

## JPA Entity:

```
package com.customer.model;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
```

@Entity

@Data

@AllArgsConstructor

@NoArgsConstructor

public class Customer {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String address;

}

## ***CustomerRepository* Interface:**

```
package com.customer.repository;
```

```
import com.customer.model.Customer;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
```

```
@Repository
```

```
public interface CustomerRepository
    extends JpaRepository<Customer, Long> {
}
```

### **CustomerAlreadyExistsException:**

```
package com.customer.exception;
```

```
public class CustomerAlreadyExistsException
    extends RuntimeException {

    private String message;

    public CustomerAlreadyExistsException() {}

    public CustomerAlreadyExistsException(String msg)
    {
        super(msg);
        this.message = msg;
    }
}
```

### **NoSuchCustomerExistsException:**

```
package com.customer.exception;
```

```
public class NoSuchCustomerExistsException
    extends RuntimeException {

    private String message;

    public NoSuchCustomerExistsException() {}

    public NoSuchCustomerExistsException(String msg)
    {
        super(msg);
        this.message = msg;
    }
}
```

### ***CustomerService:***

```
package com.customer.service;
import com.customer.model.Customer;

public interface CustomerService {

    // Method to get customer by its Id
    Customer getCustomer(Long id);

    // Method to add a new Customer
    // into the database
    String addCustomer(Customer customer);
}
```

```

        // Method to update details of a Customer
        String updateCustomer(Customer customer);
    }

// Step 4: Service class Implementation
// which defines the main logic
package com.customer.service;

// Importing required packages
import com.customer.exception.CustomerAlreadyExistsException;
import com.customer.exception.NoSuchCustomerExistsException;
import com.customer.model.Customer;
import com.customer.repository.CustomerRepository;
import java.util.NoSuchElementException;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class CustomerServiceImpl
    implements CustomerService {

    @Autowired
    private CustomerRepository customerRepository;

    // Method to get customer by Id.Throws
    // NoSuchElementException for invalid Id

```

```

public Customer getCustomer(Long id)
{
    return customerRespository.findById(id).orElseThrow(
        ()
        -> new NoSuchElementException(
            "NO CUSTOMER PRESENT WITH ID = " + id));
}

```

// Method to add new customer details to database.Throws  
 // CustomerAlreadyExistsException when customer detail  
 // already exist

```

public String addCustomer(Customer customer)
{
    Customer existingCustomer
        = customerRespository.findById(customer.getId())
        .orElse(null);
    if (existingCustomer == null) {
        customerRespository.save(customer);
        return "Customer added successfully";
    }
    else
        throw new CustomerAlreadyExistsException(
            "Customer already exists!!");
}

```

```

public String updateCustomer(Customer customer)
{
    Customer existingCustomer
        = customerRespository.findById(customer.getId())
            .orElse(null);
    if (existingCustomer == null)
        throw new NoSuchCustomerExistsException(
            "No Such Customer exists!!");
    else {
        existingCustomer.setName(customer.getName());
        existingCustomer.setAddress(
            customer.getAddress());
        customerRespository.save(existingCustomer);
        return "Record updated Successfully";
    }
}
}

```

### ***CustomerController:***

// Step 5: Creating Rest Controller CustomerController which

// defines various API's.

```
package com.customer.controller;
```

// Importing required packages

```
import com.customer.exception.CustomerAlreadyExistsException;
```

```
import com.customer.exception.ErrorResponse;
```

```
import com.customer.model.Customer;
```

```
import com.customer.service.CustomerService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseStatus;
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
```

```
public class CustomerController {
```

```
    @Autowired private CustomerService customerService;
```

```
    // Get Customer by Id
```

```
    @GetMapping("/getCustomer/{id}")
```

```
    public Customer getCustomer(@PathVariable("id") Long id)
```

```
    {
```

```
        return customerService.getCustomer(id);
```

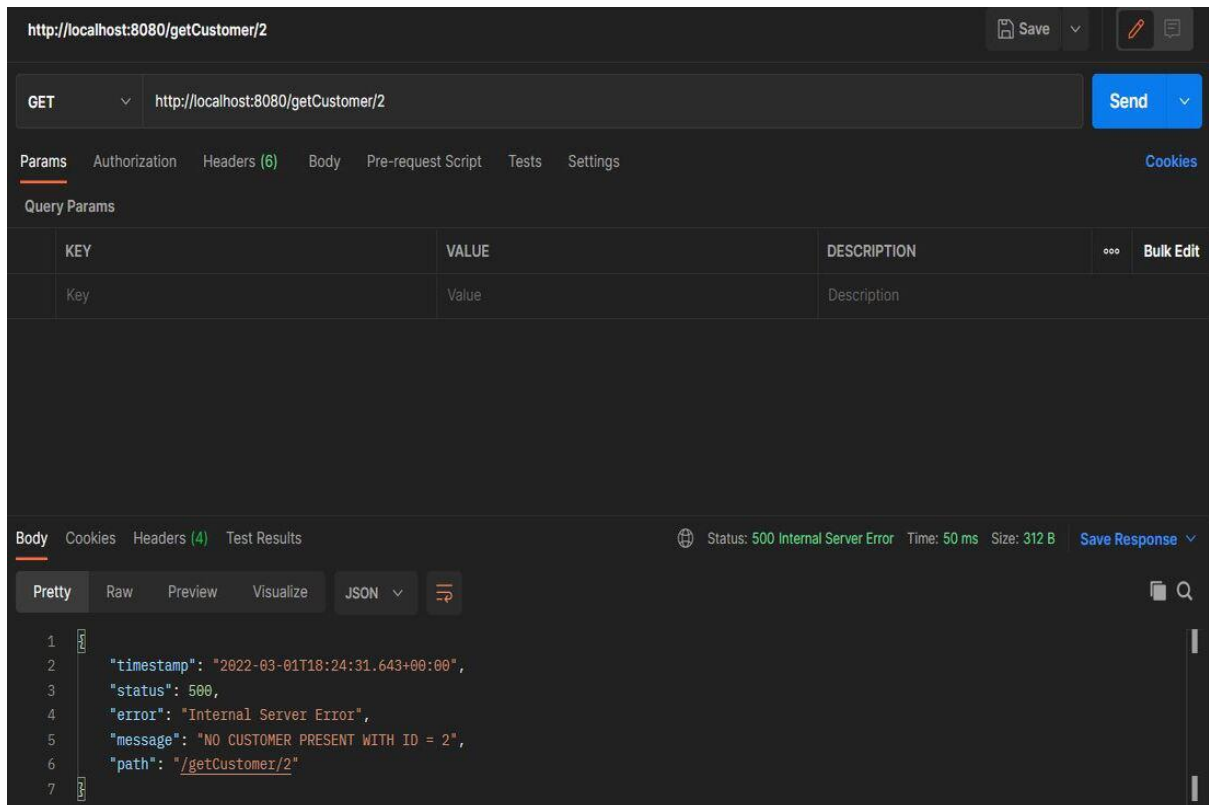
```
    }
```

```
    // Add new Customer
```

```
@PostMapping("/addCustomer")
public String
addcustomer(@RequestBody Customer customer)
{
    return customerService.addCustomer(customer);
}

// Update Customer details
@PutMapping("/updateCustomer")
public String
updateCustomer(@RequestBody Customer customer)
{
    return customerService.updateCustomer(customer);
}
}
```





### ***ErrorResponse :***

```
// Custom Error Response Class
```

```
package com.customer.exception;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Data;
```

```
import lombok.NoArgsConstructor;
```

```
@Data
```

```
@AllArgsConstructor
```

```
@NoArgsConstructor
```

```
public class ErrorResponse {
```

```

        private int statusCode;

        private String message;

        public ErrorResponse(String message)
        {
            super();
            this.message = message;
        }
    }

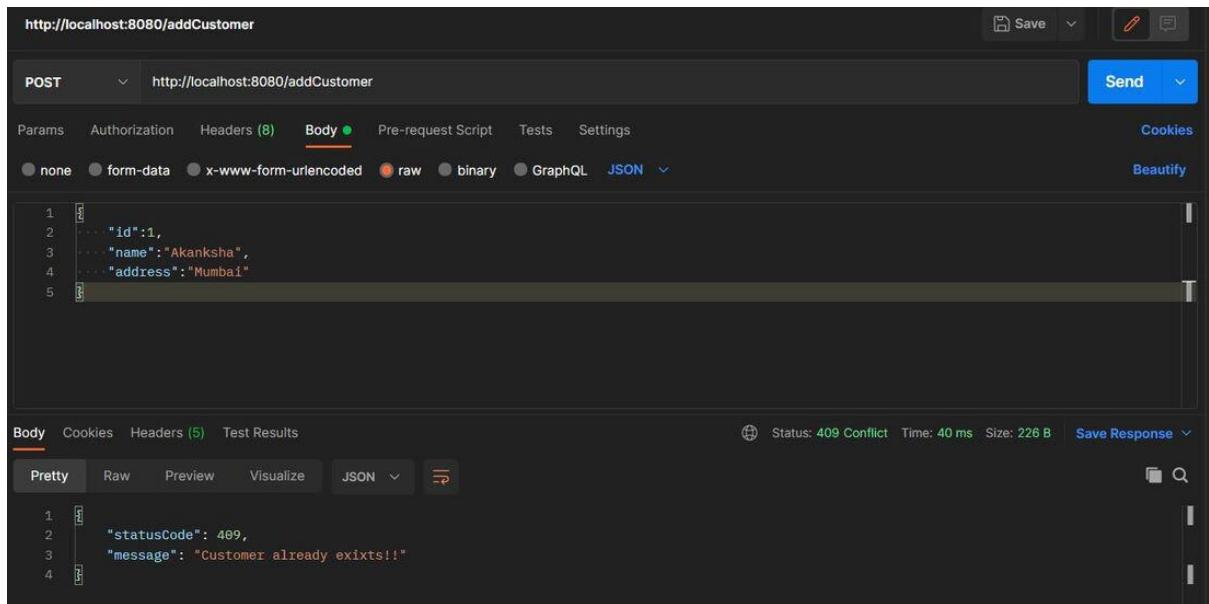
```

#### **CustomerAlreadyExistsException:**

```

// Exception Handler method added in CustomerController to
// handle CustomerAlreadyExistsException exception
@ExceptionHandler(value
                    = CustomerAlreadyExistsException.class)
@ResponseStatus(HttpStatus.CONFLICT)
public ErrorResponse
handleCustomerAlreadyExistsException(
    CustomerAlreadyExistsException ex)
{
    return new ErrorResponse(HttpStatus.CONFLICT.value(),
                            ex.getMessage());
}

```



## Global Exception Handler:

// Class to handle exception Globally

```
package com.customer.exception;
```

```
import org.springframework.http.HttpStatus;
```

```
import org.springframework.web.bind.annotation.ControllerAdvice;
```

```
import org.springframework.web.bind.annotation.ExceptionHandler;
```

```
import org.springframework.web.bind.annotation.ResponseBody;
```

```
import org.springframework.web.bind.annotation.ResponseStatus;
```

```
@ControllerAdvice
```

```
public class GlobalExceptionHandler {
```

```
    @ExceptionHandler(value
```

```
        = NoSuchCustomerExistsException.class)
```

```
    @ResponseStatus(HttpStatus.BAD_REQUEST)
```

```

public @ResponseBody ErrorResponse
handleException(NoSuchCustomerExistsException ex)
{
    return new ErrorResponse(
        HttpStatus.NOT_FOUND.value(), ex.getMessage());
}
}

```

