Web API Design with SpringBoot Week 2 Coding Assignment

**SRI SRINIVASAN**

**Points possible:** 70

|  |  |  |
| --- | --- | --- |
| Category | Criteria | % of Grade |
| Functionality | Does the code work? | 25 |
| Organization | Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear. | 25 |
| Creativity | Student solved the problems presented in the assignment using creativity and out of the box thinking. | 25 |
| Completeness | All requirements of the assignment are complete. | 25 |

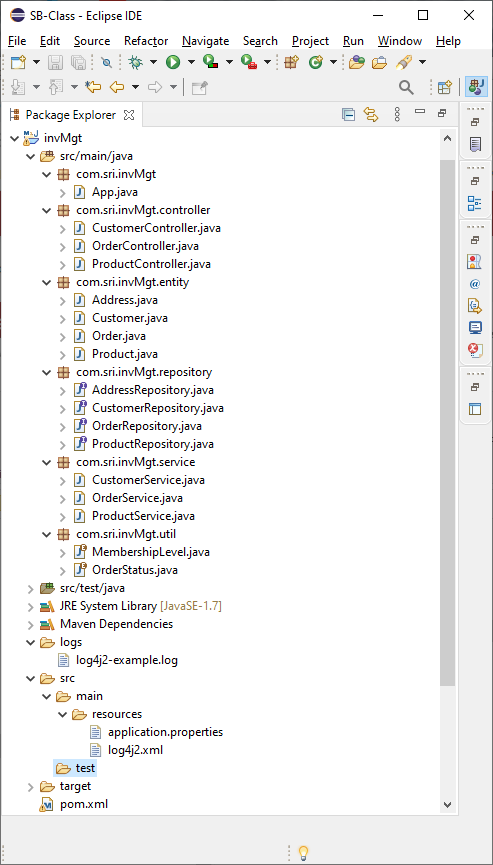
**Instructions:** In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week’s assignments and push this document, with your Java project code, to the repository. Add the URL for this week’s repository to this document where instructed and submit this document to your instructor when complete.

**Coding Steps:**

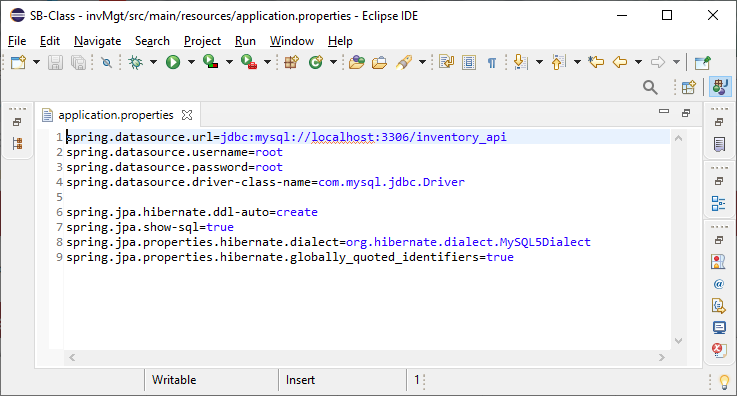
Follow the tutorial in the homework section to build an API. Paste screenshots of your code and your postman requests and responses to show the API works. Push your project to GitHub and paste the link below.

**Screenshots of Code:**

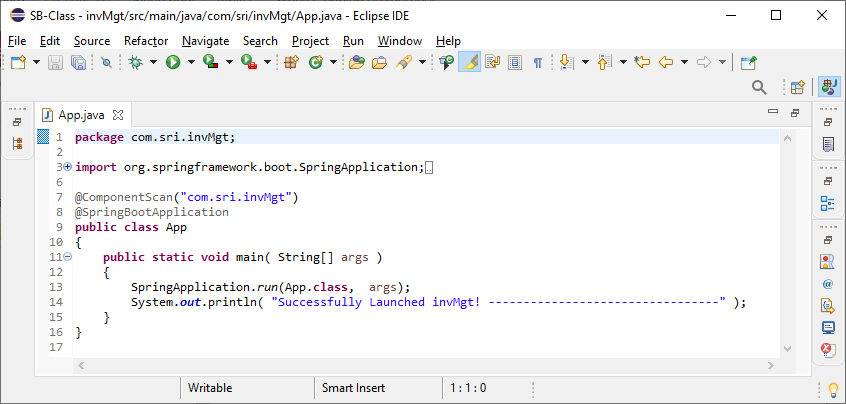
Project Structure



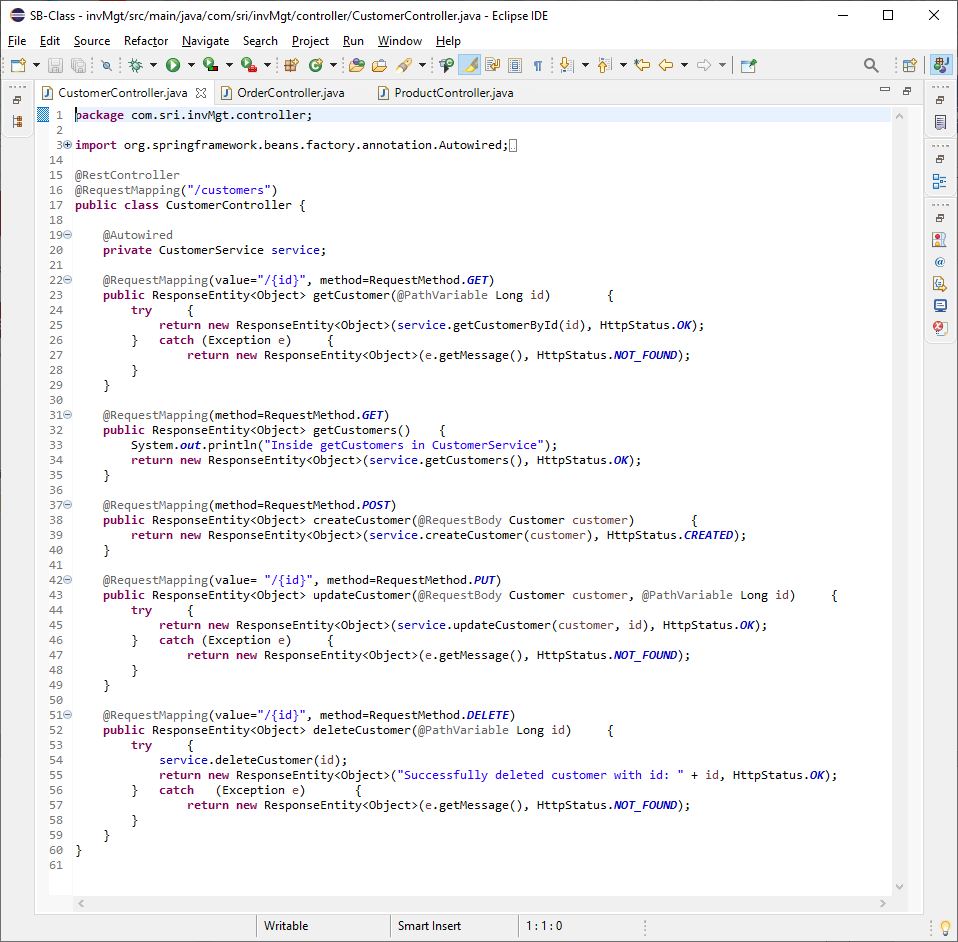
Application.properties



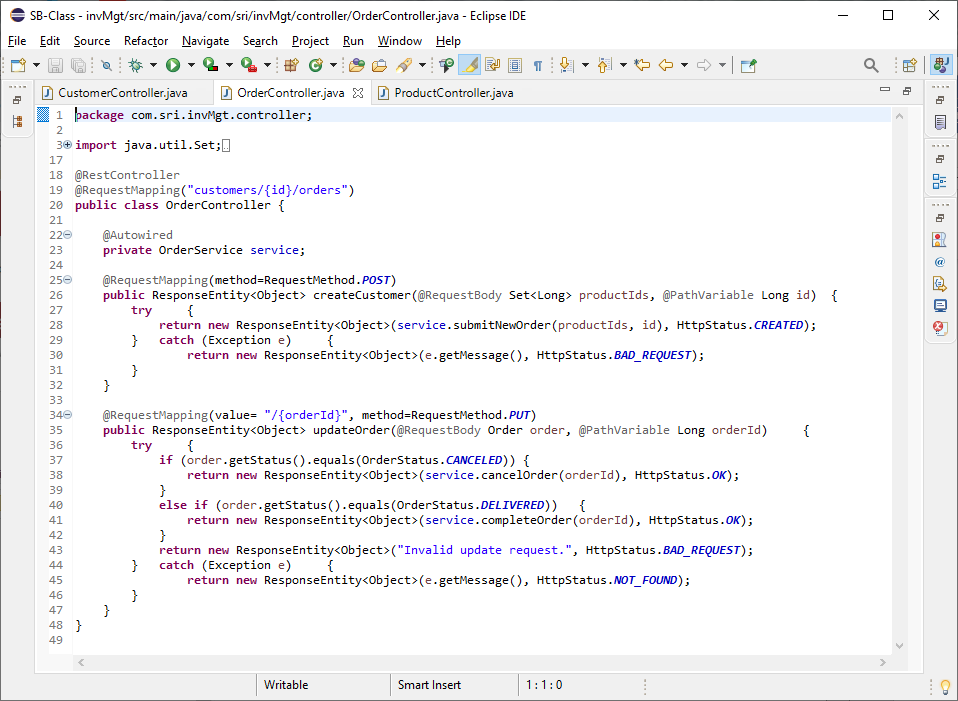
App.java



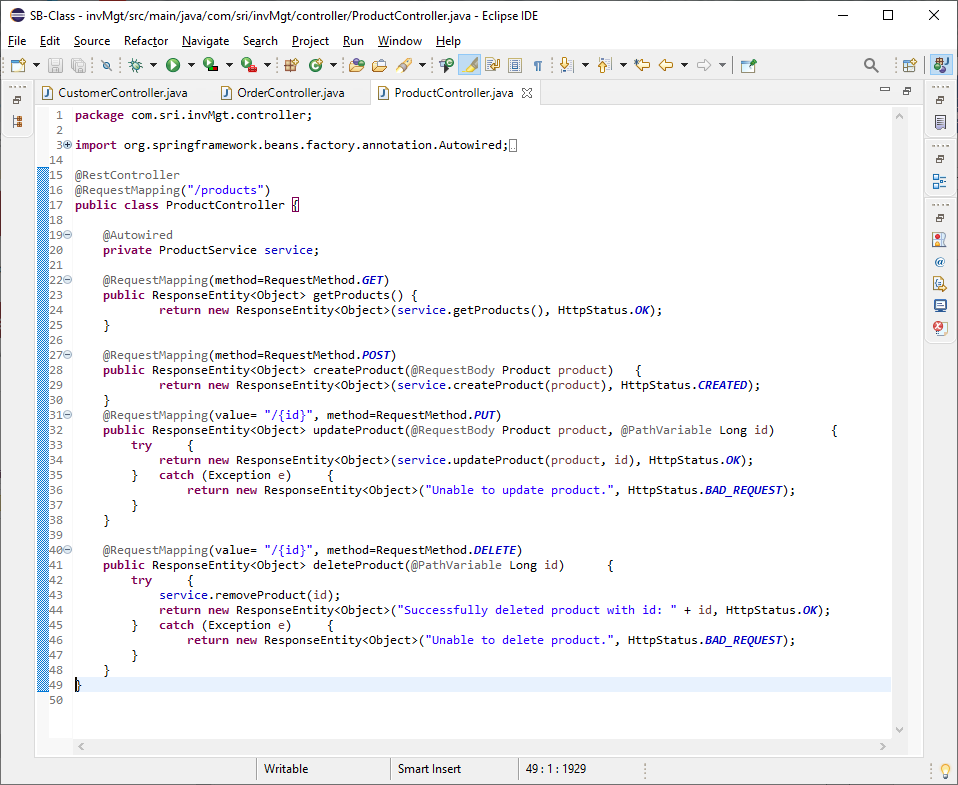
CustomerController.java(com.sri.invMgt.controller)



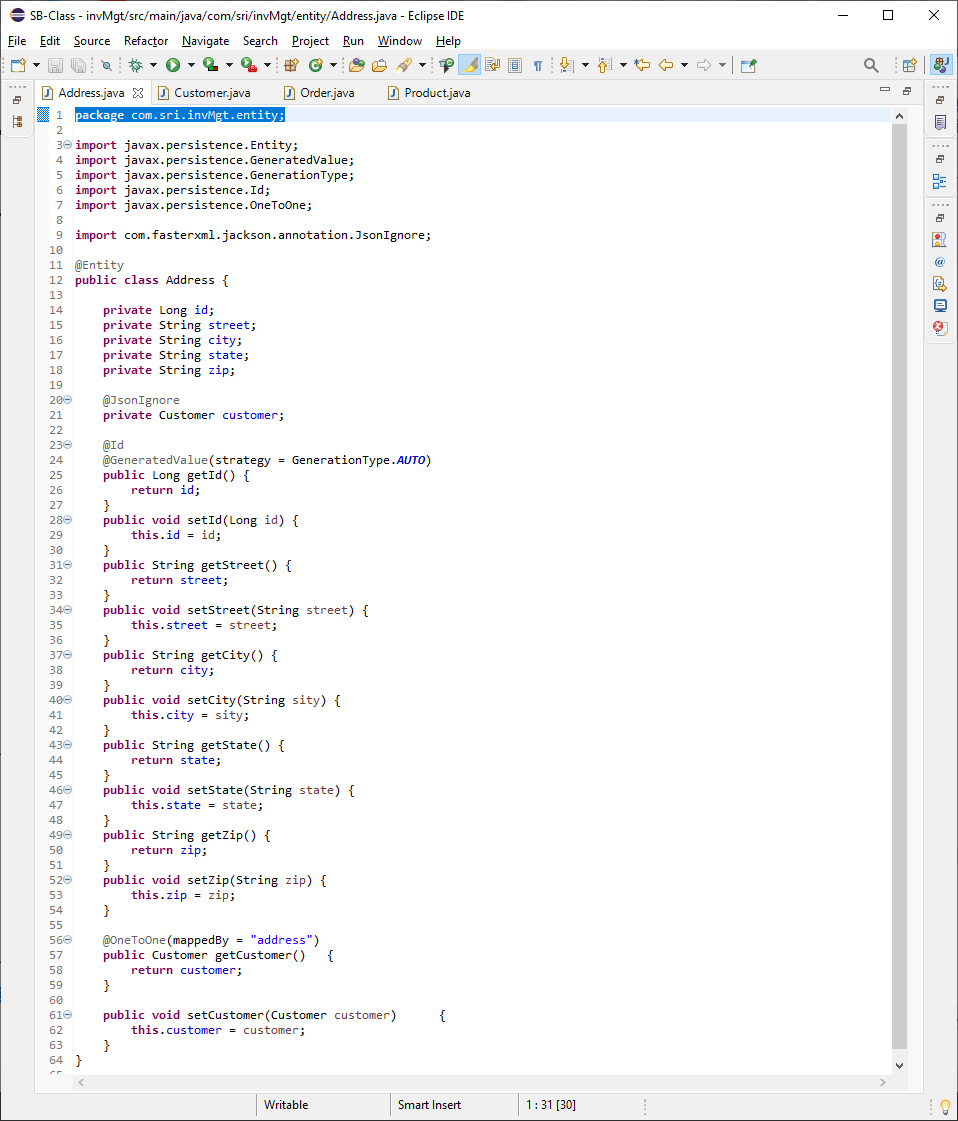
OrderConroller.java(com.sri.invMgt.controller)



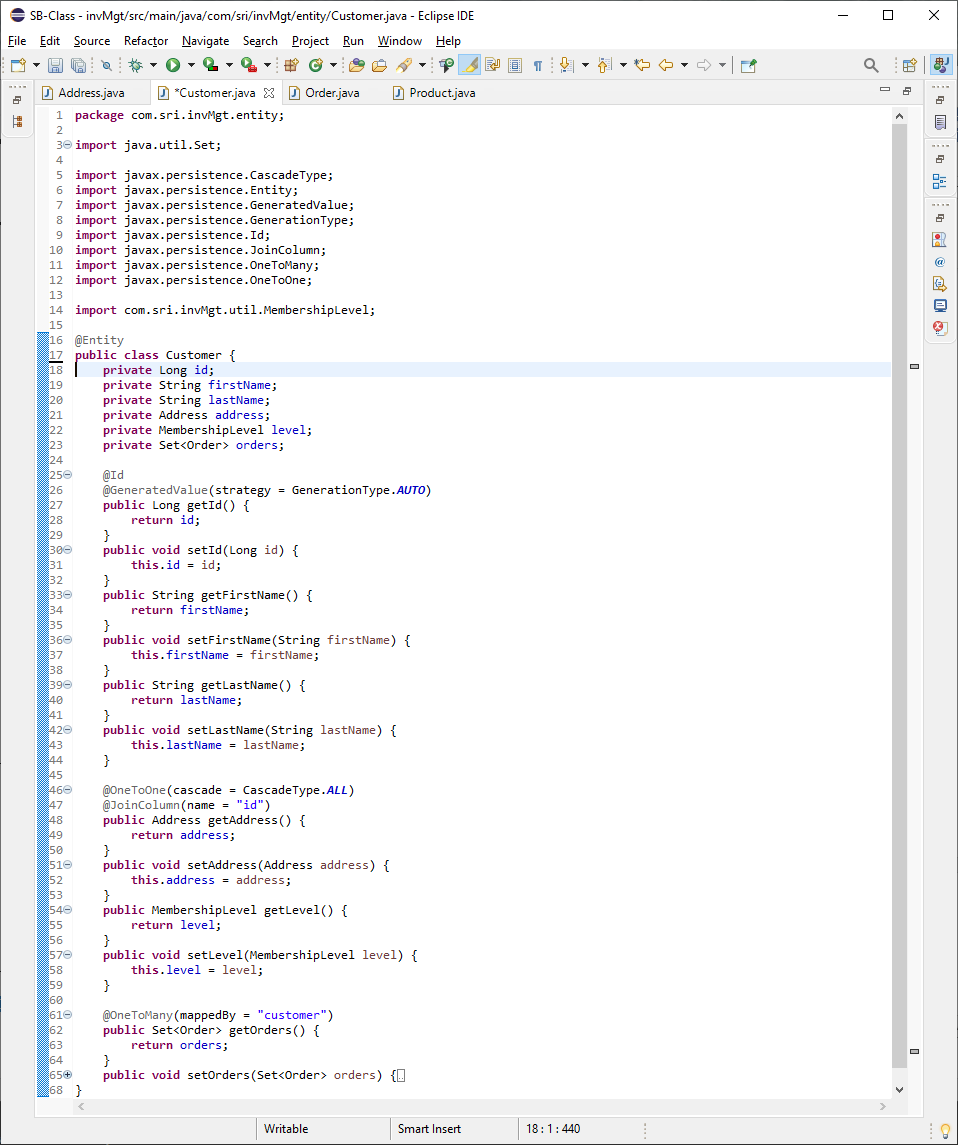
ProductController.java(com.sri.invMgt.controller)



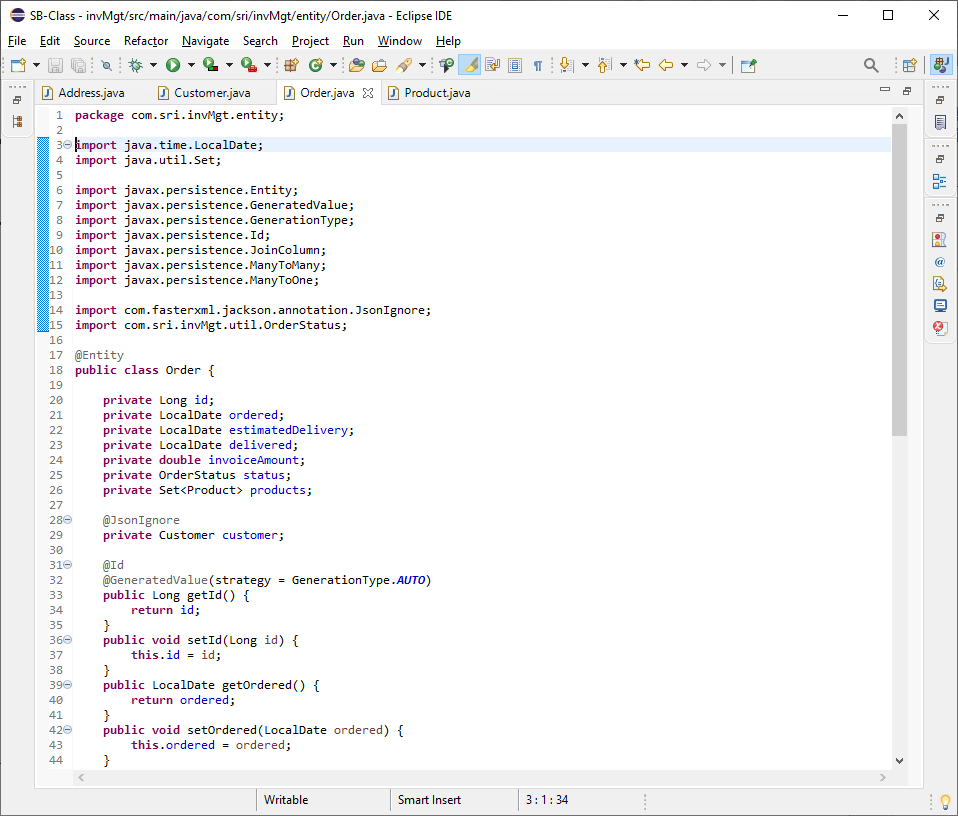
Address.java (package com.sri.invMgt.entity)

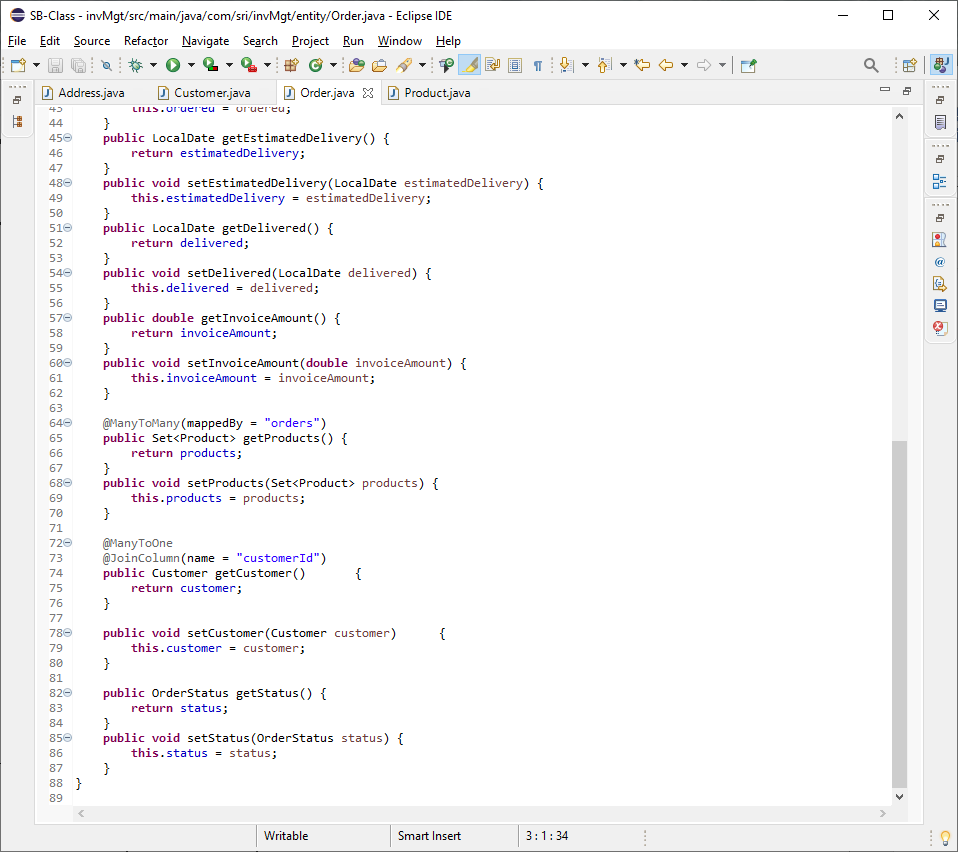


Customer.java (package com.sri.invMgt.entity)

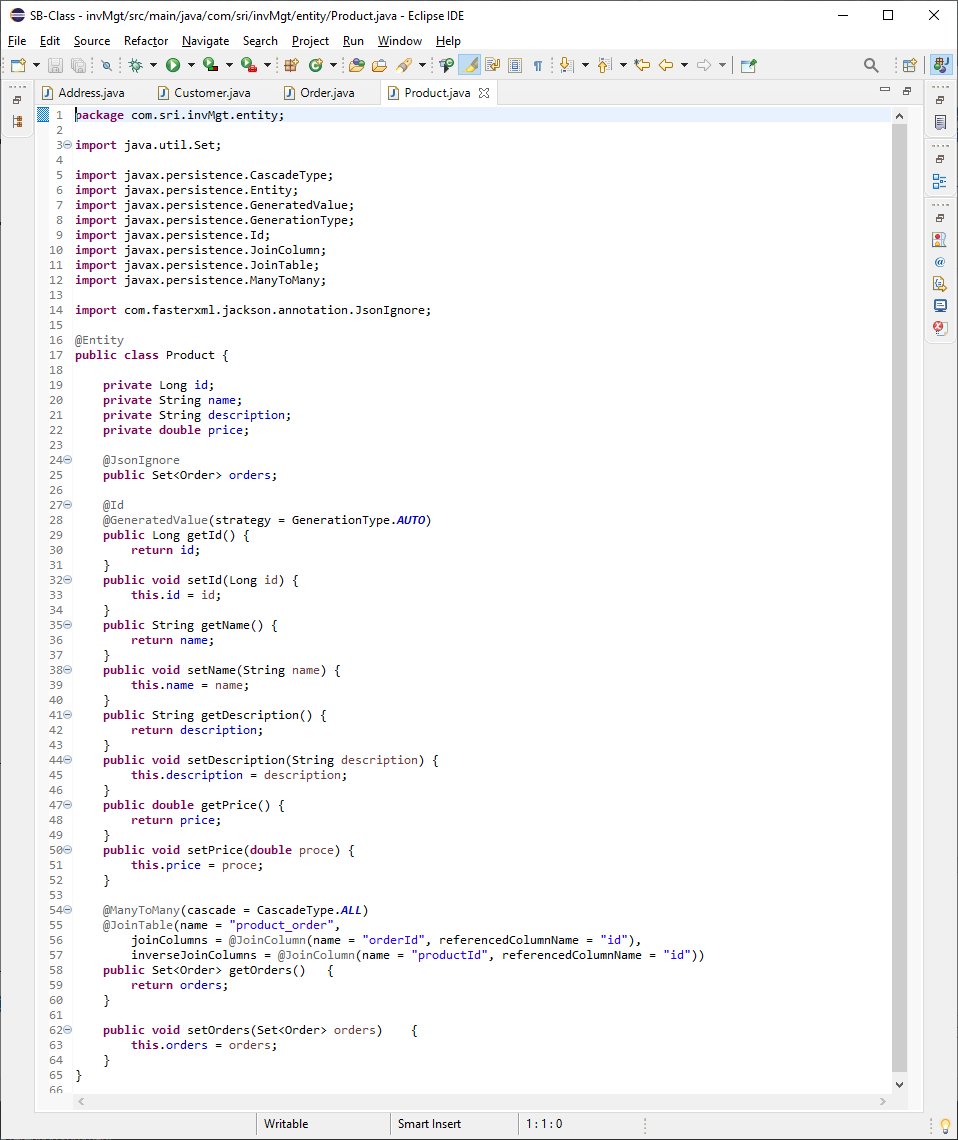


Order.java (package com.sri.invMgt.entity)

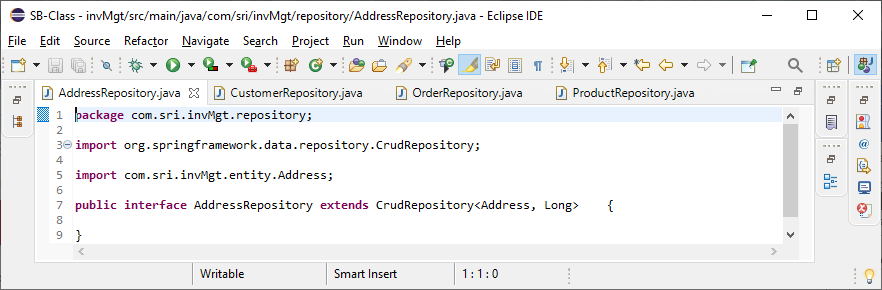




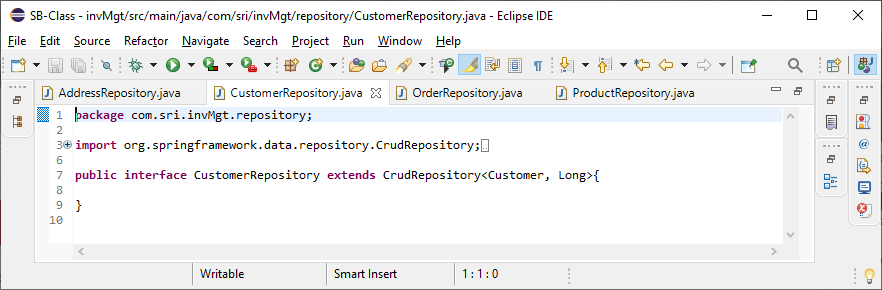
Product.java (package com.sri.invMgt.entity)



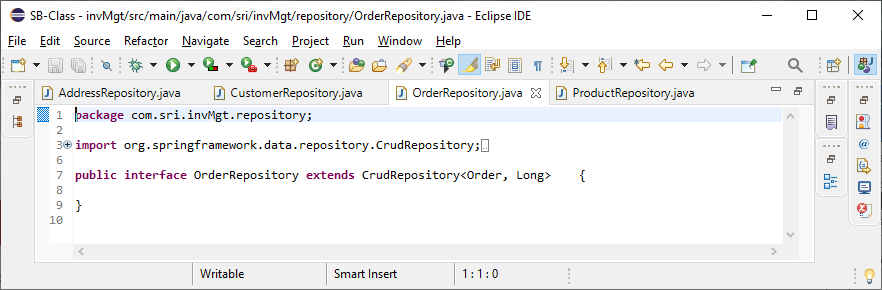
AddressRepository(package com.sri.invMgt.repository)



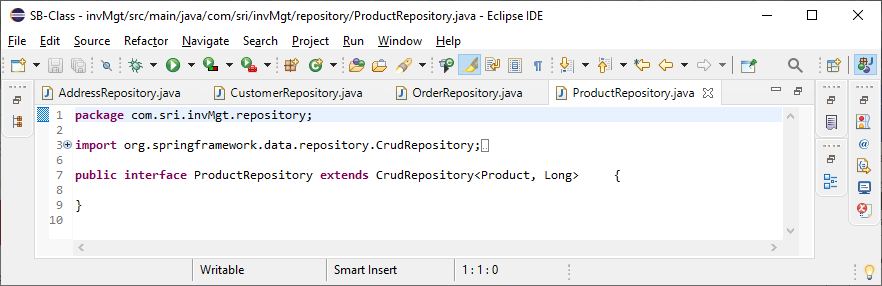
CustomerRepository(package com.sri.invMgt.repository)



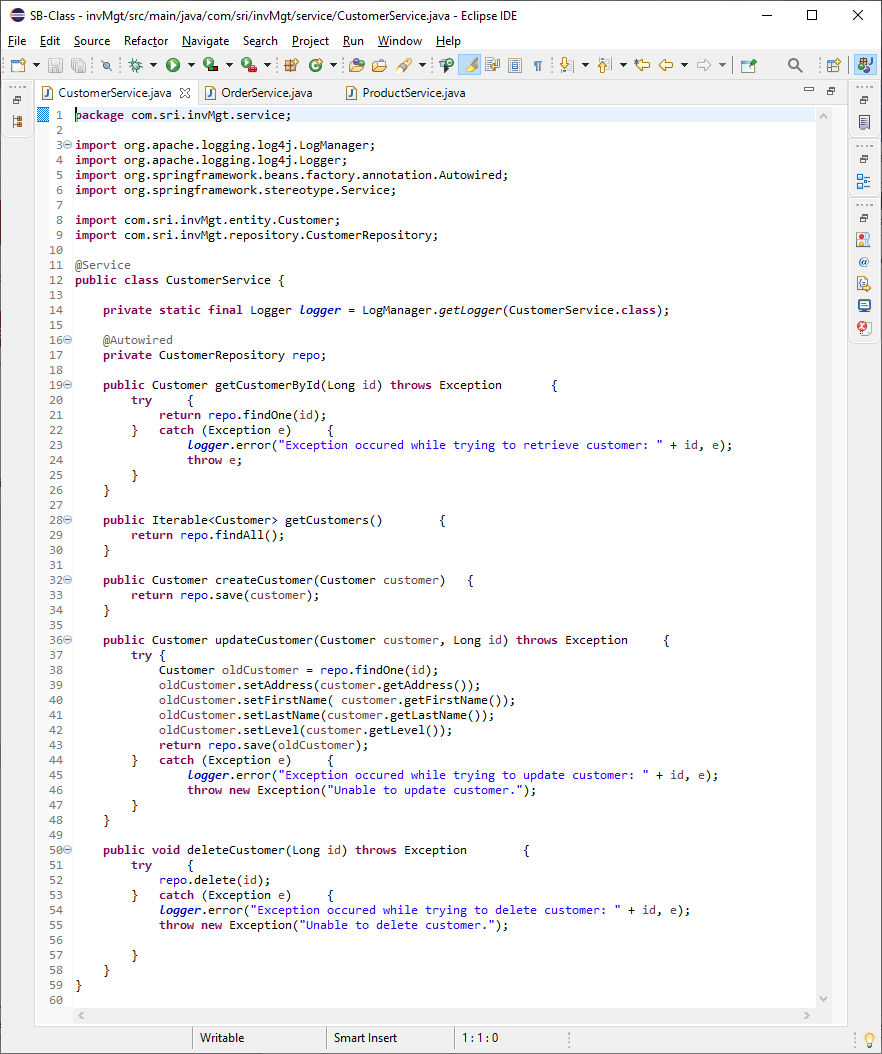
OrderRepository.java(package com.sri.invMgt.repository)



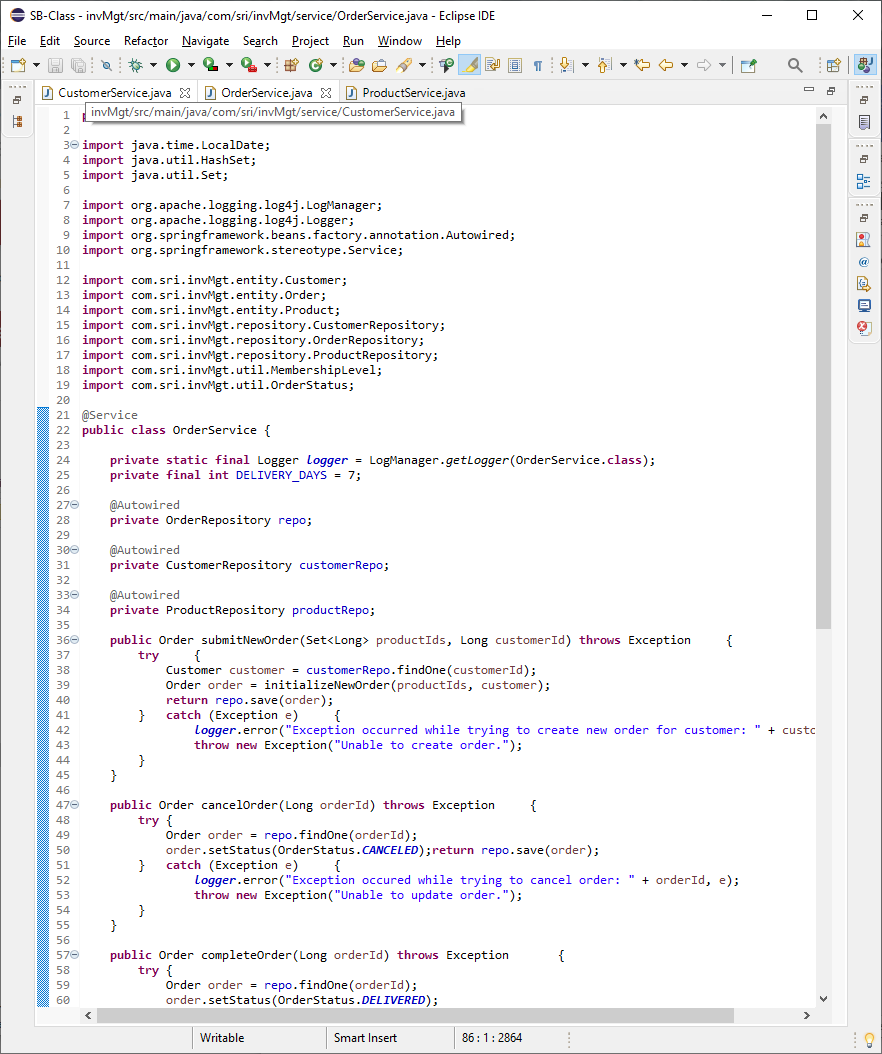
ProductRepository.java(package com.sri.invMgt.repository)

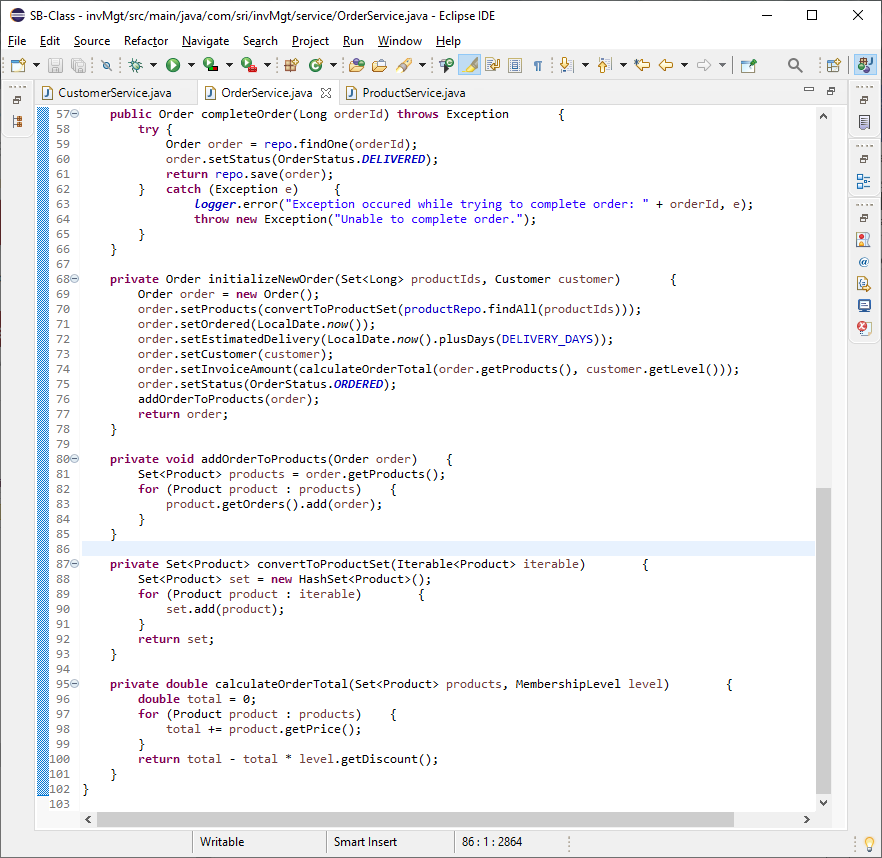


CustomerService(package com.sri.invMgt.service)

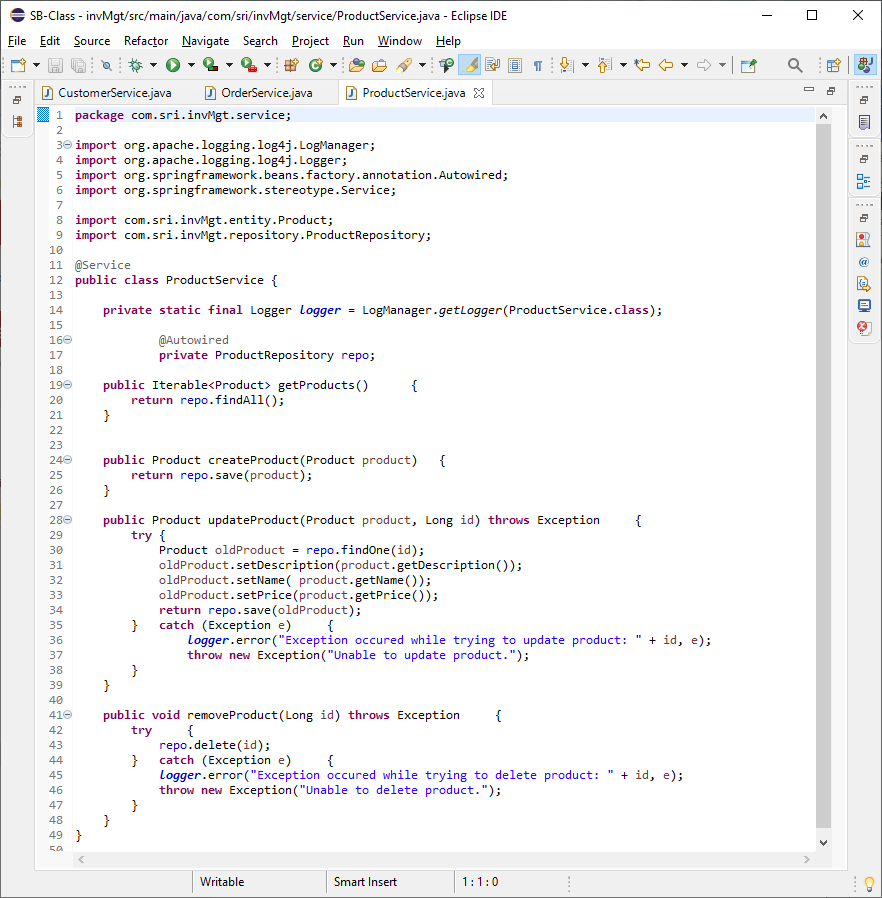


OrderService(package com.sri.invMgt.service)

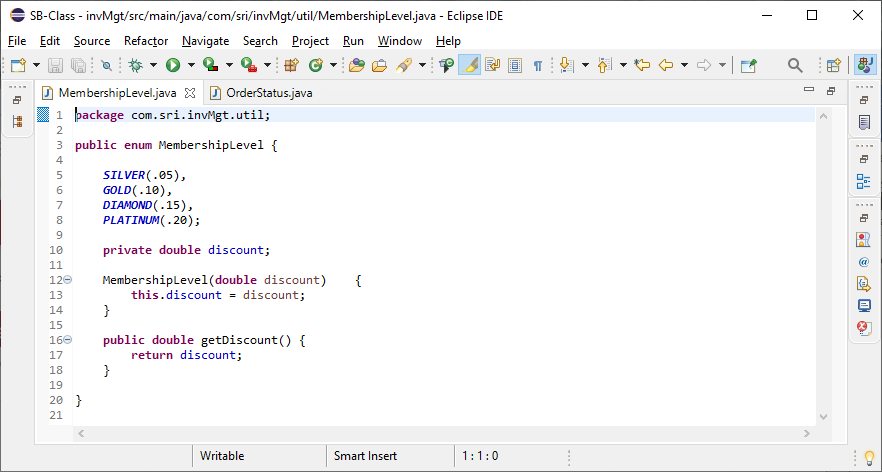




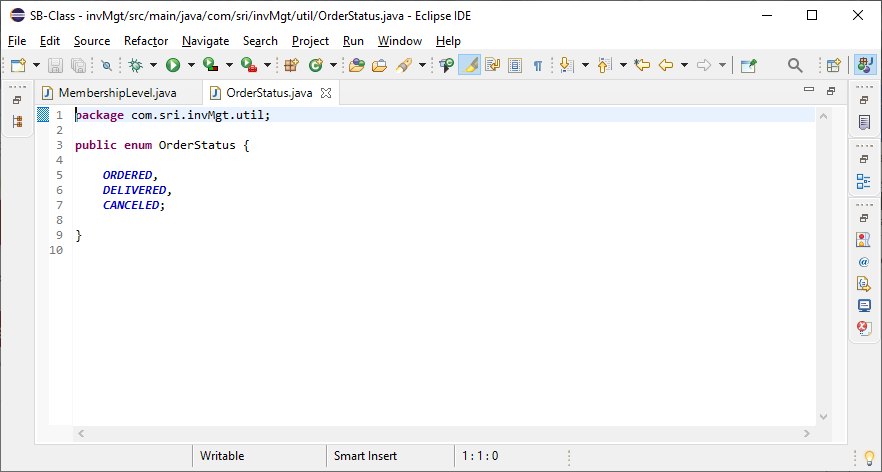
ProductService(package com.sri.invMgt.service)



MembershipLevel.java(package com.sri.invMgt.util)



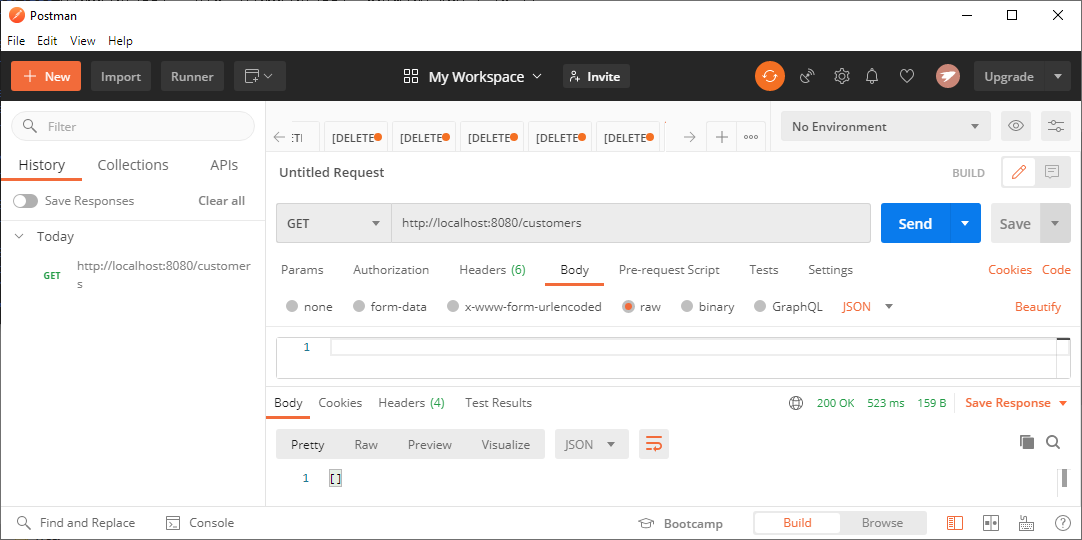
OrderStatus.java(package com.sri.invMgt.util)



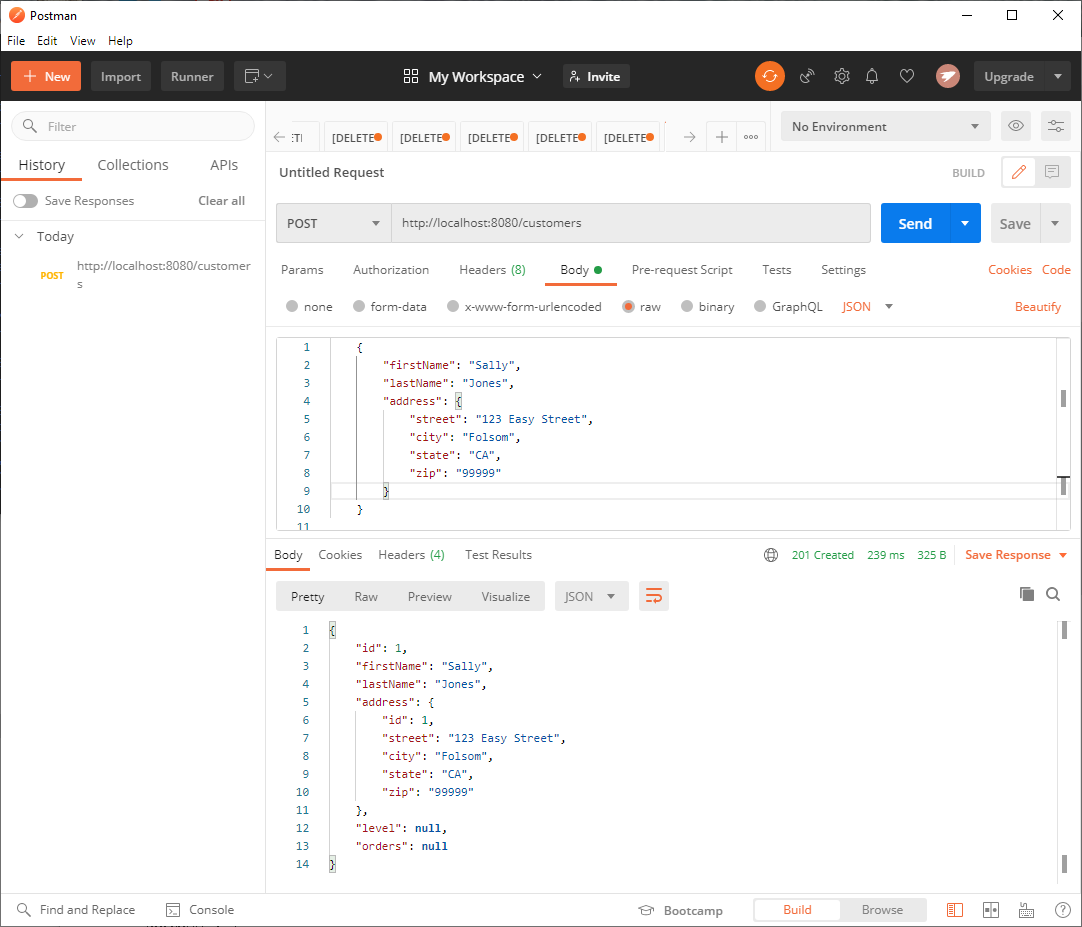
**Screenshots of Running Application:**

SCREEN SHOTS FOR POSTMAN TESTING:

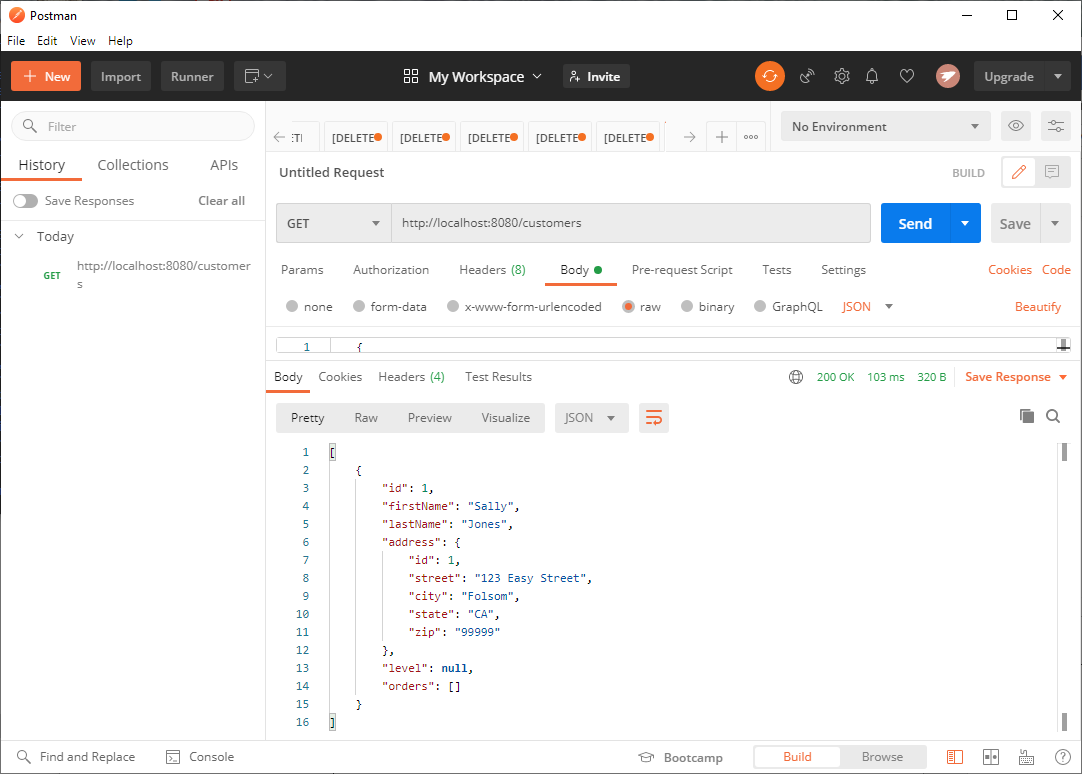
GET command from customers (initial): [Endpoint]: customers



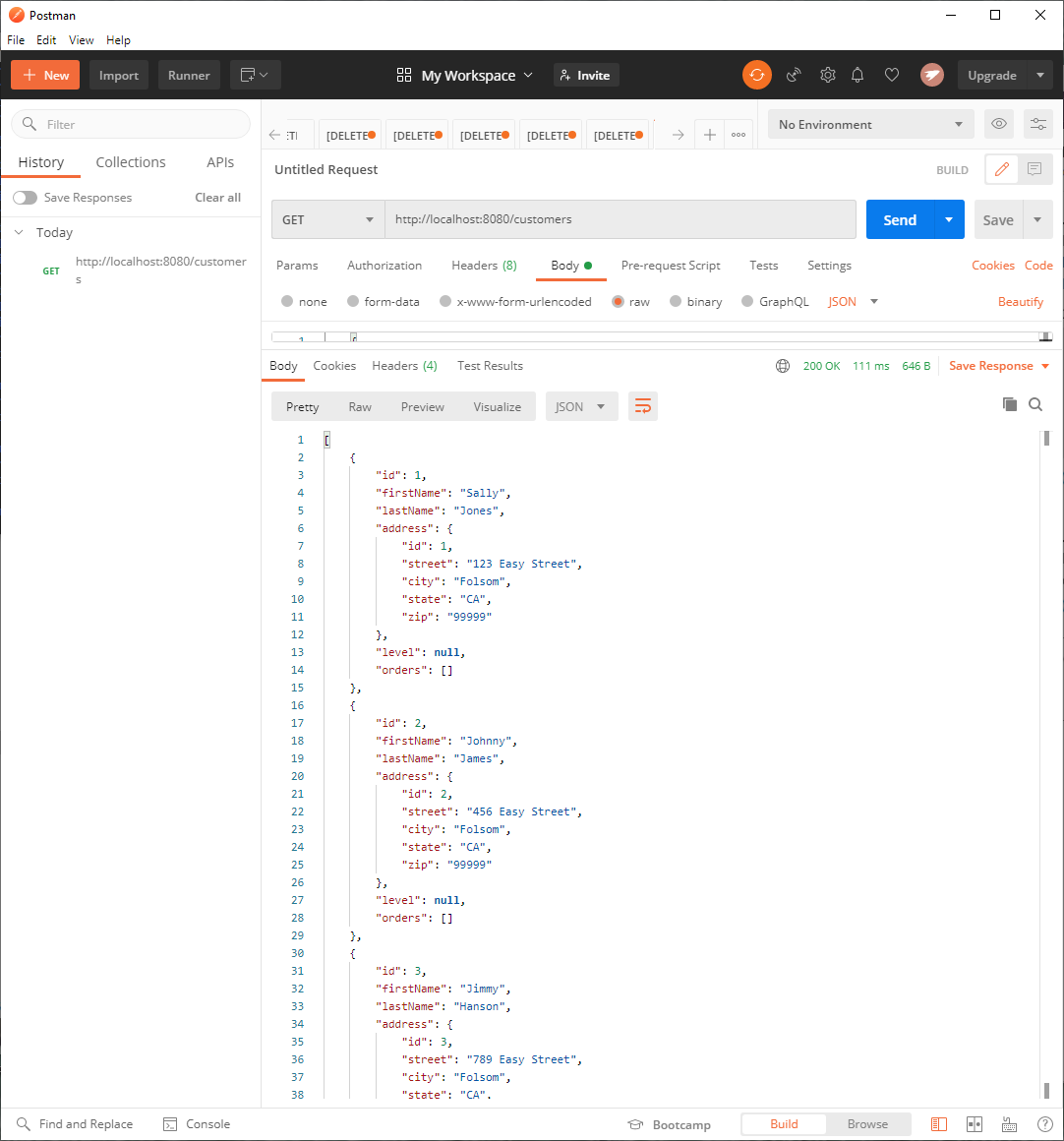
Adding (CREATE)customer using POST command: [Endpoint]: customers



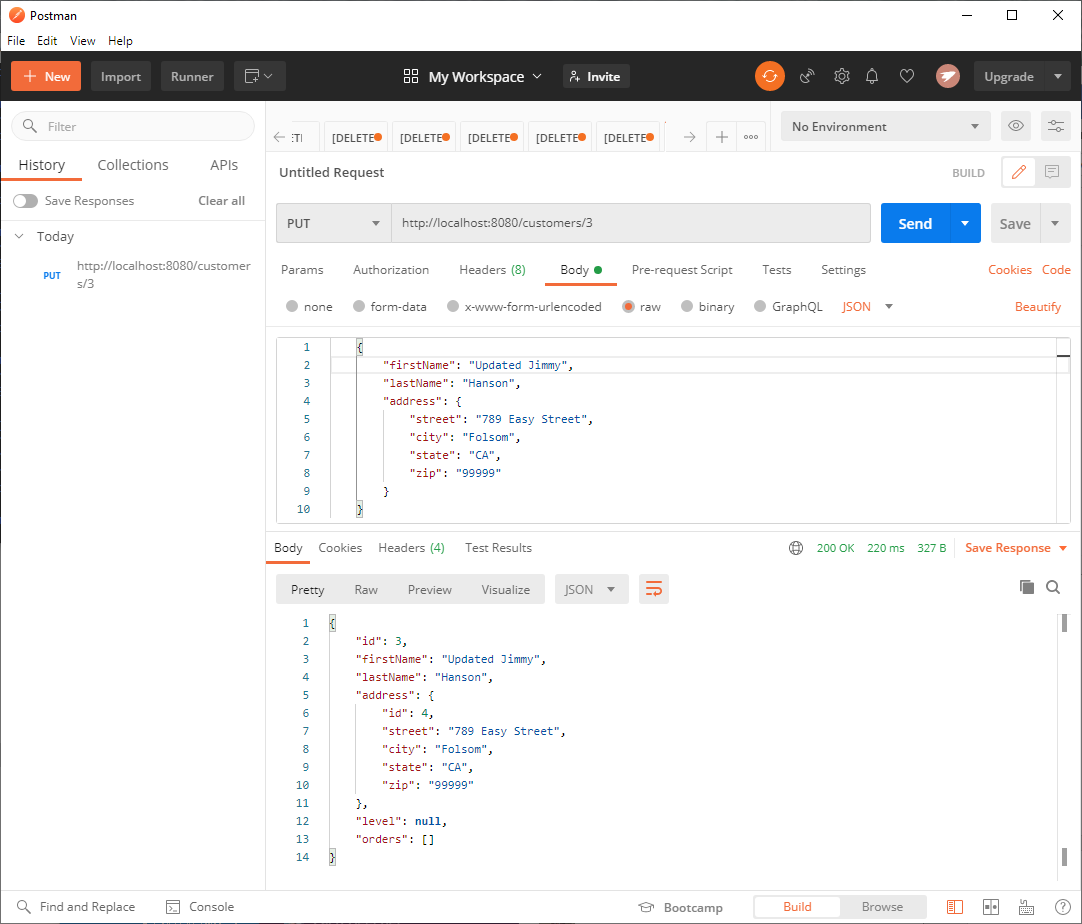
GET (READ) command from customers (after customer 1 is added): [Endpoint]: customers



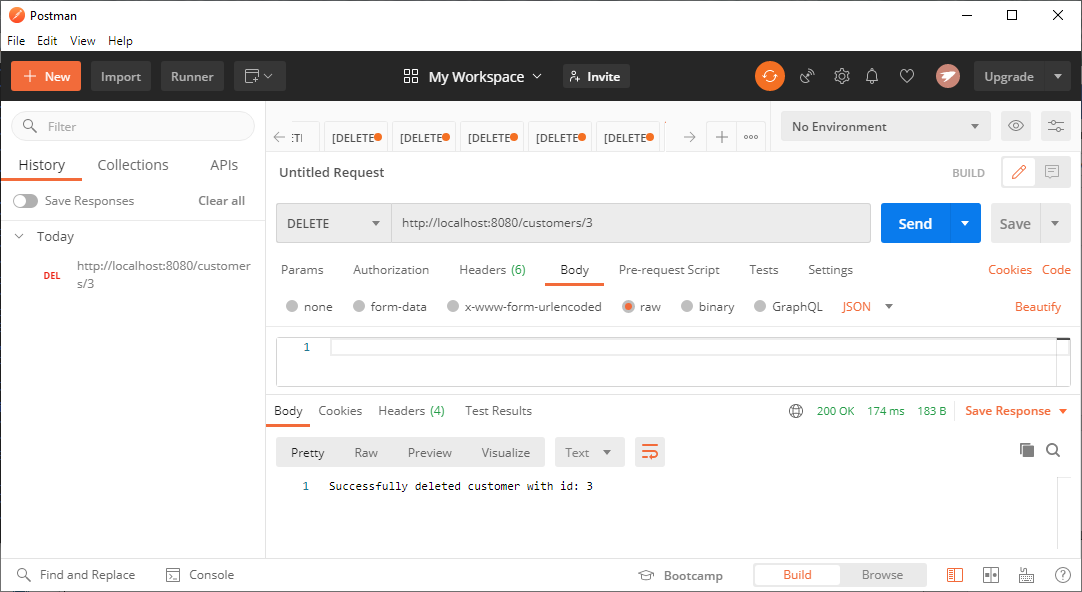
GET (READ) command from customers (total of 3 added): [Endpoint]: customers



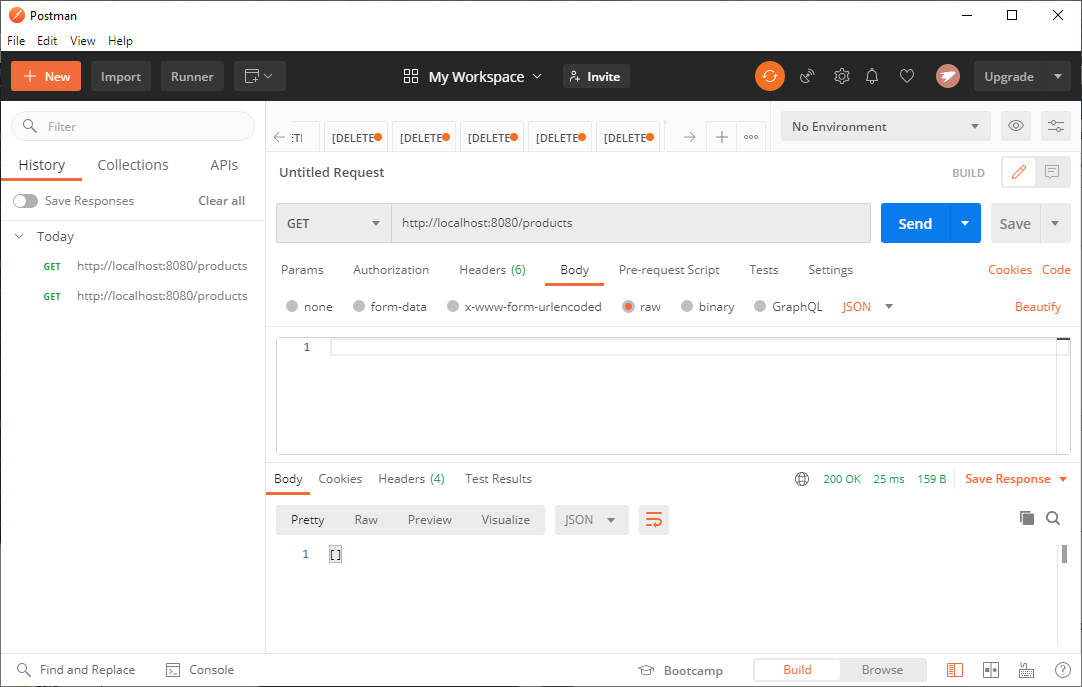
PUT (UPDATE) command from customers (update customer id #3): [Endpoint]: customers



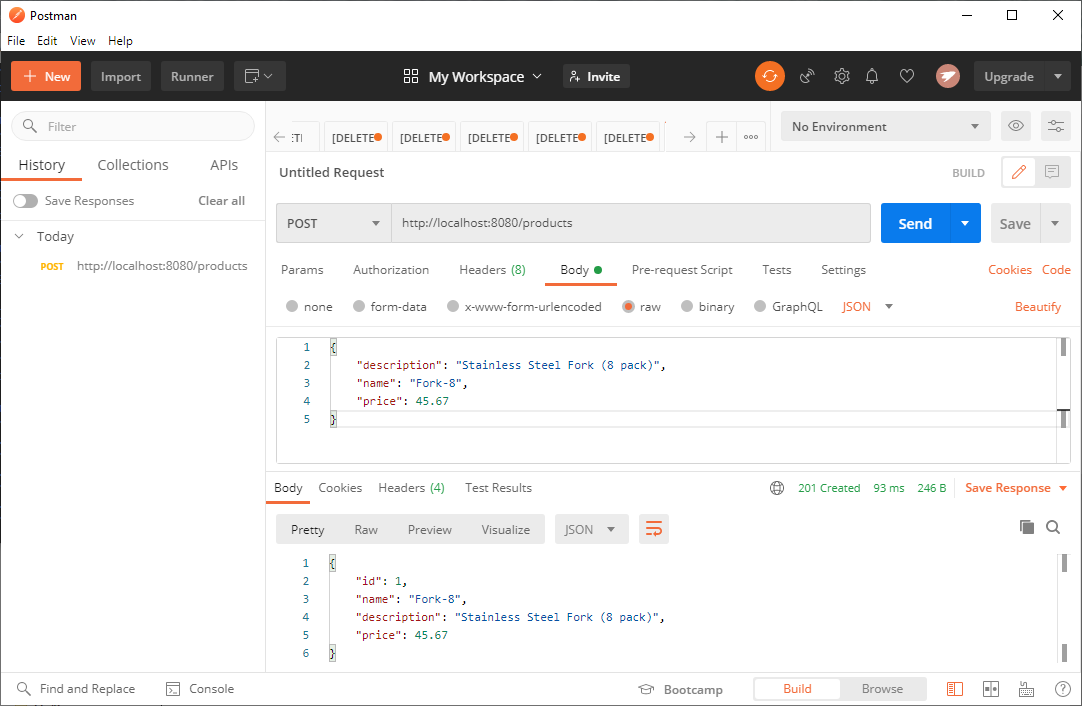
DELETE (DELETE) command from customers (delete customer id #3): [Endpoint]: customers/3



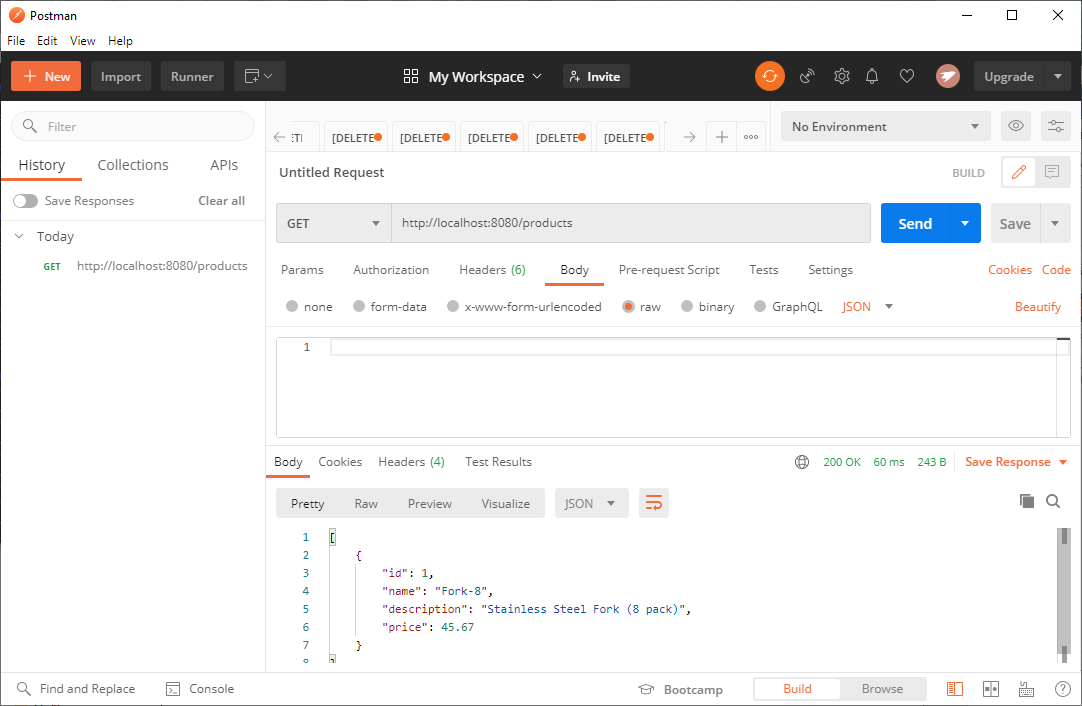
GET command from products (initial): [Endpoint]: products



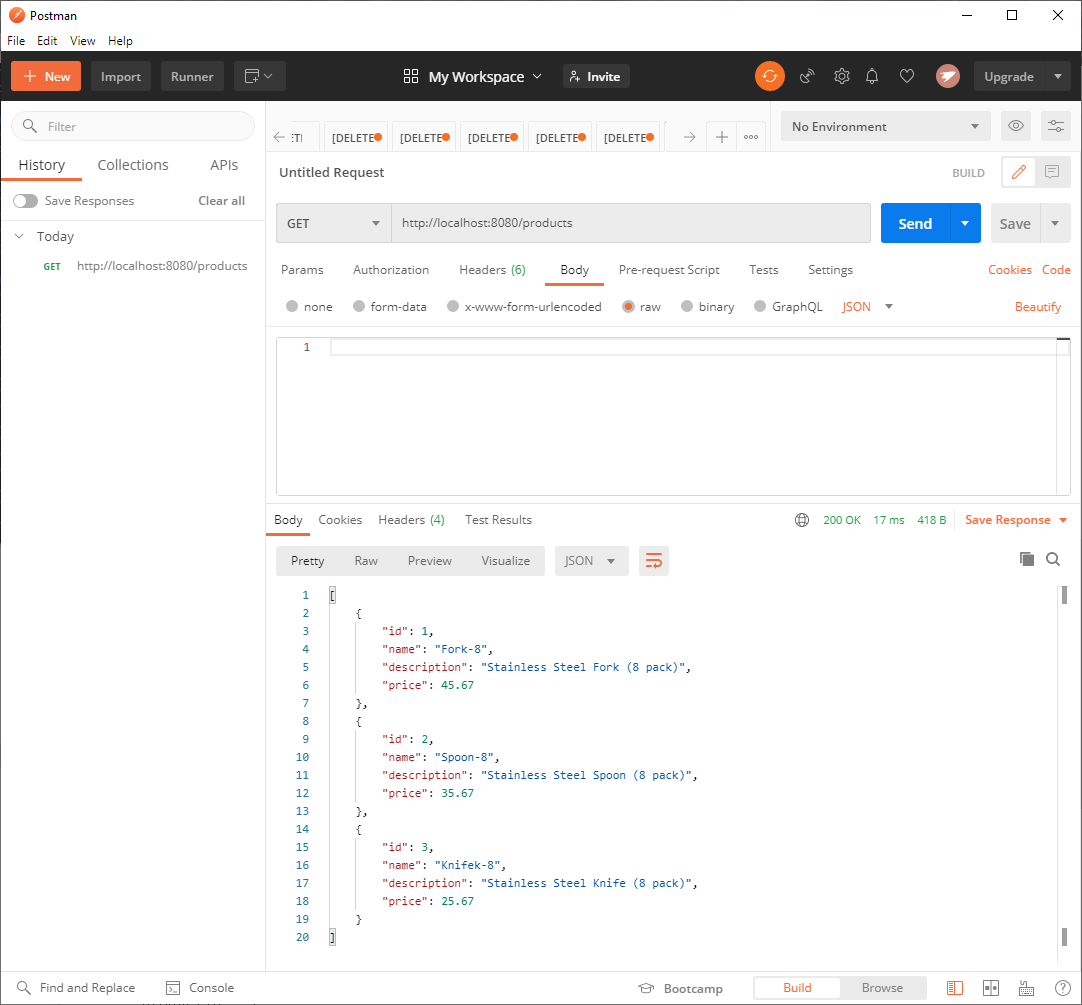
Adding (CREATE) products using POST command: [Endpoint]: products



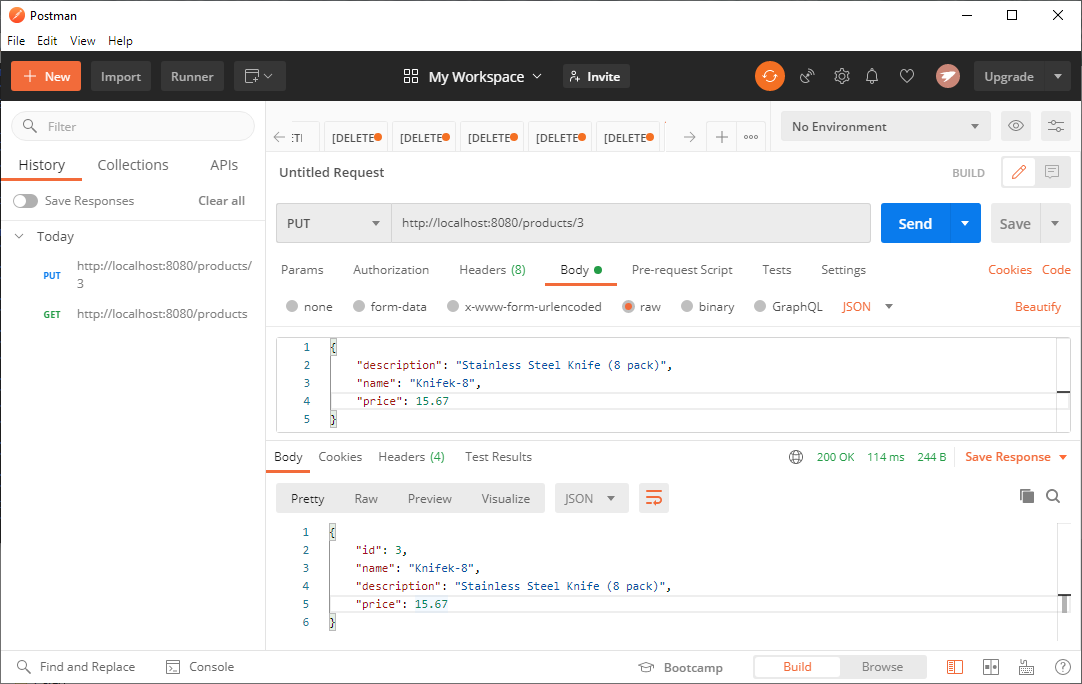
GET (READ) command from products (after product 1 is added): [Endpoint]: products



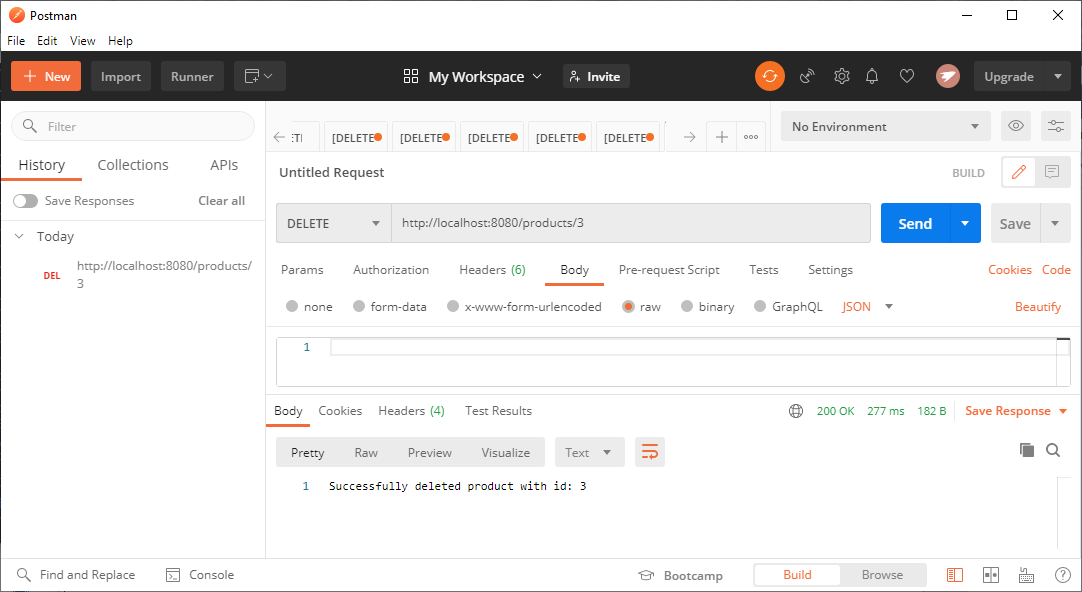
GET (READ) command from products (total of 3 added): [Endpoint]: products



PUT (UPDATE) command from products (update product id #3): [Endpoint]: products

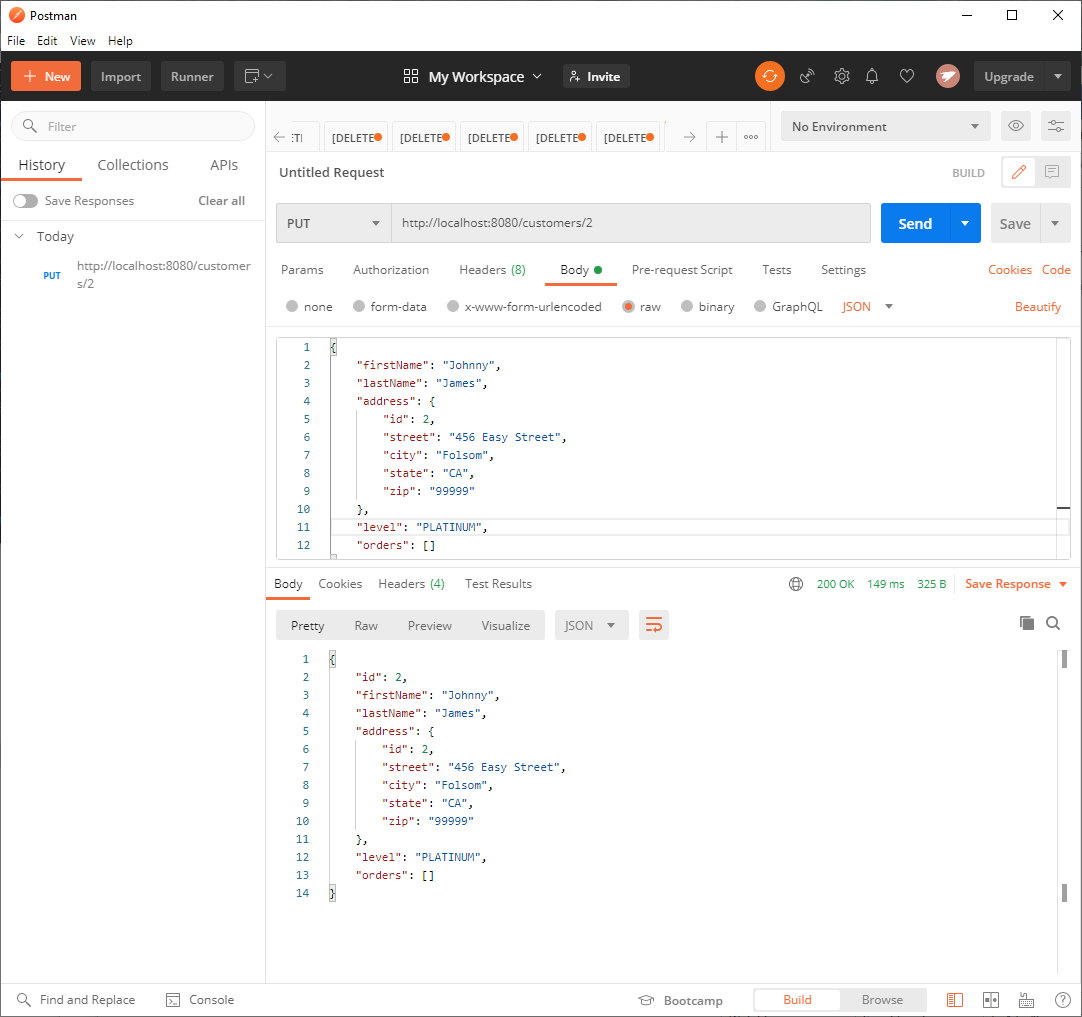


DELETE (DELETE) command from products (delete product id #3): [Endpoint]: products/3



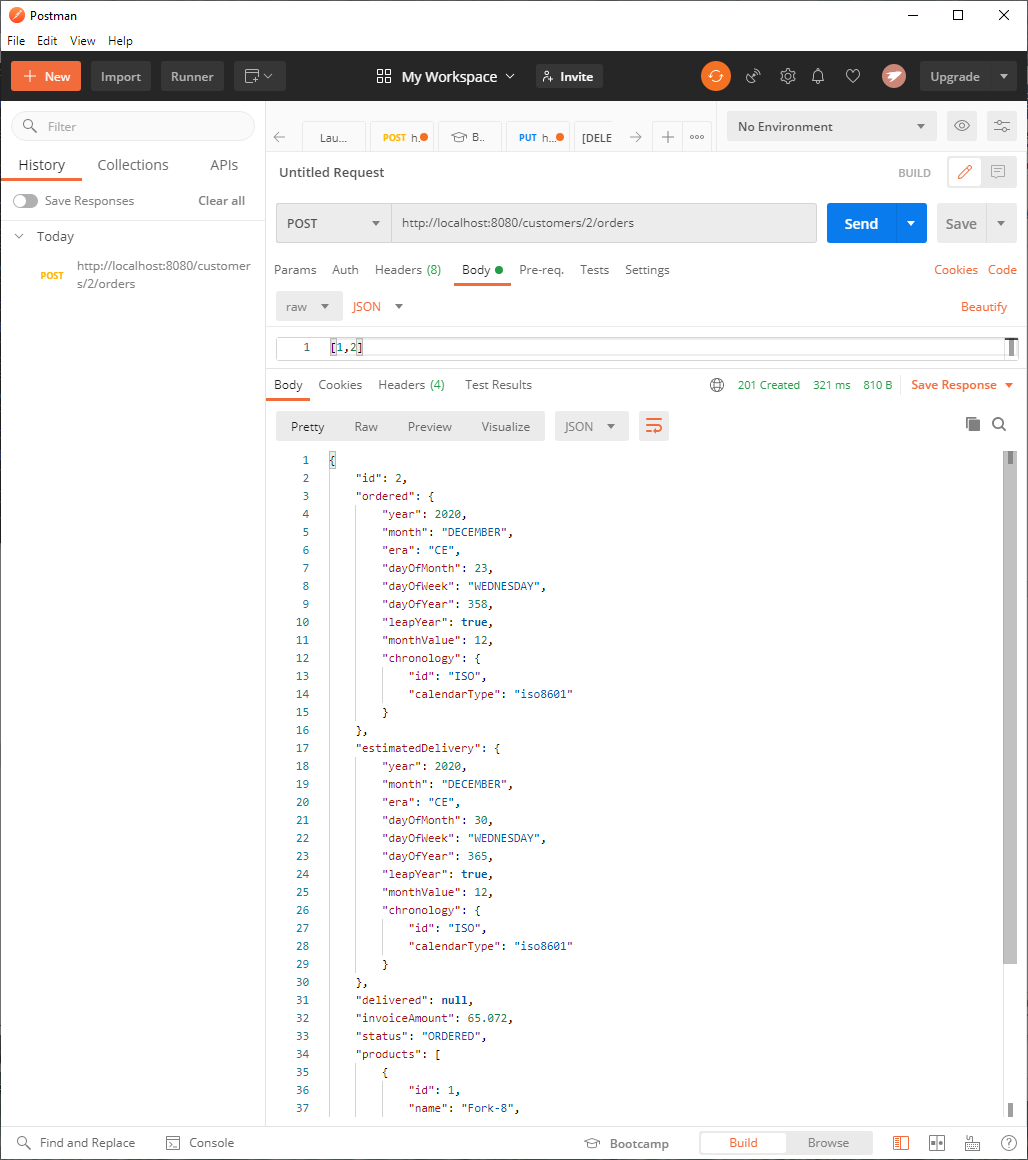
PUT (UPDATE) command to establish Membership Level (add “PLATINUM” level to customer id #2):

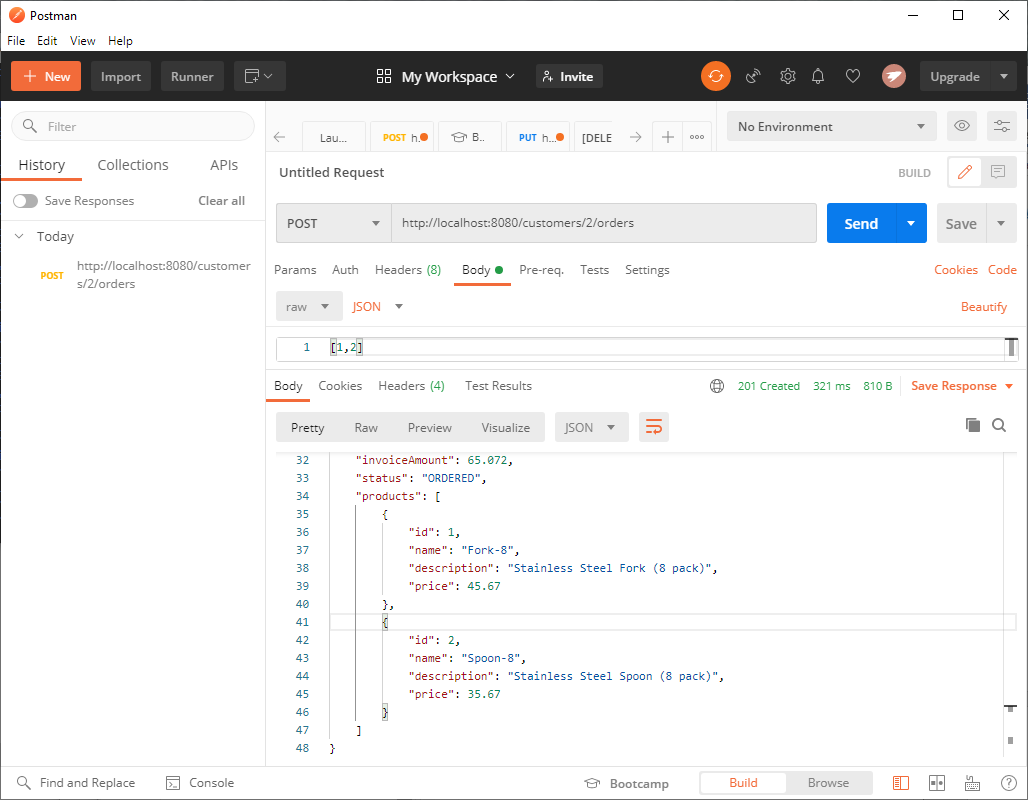
[Endpoint]: customers/2



POST(CREATE) command to add new order (add order to customer id #2):

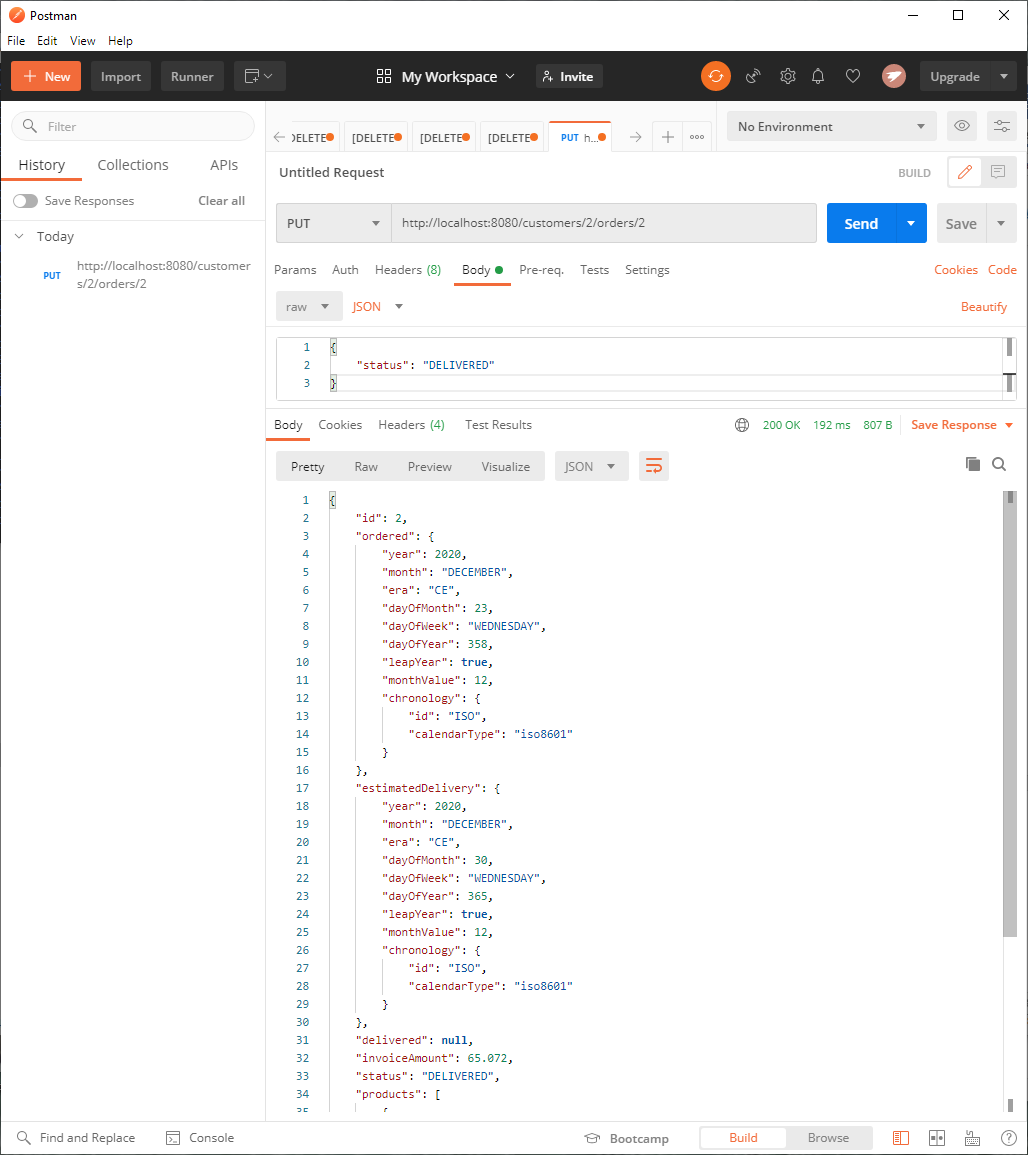
[Endpoint]: customers/2/orders (add product id 1 & 2)





PUT(UPDATE) command to update order (update order status to customer id #2 as “DELIVERED”):

[Endpoint]: customers/2/orders/2 (update status to “DELIVERED”)



**URL to GitHub Repository:**

[**https://github.com/srikripa/SB-W2-Repository.git**](https://github.com/srikripa/SB-W2-Repository.git)