Spring Rest using Spring Boot 3:

Exercise 1: Create a Spring Web Project using Maven

//SpringLearnApplication.java

package com.cognizant;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

System.out.println("Starting SpringLearnApplication...");

SpringApplication.run(SpringLearnApplication.class, args);

System.out.println("SpringLearnApplication Started Successfully.");

}

}

//pom.xml

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

//pgsql

Tomcat started on port(s): 8080 (http) with context path ''

Started SpringLearnApplication in 2.345 seconds (JVM running for 2.987)

SpringLearnApplication Started Successfully.

Exercise 2: Display Employee List and Edit Employee form using RESTful Web Service In the previous angular module, we developed a screen that lists employees and it was populated with hard coded values. Now this angular application has be changed to get the data from RESTful Web Service developed in Spring:

//Employee.java

package com.cognizant.springlearn.model;

public class Employee {

private int id;

private String name;

private String department;

// Getters & setters

public int getId() { return id; }

public void setId(int id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String getDepartment() { return department; }

public void setDepartment(String department) { this.department = department; }

}

//employee.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="employeeList" class="java.util.ArrayList">

<constructor-arg>

<list>

<bean class="com.cognizant.springlearn.model.Employee">

<property name="id" value="1" />

<property name="name" value="John Doe" />

<property name="department" value="HR" />

</bean>

<bean class="com.cognizant.springlearn.model.Employee">

<property name="id" value="2" />

<property name="name" value="Jane Smith" />

<property name="department" value="Finance" />

</bean>

</list>

</constructor-arg>

</bean>

</beans>

//EmployeeServise.java

package com.cognizant.springlearn.service;

import java.util.List;

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

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

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.ImportResource;

import org.springframework.stereotype.Service;

import com.cognizant.springlearn.model.Employee;

@Service

@ImportResource("classpath:employee.xml")

public class EmployeeService {

@Autowired

ApplicationContext context;

@SuppressWarnings("unchecked")

public List<Employee> getAllEmployees() {

return (List<Employee>) context.getBean("employeeList");

}

public Employee getEmployeeById(int id) {

List<Employee> employees = getAllEmployees();

for (Employee emp : employees) {

if (emp.getId() == id) {

return emp;

}

}

return null;

}

}

//EmployeeController.java

package com.cognizant.springlearn.controller;

import java.util.List;

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

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

import com.cognizant.springlearn.model.Employee;

import com.cognizant.springlearn.service.EmployeeService;

@RestController

@RequestMapping("/employees")

@CrossOrigin(origins = "\*") // Allow Angular calls

public class EmployeeController {

@Autowired

EmployeeService employeeService;

@GetMapping

public List<Employee> getAllEmployees() {

return employeeService.getAllEmployees();

}

@GetMapping("/{id}")

public Employee getEmployeeById(@PathVariable int id) {

return employeeService.getEmployeeById(id);

}

}

//SpringLearnApplication.java

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

///employee.service.ts

import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http';

import { Observable } from 'rxjs';

@Injectable({

providedIn: 'root'

})

export class EmployeeService {

private baseUrl = 'http://localhost:8080/employees';

constructor(private http: HttpClient) { }

getEmployees(): Observable<any> {

return this.http.get(this.baseUrl);

}

getEmployeeById(id: number): Observable<any> {

return this.http.get(`${this.baseUrl}/${id}`);

}

}

//employee-list.component.ts

import { Component, OnInit } from '@angular/core';

import { EmployeeService } from '../employee.service';

import { Router } from '@angular/router';

@Component({

selector: 'app-employee-list',

template: `

<h2>Employee List</h2>

<table>

<tr \*ngFor="let emp of employees">

<td>{{emp.id}}</td>

<td>{{emp.name}}</td>

<td>{{emp.department}}</td>

<td><button (click)="editEmployee(emp.id)">Edit</button></td>

</tr>

</table>

`

})

export class EmployeeListComponent implements OnInit {

employees: any = [];

constructor(private employeeService: EmployeeService, private router: Router) { }

ngOnInit() {

this.employeeService.getEmployees().subscribe(data => {

this.employees = data;

});

}

editEmployee(id: number) {

this.router.navigate(['/edit', id]);

}

}

//edit-employee.component.ts

import { Component, OnInit } from '@angular/core';

import { ActivatedRoute } from '@angular/router';

import { EmployeeService } from '../employee.service';

@Component({

selector: 'app-edit-employee',

template: `

<h2>Edit Employee</h2>

<form \*ngIf="employee">

<label>ID: </label>{{employee.id}}<br>

<label>Name:</label>

<input [(ngModel)]="employee.name" name="name"><br>

<label>Department:</label>

<input [(ngModel)]="employee.department" name="department"><br>

<button>Save</button>

</form>

`

})

export class EditEmployeeComponent implements OnInit {

employee: any;

constructor(private route: ActivatedRoute, private employeeService: EmployeeService) { }

ngOnInit() {

const id = +this.route.snapshot.paramMap.get('id')!;

this.employeeService.getEmployeeById(id).subscribe(data => {

this.employee = data;

});

}

}

Output:

GET http://localhost:8080/employees

GET <http://localhost:8080/employees/1>

Exercise 3: Create static employee list data using spring xml configuration:

//employee.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<!-- Departments -->

<bean id="hrDept" class="com.cognizant.springlearn.model.Department">

<property name="id" value="1"/>

<property name="name" value="HR"/>

</bean>

<bean id="financeDept" class="com.cognizant.springlearn.model.Department">

<property name="id" value="2"/>

<property name="name" value="Finance"/>

</bean>

<bean id="techDept" class="com.cognizant.springlearn.model.Department">

<property name="id" value="3"/>

<property name="name" value="Technology"/>

</bean>

<!-- Skills -->

<bean id="javaSkill" class="com.cognizant.springlearn.model.Skill">

<property name="id" value="1"/>

<property name="name" value="Java"/>

</bean>

<bean id="angularSkill" class="com.cognizant.springlearn.model.Skill">

<property name="id" value="2"/>

<property name="name" value="Angular"/>

</bean>

<!-- Employees -->

<bean id="employee1" class="com.cognizant.springlearn.model.Employee">

<property name="id" value="1"/>

<property name="name" value="John Doe"/>

<property name="salary" value="50000"/>

<property name="department" ref="hrDept"/>

<property name="skills">

<list>

<ref bean="javaSkill"/>

</list>

</property>

</bean>

<bean id="employee2" class="com.cognizant.springlearn.model.Employee">

<property name="id" value="2"/>

<property name="name" value="Jane Smith"/>

<property name="salary" value="60000"/>

<property name="department" ref="financeDept"/>

<property name="skills">

<list>

<ref bean="angularSkill"/>

</list>

</property>

</bean>

<bean id="employee3" class="com.cognizant.springlearn.model.Employee">

<property name="id" value="3"/>

<property name="name" value="Alice Johnson"/>

<property name="salary" value="70000"/>

<property name="department" ref="techDept"/>

<property name="skills">

<list>

<ref bean="javaSkill"/>

<ref bean="angularSkill"/>

</list>

</property>

</bean>

<bean id="employee4" class="com.cognizant.springlearn.model.Employee">

<property name="id" value="4"/>

<property name="name" value="Bob Williams"/>

<property name="salary" value="55000"/>

<property name="department" ref="techDept"/>

<property name="skills">

<list>

<ref bean="javaSkill"/>

</list>

</property>

</bean>

<!-- Employee List -->

<bean id="employeeList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="employee1"/>

<ref bean="employee2"/>

<ref bean="employee3"/>

<ref bean="employee4"/>

</list>

</constructor-arg>

</bean>

</beans>

//EmployeeDao.java

package com.cognizant.springlearn.dao;

import java.util.ArrayList;

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

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.ImportResource;

import org.springframework.stereotype.Repository;

import com.cognizant.springlearn.model.Employee;

@Repository

@ImportResource("classpath:employee.xml")

public class EmployeeDao {

private static ArrayList<Employee> EMPLOYEE\_LIST;

@Autowired

public EmployeeDao(ApplicationContext context) {

// Read from XML config and assign to static list

EMPLOYEE\_LIST = (ArrayList<Employee>) context.getBean("employeeList");

}

public ArrayList<Employee> getAllEmployees() {

return EMPLOYEE\_LIST;

}

}

Output:

@Autowired

EmployeeDao employeeDao;

@GetMapping("/employees")

public List<Employee> getEmployees() {

return employeeDao.getAllEmployees();

}

API Output:

[

{ "id": 1, "name": "John Doe", "department": {...}, "skills": [...] },

...

]

Exercise 4: Create REST service to gets all employees:

//EmployeeService.java

package com.cognizant.springlearn.service;

import java.util.ArrayList;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.cognizant.springlearn.dao.EmployeeDao;

import com.cognizant.springlearn.model.Employee;

@Service //

public class EmployeeService {

@Autowired

private EmployeeDao employeeDao;

@Transactional //

public ArrayList<Employee> getAllEmployees() {

return employeeDao.getAllEmployees();

}

}

//EmployeeController.java

package com.cognizant.springlearn.controller;

import java.util.ArrayList;

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

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

import com.cognizant.springlearn.model.Employee;

import com.cognizant.springlearn.service.EmployeeService;

@RestController

@RequestMapping("/employees")

@CrossOrigin(origins = "\*") // Allow frontend calls

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@GetMapping

public ArrayList<Employee> getAllEmployees() {

return employeeService.getAllEmployees();

}

}

//scss

Tomcat started on port(s): 8080 (http) with context path ''

Started SpringLearnApplication...

//json response

[

{

"id": 1,

"name": "John Doe",

"salary": 50000,

"department": { "id": 1, "name": "HR" },

"skills": [

{ "id": 1, "name": "Java" }

]

},

{

"id": 2,

"name": "Jane Smith",

"salary": 60000,

"department": { "id": 2, "name": "Finance" },

"skills": [

{ "id": 2, "name": "Angular" }

]

},

...

]

Exercise 5: Create REST service for department. Create a new service to get all the departments.

//department.xml

<!-- Department beans -->

<bean id="hrDept" class="com.cognizant.springlearn.model.Department">

<property name="id" value="1"/>

<property name="name" value="HR"/>

</bean>

<bean id="financeDept" class="com.cognizant.springlearn.model.Department">

<property name="id" value="2"/>

<property name="name" value="Finance"/>

</bean>

<bean id="techDept" class="com.cognizant.springlearn.model.Department">

<property name="id" value="3"/>

<property name="name" value="Technology"/>

</bean>

<!-- Department list -->

<bean id="departmentList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="hrDept"/>

<ref bean="financeDept"/>

<ref bean="techDept"/>

</list>

</constructor-arg>

</bean>

//DepartmentDao.java

package com.cognizant.springlearn.dao;

import java.util.ArrayList;

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

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.ImportResource;

import org.springframework.stereotype.Repository;

import com.cognizant.springlearn.model.Department;

@Repository

@ImportResource("classpath:employee.xml") // Or department.xml if separate

public class DepartmentDao {

private static ArrayList<Department> DEPARTMENT\_LIST;

@Autowired

public DepartmentDao(ApplicationContext context) {

// Load department list from XML

DEPARTMENT\_LIST = (ArrayList<Department>) context.getBean("departmentList");

}

public ArrayList<Department> getAllDepartments() {

return DEPARTMENT\_LIST;

}

}

//DepartmentService.java

package com.cognizant.springlearn.service;

import java.util.ArrayList;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.cognizant.springlearn.dao.DepartmentDao;

import com.cognizant.springlearn.model.Department;

@Service

public class DepartmentService {

@Autowired

private DepartmentDao departmentDao;

@Transactional

public ArrayList<Department> getAllDepartments() {

System.out.println("Inside DepartmentService.getAllDepartments()");

return departmentDao.getAllDepartments();

}

}

//DepartmentController.java

package com.cognizant.springlearn.controller;

import java.util.ArrayList;

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

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

import com.cognizant.springlearn.model.Department;

import com.cognizant.springlearn.service.DepartmentService;

@RestController

@RequestMapping("/departments")

@CrossOrigin(origins = "\*")

public class DepartmentController {

@Autowired

private DepartmentService departmentService;

@GetMapping

public ArrayList<Department> getAllDepartments() {

System.out.println("Inside DepartmentController.getAllDepartments()");

return departmentService.getAllDepartments();

}

}

//Department.java

package com.cognizant.springlearn.model;

public class Department {

private int id;

private String name;

// Getters & Setters

}

//scss

Tomcat started on port(s): 8080 (http)

Started SpringLearnApplication...

Output:

[

{"id":1,"name":"HR"},

{"id":2,"name":"Finance"},

{"id":3,"name":"Technology"}

]

Exercise 6: Spring Core – Load Country from Spring Configuration XML An airlines website is going to support booking on four countries. There will be a drop down on the home page of this website to select the respective country. It is also important to store the two-character ISO code of each country

//country.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="country" class="com.cognizant.springlearn.model.Country">

<property name="code" value="IN"/>

<property name="name" value="India"/>

</bean>

</beans>

//Country.java

package com.cognizant.springlearn.model;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class Country {

private static final Logger LOGGER = LoggerFactory.getLogger(Country.class);

private String code;

private String name;

public Country() {

LOGGER.debug("Inside Country Constructor.");

}

public String getCode() {

LOGGER.debug("Inside getCode()");

return code;

}

public void setCode(String code) {

LOGGER.debug("Inside setCode()");

this.code = code;

}

public String getName() {

LOGGER.debug("Inside getName()");

return name;

}

public void setName(String name) {

LOGGER.debug("Inside setName()");

this.name = name;

}

@Override

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}

//SpringLearnApplication.java

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.cognizant.springlearn.model.Country;

@SpringBootApplication

public class SpringLearnApplication {

private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);

public static void displayCountry() {

LOGGER.debug("START");

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

Country country = context.getBean("country", Country.class);

LOGGER.debug("Country : {}", country.toString());

LOGGER.debug("END");

}

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

displayCountry();

}

}

//pgsql

DEBUG SpringLearnApplication: START

DEBUG Country: Inside Country Constructor.

DEBUG Country: Inside setCode()

DEBUG Country: Inside setName()

DEBUG Country: Inside getCode()

DEBUG Country: Inside getName()

DEBUG SpringLearnApplication: Country : Country [code=IN, name=India]

DEBUG SpringLearnApplication: END

//<bean>

<bean id="country" class="com.cognizant.springlearn.model.Country">

//<property>

<property name="code" value="IN"/>

//ApplicationContext.java

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

Exercise 7: Hello World RESTful Web Service:

//HelloController.java

package com.cognizant.springlearn.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

@RestController

public class HelloController {

private static final Logger LOGGER = LoggerFactory.getLogger(HelloController.class);

@GetMapping("/hello")

public String sayHello() {

LOGGER.info("START");

LOGGER.info("END");

return "Hello World!!";

}

}

//http://localhost:8083/hello

Date

Content-Type: text/plain;charset=UTF-8

Content-Length: 13

Connection: keep-alive

Exercise 8: REST - Country Web Service Write a REST service that returns India country details:

//CountryController.java

package com.cognizant.springlearn.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

import com.cognizant.springlearn.model.Country;

@RestController

public class CountryController {

private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START");

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

Country country = context.getBean("country", Country.class);

LOGGER.debug("Country : {}", country.toString());

LOGGER.info("END");

return country;

}

}

//json

{

"code": "IN",

"name": "India"

}

//xml

<bean id="country" class="com.cognizant.springlearn.model.Country">

<property name="code" value="IN"/>

<property name="name" value="India"/>

</bean>

Exercise 9: REST - Get country based on country code Write a REST service that returns a specific country based on country code:

//country.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<!-- Individual countries -->

<bean id="countryIN" class="com.cognizant.springlearn.model.Country">

<property name="code" value="IN"/>

<property name="name" value="India"/>

</bean>

<bean id="countryUS" class="com.cognizant.springlearn.model.Country">

<property name="code" value="US"/>

<property name="name" value="United States"/>

</bean>

<bean id="countryDE" class="com.cognizant.springlearn.model.Country">

<property name="code" value="DE"/>

<property name="name" value="Germany"/>

</bean>

<bean id="countryJP" class="com.cognizant.springlearn.model.Country">

<property name="code" value="JP"/>

<property name="name" value="Japan"/>

</bean>

<!-- Country list -->

<bean id="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="countryIN"/>

<ref bean="countryUS"/>

<ref bean="countryDE"/>

<ref bean="countryJP"/>

</list>

</constructor-arg>

</bean>

</beans>

//CountryDao.java

package com.cognizant.springlearn.dao;

import java.util.List;

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

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.ImportResource;

import org.springframework.stereotype.Repository;

import com.cognizant.springlearn.model.Country;

@Repository

@ImportResource("classpath:country.xml")

public class CountryDao {

private static List<Country> COUNTRY\_LIST;

@Autowired

public CountryDao(ApplicationContext context) {

COUNTRY\_LIST = (List<Country>) context.getBean("countryList");

}

public List<Country> getAllCountries() {

return COUNTRY\_LIST;

}

}

//CountryService.java

package com.cognizant.springlearn.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.cognizant.springlearn.dao.CountryDao;

import com.cognizant.springlearn.model.Country;

@Service

public class CountryService {

@Autowired

private CountryDao countryDao;

public Country getCountry(String code) {

List<Country> countryList = countryDao.getAllCountries();

// Using Lambda to match case-insensitive code

return countryList.stream()

.filter(c -> c.getCode().equalsIgnoreCase(code))

.findFirst()

.orElse(null);

}

}

//CountryController.java

package com.cognizant.springlearn.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.service.CountryService;

@RestController

public class CountryController {

private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);

@Autowired

private CountryService countryService;

@GetMapping("/countries/{code}")

public Country getCountry(@PathVariable String code) {

LOGGER.info("START");

Country country = countryService.getCountry(code);

LOGGER.debug("Country: {}", country);

LOGGER.info("END");

return country;

}

}

//json

{

"code": "IN",

"name": "India"

}

Exercise 10: Create authentication service that returns JWT:

<!-- pom.xml -->

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt</artifactId>

<version>0.9.1</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

//AuthenticationController.java

package com.cognizant.springlearn.controller;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import javax.servlet.http.HttpServletRequest;

import java.util.Base64;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

@RestController

public class AuthenticationController {

private static final Logger LOGGER = LoggerFactory.getLogger(AuthenticationController.class);

// Secret key for JWT signing

private static final String SECRET\_KEY = "secretkey123";

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

public Map<String, String> authenticate(HttpServletRequest request) {

LOGGER.info("START /authenticate");

// Get "Authorization" header

String authHeader = request.getHeader("Authorization");

if (authHeader == null || !authHeader.startsWith("Basic ")) {

throw new RuntimeException("Missing or invalid Authorization header");

}

// Decode Base64 username:password

String base64Credentials = authHeader.substring("Basic ".length());

byte[] credDecoded = Base64.getDecoder().decode(base64Credentials);

String credentials = new String(credDecoded);

String[] values = credentials.split(":", 2);

String username = values[0];

String password = values[1];

LOGGER.info("Username: {}", username);

LOGGER.info("Password: {}", password);

// TODO: In real app, validate user/password here

// For demo, accept any user/pass

// Generate JWT

String jwt = Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date())

.setExpiration(new Date(System.currentTimeMillis() + 60 \* 60 \* 1000)) // 1 hour

.signWith(SignatureAlgorithm.HS256, SECRET\_KEY)

.compact();

Map<String, String> tokenMap = new HashMap<>();

tokenMap.put("token", jwt);

LOGGER.info("END /authenticate");

return tokenMap;

}

}

//SecurityConfig.java

package com.cognizant.springlearn.config;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.configuration.\*;

@Configuration

@EnableWebSecurity

public class SecurityConfig extends WebSecurityConfigurerAdapter {

@Override

protected void configure(HttpSecurity http) throws Exception {

http

.csrf().disable()

.authorizeRequests()

.antMatchers("/authenticate").permitAll()

.anyRequest().authenticated();

}

}

//json

{"token":"<long.jwt.token.string>"}

//Inpostman

Authorization: Basic dXNlcjpwd2Q=

Content-Type: application/json