1)Create a RESTful web service that returns "Hello World" message

#Main Class

package com.springrest.Springrest;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringRestApplication {

public static void main(String[] args) {

SpringApplication.run(SpringRestApplication.class, args);

}

}

#Class HelloController

package com.springrest.Springrest.controller;

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

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

@RestController

public class HelloController {

@RequestMapping("/india")

public String sayHello() {

return"hello india";

}

}

2 )Create a RESTful web service that authenticates anuser. User will specify his/her credentials i.e. username and password. If username and password are correct, It should return "valid user" message, else "Invalid user" message.

#Class UserController

package com.springrest.Springrest.Q2

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

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

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

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

@RestController

public class UserController {

@RequestMapping(value ="/user/user={username}/pass={password}", method = RequestMethod.GET)

public ResponseEntity<String>AuthUserget(@PathVariable("username") String uname,@PathVariable("passwprd") String pwd)

{

System.out.println(uname.toString());

System.out.println(pwd.toString());

if(uname.toString().equals("admin") && pwd.toString().equals("admin123")) {

return new ResponseEntity<String>("Valid User",HttpStatus.FOUND);

}

else {

return new ResponseEntity<String>("InValid User",HttpStatus.NOT\_FOUND);

}

}

}

3)Create a RESTful web service that returns state, city and country information when user passes zipcode. You can send state, city and country information in JSON format.Sample Input: 99501Sample output: {"state": "AK", City: "ANCHORAGE", "country: "US"}

#Class Address

**package** com.springrest.Springrest.Q3;

**public** **class** Address {

String zip;

String city;

String state;

String country;

**public** String getZip() {

**return** zip;

}

**public** **void** setZip(String zip) {

**this**.zip = zip;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getState() {

**return** state;

}

**public** **void** setState(String state) {

**this**.state = state;

}

**public** String getCountry() {

**return** country;

}

**public** **void** setCountry(String country) {

**this**.country = country;

}

}

#Class :AddressController

package com.springrest.Springrest.Q3;

import java.util.ArrayList;

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

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

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

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

@RestController

public class AddressController {

@RequestMapping(value = "/zip", method= RequestMethod.GET)

public ArrayList<Address> getStudentsList(){

Address zip1 = new Address();

zip1.setZip("552344");

zip1.setState("UtterPradesh");

zip1.setCity("Ayodhya");

zip1.setCountry("India");

Address zip2 = new Address();

zip2.setZip("271307");

zip2.setState("Allahabad");

zip2.setCity("Banglor");

zip2.setCountry("Grugram");

ArrayList<Address> ziplist = new ArrayList<Address>();

ziplist.add(zip1);

return ziplist;

}

@RequestMapping(value="/zip/{zip}", method=RequestMethod.GET)

public Address getZipCode(@PathVariable("zip") String zip) {

Address zip1=new Address();

zip1.setZip("552344");

zip1.setState("UtterPradesh");

zip1.setCity("Ayodhya");

zip1.setCountry("India");

return zip1;

}

}

4)Create a RESTful web service that validates the credit card. It means we need to check the type of credit card like American Express, Discover, Visa etc and it is valid or not.Valid length: 16 digits.First 6 digits must be in one of the following ranges:601100 through 601109601120 through 601149601174601177 through 601179601186 through 601199644000 through 659999enRouteValid length: 15 digits. First four digits must be 2014 or 2149.JCBValid length: 16 to 19 digits.First 4 digits must be in the range 3528 through 3589. MasterCardValid length: 16 digits.First digit must be 5 and second digit must be in the range 1 through 5 inclusive. The range is 510000 through 559999.First digit must be 2 and second digit mustbe in the range 2 through 7 inclusive. The range is 222100 through 272099

5)Develop RESTful web services for "Employee Management System" that manages the information about employees 1. Add anew employee2. Searching for specific employee3. Deleting an existing employee4. Finding all employees5. Editing/updating employee information.Create aEmployee domain model class having following properties: employeeId, employeeName, employeeDepartment, employeeDesignation, employeeSalary. Employee Id should be generated automatically at database level.Develop controller, service and repository layersclasses.Use CrudRepository from Spring Data

#Class:Employee.java

package com.springrest.Springrest.Q5;

import java.util.List;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class Employee {

@Id

int employeeId;

String employeeName;

String employeeDepartment;

String employeeDesignation;

int employeeSalary;

public Employee() {

}

public Employee(int employeeId, String employeeName, String employeeDepartment, String employeeDesignation,

int employeeSalary) {

super();

this.employeeId = employeeId;

this.employeeName = employeeName;

this.employeeDepartment = employeeDepartment;

this.employeeDesignation = employeeDesignation;

this.employeeSalary = employeeSalary;

}

public int getEmployeeId() {

return employeeId;

}

public void setEmployeeId(int employeeId) {

this.employeeId = employeeId;

}

public String getEmployeeName() {

return employeeName;

}

public void setEmployeeName(String employeeName) {

this.employeeName = employeeName;

}

public String getEmployeeDepartment() {

return employeeDepartment;

}

public void setEmployeeDepartment(String employeeDepartment) {

this.employeeDepartment = employeeDepartment;

}

public String getEmployeeDesignation() {

return employeeDesignation;

}

public void setEmployeeDesignation(String employeeDesignation) {

this.employeeDesignation = employeeDesignation;

}

public int getEmployeeSalary() {

return employeeSalary;

}

public void setEmployeeSalary(int employeeSalary) {

this.employeeSalary = employeeSalary;

}

public List<Employee> getAllEmp() {

// TODO Auto-generated method stub

return null;

}

}

#Class:EmployeeController

package com.springrest.Springrest.Q5;

import java.util.Arrays;

import java.util.List;

import java.util.Optional;

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

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

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

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

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

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

//import com.example.assign5.emp.\*;

@RestController

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@RequestMapping("/emp")

public List<Employee> getAllEmp(){

return employeeService.getAllEmp();

}

@RequestMapping("/emp/{employeeId}")

public Optional<Employee> getEmp(@PathVariable int employeeId) {

return employeeService.getEmp(employeeId);

}

@RequestMapping(method = RequestMethod.POST, value="/emp")

public void addEmp(@RequestBody Employee emp) {

employeeService.addEmp(emp);

}

@RequestMapping(method = RequestMethod.PUT, value="/emp/{employeeId}")

public void updateEmp(@RequestBody Employee emp, @PathVariable int employeeId) {

employeeService.updateEmp(employeeId,emp);

}

@RequestMapping(method = RequestMethod.DELETE, value="/emp/{employeeId}")

public void deleteEmp(@PathVariable int employeeId) {

employeeService.deleteEmp(employeeId);

}

}

#Interface:EmployeeRepository

**package** com.springrest.Springrest.Q5;

**import** org.springframework.data.repository.CrudRepository;

**public** **interface** EmployeeRepository **extends** CrudRepository<Employee, Integer>{

}

#Class:EmployeeService

package com.springrest.Springrest.Q5;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

/\*private List<Employee> emps = new ArrayList<>(Arrays.asList(

new Employee(12,"Rob","HR","Manager",30000),

new Employee(13,"Sam","Training","Trainee",15000),

new Employee(14,"Dan","IT","Developer",40000)

));\*/

// 4. finding all employees

public List<Employee> getAllEmp(){

//return emps;

List<Employee> emps = new ArrayList<>();

employeeRepository.findAll()

.forEach(emps::add);

return emps;

}

//2. Searching for specific employee

public Optional<Employee> getEmp(int employeeId) {

//return emps.stream().filter(e -> e.getEmployeeId() == (employeeId)).findFirst().get();

return employeeRepository.findById(employeeId);

}

//1. adding new employee

public void addEmp(Employee emp) {

//emps.add(emp);

employeeRepository.save(emp);

}

//5. update the employee information

public void updateEmp(int employeeId, Employee emp) {

/\* for(int i = 0; i < emps.size();i++) {

Employee e = emps.get(i);

if(e.getEmployeeId()==(employeeId)) {

emps.set(i, emp);

return;

}

}\*/

employeeRepository.save(emp);

}

// 3. delete the employee

public void deleteEmp(int employeeId) {

// emps.removeIf(e -> e.getEmployeeId() == (employeeId));

employeeRepository.deleteById(employeeId);;

}

}

6)Create a Calculator RESTful service that provides following functionality.1.Addition of the 2 numbers2.Subtraction of the 2 numbers3.Multiplication of 2 numbers4.Division of 2 numbers5.Finding square root of a number.Consume the above RESTful web service by using RestTemplate

7)Design and develop RESTful web service as follows:1.A user can place an order2.A user can update an order3.A user can view specific order4.A user can view all the orders5.A user can delete a specific order.Note: Use MongoRepository of Springdatato store order details

#Class:Orders

**package** com.control.SpringAssignment7.new1;

**import** org.springframework.data.annotation.Id;

**import** org.springframework.data.mongodb.core.mapping.Document;

**import** org.springframework.data.mongodb.core.mapping.Field;

@Document

**public** **class** Orders {

@Id

**private** String order\_id;

@Field

**private** String order\_name;

@Field

**private** String order\_price;

**public** Orders()

{

}

**public** Orders(String order\_name, String order\_price)

{

**this**.order\_name = order\_name;

**this**.order\_price = order\_price;

}

**public** String getOrder\_id()

{

**return** order\_id;

}

**public** **void** setOrder\_id(String order\_id)

{

**this**.order\_id = order\_id;

}

**public** String getOrder\_name()

{

**return** order\_name;

}

**public** **void** setOrder\_name(String order\_name)

{

**this**.order\_name = order\_name;

}

**public** String getOrder\_price()

{

**return** order\_price;

}

**public** **void** setOrder\_price(String order\_price)

{

**this**.order\_price = order\_price;

}

@Override

**public** String toString()

{

**return** String.*format*("Orders[order\_id='%s', order\_name = '%s', order\_price = '%s']", order\_id, order\_name,order\_price);

}

}

#Class:OrdersController

**package** com.control.SpringAssignment7.new1;

**import** java.util.List;

**import** java.util.Optional;

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

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.RequestBody;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.bind.annotation.RestController;

@RestController

**public** **class** OrdersController {

@Autowired

**private** OrdersService ordersService;

@RequestMapping("/orders")

**public** List<Orders> getAllOrders()

{

**return** ordersService.getAllOrders();

}

@RequestMapping("/orders/{order\_id}")

**public** Optional<Orders> getOrders(@PathVariable String order\_id)

{

**return** ordersService.getOrders(order\_id);

}

@RequestMapping(method = RequestMethod.***POST***, value = "/orders")

**public** **void** addOrders(@RequestBody Orders order)

{

ordersService.addOrders(order);

}

@RequestMapping(method = RequestMethod.***PUT***, value = "/orders/{order\_id}")

**public** **void** updateOrders(@RequestBody Orders order, @PathVariable String order\_id)

{

ordersService.updateOrders(order,order\_id);

}

@RequestMapping(method = RequestMethod.***DELETE***, value = "/orders/{order\_id}")

**public** **void** deleteOrders(@PathVariable String order\_id)

{

ordersService.deleteOrders(order\_id);

}

}

#Class:OrdersService

**package** com.control.SpringAssignment7.new1;

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Optional;

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

**import** org.springframework.stereotype.Service;

@Service

**public** **class** OrdersService {

@Autowired

**private** OrdersRepository ordersRepo;

**public** List<Orders> getAllOrders()

{

List<Orders> order = **new** ArrayList<>();

ordersRepo.findAll()

.forEach(order::add);

**return** order;

}

**public** Optional<Orders> getOrders(String order\_id)

{

**return** ordersRepo.findById(order\_id);

}

**public** **void** addOrders(Orders order)

{

ordersRepo.save(order);

}

**public** **void** updateOrders(Orders order, String order\_id) {

ordersRepo.save(order);

}

**public** **void** deleteOrders(String order\_id) {

ordersRepo.deleteById(order\_id);

}

}

#Interface:OrdersRepository

**package** com.control.SpringAssignment7.new1;

**import** org.springframework.data.mongodb.repository.MongoRepository;

**public** **interface** OrdersRepository **extends** MongoRepository<Orders, String>{

}

8)Design and develop RESTful web service as follows:1.An admin can add a new product.2.An admin can update details of existing product.3.An admin can delete existing product

4.An admin can view specific product5.An admin can view all the productsNote: Use MongoRepository of Springdatato store product details.

9)Design and develop RESTful web service as follows:1.Add a new customer information.2.Update customer information.3.Delete existing customer information.4.Fetch information of specific customer.5.Fetch information of all customers.Note: Use CrudRepository ofSpring Data to store customer details

#Class:Customer

package com.springrest.Springrest.Q9;

import javax.persistence.Entity;

import org.springframework.data.annotation.Id;

@Entity

public class Customer {

@Id

private String cust\_id;

private String cust\_name;

private String cust\_phone;

public Customer()

{

}

public Customer(String cust\_id, String cust\_name, String cust\_phone)

{

this.setCust\_id(cust\_id);

this.setCust\_name(cust\_name);

this.setCust\_phone(cust\_phone);

}

public String getCust\_id() {

return cust\_id;

}

public void setCust\_id(String cust\_id) {

this.cust\_id = cust\_id;

}

public String getCust\_name() {

return cust\_name;

}

public void setCust\_name(String cust\_name) {

this.cust\_name = cust\_name;

}

public String getCust\_phone() {

return cust\_phone;

}

public void setCust\_phone(String cust\_phone) {

this.cust\_phone = cust\_phone;

}

}

#Class:UserController

package com.springrest.Springrest.Q9;

import java.util.List;

import java.util.Optional;

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

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

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

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

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

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

@RestController

public class CustomerController {

@Autowired

private CustomerService customerService;

@RequestMapping("/cus")

public List<Customer> getAllCus(){

return customerService.getAllCus();

}

@RequestMapping("/cus/{cust\_id}")

public Optional<Customer> getCus(@PathVariable String cust\_id) {

return customerService.getCus(cust\_id);

}

@RequestMapping(method = RequestMethod.POST, value="/cus")

public void addCus(@RequestBody Customer cust) {

customerService.addCus(cust);

}

@RequestMapping(method = RequestMethod.PUT, value="/cus/{cust\_id}")

public void updateCus(@RequestBody Customer cust, @PathVariable String cust\_id) {

customerService.updateCus(cust\_id,cust);

}

@RequestMapping(method = RequestMethod.DELETE, value="/cus/{cust\_id}")

public void deleteCus(@PathVariable String cust\_id) {

customerService.deleteCus(cust\_id);

}

}

#Class:UserService

package com.springrest.Springrest.Q9;

import java.util.ArrayList;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

@Service

public class CustomerService {

@Autowired

private CustomerRepository customerRepository;

// 4. finding all employees

public List<Customer> getAllCus(){

List<Customer> custo = new ArrayList<>();

customerRepository.findAll()

.forEach(custo::add);

return custo;

}

//2. Searching for specific employee

public Optional<Customer> getCus(String cust\_id) {

return customerRepository.findById(cust\_id);

}

//1. adding new employee

public void addCus(Customer cus) {

customerRepository.save(cus);

}

//5. update the employee information

public void updateCus(String cust\_id, Customer cus) {

customerRepository.save(cus);

}

// 3. delete the employee

public void deleteCus(String cust\_id) {

customerRepository.deleteById(cust\_id);;

}

}

#interface:CustomerRepository

**package** com.springrest.Springrest.Q9;

**import** org.springframework.data.repository.CrudRepository;

**public** **interface** CustomerRepository **extends** CrudRepository<Customer, String>{

}

10)Test all the above RESTful web services by using MockMvc.