**TREND NXT JAX RS – JAVA API FOR RESTFUL WEB SERVICES**

**TOP GEAR ASSIGNMENT REPORT SUBMISSION**

* **Submitted by**

Tara Venkata Sai Rahul Muktineni

(Employee Id: 20236520)

(Employee Email: tara.muktineni@wipro.com)

Below are the tools and technologies that are used during the development of this assignment.

**Programming Language:** Java (jdk-17.0.2)

**IDE:** Eclipse IDE for Java Developers 2021-12 (4.22.0)

**Framework:** Spring Boot 2.5

**JAX RS Implementation**: Jersey

**Rest UI Client:** Postman

**Database**: MySQL

**TOPIC 6: HTTP Methods/Request/Response Annotations**

**Assignment 1:**

“GET” method that returns all the movie objects in “text/xml” format.

Graphical user interface, text, application, email

Description automatically generated

“GET” method that returns all the movie objects in “application/xml” format.

Graphical user interface, text, application, email

Description automatically generated

“GET” method that returns all the movie objects in “application/json” format.

Graphical user interface, text, application, email

Description automatically generated

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

Graphical user interface, text, application, email

Description automatically generated

**Assignment 2:**

“GET” method that returns all the existing student details in “text/xml” format.

Graphical user interface, text, application, email

Description automatically generated

“GET” method that returns all the existing student details in “application/xml” format.

Graphical user interface, text, application, email

Description automatically generated

“GET” method that returns all the existing student details in “application/json” format.

Graphical user interface, text, application, email

Description automatically generated

“GET” method that returns one existing student details based on the studentId in “application/json” format and the studentId is passed as a path parameter.

Graphical user interface, application, email, Teams

Description automatically generated

“GET” method that returns all student details who have scored greater than a particular mark in “application/json” format. Studentmarks is passed as a path parameter.

Graphical user interface, text, application, email

Description automatically generated

“POST” method that allows us to insert/add a new student entry. Student details are passed through a form in “application/json” format.

Graphical user interface, application, Teams

Description automatically generated

“PUT” method that allows us to update/edit an existing student entry and allows to update/edit details in “application/json” format.

Graphical user interface, text, application, email, Teams

Description automatically generated

“DELETE” method that allows us to delete an existing student entry in “application/json” format.

Graphical user interface, text, application, email

Description automatically generated

**Assignment 3:**

“GET” method that returns all the students belonging to a particular class in “application/json” format. StudentClass is passed as a path parameter.

Graphical user interface, application, Teams

Description automatically generated

“GET” method that returns all the students belonging to a particular class in “application/json” format. StudentClass is passed as a query parameter.

A screenshot of a computer

Description automatically generated

“GET” method that returns all the students belonging to a particular class in “application/json” format. Method supports StudentClass being passed as a path parameter, query parameter and matrix parameter.

A screenshot of a computer

Description automatically generated

“GET” method that returns one existing student details based on the studentId in “application/json” format and the studentId is passed as a query parameter.

A screenshot of a computer

Description automatically generated

“GET” method that returns one existing student details based on the studentId in “application/json” format. The method supports studentId being passed as a path parameter, query parameter and matrix parameter.

A screenshot of a computer

Description automatically generated

“GET” method that returns all student details who have scored greater than a particular mark in “application/json” format. StudentMarks is passed as a query parameter.

Graphical user interface, text, application, email

Description automatically generated

“GET” method that returns all student details who have scored greater than a particular mark in “application/json” format. Method supports StudentMarks being passed as a path parameter, query parameter and matrix parameter.

Graphical user interface, text, application

Description automatically generated

**Assignment 4:**

“POST” method that validates the given credit card number and returns TRUE if the credit card number is valid (ending with an even number) in “application/json” format.

Graphical user interface, text, application, email

Description automatically generated

“POST” method that validates the given credit card number and returns FALSE if the credit card number is invalid (ending with an odd number) in “application/json” format.

Graphical user interface, text, application, email

Description automatically generated

**TOPIC 7: Developing Rest Service API**

**&**

**TOPIC 8: Developing Rest Client**

Employee Details are stored in MySQL database and can be seen as shown in the below image.

Graphical user interface

Description automatically generated

**Figure:** MySQL Workbench

As part of the assignment, I have used **Hibernate** as the ORM implementation in order to work with the database.

**Assignment 1:**

“GET” method that returns all the employee details in “text/xml” format.

Graphical user interface, text, application, email

Description automatically generated

This API returns all the employee details in text/xml format as requested by the client using Accept header. The response code is 200 and the response status is OK indicating that it is a successful request, and the response is displayed in the response body in text/xml format.

“GET” method that returns all the employee details in “application/xml” format.

Graphical user interface, text, application, email

Description automatically generated

This API returns all the employee details in “application/xml” format as requested by the client using Accept header. The response code is 200 and the response status is OK indicating that it is a successful request, and the response is displayed in the response body in application/xml format.

“GET” method that returns all the employee details in “application/json” format.

Graphical user interface, application, email

Description automatically generated

This API returns all the employee details in application/json format as requested by the client using Accept header. The response code is 200 and the response status is OK indicating that it is a successful request, and the response is displayed in the response body in application/json format.

“GET” method that returns all the employee details based on a particular empLocation in “text/xml” format. empLocation is passed as a path parameter.

Graphical user interface, text, application, email

Description automatically generated

This API returns all the employee details who are working in the given employee location in text/xml format as requested by the client using Accept header. The response code is 200 and the response status is OK indicating that it is a successful request, and the response is displayed in the response body in text/xml format.

“GET” method that returns all the employee details based on a particular empLocation in “application/xml” format. empLocation is passed as a path parameter.

Graphical user interface, text, application, email

Description automatically generated

This API returns all the employee details who are working in the given employee location in application/xml format as requested by the client using Accept header. The response code is 200 and the response status is OK indicating that it is a successful request, and the response is displayed in the response body in application/xml format.

“GET” method that returns all the employee details based on a particular empLocation in “application/json” format. empLocation is passed as a path parameter.

Graphical user interface, text, application, email

Description automatically generated

This API returns all the employee details who are working in the given employee location in application/json format as requested by the client using Accept header. The response code is 200 and the response status is OK indicating that it is a successful request, and the response is displayed in the response body in application/json format.

“GET” method that returns all the employee details based on a particular empBand in “text/xml” format. empBand is passed as a path parameter.

Graphical user interface, text, application, email

Description automatically generated

This API returns all the employee details who belong to the given employee band in text/xml format as requested by the client using Accept header. The response code is 200 and the response status is OK indicating that it is a successful request, and the response is displayed in the response body in text/xml format.

“GET” method that returns all the employee details based on a particular empBand in “application/xml” format. empBand is passed as a path parameter.

Graphical user interface, text, application, email

Description automatically generated

This API returns all the employee details who belong to the given employee band in application/xml format as requested by the client using Accept header. The response code is 200 and the response status is OK indicating that it is a successful request, and the response is displayed in the response body in application/xml format.

“GET” method that returns all the employee details based on a particular empBand in “application/json” format. empBand is passed as a path parameter.

Graphical user interface, text, application, email

Description automatically generated

This API returns all the employee details who belong to the given employee band in application/json format as requested by the client using Accept header. The response code is 200 and the response status is OK indicating that it is a successful request, and the response is displayed in the response body in application/json format.

“POST” method that adds an Employee entry into the database. The data is passed in an XML format in the request body.

Graphical user interface, text, application, email

Description automatically generated

This API adds the employee details provided by the client in application/xml format in the request body into the database. The response code is 200 and the response status is OK indicating that it is a successful request, and the response which is the employee entry added by the client is displayed in the response body in application/json format.

“UPDATE” method that updates an Employee entry in the database. empId will be passed as a path parameter and the new Employee details will be passed in an XML format in the request body.

Graphical user interface, text, application, email

Description automatically generated

This API updates the employee details provided by the client in application/xml format in the request body into the database. The employee id which is used to update the employee entry is provided in the request URL as a path parameter. The response code is 200 and the response status is OK indicating that it is a successful request, and the response which is the employee entry updated by the client is displayed in the response body in application/json format.

“GET” method that returns the employee details based on a particular empId in “application/json” format. empId is passed as a path parameter.

Graphical user interface, text, application, email

Description automatically generated

This API returns the employee details of the employee whose employee id is provided in the request URL as a path parameter. The response code is 200 and the response status is OK indicating that it is a successful request, and the response is displayed in the response body in application/json format.

“DELETE” method that deletes the Employee record from the database based on the empId. The empId is passed as a path parameter.

**Note:** However, this feature is not implemented in the code and the request is executed in order to view the error code and status.

Error Code: 405

Error Status: Method Not Allowed

Graphical user interface, text, application, email

Description automatically generated

This API is NOT SUPPORTED by the application and hence it does not perform any CRUD operation on the Employee table in the database. As the method is not available to perform any action, the response code is 405 and the response status is METHOD NOT ALLOWED indicating that it is an unsuccessful request, and the error response is displayed in the response body in application/json format.

**Assignment 2:**

A client application is developed using Jersey framework. The application consists of a single class named “RestClient.java” which programmatically consumes all the available methods that are created as part of Assignment 1 and displays the responses in the console as shown in the below figure. The class file is part of the “assignment” application (residing in com.restful.assignment.service package) that is uploaded as part of the submission.

Text

Description automatically generated

**Figure:** Application responses to the requests sent by the RestClient client

The entire response is uploaded as a file named “RestClient-responses” as part of the submission.

Below is the image displaying the data available in the MySQL database before the RestClient client consumes all the API’s provided by the application. There are 7 records in the Employee table.

Graphical user interface

Description automatically generated with medium confidence

Below is the image displaying the data available in the MySQL database after the RestClient client consumes all the API’s provided by the application. As part of the API’s, there’s an “Add” employee and “Update” employee API’s that are available. The RestClient consumes the “Add” employee API and creates a new employee record with empId “EM010” in the employee table. The RestClient consumes the “Update” employee API and updates the employee record with empId “EM010”. The EM010 record below is the cumulative output of Add and Update employee API’s being executed sequentially.

Table

Description automatically generated with medium confidence

**Assignment 3:**

I have used Postman as the Rest UI client to test and consume all the RESTful web services created as part of this assignment. The Postman requests and responses from all the web services are attached as screenshots in this document.

I have analyzed the outputs and errors for all the web services as part of Assignment 1 as shown above.

The Postman collection suite is uploaded as part of the submission.

**NOTE:**

As part of Topic 6 Assignment 3, I have created a GET method to accept the StudentClass as a path parameter, query parameter and matrix parameter. I have also created separate GET methods that takes StudentClass as a query parameter and StudentClass as a matrix parameter. The reason I have created separate methods and a combined method is because I was confused when I read “We should also support GET method to receive parameters as both query param and matrix param as well”. Hence, I created separate GET methods for each type of parameter and a single GET method that takes StudentClass as path parameter, query parameter and matrix parameter.

Similar implementation is done for GET methods for studentId as a parameter and studentMarks as a parameter.