**Spring Core and Maven**

**Exercise 1: Configuring a Basic Spring Application**

package com.example;

public class HelloService {

public void sayHello() {

System.out.println("Hello from Spring!");

}

}

Beans.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="helloService" class="com.example.HelloService"/>

</beans>

package com.example;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

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

HelloService service = (HelloService) context.getBean("helloService");

service.sayHello();

}

}

**Exercise 2: Implementing Dependency Injection**

package com.example;

public class Student {

private String name;

public void setName(String name) {

this.name = name;

}

public void displayInfo() {

System.out.println("Student Name: " + name);

}

}

package com.example;

public class College {

private Student student;

public void setStudent(Student student) {

this.student = student;

}

public void show() {

student.displayInfo();

}

}

<bean id="studentBean" class="com.example.Student">

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

</bean>

<bean id="collegeBean" class="com.example.College">

<property name="student" ref="studentBean"/>

</bean>

package com.example;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

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

College college = (College) context.getBean("collegeBean");

college.show();

}

}

**Spring Data JPA with Spring Boot, Hibernate**

Implement services for managing Country

Find a country based on country code

Add a new country

Country.java

package com.example.countryservice.entity;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

@Entity

public class Country {

@Id

private String code;

private String name;

public Country() {}

public Country(String code, String name) {

this.code = code;

this.name = name;

}

public String getCode() { return code; }

public void setCode(String code) { this.code = code; }

public String getName() { return name; }

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

}

CountryRepostory.java

package com.example.countryservice.repository;

import com.example.countryservice.entity.Country;

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

public interface CountryRepository extends JpaRepository<Country, String> {

}

CountryService.java

package com.example.countryservice.service;

import com.example.countryservice.entity.Country;

import com.example.countryservice.repository.CountryRepository;

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

import org.springframework.stereotype.Service;

import java.util.Optional;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

public Optional<Country> getCountryByCode(String code) {

return countryRepository.findById(code);

}

public Country addCountry(Country country) {

return countryRepository.save(country);

}

}

CountryController.java

package com.example.countryservice.controller;

import com.example.countryservice.entity.Country;

import com.example.countryservice.service.CountryService;

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

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

import java.util.Optional;

@RestController

@RequestMapping("/countries")

public class CountryController {

@Autowired

private CountryService countryService;

@GetMapping("/{code}")

public Optional<Country> getCountry(@PathVariable String code) {

return countryService.getCountryByCode(code);

}

@PostMapping

public Country addCountry(@RequestBody Country country) {

return countryService.addCountry(country);

}

}

CountryServiceApplication.java

package com.example.countryservice;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountryServiceApplication {

public static void main(String[] args) {

SpringApplication.run(CountryServiceApplication.class, args);

}

}