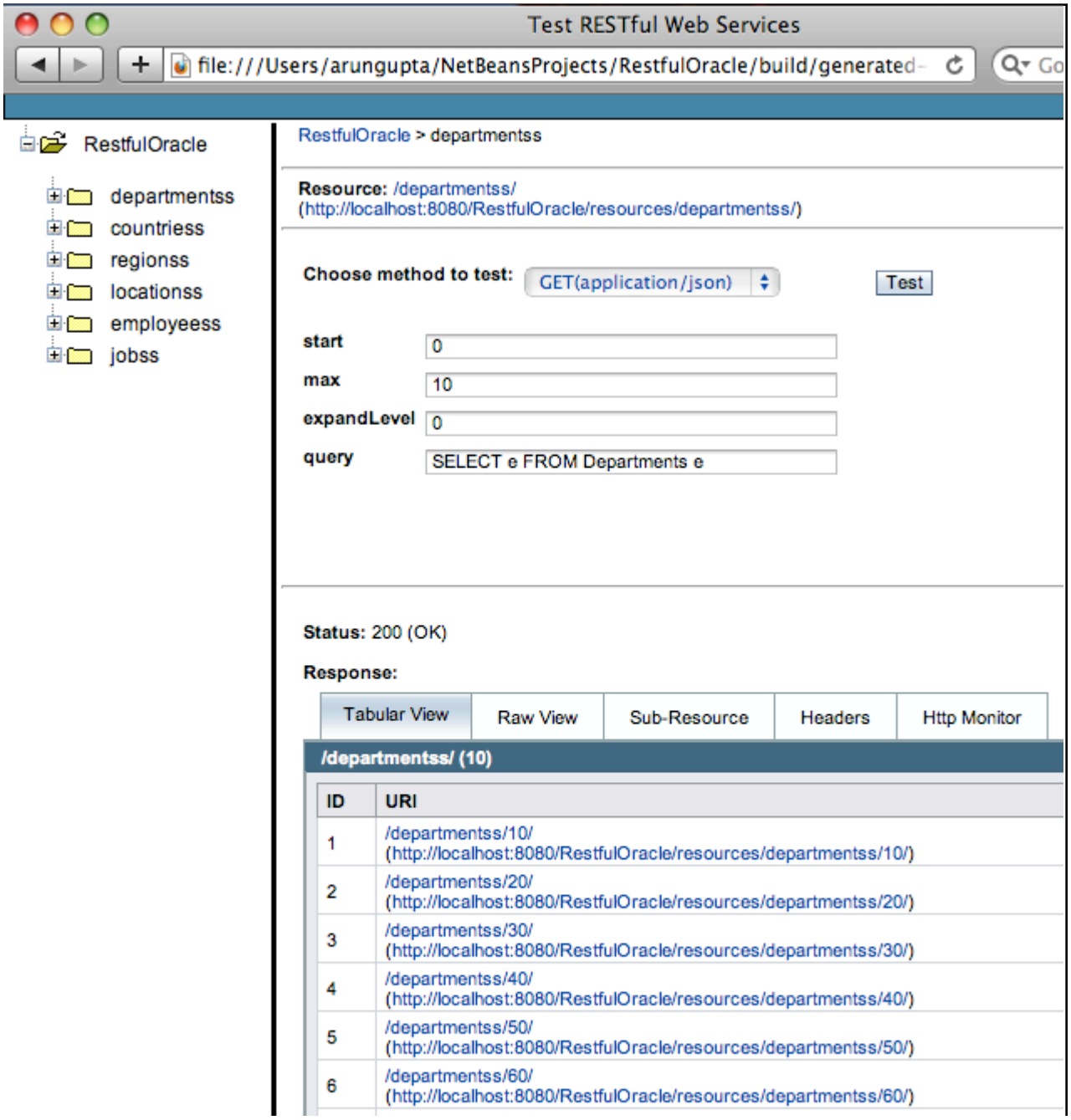


This deploys the created Web application on the selected GlassFish build and displays the following page in the default browser:

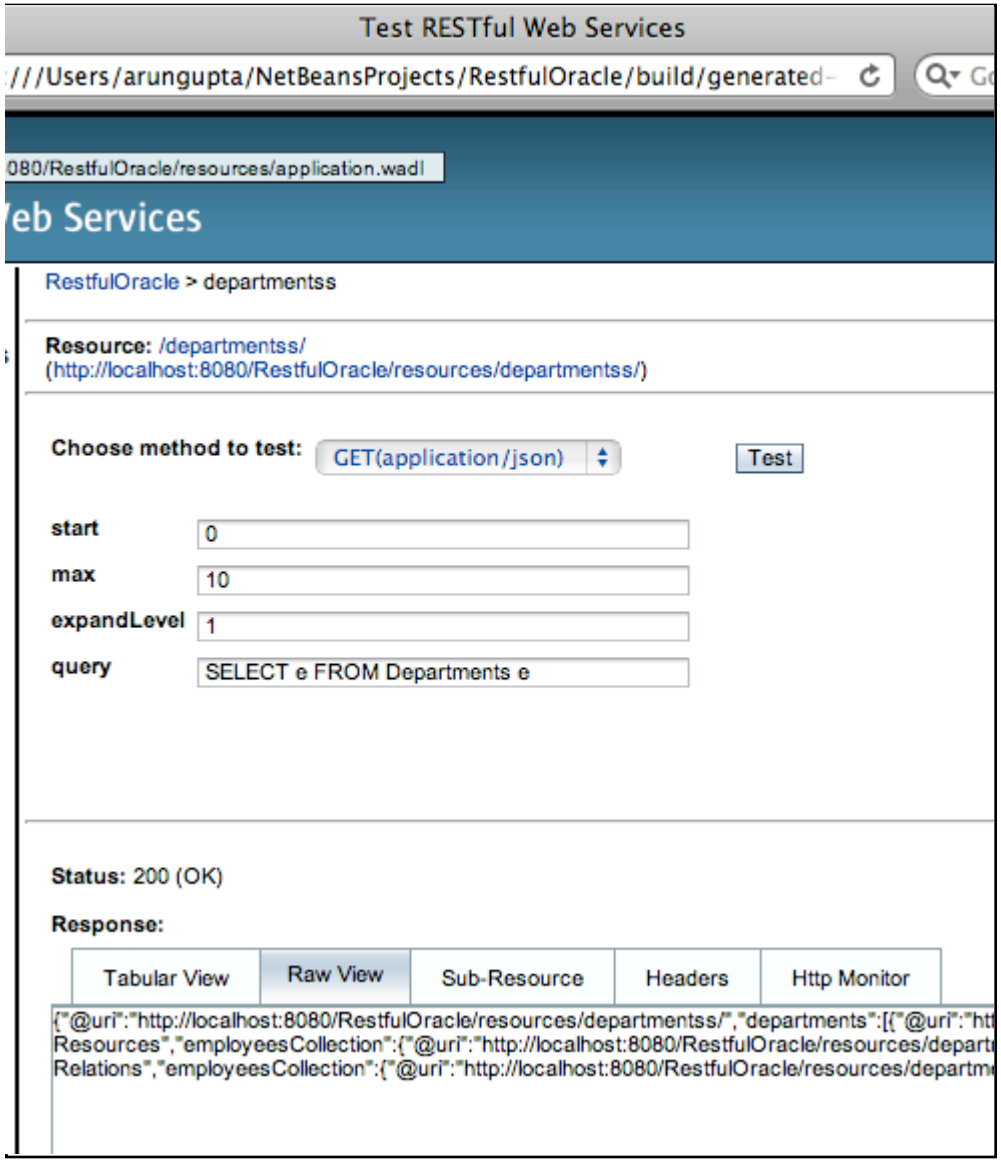
2. Click on "deparmentss" and then on "Test" button to see the output as:


Test RESTful Web Services
file:///Users/arungupta/NetBeansProjects/RestfulOracle/build/generated
WADL : http://localhost:8080/RestfulOracle/resources/application.wadl
Test RESTful Web Services
RestfulOracle
departmentss
countriess
regionss
locationss
employeess
jobss
RestfulOracle > departmentss
Resource: /departmentss/
(http://localhost:8080/RestfulOracle/resources/departmentss/)
Choose method to test: GET(application/json)
Test
start 0
max 10
expandLevel 1
query SELECT e FROM Departments e
Status: 200 (OK)
Response:
Tabular View
Raw View
Sub-Resource
Headers
Http Monitor
/departmentss/ (144)
ID
URI
1 /departmentss/10/
(http://localhost:8080/RestfulOracle/resources/departmentss/10/)
2 /departmentss/10/employeesCollection/
(http://localhost:8080/RestfulOracle/resources/departmentss/10/employee)
3 /departmentss/10/employeesCollection/200/
(http://localhost:8080/RestfulOracle/resources/departmentss/10/employee)
4 /departmentss/10/locationId/
(http://localhost:8080/RestfulOracle/resources/departmentss/10/location)

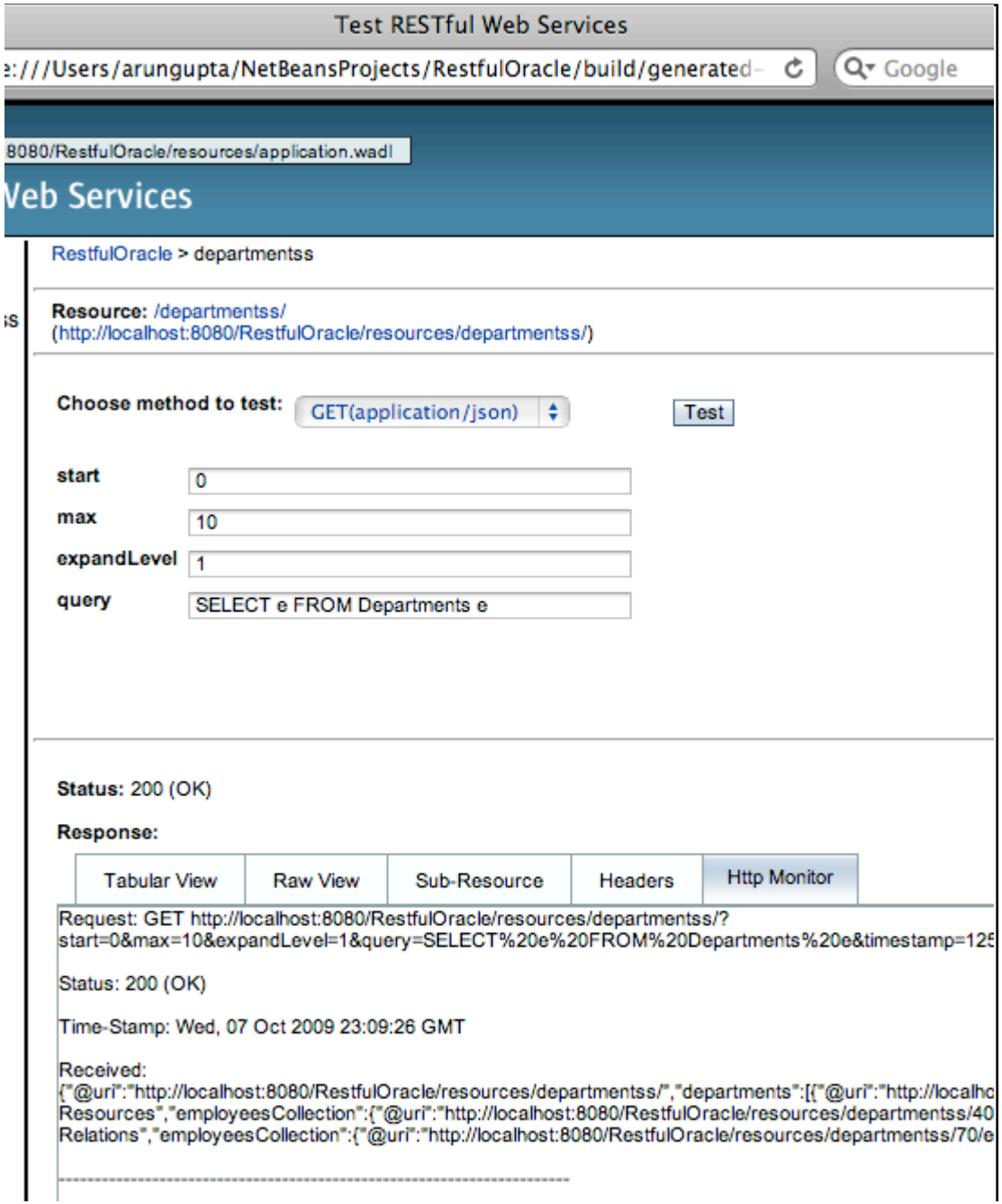
Clicking the "Test" button issues a GET request to "http://localhost:8080/RestfulOracle/resources/departmentss". This uses the generated JAX-RS wrapper classes to talk to the database using JPA entity classes and query the first 10 rows from the "DEPARTMENTS" table. The response is then JSON formatted using JAX-RS wrapper classes and is returned to the requesting page which then displays it nicely formatted in the table. It also shows l-level deep department's relationship to other entities. If the "expandLevel" on the above page is set to "0", then the following output is shown:


Test RESTful Web Services
file:///Users/arungupta/NetBeansProjects/RestfulOracle/build/generated
RestfulOracle
departmentss
countriess
regionss
locationss
employeess
jobss
RestfulOracle > departmentss
Resource: /departmentss/
(http://localhost:8080/RestfulOracle/resources/departmentss/)
Choose method to test: GET(application/json)
Test
start 0
max 10
expandLevel 0
query SELECT e FROM Departments e
Status: 200 (OK)
Response:
Tabular View
Raw View
Sub-Resource
Headers
Http Monitor
/departmentss/ (10)
ID
URI
1 /departmentss/10/ (http://localhost:8080/RestfulOracle/resources/departmentss/10/)
2 /departmentss/20/ (http://localhost:8080/RestfulOracle/resources/departmentss/20/)
3 /departmentss/30/ (http://localhost:8080/RestfulOracle/resources/departmentss/30/)
4 /departmentss/40/ (http://localhost:8080/RestfulOracle/resources/departmentss/40/)
5 /departmentss/50/ (http://localhost:8080/RestfulOracle/resources/departmentss/50/)
6 /departmentss/60/ (http://localhost:8080/RestfulOracle/resources/departmentss/60/)

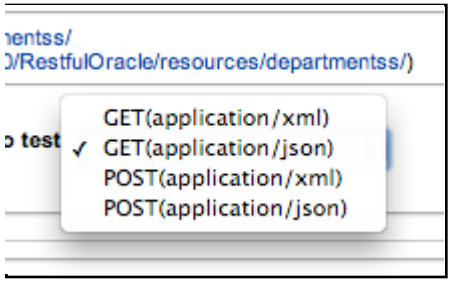
The "Raw View" (JSON data) of the original output looks like:



Notice this is the raw JSON output generated by the JAX-RS wrapper classes. The "Http Monitor" traffic looks like:


Test RESTful Web Services
e:///Users/arungupta/NetBeansProjects/RestfulOracle/build/generated
Google
8080/RestfulOracle/resources/application.wadl
Web Services
RestfulOracle > departmentss
Resource: /departmentss/
(http://localhost:8080/RestfulOracle/resources/departmentss/)
Choose method to test:
GET(application/json)
Test
start
0
max
10
expandLevel
1
query
SELECT e FROM Departments e
Status: 200 (OK)
Response:
Tabular View
Raw View
Sub-Resource
Headers
Http Monitor
Request: GET http://localhost:8080/RestfulOracle/resources/departmentss/?
start=0&max=10&expandLevel=1&query=SELECT%20e%20FROM%20Departments%20e&timestamp=125
Status: 200 (OK)
Time-Stamp: Wed, 07 Oct 2009 23:09:26 GMT
Received:
{"@uri":"http://localhost:8080/RestfulOracle/resources/departmentss/","departments":[{"@uri":"http://localhc
Resources","employeesCollection":{"@uri":"http://localhost:8080/RestfulOracle/resources/departmentss/40
Relations","employeesCollection":{"@uri":"http://localhost:8080/RestfulOracle/resources/departmentss/70/e

The format of data returned can be changed from "application/json" to "application/xml" as shown below:



And even a POST request can be generated.

Do you have the need to expose your Oracle database tables as RESTful entities ?

A complete archive of all the TOTDs is available here.

This and other similar applications will be demonstrated at the upcoming Oracle Open World.

Technorati: totd oracle database glassfish v3 netbeans javaee jax-rs jpa rest

Share and Enjoy:

Related posts:

1. TOTD #107: Connect to Oracle database using NetBeans
2. TOTD #108: Java EE 6 web application (JSF 2.0 + JPA 2.0 + EJB 3.1) using Oracle, NetBeans, and GlassFish
3. RESTful representation of “sakila” using GlassFish and NetBeans IDE
4. TOTD #25: Rails application with PostgreSQL database using NetBeans
5. TOTD #106: How to install Oracle Database 10g on Mac OS X (Intel) ?

Comments (0)

No Comments »

No comments yet.

RSS feed for comments on this post. TrackBack URL

Leave a comment

Logged in as arungupta. Log out »

Submit Comment

- Pages:
  - About
    - speaking-credentials
- Search:

Search

- Total published items: 988
- Categories:
  - general
  - glassfish
    - admin
    - eclipse
    - frameworks
      - django
      - grails
      - javaee
        - javaserverfaces
      - jmaki
      - rails
      - sinatra
      - wicket
    - lotd
    - netbeans
    - screencast
    - webservices
  - personal
  - photography
  - running
    - marathons
  - totd
  - Uncategorized
  - web2.0

- 
  Arun Gupta
  arungupta

  @ILoveNewDelhi It's not "the" national animal, it's the aquatic national animal. "The" national animal is still Tiger.
  4 hours ago

twitter
Join the conversation

- 

- del.icio.us
  - Troubleshooting a MacBook, MacBook Air or MacBook Pro that won't turn on 5 hours ago
  - Developing RIA Web Applications with Oracle ADF 2009/10/08
  - GE2ORGE MAGGESSY: JDeveloper 11g on Mac OS 10.5 and above 2009/10/07
  - All-in-one Rails Server - RightScale Support 2009/10/01
  - JRuby on Rails: Deploying to Oracle Containers for Java EE (OC4J) - Oracle Wiki 2009/10/01
- Archives:
  - October 2009
  - September 2009
  - August 2009

- July 2009
- June 2009
- May 2009
- April 2009
- March 2009
- February 2009
- January 2009
- December 2008
- November 2008
- October 2008
- September 2008
- August 2008
- July 2008
- June 2008
- May 2008
- April 2008
- March 2008
- February 2008
- January 2008
- December 2007
- November 2007
- October 2007
- September 2007
- August 2007
- July 2007
- June 2007
- May 2007
- April 2007
- March 2007
- February 2007
- January 2007
- December 2006
- November 2006
- October 2006

- September 2006
- August 2006
- July 2006
- June 2006
- May 2006
- April 2006
- March 2006
- September 2005
- August 2005

- Meta:
  - Site Admin
  - Log out
  - RSS
  - Comments RSS
  - Valid XHTML
  - XFN
  - WP

Technorati Widget

Technorati authority

2

Search recent posts

Technorati member

» Member profile
» Linking blogs

Top tags

admin eclipse galileo glassfish jax-ws lotd management monitoring netbeans persistence personal rails totd uncategorized webinar wicket clustering frameworks javaee javaserverfaces jersey jpa loadbalancing metro rest running v3 webservices 72

Blog reactions

TOTD #112: Exposing Oracle database tables as RESTful entities using JAX-RS, GlassFish, and NetBeans
Content available at:
in Arun Gupta, Miles to go ...

TOTD #111: Rails Scaffold for a pre-existing table using Oracle and GlassFish
Content available at:
in Arun Gupta, Miles to go ...

TOTD #110: JRuby on Rails application using Oracle on GlassFish
Content available at:
in Arun Gupta, Miles to go ...

Favorites

*Powered by **WordPress***
55608 visits from Sep 11, 2009