**Micro services** **– L2: Hands-on Assignments**

**Estimated Efforts: 8 PDs**

|  |  |
| --- | --- |
| **Created By** | **Reviewed By** |
| **Bijulal G S** | **Savitha Sridhar** |

**Author: Bijulal.s63@wipro.com**

**Top Gear VDI:** Spring, SOA, Hibernate

**Date: 26 JUNE 2020**

**TOC:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic No** | **Topic Name** | **Sub Topics** | **Min No of Assignments to be Done** |
| 1 | Spring Cloud Eureka | What is Eureka, What is micro services phone book concept, how spring cloud eureka avoids hard coding ports and supports in Load Balancing | 1 |
| 2 | Spring cloud Gateway by using NETFLIX ZUUL | Setting up gateway with ZUUL, enabling micro services programmatic communication, avoiding hard coding | 3 |
| 3 | Netflix Open Feign | Setting declarative micro services Communication | 1 |
| 4 | Using Hystrix | Setting up application tolerant ,setting up hystrix dashboard, setting circuit breaker pattern | 1 |
| 5 | SLEUTH and ZIPKIN for distributed tracing | Setting SLEUTH, Setting LOGGING , Setting ZIPKIN | 1 |
| 6 | Micro service security | Default Security, Storing user credentials in DB | 1 |
| **Total Min No of Assignments to be Done** | | | **8** |

**Topic 1: Spring cloud - EUREKA**

**Assignment 1:**

Create EUREKA based registry that will register in port no 8761 also add a new service into this registry. Try with spring initializer

**Assignment 2:**

Create EUREKA based registry that will register in port no 8761 also add a new service into this registry. Try with Spring CLI (Command Line Interface)

**Topic 2: Spring cloud gateway by using NETFLIX ZUUL**

**Assignment 1:**

Create a new Micro service by using spring data JPA to connect with a Database and create CRUD service, connect this service with registry and create ZUUL routing

Gateway

Registry Service

CRUDSERVER

Database

CRUDSERVER exposes web services for all other micro services, all other micro services uses those exposed services

Deliverable needed here:

1. Gateway
2. Registry
3. CRUDSERVER with spring data JPA integration

**Assignment 2:**

Modify above assinment1 of Topic2 by adding two more services Customer Service and payment service that share services from CRUDSEVER, here you can use hard coding of Services and IP address can use

Find below diagram and requirements

Registry Service

Customer Service

Database

Payment Service

CRUDSERVER

Gateway

Deliverable needed here

1. CRUDSERVER functionalities are below at this point

Should have four web services as below

|  |  |  |
| --- | --- | --- |
| Type | Schema/Model/Description | Response |
| POST | customerID(Auto generated)  customerName  customerType(platinum,gold,silver)  Address 1  Address 2  City  State/Province  Postal Code  Country  E-Mail  Phone  Fax | **201 Created** |
| GET | GET all customers | **200 OK** |
| GET | GET a customer by ID | **200 OK** |
| PUT | Edit a customer with Customer payload | **200 ok** |
| DELETE | Delete a customer | **200 ok** |
|  |  |  |

Newly created Customer micro service will communicate to CRUDSERVER for its CRUD activities, here you can use hard coding of IP

1. CRUDSERVER functionalities are below at this point

Should have another four web services as below for payments

|  |  |  |
| --- | --- | --- |
| type | Schema/Model/Description | Response |
| POST | paymentID  Amount  creditType(IMPS,NEFT,direct)  description  customerID(one to many payments, a customer can have multiple payments) | **201 Created** |
| GET | GET all payments | **200 OK** |
| GET | GET a payment | **200 OK** |
| PUT | Edit a payment with payment payload | **200 ok** |
| DELETE | Delete a payment | **200 ok** |
| GET | Get all payments by a customer id | **200 ok** |

**Assignment 3:**

Modify above project by adding a Client Side Load Balancer with Netflix ribbon, avoid all hard coding of services and ports

**Topic 3: Setting declarative micro services Communication**

**Assignment 1:**

Modify assignment2 topic 2, use declarative way of communicating with services, avoid using Rest Template, use open feign implementation

**Topic 4: Using Hystrix**

**Assignment 1:**

Modify topic3 assignment1 by adding hystrix features, by adding circuit breaker feature

**Assignment 2:**

Modify application by adding a new hystrix Dashboard feature

**Topic 5: SLEUTH and ZIPKIN for distributed tracing**

**Assignment 1:**

Centralize traces by adding SLEUTH and setup ZIPKIN for analyzing traces for all services

**Topic 6: Micro service security**

**Assignment 1:**

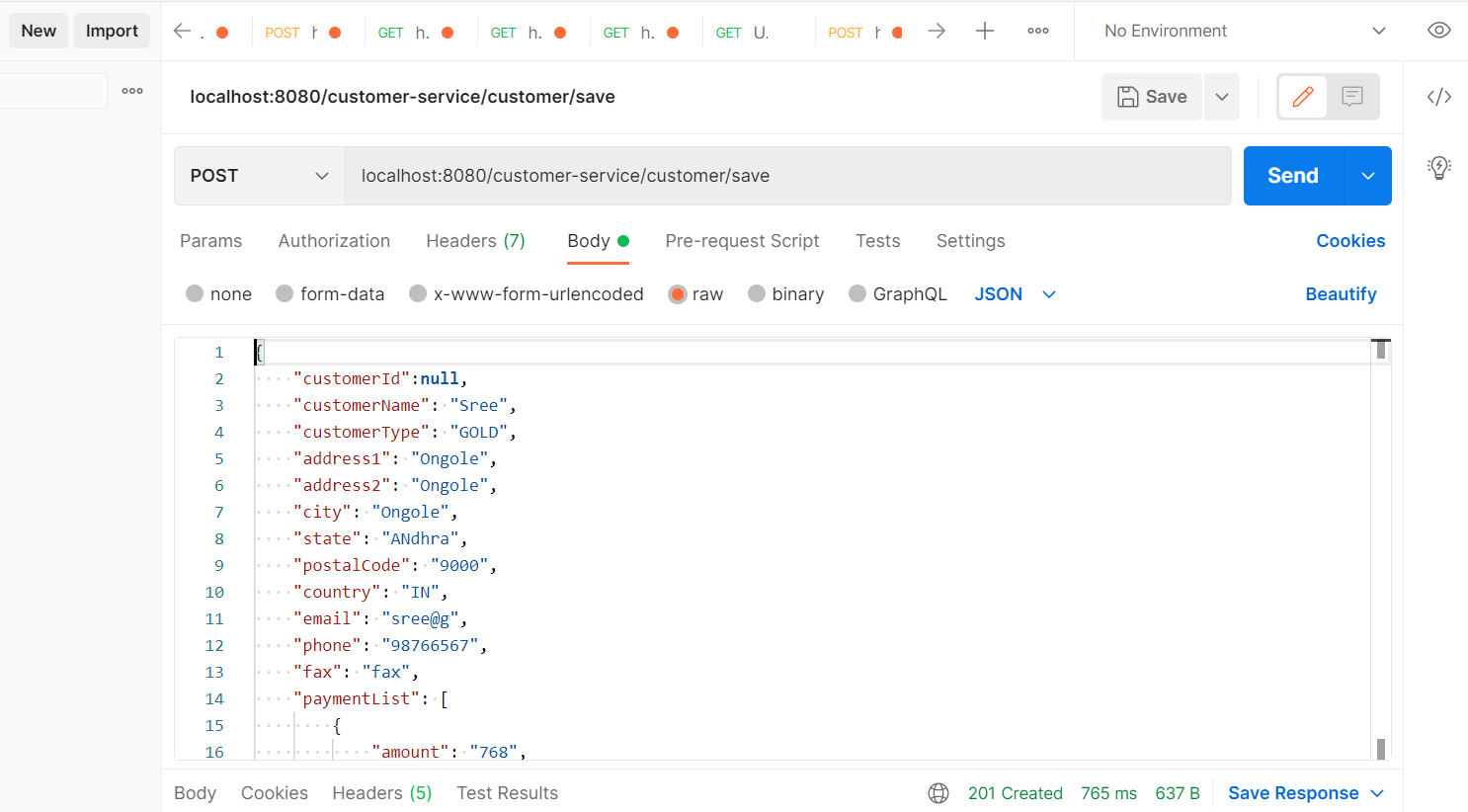
Create a new micro service and set default basic Authentication and change its username and password by hard coded values?

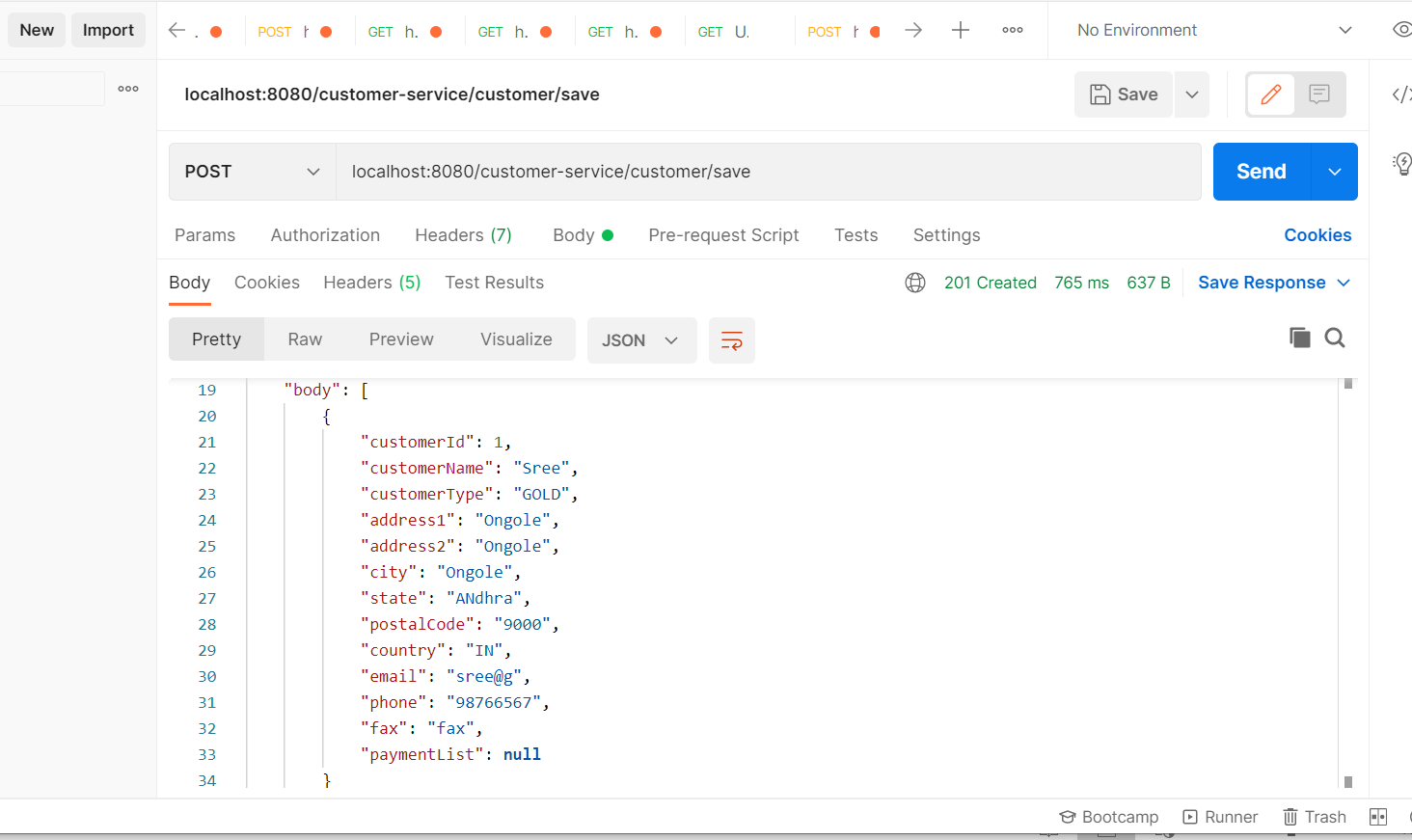
Functionality can be anything like a hello world JSON response

**Assignment 2:**

Create a new micro service and make basic authentication. User name and password should be stored in a database

Functionality can be anything like a hello world JSON response





{

"headers": {

"connection": [

"keep-alive"

],

"content-type": [

"application/json"

],

"date": [

"Mon, 02 Aug 2021 05:08:40 GMT"

],

"keep-alive": [

"timeout=60"

],

"transfer-encoding": [

"chunked"

]

},

"body": [

{

"customerId": 1,

"customerName": "Sree",

"customerType": "GOLD",

"address1": "Ongole",

"address2": "Ongole",

"city": "Ongole",

"state": "ANdhra",

"postalCode": "9000",

"country": "IN",

"email": "sree@g",

"phone": "98766567",

"fax": "fax",

"paymentList": [

{

"paymentId": 1,

"amount": 768,

"creditType": "Cash",

"description": "dfrtyuuii",

"customerDto": null

}

]

},

{

"customerId": 2,

"customerName": "Sree",

"customerType": "GOLD",

"address1": "Ongole",

"address2": "Ongole",

"city": "Ongole",

"state": "ANdhra",

"postalCode": "9000",

"country": "IN",

"email": "sree@g",

"phone": "98766567",

"fax": "fax",

"paymentList": []

}

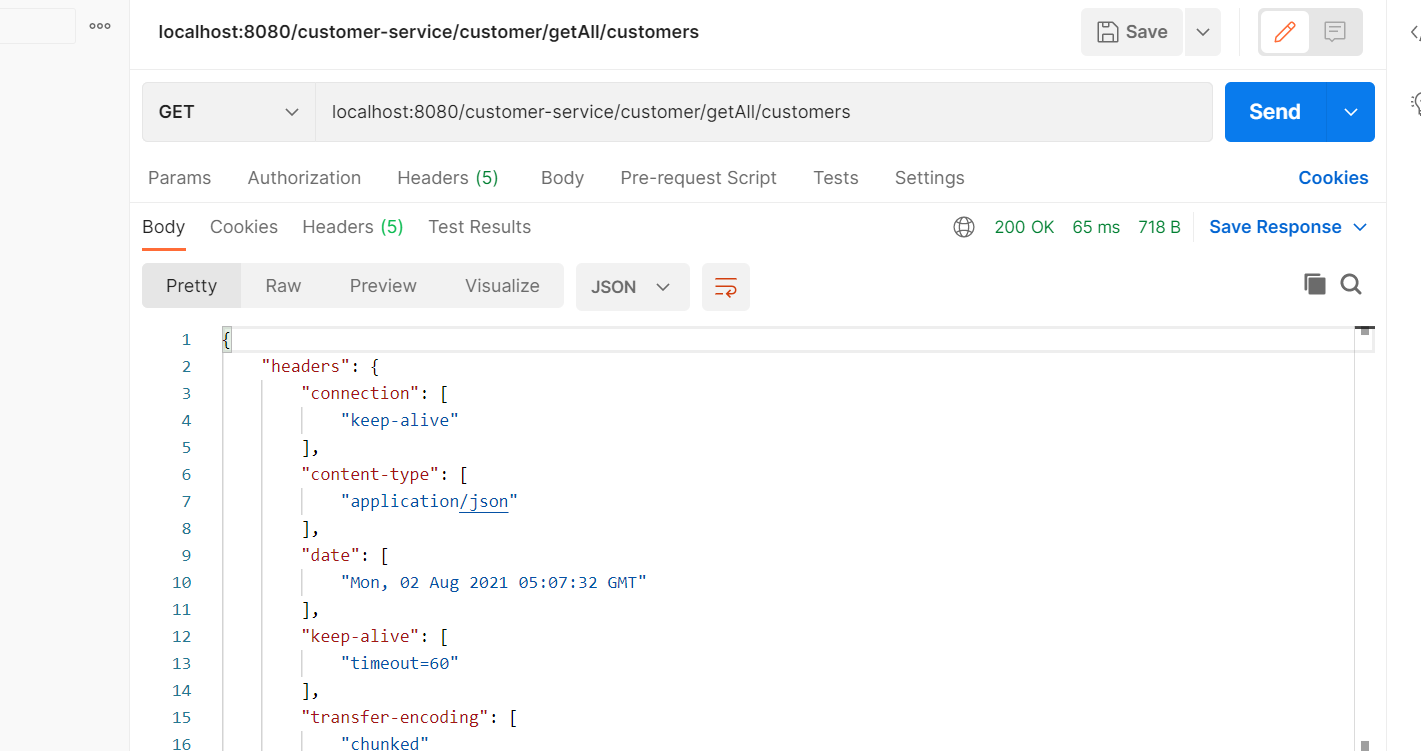
],

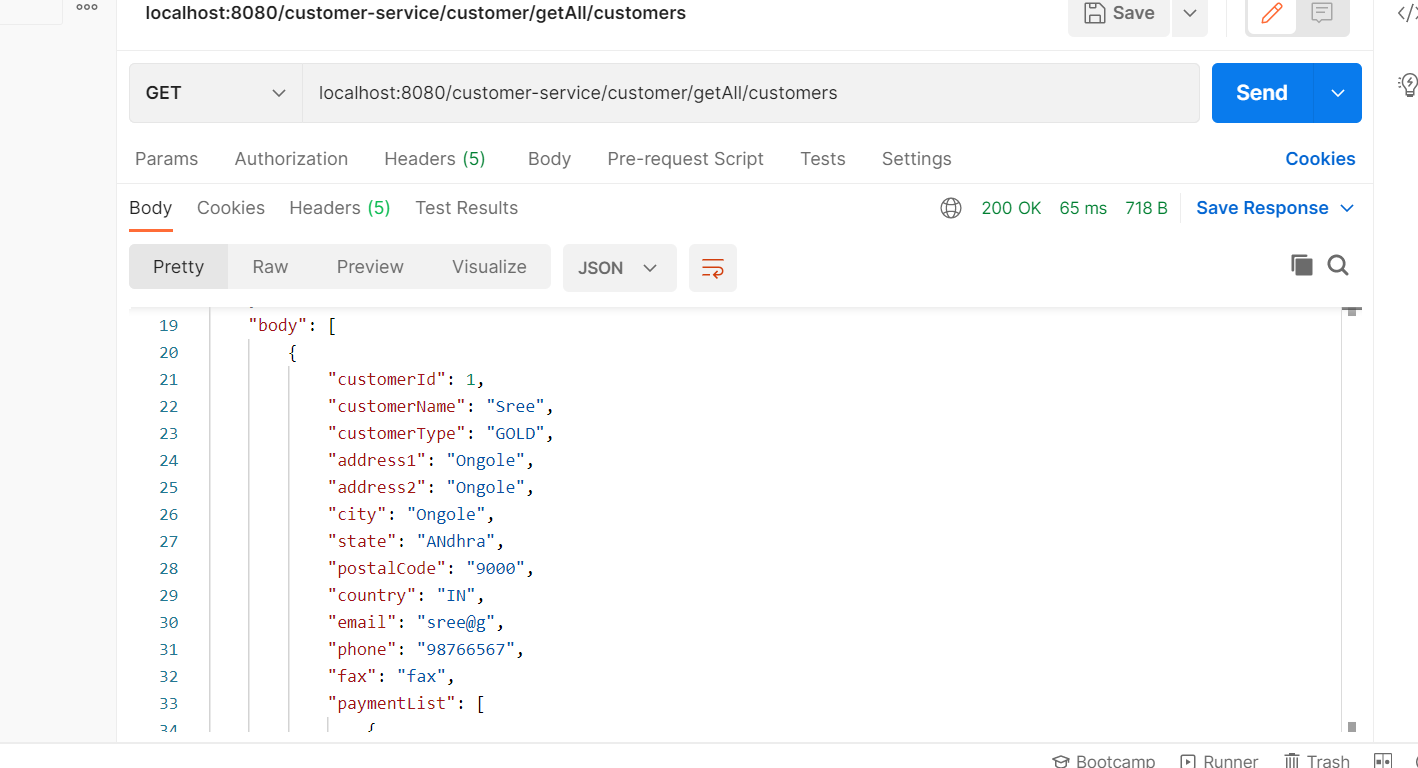
"statusCode": "CREATED",

"statusCodeValue": 201

}

**Get All Customers :**





{

"headers": {

"connection": [

"keep-alive"

],

"content-type": [

"application/json"

],

"date": [

"Mon, 02 Aug 2021 05:07:32 GMT"

],

"keep-alive": [

"timeout=60"

],

"transfer-encoding": [

"chunked"

]

},

"body": [

{

"customerId": 1,

"customerName": "Sree",

"customerType": "GOLD",

"address1": "Ongole",

"address2": "Ongole",

"city": "Ongole",

"state": "ANdhra",

"postalCode": "9000",

"country": "IN",

"email": "sree@g",

"phone": "98766567",

"fax": "fax",

"paymentList": [

{

"paymentId": 1,

"amount": 768,

"creditType": "Cash",

"description": "dfrtyuuii",

"customerDto": null

}

]

}

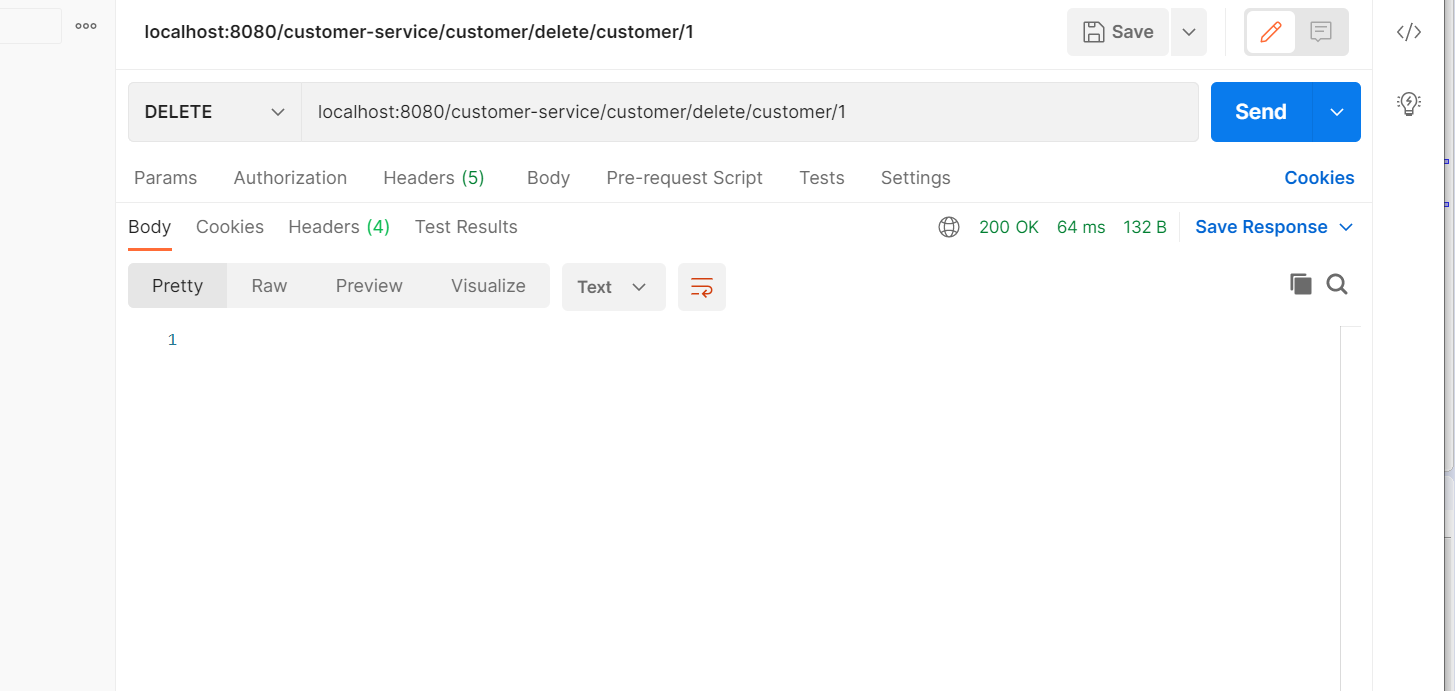
],

"statusCode": "OK",

"statusCodeValue": 200

}

**3.Delete customer by ID**



**4. update customer:**

