1. Customerservice.java

package com.tns.Customerservice;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class Customerservice

{

public static void main(String[] args)

{

SpringApplication.run(Customerservice.class, args);

}

}

2.Customer.java

package com.tns.Customerservice;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Customer

{

private Integer C\_id;

private String C\_name;

public Customer()

{

super();

}

public Customer(Integer c\_id, String c\_name)

{

super();

c\_id = c\_id;

c\_name = c\_name;

}

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

public Integer getS\_id() {

return c\_id;

}

public void setS\_id(Integer c\_id) {

c\_id = c\_id;

}

public String getS\_name() {

return c\_name;

}

public void setS\_name(String c\_name)

{

c\_name =c\_name;

}

@Override

public String toString()

{

return "Customer[Customer id:"+c\_id+" Customer name"+c\_name+"]";

}

}

3.Customer\_Service\_Repository.java (

package co\_c.tns.Custommerservice;

import org.springframework.data.jpa.repository.JpaRepository;

public interface Customer\_Service\_Repository extends JpaRepository<Customer, Integer>

{

}

4. Customer\_Service.java

package com.tns.Customerservice;

import java.util.List;

import javax.transaction.Transactional;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

@Service

@Transactional

public class customer\_Service

{

@Autowired

private customer\_Service\_Repository repo;

public List<Customer> listAll()

{

return repo.findAll();

}

public void save(Customer cus)

{

repo.save(cus);

}

public Customer get(Integer c\_id)

{

return repo.findById( c\_id).get();

}

public void delete(Integer c\_id)

{

repo.deleteById( c\_id);

}

}

5. Customer\_service\_Controller.java

package com.tns.Customerservice;

import javax.persistence.NoResultException;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.DeleteMapping;

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.RestController;

@RestController

public class Customer\_service\_Controller

{

@Autowired(required=true)

private Customer\_Service service;

@GetMapping("/customerservice")

public java.util.List<Customer> list()

{

return service.listAll();

}

@GetMapping("/customerservice/{s\_id}")

public ResponseEnPathVCustomer> get(@PathVariable Integer s\_id)

{

try

{

Customer cus=service.get(c\_id);

return new ResponseEntity<Customer>(cus,HttpStatus.OK);

}

catch(NoResultException e)

{

return new ResponseEntity<customer>(HttpStatus.NOT\_FOUND);

}

}

@PostMapping("/customerservice")

public void add(@RequestBody Customer cus)

{

service.save(cus);

}

@PutMapping("/customerservice/{c\_id}")

public ResponseEntity<?> update(@RequestBody Customer cus, @PathVariable Integer C\_id)

{

Student existstud=service.get(c\_id);

service.save(existstud);

return new ResponseEntity<>(HttpStatus.OK);

}

@DeleteMapping("/customerservice/{c\_id}")

public void delete(@PathVariable Integer c\_id)

{

service.delete(c\_id);

}

}

-------------------------------------------------------------------------------

Go to src/resource folder

-application.properties

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://localhost:3306/Customerservice

spring.datasource.username=root

spring.datasource.password=Fathima@123

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect

server.port=8080