Taco Loco Backend Challenge

By Ayesha Khader

Purpose

A backend service has been created in order to assist truck drivers and to help them manage customer addresses for their taco deliveries. The Delivery Service enables a truck driver to look up the entire deliveries database, add a new customer's name and address, update a delivery and delete a delivery. The database includes the delivery IDs, the customer names and the customer addresses.

Coding Languages and Techniques

Primarily, the code has been written in Java and has been incorporated with JSON. Moreover, another programming language that has been extensively used is XML. Furthermore, the techniques mainly used in writing the code are: Jersey RESTful API and JAXB. JAXB has been used to convert Java objects to XML and vice versa. Similarly, Jersey uses MOXy to convert Java objects to JSON and vice versa.

Prerequisites

The following prerequisites are required to run the program:

- Eclipse IDE for Enterprise Java Developers
- Advanced REST Client Add-on for Chrome or Firefox

Functioning of The Taco Loco Delivery Service

In order to run the as application, clone the GitHub directory and open the *DeliveryService* project on Eclipse. Next, right click on the project name in the *Project Explorer*

and click on *Run As* and *Run on Server*, as shown in Figure 1 below. *Apache Tomcat version 9.0* has been used as the server.

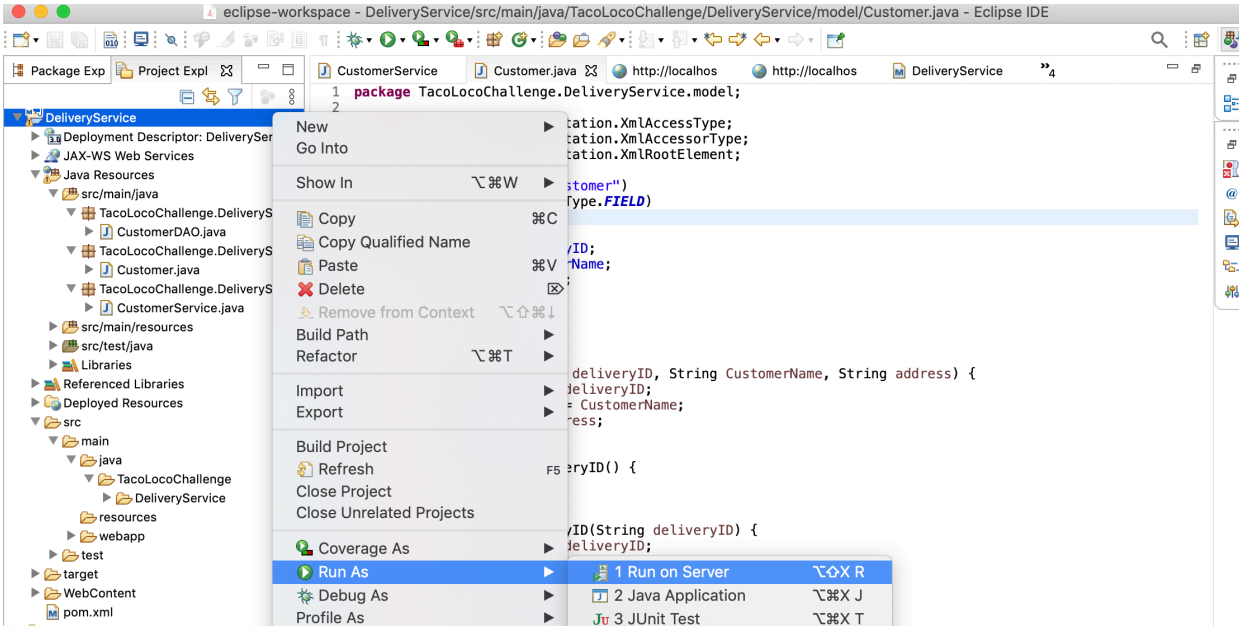

Figure 1: Running the Taco Loco Delivery Service Program

After the server starts running, the user will be navigated to the screen shown in Figure 2, below.

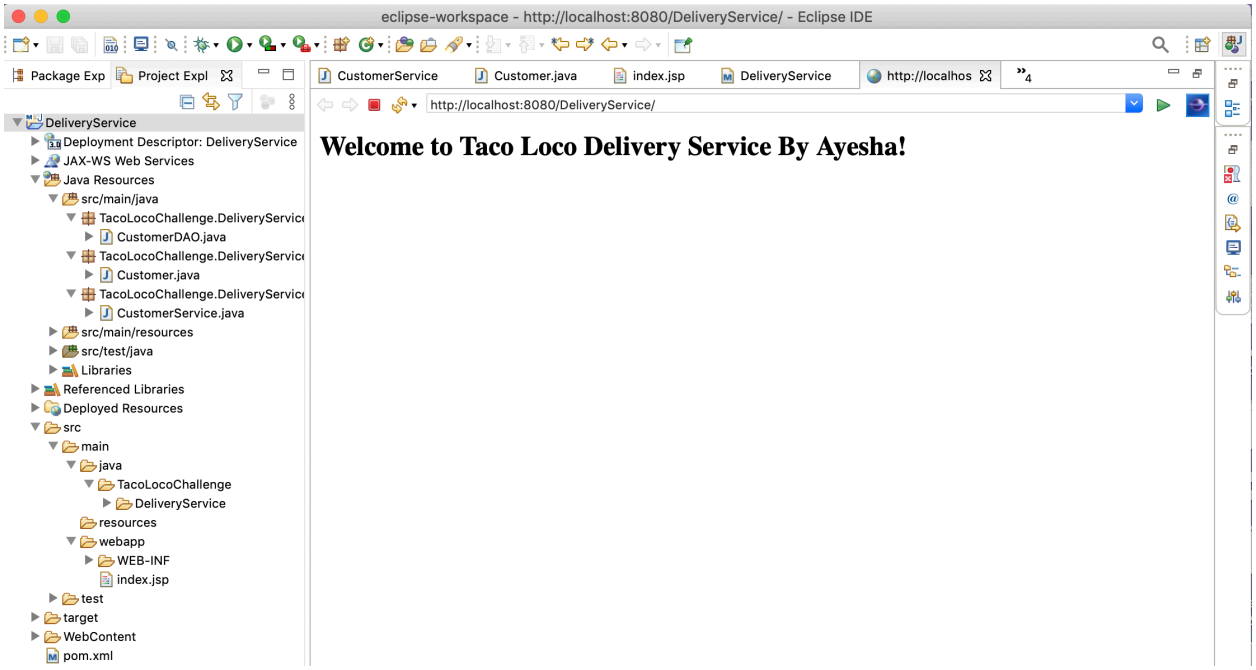

Figure 2: Delivery Service Running on The Server

Next, launch the Advanced REST Client Add-on from either Chrome or Firefox.

Looking Up All Deliveries in The Database

In order to lookup all deliveries in the database, the following path should be used:

http://localhost:8080/DeliveryService/rest/customers

Set the Method field to GET and click on the SEND button. This will display the entire database, as shown in Figure 3 below.

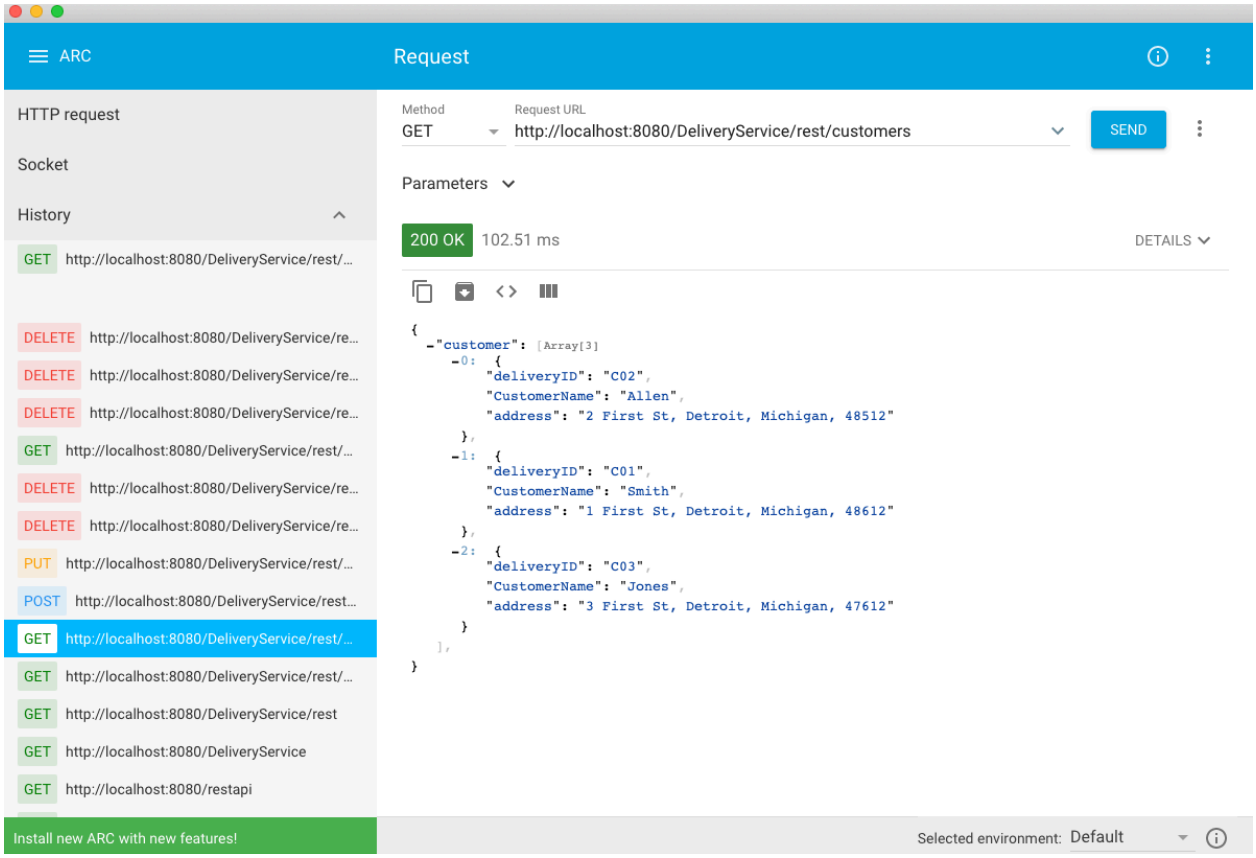

Figure 3: Looking Up All Deliveries in The Database

Looking Up A Delivery in The Database by Delivery ID

In order to lookup a delivery in the database by delivery ID, add the DeliveryID at the end of the URL path.

For example, to look for DeliveryID number “C03”, the following path should be used:

http://localhost:8080/DeliveryService/rest/customers/C03

Set the Method field to GET and click on the SEND button. This will display the customer with that particular ID, as shown in Figure 4 below.

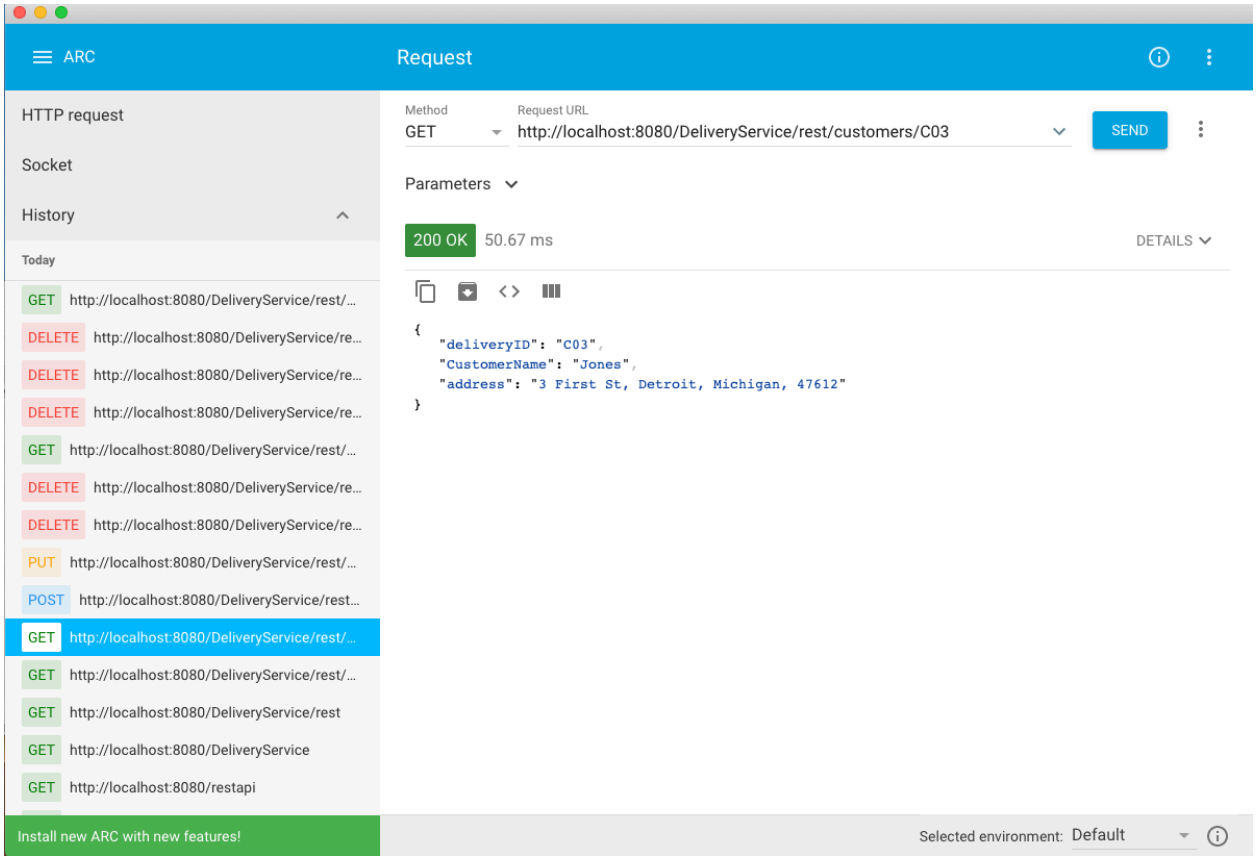

Figure 4: Looking Up A Delivery by the DeliveryID

Adding A New Delivery in The Database

In order to add a new delivery, the following path should be used:

http://localhost:8080/DeliveryService/rest/customers

Set the Method field to POST and expand the parameters section. The Headers section should be updated similar to Figure 5 shown below.

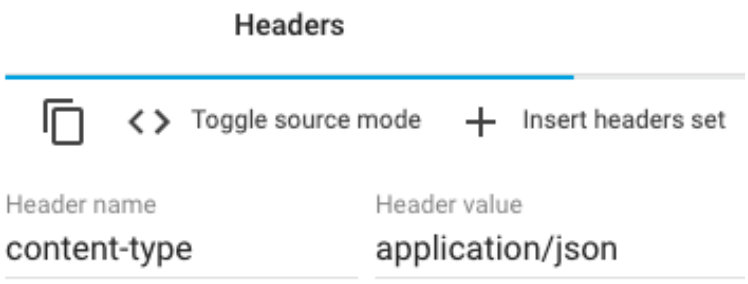

Figure 5: Headers Section

Update the body to add the new delivery's information and click on the SEND button. An example is shown in Figure 6 below.

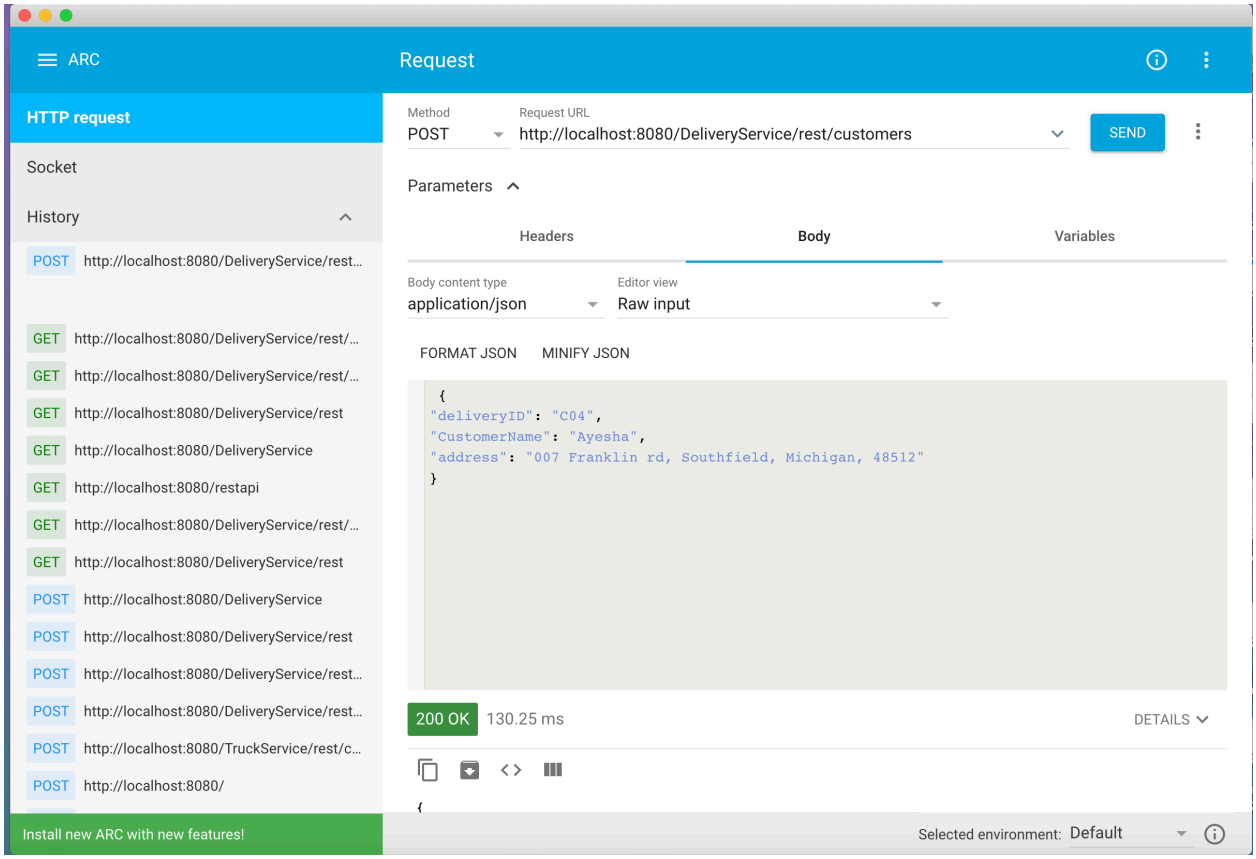

Figure 6: Adding A New Delivery to The Database

Now, if a GET request is sent again to display all the deliveries in the database, the new delivery and the customer's information should be added, as shown in Figure 7 below.

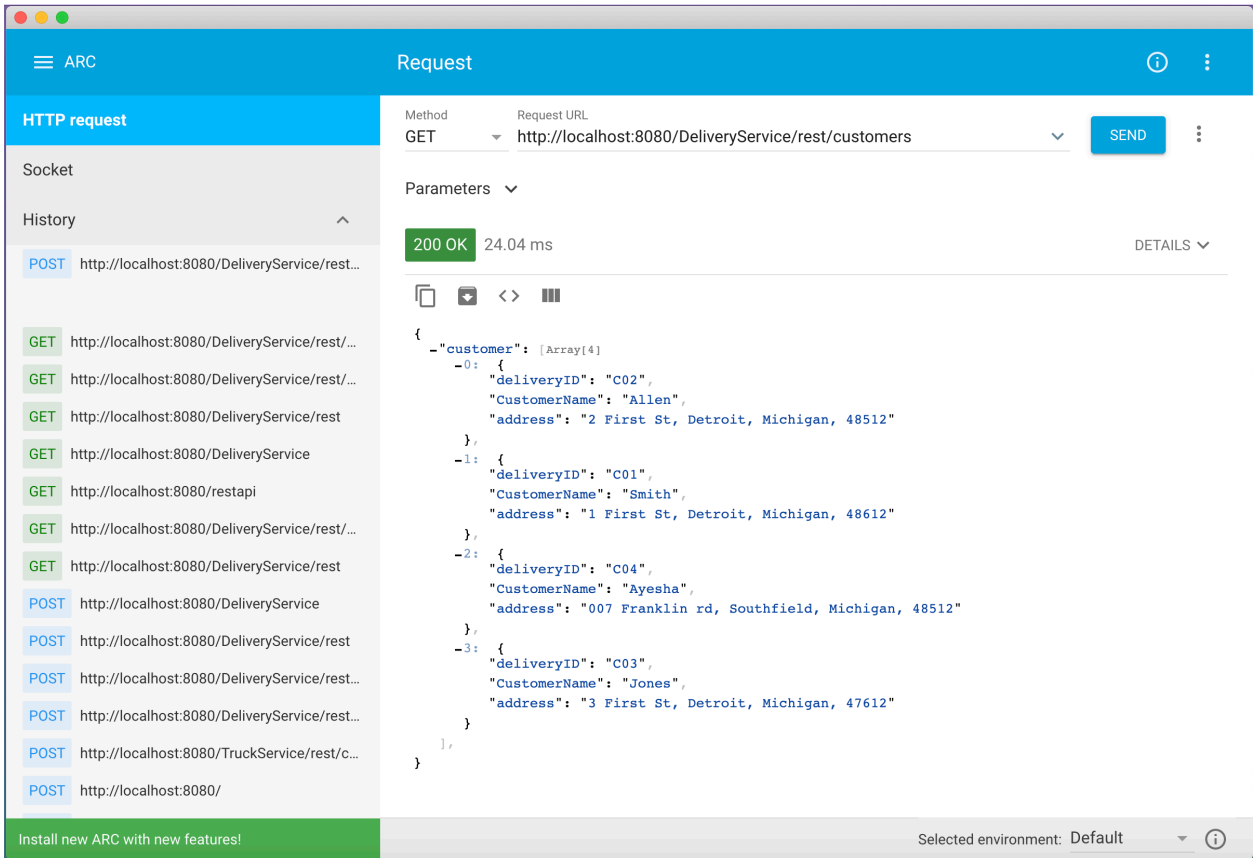

Figure 7: Database Updated with The New Customer's Delivery Information

Updating A Delivery in The Database

In order to update a delivery, the following path should be used:

http://localhost:8080/DeliveryService/rest/customers

Set the Method field to PUT and expand the parameters section. The Headers section should be updated similar to Figure 8 shown below.

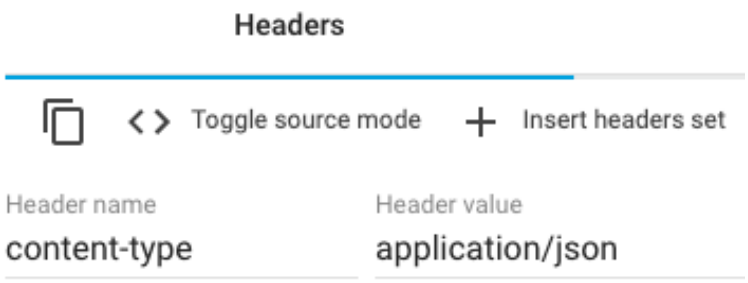

Figure 8: Headers Section

Update the body to update the delivery's information and click on the SEND button. An example is shown in Figure 9 below. Here, the customer's building number has been updated from 007 to 168 in the address field.

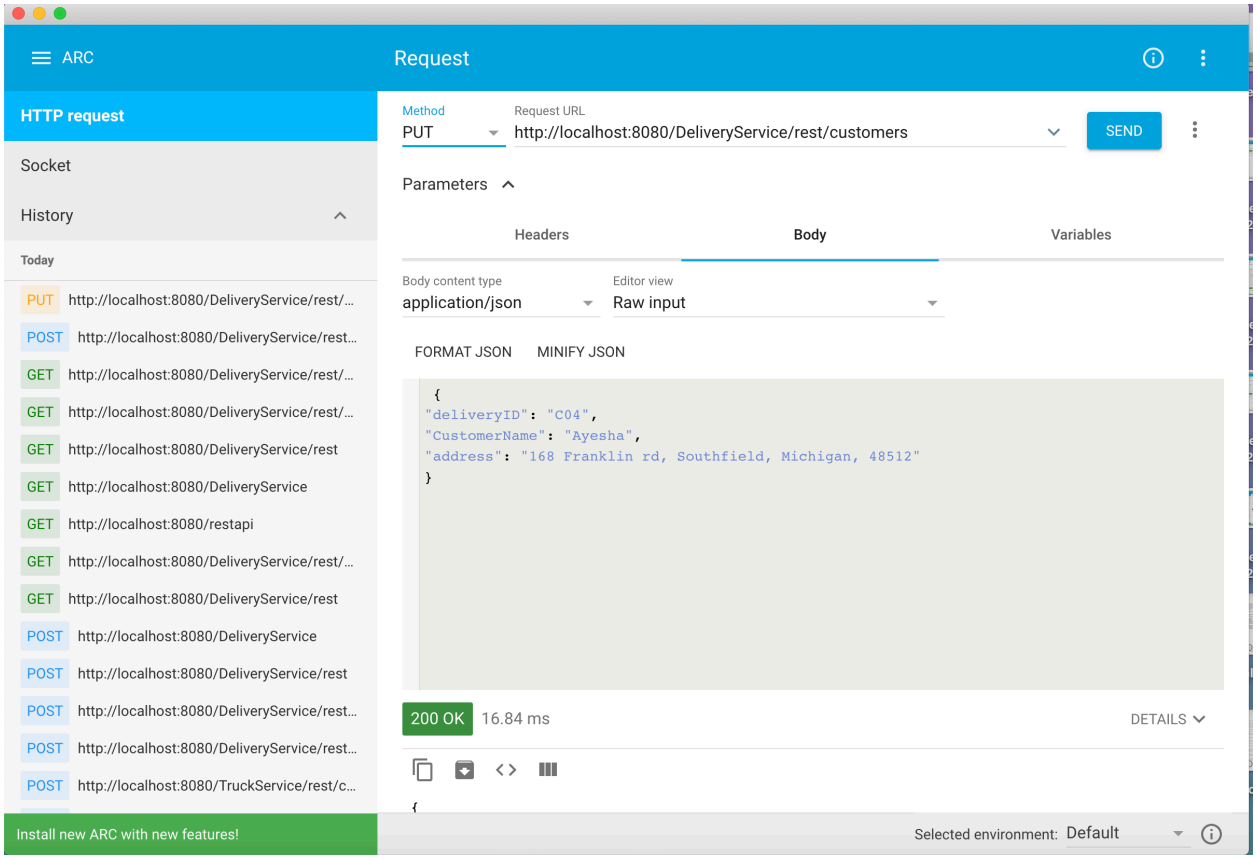

Figure 9: Updating a Delivery

Now, if a GET request is sent again, to display all the deliveries in the database, the new delivery and the customer's information should be updated, as shown in Figure 10 below.

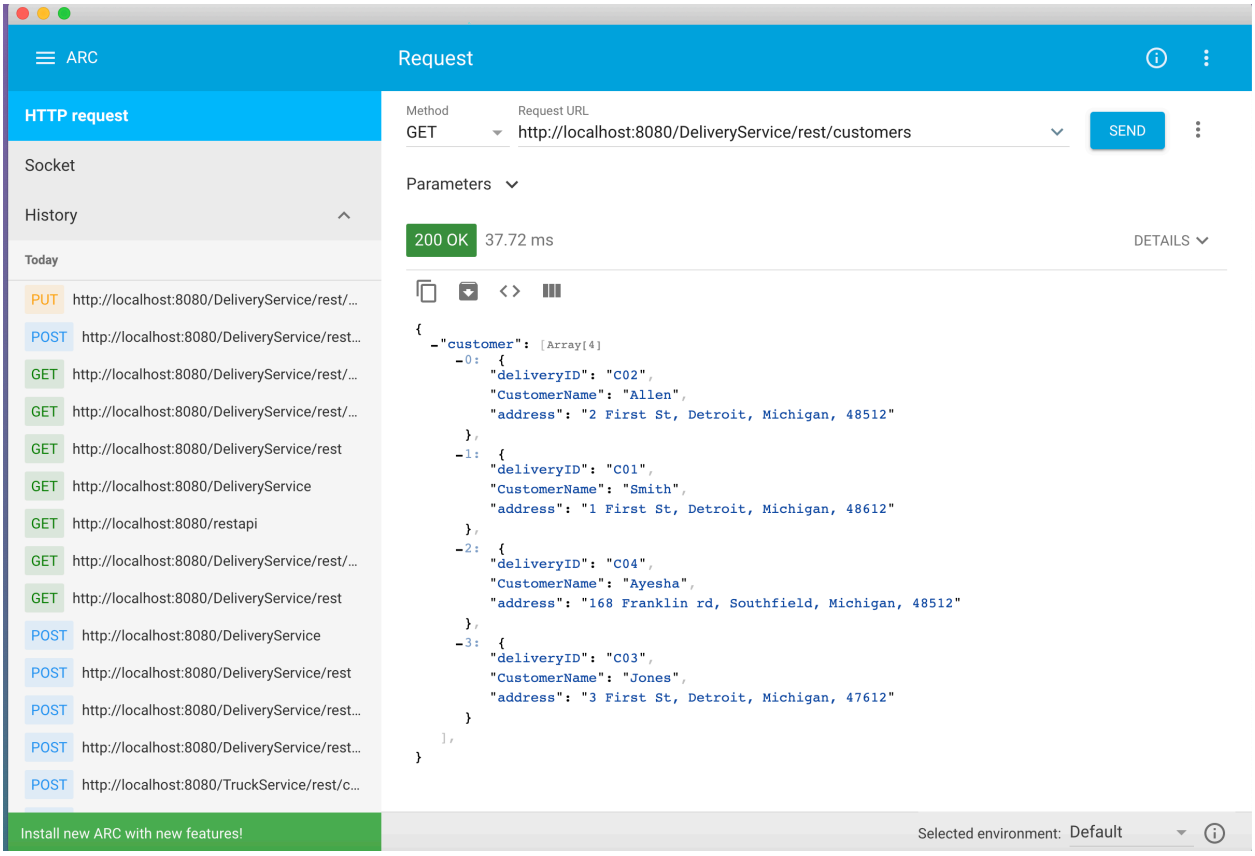

Figure 10: Updated Customer's Address in the Delivery Database

Deleting A Delivery from The Database

In order to delete a delivery in the database, add the DeliveryID at the end of the URL path.

For example, to delete the DeliveryID number "C04", the following path should be used:

http://localhost:8080/DeliveryService/rest/customers/C04

Set the Method field to DELETE and click on the SEND button, as shown in Figure 11 below.

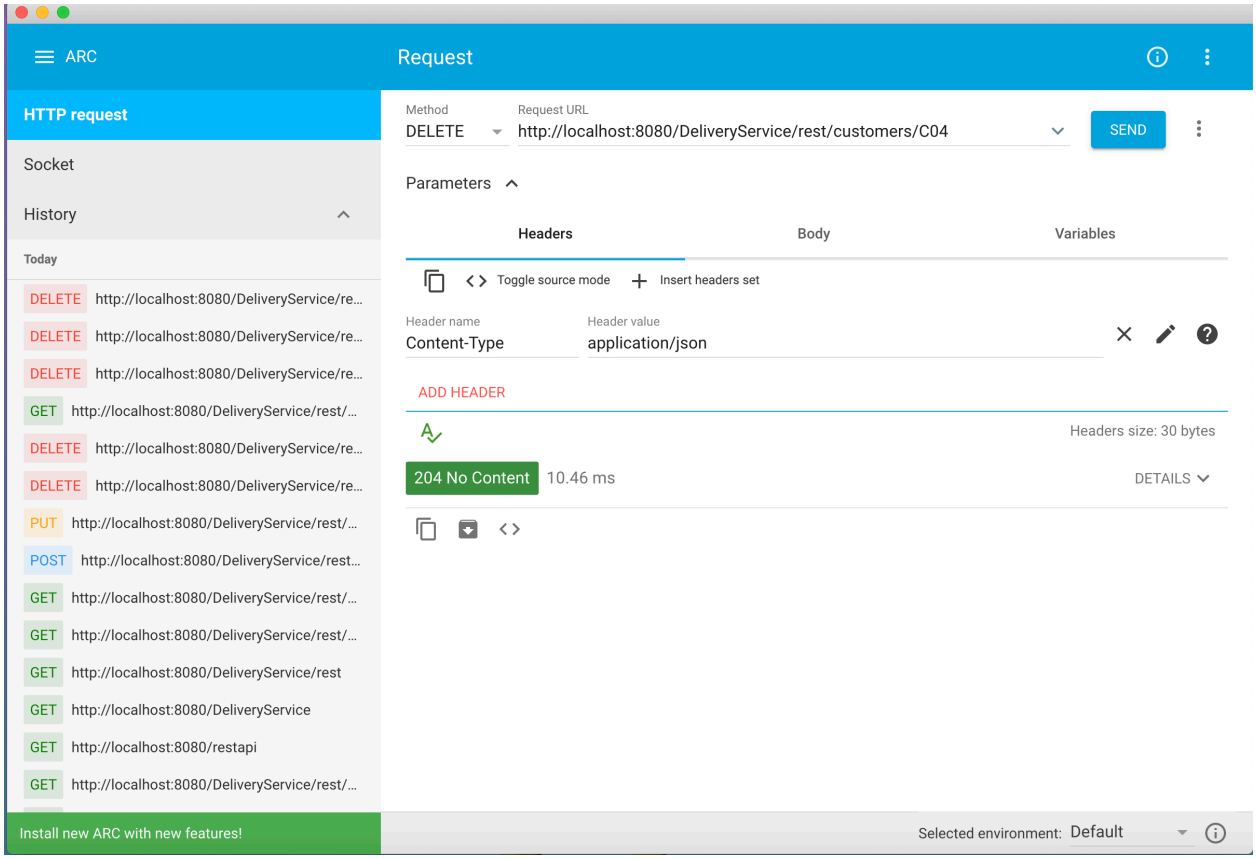

Figure 11: Deleting A Delivery from the Database

Now, if a GET request is sent again, to display all the deliveries in the database, the customer with that particular ID will be deleted, as shown in Figure 12.

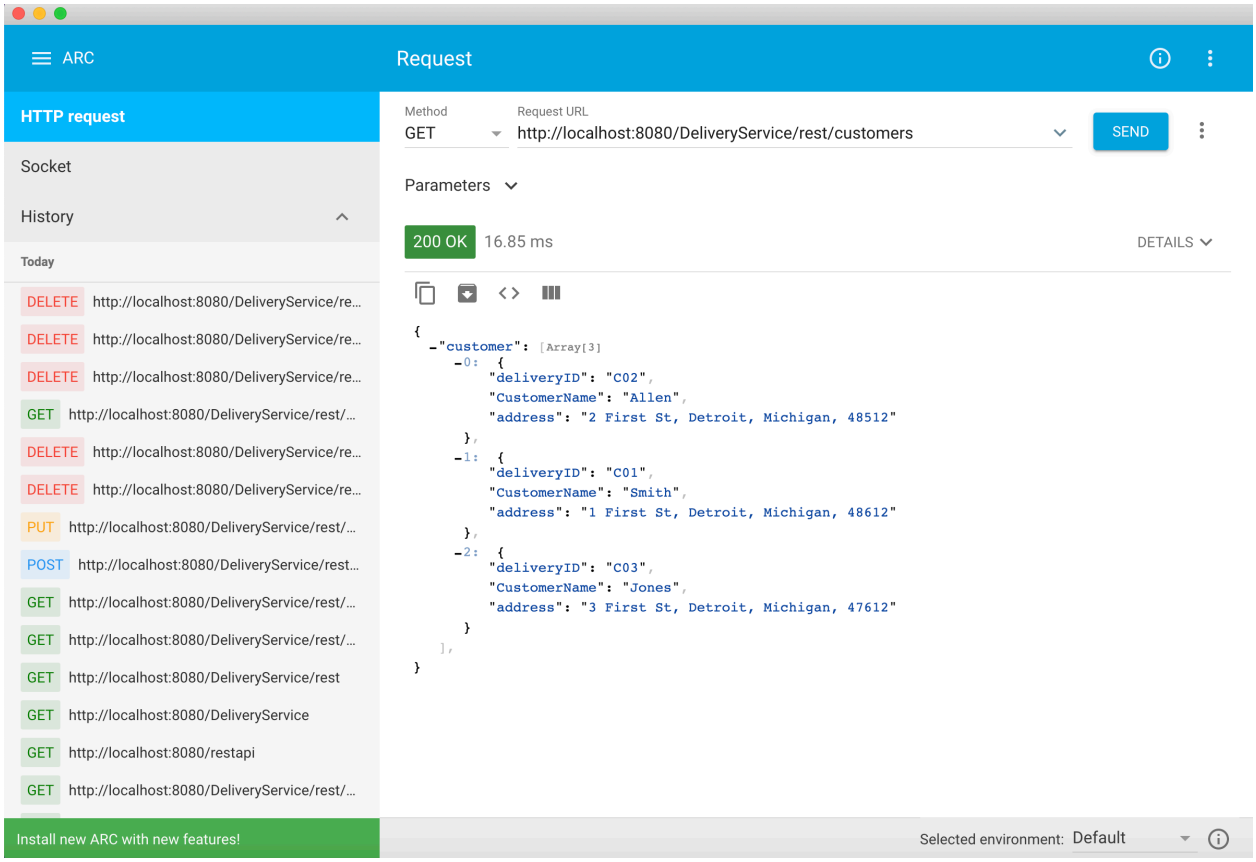

Figure 12: Deleted DeliveryID "C04" from the Delivery Database