**WEEK-4**



**Exercise-1**: **Create a Spring Web Project using Maven**

**pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

         http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>

  <artifactId>spring-data-jpa-demo</artifactId>

  <version>1.0.0</version>

  <packaging>jar</packaging>

  <parent>

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

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

    <version>3.2.5</version>

    <relativePath/>

  </parent>

  <properties>

    <java.version>21</java.version>

  </properties>

  <dependencies>

  <dependency>

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

    <artifactId>spring-boot-starter-data-jpa</artifactId>

  </dependency>

  <dependency>

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

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

  </dependency>

  <dependency>

    <groupId>com.h2database</groupId>

    <artifactId>h2</artifactId>

    <scope>runtime</scope>

  </dependency>

  <dependency>

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

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

    <scope>test</scope>

  </dependency>

</dependencies>

  <build>

    <plugins>

      <plugin>

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

        <artifactId>spring-boot-maven-plugin</artifactId>

      </plugin>

    </plugins>

  </build>

</project>

**SpringDataJpaDemoApplication.java**

package com.example.spring\_data\_jpa\_demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringDataJpaDemoApplication {

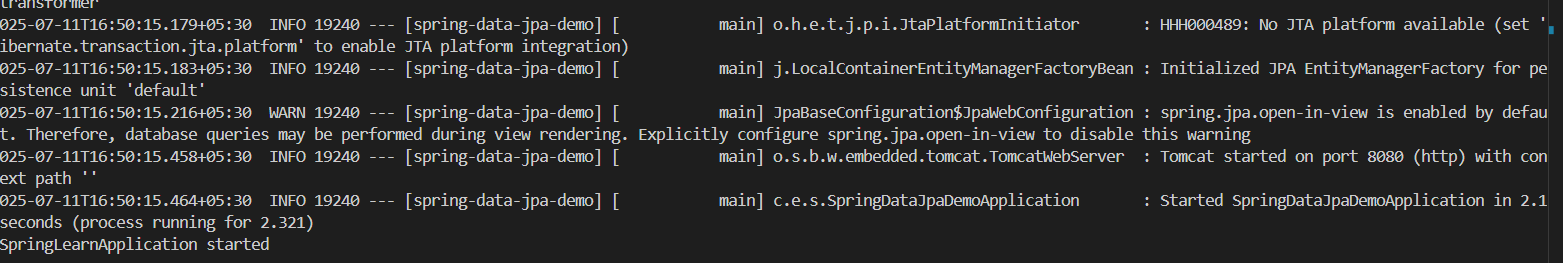
    public static void main(String[] args) {

        SpringApplication.run(SpringDataJpaDemoApplication.class, args);

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

    }}

**OUTPUT:**



**Exercise 2: Spring Core – Load Country from Spring Configuration XML**

**pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

         http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>

  <artifactId>spring-data-jpa-demo</artifactId>

  <version>1.0.0</version>

  <packaging>jar</packaging>

  <parent>

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

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

    <version>3.2.5</version>

    <relativePath/>

  </parent>

  <properties>

    <java.version>21</java.version>

  </properties>

  <dependencies>

  <dependency>

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

    <artifactId>spring-boot-starter-data-jpa</artifactId>

  </dependency>

  <dependency>

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

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

  </dependency>

  <dependency>

    <groupId>com.h2database</groupId>

    <artifactId>h2</artifactId>

    <scope>runtime</scope>

  </dependency>

  <dependency>

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

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

    <scope>test</scope>

  </dependency>

</dependencies>

  <build>

    <plugins>

      <plugin>

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

        <artifactId>spring-boot-maven-plugin</artifactId>

      </plugin>

    </plugins>

  </build>

</project>

**SpringLearnApplication.java**

package com.example.spring\_data\_jpa\_demo;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.SpringApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

@SpringBootApplication

public class SpringLearnApplication {

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

    public static void main(String[] args) {

        SpringApplication.run(SpringLearnApplication.class, args);

        LOGGER.debug("SpringLearnApplication started");

        displayCountry();

    }

    public static void displayCountry() {

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

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

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

    }

}

**Country.java**

package com.example.spring\_data\_jpa\_demo;

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("Getting country code: {}", code);

        return code;

    }

    public void setCode(String code) {

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

        this.code = code;

    }

    public String getName() {

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

        return name;

    }

    public void setName(String name) {

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

        this.name = name;

    }

    @Override

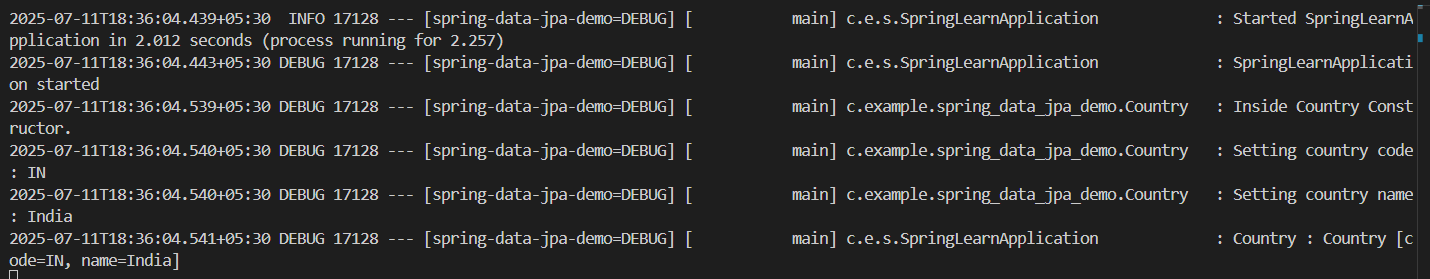
    public String toString() {

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

    }

}

**OUTPUT:**



**Exercise 3: Hello World RESTful Web Service**

**SpringLearnApplication.java**

package com.example.spring\_data\_jpa\_demo;

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);

        }

}

**HelloController.java**

package com.example.spring\_data\_jpa\_demo;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

@RestController

public class HelloController {

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

    @GetMapping("/hello")

    public String sayHello() {

        LOGGER.info("START - sayHello()");

        String response = "Hello World!!";

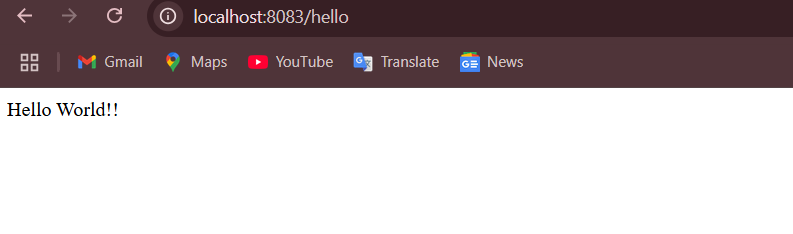
        LOGGER.info("END - sayHello()");

        return response;

    }

}

**OUTPUT:**



**Exercise 4: REST - Country Web Service**

**CountryController.java**

package com.example.spring\_data\_jpa\_demo;

import com.example.spring\_data\_jpa\_demo.Country;

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.RequestMapping;

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

@RestController

public class CountryController {

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

    @RequestMapping("/country")

    public Country getCountryIndia() {

        LOGGER.info("START - getCountryIndia()");

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

        Country country = (Country) context.getBean("in");

        LOGGER.info("END - getCountryIndia()");

        return 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="in" class="com.example.spring\_data\_jpa\_demo.Country">

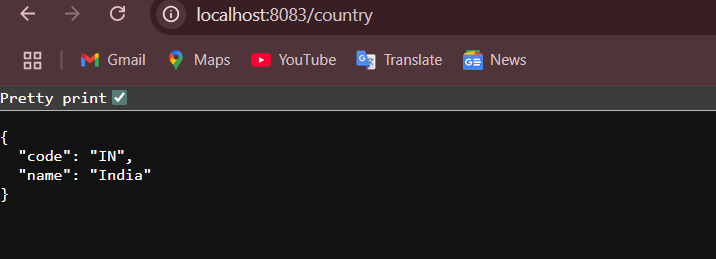
        <property name="code" value="IN"/>

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

    </bean>

</beans>

**OUTPUT:**



**Exercise 5: REST - Get 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">

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

        <constructor-arg>

            <list>

                <bean class="com.example.spring\_data\_jpa\_demo.Country">

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

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

                </bean>

                <bean class="com.example.spring\_data\_jpa\_demo.Country">

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

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

                </bean>

                <bean class="com.example.spring\_data\_jpa\_demo.Country">

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

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

                </bean>

            </list>

        </constructor-arg>

    </bean>

</beans>

**Country.java**

package com.example.spring\_data\_jpa\_demo;

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("Getting country code: {}", code);

        return code;

    }

    public void setCode(String code) {

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

        this.code = code;

    }

    public String getName() {

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

        return name;

    }

    public void setName(String name) {

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

        this.name = name;

    }

    @Override

    public String toString() {

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

    }

}

**CountryService.java**

package com.example.spring\_data\_jpa\_demo;

import com.example.spring\_data\_jpa\_demo.Country;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

    public Country getCountry(String code) {

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

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

        return countryList.stream()

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

                .findFirst()

                .orElse(null);

    }

}

**CountryController.java**

package com.example.spring\_data\_jpa\_demo;

import com.example.spring\_data\_jpa\_demo.Country;

import com.example.spring\_data\_jpa\_demo.CountryService;

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

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

@RestController

public class CountryController {

    @Autowired

    private CountryService countryService;

    @GetMapping("/countries/{code}")

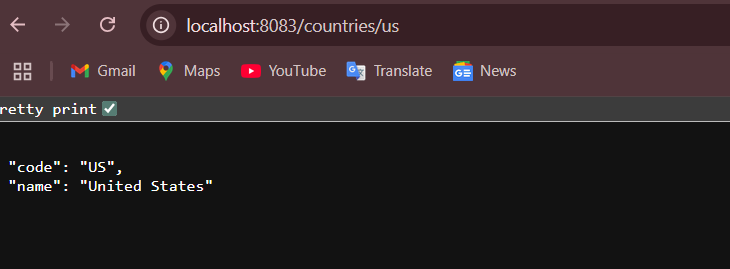
    public Country getCountry(@PathVariable String code) {

        return countryService.getCountry(code);

    }

}

**OUTPUT:**

****

**Exercise 6: Create authentication service that returns JWT**

**DemoApplication.java**

package com.example.spring\_data\_jpa\_demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DemoApplication {

    public static void main(String[] args) {

        SpringApplication.run(DemoApplication.class, args);

    }}

**SecurityConfig.java**

package com.example.spring\_data\_jpa\_demo;

import org.springframework.context.annotation.Configuration;

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

import org.springframework.context.annotation.Bean;

import org.springframework.security.web.SecurityFilterChain;

@Configuration

public class SecurityConfig {

    @Bean

    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

        http.csrf(csrf -> csrf.disable())

            .authorizeHttpRequests(auth -> auth

                .requestMatchers("/authenticate").permitAll()

                .anyRequest().authenticated());

        return http.build();

    }}

**JwtUtil.java**

package com.example.spring\_data\_jpa\_demo;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import io.jsonwebtoken.security.Keys;

import org.springframework.stereotype.Component;

import java.util.Date;

import java.security.Key;

@Component

public class JwtUtil {

    private static final Key SECRET\_KEY = Keys.secretKeyFor(SignatureAlgorithm.HS256);

    private static final long EXPIRATION\_TIME = 1000 \* 60 \* 60;

    public String generateToken(String username) {

        return Jwts.builder()

                .setSubject(username)

                .setIssuedAt(new Date(System.currentTimeMillis()))

                .setExpiration(new Date(System.currentTimeMillis() + EXPIRATION\_TIME))

                .signWith(SECRET\_KEY)

                .compact();

    }}

**AuthController.java**

package com.example.spring\_data\_jpa\_demo;

import com.example.spring\_data\_jpa\_demo.JwtUtil;

import jakarta.servlet.http.HttpServletRequest;

import org.springframework.http.ResponseEntity;

import org.springframework.util.Base64Utils;

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

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

@RestController

public class AuthController {

    private final JwtUtil jwtUtil;

    public AuthController(JwtUtil jwtUtil) {

        this.jwtUtil = jwtUtil;}

    @GetMapping("/authenticate")

    public ResponseEntity<?> authenticate(HttpServletRequest request) {

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

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

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

            String credentials = new String(Base64Utils.decodeFromString(base64Credentials));

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

            String username = values[0];

            String password = values[1];

            if ("user".equals(username) && "pwd".equals(password)) {

                String token = jwtUtil.generateToken(username);

                return ResponseEntity.ok().body("{\"token\":\"" + token + "\"}");

            }}

        return ResponseEntity.status(401).body("{\"error\":\"Unauthorized\"}");

    }}

**pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

         http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>

  <artifactId>spring-data-jpa-demo</artifactId>

  <version>1.0.0</version>

  <packaging>jar</packaging>

  <properties>

    <java.version>21</java.version>

  </properties>

  <dependencies>

  <dependency>

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

    <artifactId>spring-boot-starter-data-jpa</artifactId>

  </dependency>

  <dependency>

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

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

  </dependency>

  <dependency>

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

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

  </dependency>

  <dependency>

    <groupId>com.h2database</groupId>

    <artifactId>h2</artifactId>

    <scope>runtime</scope>

  </dependency>

  <dependency>

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

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

    <scope>test</scope>

  </dependency>

  <dependency>

    <groupId>io.jsonwebtoken</groupId>

    <artifactId>jjwt-api</artifactId>

    <version>0.11.5</version>

  </dependency>

  <dependency>

    <groupId>io.jsonwebtoken</groupId>

    <artifactId>jjwt-impl</artifactId>

    <version>0.11.5</version>

    <scope>runtime</scope>

  </dependency>

  <dependency>

    <groupId>io.jsonwebtoken</groupId>

    <artifactId>jjwt-jackson</artifactId>

    <version>0.11.5</version>

    <scope>runtime</scope>

  </dependency>

</dependencies>

  <build>

    <plugins>

      <plugin>

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

        <artifactId>spring-boot-maven-plugin</artifactId>

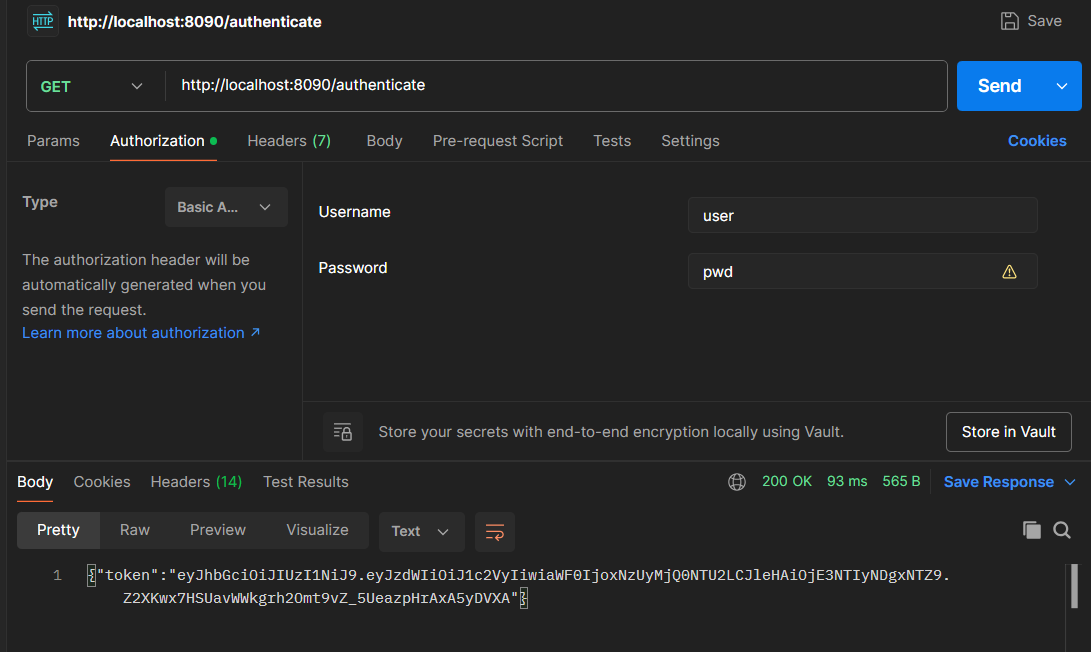
      </plugin>

    </plugins>

  </build>

</project>

**OUTPUT:**

****