

CEGEP VANIER COLLEGE

CENTRE FOR CONTINUING EDUCATION

Web Services

420-941-VA

Teacher: Samir Chebbine

Lab 4

Oct 21, 2024

Lab 4: Jersey Client Consuming REST Web Services

Complete all these following programs in class. All *missing coding statements* are presented during class time and in Presentation 4.

Create and Submit a Word file **Lab4WebServicesYourName.doc** which contains Answers of theory questions if any and output screenshots for every Java EE Project. Submit the Java projects too and submit the whole Lab 4 as compressed zip file

1. Creating Client Maven Project

- Create a Maven project called **ClientHelloRESTProject**.
- Add **Maven Project dependencies** as stated in my Presentation 4 **pom.xml**. Create new package called **clientHelloREST** as shown in Figure 1.
- Run Jersey client project **ClientHelloRESTProject** as Java application consuming REST Web Services **WebHelloRESTProject** of Lab 3.

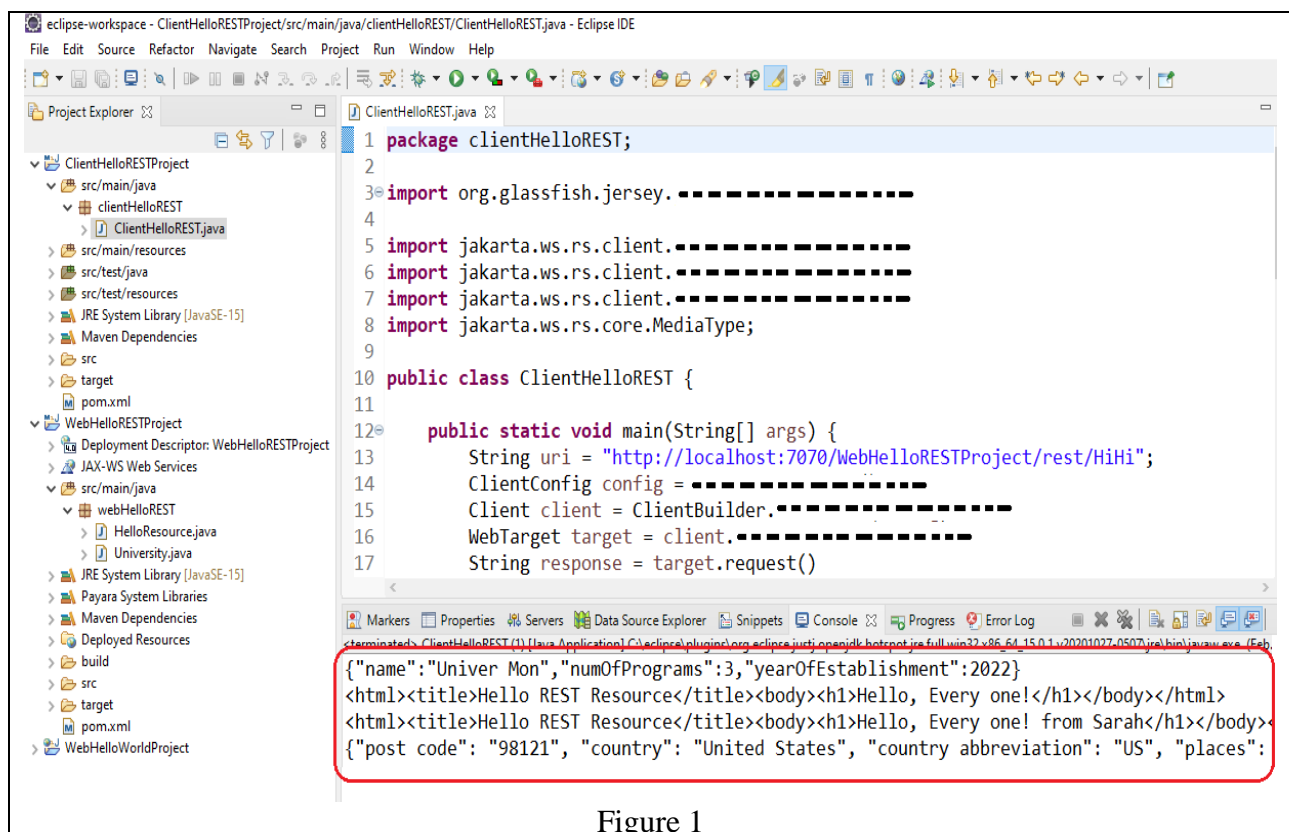


Figure 1

- Add appropriate statements to consume **one of the public REST Services** you provided in Lab 2. Provide screenshots in word document showing the output as in Figure 1.

2. Maven Project: ClientMathOperationsRESTProject

- Create a new **Maven project** called **ClientMathOperationsRESTProject**.
- Add **Maven Project dependencies** in **pom.xml**. Create new package called **clientMathOperationsREST**.
- Create a console Java Client **ClientMathOperationsREST.java** using Jersey Client Framework to consume the REST Web Service **ClientMathOperationsRESTProject** developed in **Lab 3**. You need to deploy **ClientMathOperationsRESTProject** within GalssFish Server.
- Add appropriate statements in Java Client to invoke the REST resource using **query string parameters** **x**, **y**, **z** that calls implemented methods **calculateSum()/calculatePrd()** in path URL mapping ("**MathOp**"), display media type **HTML** as shown hereafter.
- Add appropriate statements in Java Client to invoke the REST resource using path URL mapping ("**/displayListZYZ...**"), include appropriate media type and display the output as shown hereafter.
- Add appropriate statements in Java Client to invoke the REST resource using path URL mapping ("**/OpHashMap ...**") using path parameter **x** as search parameter to access REST resource searching into Hash Map, include appropriate media type and display the output as shown hereafter.

```
<terminated> ClientMathOperationsREST [Java Application] C:\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_15.0.1.v20201027-0507\jre\bin\javaw.exe (Feb. 20, 2024, 2:29:01 p.m.)
Please Enter x: 1
Please Enter y: 2
Please Enter z: 3
<html><title>Hello Math Op Resource</title><body><h1>Calculate (x+2*y+3*z) Output is: 14.0</h1>
<br/><br/><h1>Calculate (x*2*y*3*z) Output is: 36.0</h1></body></html>

<html><title>Hello REST Resource</title><body><h3>listXYZ Array List:<br/> Array List Element: 0:
MathOp [x=1.0, y=2.0, z=3.0]<br/> Array List Element: 1:MathOp [x=4.0, y=5.0, z=6.0]
<br/> Array List Element: 2:MathOp [x=7.0, y=8.0, z=9.0]<br/></h3></body></html>

Please Enter x for search into HashMap Collection: 7
{"x":7.0,"y":8.0,"z":9.0}
```

3. Creating Graphical Client Maven Project

- Create a Maven project called **ClientGraphicHelloRESTProject**
- Add **Maven Project dependencies** in **pom.xml**. Create new package called **clientGraphicHelloREST** as shown hereafter.
- Run Jersey client project **ClientGraphicHelloRESTProject** as Java application consuming REST Web Services **WebHelloRESTProject** of Lab 3.

4. Maven Dynamic Web Project: WebFacultyRESTProject

- Create a new Dynamic Web project called **WebFacultyRESTProject** and convert it into **Maven Project**.
- Add **Maven Project dependencies** in **pom.xml**. Create new package called **webFacultyREST**.
- Deploy **WebFacultyRESTProject** within GalssFish Server to be executed by **Servlet class ServletContainer** specified in **web.xml**
- You need to develop a **Java class** called **Faculty** (see **Block3**), which includes the following members:
 - The private data members: `f_Id` (Integer), `f_Lname` (String), `f_Fname` (String), `f_zipcodeBirth` (String), `f_Salary` (double), `f_BonusRate` (double).
 - Add a method (`doCalc_Bonus()`) that calculates, and returns faculty bonus (faculty bonus = (`f_Salary` * `f_BonusRate` / 100))
- Create a new REST Resource class **WebFacultyResource.java**
 - Add a path URL mapping ("WebFaculty") to access REST resource using appropriate Java REST annotation.
 - Add a method **displayHTMLFacultyInfo()** that returns a HTML media type using appropriate Java REST annotations.
 - Add appropriate statements in **displayHTMLFacultyInfo()** to instantiate a data structure **HashMap** referenced by (`facultyHashMap`) where hash map key represents `f_id` and value hash map of `Faculty` class type. Set every component of hash map to the following values read from text file **Faculty.in**:


```
f_id=101,f_Lname=Robertson,f_Fname=Myra,f_zipcodeBirth=98121,f_Salary=60000.00,f_BonusRate=2.50
f_id=212,f_Lname=Smith,f_Fname=Neal,f_zipcodeBirth=85001,f_Salary=40000.00,f_BonusRate=3.00
f_id=315,f_Lname=Arlec,f_Fname=Lisa,f_zipcodeBirth=71601,f_Salary=55000.00,f_BonusRate=1.50
f_id=857,f_Lname=Fillipo,f_Fname=Paul,f_zipcodeBirth=90001,f_Salary=30000.00,f_BonusRate=5.00
f_id=370,f_Lname=Denkan,f_Fname=Anais,f_zipcodeBirth=15001,f_Salary=95000.00,f_BonusRate=1.50
```
 - Skip through Hash Map collection (`facultyHashMap`) and display its unsorted components and sorted components with respect to bonus value into web table respectively as shown hereafter (see **Block3 for sorting**). Check output using Postman.

5. You need to **consume public REST web service** <http://api.zippopotam.us/us/zipcode> when skipping through Hash Map collection (facultyHashMap) to display JSON output of each f_zipcodeBirth in faculty zip code column (as shown hereafter). Save your own screenshot.

← → ↻ localhost:7070/WebFacultyRESTProject/rest/WebFaculty ☆ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿

Consuming public REST Service for each f_zipcodeBirth

Print Faculty Records collection

Faculty ID	Faculty LName	Faculty FName	Faculty ZipCode	Faculty Salary	Faculty Bonus Rate	Faculty Total Bonus
370	Denkan	Anais	{"post code": "15001", "country": "United States", "country abbreviation": "US", "places": [{"place name": "Aliquippa", "longitude": "-80.3197", "state": "Pennsylvania", "state abbreviation": "PA", "latitude": "40.5921"}]}	95000.0	1.5%	1425.00\$
212	Smith	Neal	{"post code": "85001", "country": "United States", "country abbreviation": "US", "places": [{"place name": "Phoenix", "longitude": "-112.3518", "state": "Arizona", "state abbreviation": "AZ", "latitude": "33.704"}]}	40000.0	3.0%	1200.00\$
101	Robertson	Myra	{"post code": "98121", "country": "United States", "country abbreviation": "US", "places": [{"place name": "Seattle", "longitude": "-122.3447", "state": "Washington", "state abbreviation": "WA", "latitude": "47.6151"}]}	60000.0	2.5%	1500.00\$
857	Fillipo	Paul	{"post code": "90001", "country": "United States", "country abbreviation": "US", "places": [{"place name": "Los Angeles", "longitude": "-118.2479", "state": "California", "state abbreviation": "CA", "latitude": "33.9731"}]}	30000.0	5.0%	1500.00\$
315	Arlec	Lisa	{"post code": "71601", "country": "United States", "country abbreviation": "US", "places": [{"place name": "Pine Bluff", "longitude": "-91.9859", "state": "Arkansas", "state abbreviation": "AR", "latitude": "34.209"}]}	55000.0	1.5%	825.00\$

The Total Faculty Bonus is: 6450.00\$

Faculty Hash Map Info Sorted (Sorted by Value Bonus)

Faculty ID	Faculty LName	Faculty FName	Faculty ZipCode	Faculty Salary	Faculty Bonus Rate	Faculty Total Bonus
315	Arlec	Lisa	71601	55000.0	1.5%	825.00\$
212	Smith	Neal	85001	40000.0	3.0%	1200.00\$
370	Denkan	Anais	15001	95000.0	1.5%	1425.00\$
101	Robertson	Myra	98121	60000.0	2.5%	1500.00\$
857	Fillipo	Paul	90001	30000.0	5.0%	1500.00\$

5. Maven Project: ClientCarRESTProject

- Create a new **Maven project** called **ClientCarRESTProject**.
- Create a console Java Client **ClientCarREST.java** using Jersey Client Framework to consume the REST Web Service **WebCarRESTProject** developed in **Lab 3**.
- Add appropriate statements in Java Client to invoke the REST resource using path URL mapping ("WebCar"), display media type TEXT as shown hereafter.
- Add appropriate statements in Java Client to invoke the REST resource using path URL mapping ("/searchCar..."), include appropriate media type and display the output as shown hereafter.

```
Car TEXT Output:
car_vin: M198754, car_desc: Honda, car_price: 40000.0, Car Price with Discount: 36000.00$
car_vin: K1245, car_desc: Ford, car_price: 35000.0, Car Price with Discount: 31500.00$
car_vin: M98524M4, car_desc: Hyundai, car_price: 25000.0, Car Price with Discount: 22500.00$
car_vin: S741582, car_desc: Nissan, car_price: 30000.0, Car Price with Discount: 27000.00$

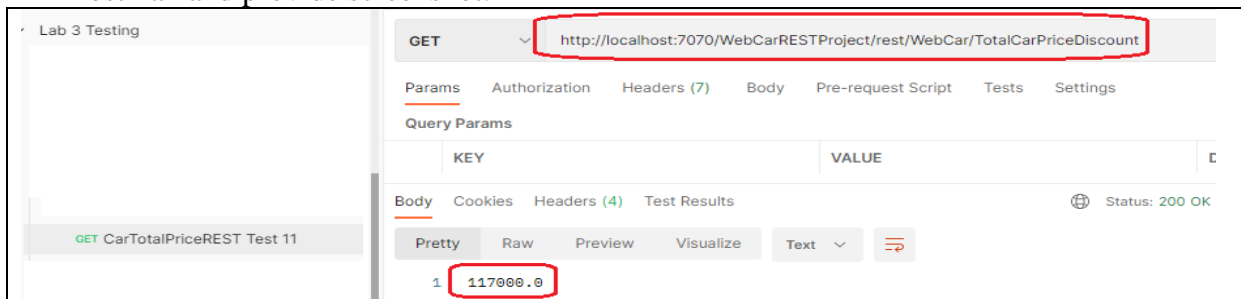
The Total Car Price after Discount is: 117000.00$

Car List JSON Output:
[{"desc": "Honda", "price": 40000.0, "vin": "M198754"}, {"desc": "Ford", "price": 35000.0, "vin": "K1245"}, {"desc": "Hyundai", "price": 25000.0, "vin": "M98524M4"}, {"desc": "Nissan", "price": 30000.0, "vin": "S741582"}]
[{"desc": "Honda", "price": 40000.0, "vin": "M198754"}, {"desc": "Ford", "price": 35000.0, "vin": "K1245"}, {"desc": "Hyundai", "price": 25000.0, "vin": "M98524M4"}, {"desc": "Nissan", "price": 30000.0, "vin": "S741582"}]

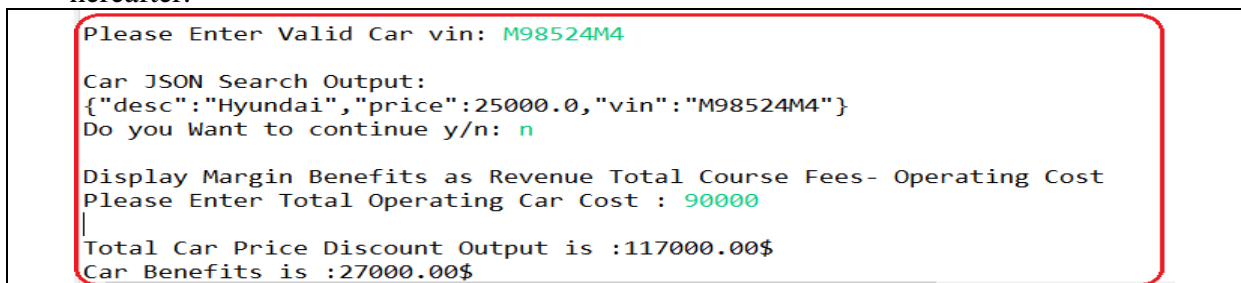
Please Enter Valid Car vin: M98524M4

Car JSON Search Output:
{"desc": "Hyundai", "price": 25000.0, "vin": "M98524M4"}
Do you Want to continue y/n: n
```

- e) In **WebCarRESTProject**: Add a new path URL mapping ("/TotalCarPriceDiscount") to access REST resource.
- f) Add a method **double calculateTotalCarPriceDiscount()** that returns total car prices after discount when skipping through car hash map collection and will be fired upon using URL mapping ("/TotalCarPriceDiscount") as shown hereafter. Check the output using Postman and provide screenshot.

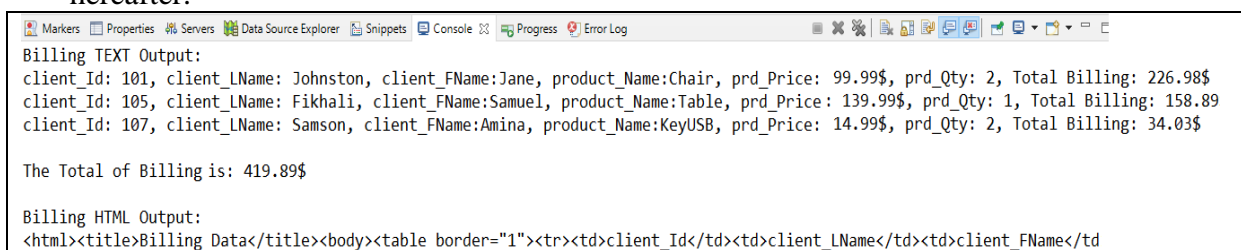


- g) We would like to calculate **Total Car Benefit** within **ClientCarRESTProject** as (Total Operating Car Cost – Total Car Price Discount). In this use case, Total Operating Car Cost is available only in Client Application **ClientCarRESTProject** as user input while Total Car Price Discount will be extracted from **consuming REST** Web Service Resource ("/TotalCarPriceDiscount"). Add appropriate statements in Java Client to invoke the REST resource using path URL mapping ("/TotalCarPriceDiscount") and calculate car benefit, accordingly, include appropriate media type and display the output as shown hereafter.

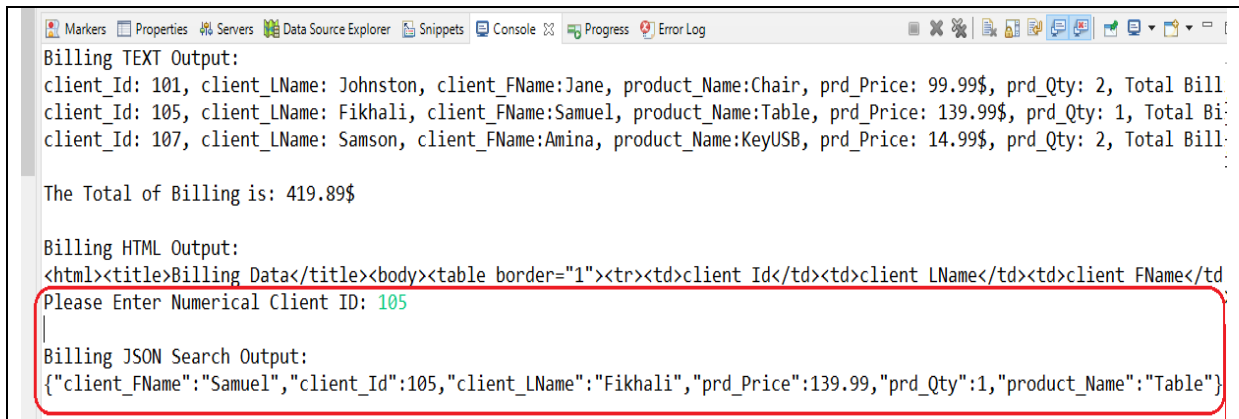


6. Maven Project: ClientBillingRESTProject

- a) Create a new **Maven project** called **ClientBillingRESTProject**.
- b) Add **Maven Project dependencies** in **pom.xml**. Create new package called **clientBillingREST**.
- c) Create a Java Client **ClientBillingREST.java** using Jersey Client 3.1.2 Framework to consume the REST Web Service **WebBillingRESTProject** developed in **Lab 3**. You need to deploy **WebBillingRESTProject** within GalssFish Server.
- d) Add appropriate statements in Java Client to invoke the REST resource using path URL mapping ("WebBilling"), include both media type plain TEXT and HTML as shown hereafter.



- e) Add appropriate statements in Java Client to invoke the REST resource using path URL mapping ("/searchBilling..."), include appropriate media type and display the output as shown hereafter.



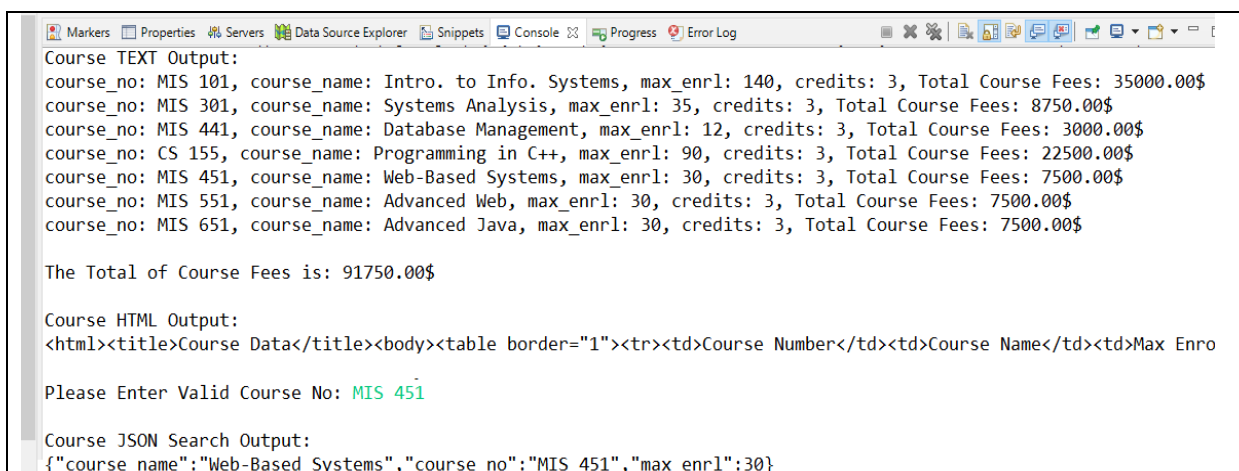
```
Markers Properties Servers Data Source Explorer Snippets Console Progress Error Log
Billing TEXT Output:
client_Id: 101, client_LName: Johnston, client_FName:Jane, product_Name:Chair, prd_Price: 99.99$, prd_Qty: 2, Total Bill
client_Id: 105, client_LName: Fikhali, client_FName:Samuel, product_Name:Table, prd_Price: 139.99$, prd_Qty: 1, Total Bi
client_Id: 107, client_LName: Samson, client_FName:Amina, product_Name:KeyUSB, prd_Price: 14.99$, prd_Qty: 2, Total Bill

The Total of Billing is: 419.89$

Billing HTML Output:
<html><title>Billing Data</title><body><table border="1"><tr><td>client_Id</td><td>client_LName</td><td>client_FName</td>
Please Enter Numerical Client ID: 105
Billing JSON Search Output:
{"client_FName":"Samuel","client_Id":105,"client_LName":"Fikhali","prd_Price":139.99,"prd_Qty":1,"product_Name":"Table"}
```

7. Maven Project: ClientCourseRESTProject

- a) Create a new **Maven project** called **ClientCourseRESTProject**.
- b) Add **Maven Project dependencies** in **pom.xml**. Create new package called **clientCourseREST**.
- c) Create a console Java Client **ClientCourseREST.java** using Jersey Client Framework to consume the REST Web Service **WebCourseRESTProject** developed in **Lab 3**. You need to deploy **WebCourseRESTProject** within GalssFish Server.
- d) Add appropriate statements in Java Client to invoke the REST resource using path URL mapping ("WebCourse"), display media type HTML as shown hereafter.
- e) Add appropriate statements in Java Client to invoke the REST resource using path URL mapping ("/searchCourse..."), include appropriate media type and display the output as shown hereafter.



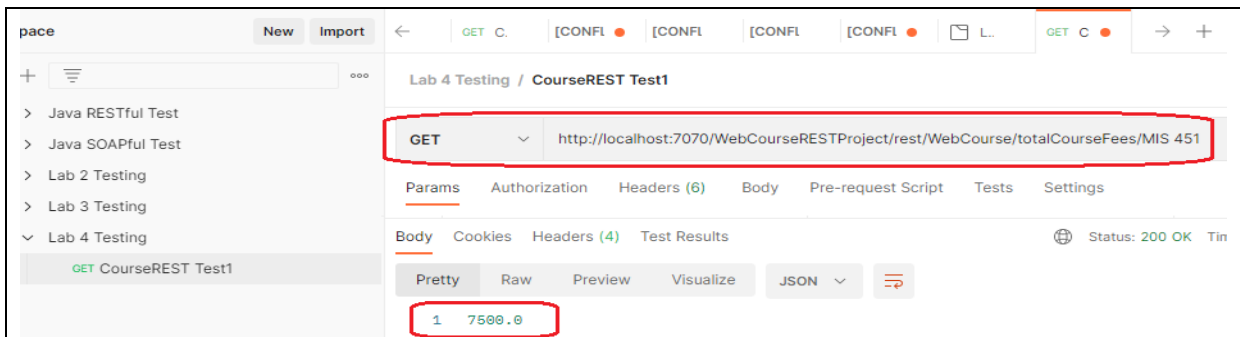
```
Markers Properties Servers Data Source Explorer Snippets Console Progress Error Log
Course TEXT Output:
course_no: MIS 101, course_name: Intro. to Info. Systems, max_enrl: 140, credits: 3, Total Course Fees: 35000.00$
course_no: MIS 301, course_name: Systems Analysis, max_enrl: 35, credits: 3, Total Course Fees: 8750.00$
course_no: MIS 441, course_name: Database Management, max_enrl: 12, credits: 3, Total Course Fees: 3000.00$
course_no: CS 155, course_name: Programming in C++, max_enrl: 90, credits: 3, Total Course Fees: 22500.00$
course_no: MIS 451, course_name: Web-Based Systems, max_enrl: 30, credits: 3, Total Course Fees: 7500.00$
course_no: MIS 551, course_name: Advanced Web, max_enrl: 30, credits: 3, Total Course Fees: 7500.00$
course_no: MIS 651, course_name: Advanced Java, max_enrl: 30, credits: 3, Total Course Fees: 7500.00$

The Total of Course Fees is: 91750.00$

Course HTML Output:
<html><title>Course Data</title><body><table border="1"><tr><td>Course Number</td><td>Course Name</td><td>Max Enro
Please Enter Valid Course No: MIS 451
Course JSON Search Output:
{"course_name":"Web-Based Systems","course_no":"MIS 451","max_enrl":30}
```

- f) In **WebCourseRESTProject**: Add a new path URL mapping ("/totalCourseFees...") with path parameter **course_no** as search String parameter to access REST resource.

- g) Add a method **double calculateTotalCourseFees(String course_no)** that returns total fees as double data type of the searched **course_no** within Array List search and will be fired upon using URL mapping ("/totalCourseFees...") as shown hereafter. Check the output using Postman and provide screenshot.



- h) We would like to calculate **Course Benefits** within **ClientCourseRESTProject** as (Total Course Fees – Total Operating Course Cost). In this use case, Total Operating Course Cost are available only in Client Application **ClientCourseRESTProject** as user input while Total Course Fees will be extracted from **consuming REST** Web Service Resource ("/totalCourseFees..."). Add appropriate statements in Java Client to invoke the REST resource using path URL mapping ("/totalCourseFees...") and calculate course benefits, accordingly, include appropriate media type and display the output as shown hereafter.

