JAX-RS - Java API for RESTful Web Services – L1: Trend.Nxt Hands-on Assignments

Estimated Efforts: 4 PDs

For detailed ToC and other Details: https://wipro365.sharepoint.com/sites/ku-practice-4043/CCX-Competency/TREND.NxT/Pages/JAX-RS-L1.aspx

Author: sridhar.savitha@wipro.com

Date: 24th May 2018

TOC:

Topic No	**Topic Name**	**Sub Topics**	**Min No of Assignments to be Done**
1	Introduction to Web Service	What is Web Service, Different type of Web Services, Soap Vs REST, Knowledge of XML/JSON, JAX-WS vs JAX-RS, Frameworks for SOAP(CXF, Axis) and REST(Jersey, Rest Easy)	0
2	JAX-RS Introduction	What is RESTful Web Service, JAX RS Version (1.x, 2.x), JAX-RS Specification (JSR 311, JSR 339), Java Servlet (Desired to skill to learn Rest), Advantage of Restful Service, REST Architectural Style, HTTP Methods, Statelessness, Idempotent Methods	0
3	Install and Setup JAX-RS	Java & Web Server, JAX-RS Libraries, Installation of Jersey in Eclipse, Configuring web.xml	0
4	REST Messages	HTTP Request, HTTP Response, Handling HTTP Response code(200,400,403,501,503 etc.,)	0
5	REST Resources	Resources, Sub-Resource, addressing Resources	0
6	HTTP Methods/Request/Response Annotations	HTTP Methods - @GET, @PUT, @POST, @DELETE and @HEAD, URI, Path Parameters - @PathParam,@FormParam,@Path,@Resource,@Consumes, @Produces, Request/Response - @MatrixParam,@QueryParam,@BeanParam,@Context Response Builder, MediaType TEXT_XML,TEXT_PLAIN,APPLICATION_JSON	4

7	Developing Rest Service API	Developing Rest Services	1
8	Developing Rest Client	Client builder API, Web target, Postman (UI Client)	2
Total Min No of Assignments to be Done			**7**

Topic 6: HTTP Methods/Request/Response Annotations

Assignment 1:

Create a RESTful Web Service to support the below requirements:

a. The Model/Entity class is as follows:

Class : Movie		
movieId	String	E.g.: M001
movieName	String	E.g.: Top Gun
movieActor	String	E.g.: Tom Cruise
movieCollection	Float	E.g.: 350000000.00

b. Store some sample Movie objects in a collection (Map/List) (around 10 movie objects)

b. The Root resource class should provide the below “GET” methods:

- “GET” method that returns all the movie objects in “text/xml” format
- “GET” method that returns all the movie objects in “application/xml” format
- “GET” method that returns all the movie objects in “application/json” format
- “GET” method that takes a “movieId” as the argument and returns only the Movie details for that MovieId in “application/xml” format.

c. Hint: Annotate the Model class with relevant JAXB annotations so that the above GET methods can return the objects directly and the objects will be automatically marshalled to XML/JSON by JAXB.

(You could use any JAX-RS RI like Jersey or RESTEasy for this program)

Assignment 2:

Create a RESTful Web Service to support the below requirements:

a. The Model/Entity class is as follows:

Class : Student		
studentId	String	E.g.: S001
studentName	String	E.g.: Steve Martin
studentClass	String	E.g.: C1
StudentTotalMarks	Float	E.g.: 90

b. Student details can be stored in a local collection or in a table in a database (choice is yours)

c. The RESTful Web Service resource class should support the below operations:

1. "GET" method that returns all the existing student details in XML/JSON format
2. "GET" method that returns one existing student details based on the studentId and the studentId is passed as a path parameter
3. "GET" method that returns all student details who have scored greater than a particular mark. Studentmarks is passed as a path parameter.
3. "POST" method that allows us to insert/add a new student entry. Student details are passed through a form.
4. "PUT" method that allows us to update/edit an existing student entry and allows to update/edit details.
5. "DELETE" method that allows us to delete an existing student entry.

d. (You could use any JAX-RS RI like Jersey or RESTEasy for this program)

Assignment 3:

Rewrite the Solution of Assignment 2 to incorporation the below changes/additions:

a. Include a "GET" method that returns all the students belonging to a particular class. StudentClass is passed as a path parameter. We should also support

"GET" method to receive parameters as both query param and matrix param as well.

b. The "GET" method that returns one existing student details based on the studentId and the "GET" method that returns all student details who have scored greater than a particular mark should also support parameters to be passed to them as query param and matrix param as well.

Assignment 4:

Create Restful Web service that will process 'credit card number' validations. To keep things simple, we'll say that any credit card ending in an even number is considered valid and any card with an odd number is invalid. (This obviously isn't true in real life; we're just simplifying the problem for demonstration purposes.) In order to process the card, we'll need to know the card number. We'll send back a true if it's valid—false if it's invalid.

Topic 7: Developing Rest Service API & Topic 8: Developing Rest Client

Assignment 1:

Create a RESTful Web Service to support "Employee" management.

The Model/Entity class is as follows:

Class : Employee		
empId	String	E.g.: EM001
empFirstName	String	E.g.: Peter
empLastName	String	E.g.: Parker
empLocation	String	E.g.: London
empBand	String	E.g.: B3

a. Employee details needs to be stored in a table in a database (You can either use JDBC API or Hibernate ORM for working with Database)

b. The RESTful Resource class should provide facilities to :

- **a.** Retrieve all employee details ("GET" method) in XML/JSON format
- **b.** Retrieve all employee details based on a particular "empLocation" ("GET" method which receives "empLocation" as a path parameter) The returned output should be in XML/JSON format
- **c.** Retrieve all employee details based on a particular "empBand" ("GET" method which receives "empBand" as a path parameter). The returned output should be XML/JSON format
- **d.** Add a new Employee entry ("POST" method). The data will be passed as XML format.
- **e.** Edit an existing Employee entry ("PUT" method). empId will be passed as a path parameter and the new details will be passed in XML format.

f. Deleting an existing Employee is NOT SUPPORTED at present.

Assignment 2:

a. Create a Jersey Client program to programmatically consume the RESTful Web Service that you created in Assignment 1. You have to test all the methods available. Try to make a delete request also and see what error codes/status codes you would get.

Assignment 3:

a. Use ‘postman’ (Rest UI client) to test/consume that RESTful Web Service that you created in Assignment 1. Analyze the outputs and errors that you might receive.

Note: Wherever applicable, capture the screen shots of all the outputs and use it during your assignment submission.