

WEEK 4 JAVA FSE DIGITAL NURTURE 4.0

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-web-demo</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>war</packaging>

  <properties>
    <java.version>1.8</java.version>
    <spring.version>5.3.30</spring.version>
  </properties>

  <dependencies>
    <!-- Spring MVC -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>${spring.version}</version>
    </dependency>

    <!-- Servlet API -->
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>4.0.1</version>
      <scope>provided</scope>
    </dependency>

    <!-- JSP -->
    <dependency>
      <groupId>javax.servlet.jsp</groupId>
      <artifactId>javax.servlet.jsp-api</artifactId>
      <version>2.3.3</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>
```

```

    <build>
        <finalName>spring-web-demo</finalName>
    </build>
</project>

```

HomeController.java

```
package com.example.demo;
```

```

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.ui.Model;

```

```

@Controller
public class HomeController {

    @RequestMapping("/")
    public String home(Model model) {
        model.addAttribute("message", "Welcome to Spring MVC Web App!");
        return "home"; // returns home.jsp
    }
}

```

Web.xml

```

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
        http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
    version="3.1">

    <display-name>Spring Web Demo</display-name>

    <!-- Spring Dispatcher Servlet -->
    <servlet>
        <servlet-name>dispatcher</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>/WEB-INF/dispatcher-servlet.xml</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>dispatcher</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

    <welcome-file-list>

```

```

    <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>

```

Dispatcher-servlet.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    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
        http://www.springframework.org/schema/mvc
        http://www.springframework.org/schema/mvc/spring-mvc.xsd">

    <mvc:annotation-driven/>

    <context:component-scan base-package="com.example.demo"/>

    <!-- View Resolver -->
    <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/views/" />
        <property name="suffix" value=".jsp" />
    </bean>
</beans>

```

Home.jsp

```

<%@ page contentType="text/html; charset=UTF-8" %>
<html>
<head>
    <title>Home Page</title>
</head>
<body>
    <h1>${message}</h1>
</body>
</html>

```

2.Spring Core – Load Country from Spring Configuration XML

Country.java

```

package com.example;

public class Country {
    private String name;
    private String capital;

```

```

// Getters and setters
public String getName() {
    return name;
}

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

public String getCapital() {
    return capital;
}

public void setCapital(String capital) {
    this.capital = capital;
}

@Override
public String toString() {
    return "Country [name=" + name + ", capital=" + capital + "]";
}
}

```

Spring-config.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.example.Country">
        <property name="name" value="India"/>
        <property name="capital" value="New Delhi"/>
    </bean>

</beans>

```

App.java

```

package com.example;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class App {
    public static void main(String[] args) {
        ApplicationContext context = new ClassPathXmlApplicationContext("spring-config.xml");
        Country country = (Country) context.getBean("country");
    }
}

```

```

        System.out.println("Country Name: " + country.getName());
        System.out.println("Capital: " + country.getCapital());
    }
}

```

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-core-country</artifactId>
    <version>1.0-SNAPSHOT</version>

    <dependencies>
        <!-- Spring Core Context -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>5.3.30</version>
        </dependency>
    </dependencies>
</project>

```

3.Hello World RESTful Web Service

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>hello-world-rest</artifactId>
    <version>1.0.0</version>
    <packaging>jar</packaging>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.1.4</version>
    </parent>

```

```

<dependencies>
  <!-- Spring Boot Web (for REST) -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
</dependencies>

<build>
  <plugins>
    <!-- Spring Boot Maven Plugin -->
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
</project>

```

HelloWorldApplication.java

```
package com.example.helloworld;
```

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
public class HelloWorldApplication {
    public static void main(String[] args) {
        SpringApplication.run(HelloWorldApplication.class, args);
    }
}

```

HelloWorldApplication.java

```
package com.example.helloworld;
```

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

```
@RestController
public class HelloWorldController {

    @GetMapping("/hello")
    public String hello() {
        return "Hello, World!";
    }
}

```

4.REST - Country Web Service

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>country-rest-service</artifactId>
  <version>1.0.0</version>
  <packaging>jar</packaging>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.1.4</version>
  </parent>

  <dependencies>
    <!-- Spring Web for REST -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
  </dependencies>
</project>
```

CountryRestApplication.java

```
package com.example.countryservice;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class CountryRestApplication {
    public static void main(String[] args) {
        SpringApplication.run(CountryRestApplication.class, args);
    }
}
```

Country.java

```
package com.example.countryservice;

public class Country {
    private String name;
```

```

private String capital;

// Constructor
public Country(String name, String capital) {
    this.name = name;
    this.capital = capital;
}

// Getters
public String getName() {
    return name;
}

public String getCapital() {
    return capital;
}
}

```

CountryController.java

```

package com.example.countryservice;

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

@RestController
public class CountryController {

    @GetMapping("/country")
    public Country getCountry() {
        return new Country("India", "New Delhi");
    }
}

```

5.REST - Get country based on country code

pom.xml

```

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

    <groupId>com.example</groupId>
    <artifactId>country-rest-service</artifactId>
    <version>1.0.0</version>

    <parent>

```



```

    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.1.4</version>
</parent>

<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
</dependencies>
</project>

```

Country.java

```

package com.example.countryservice;

public class Country {
    private String name;
    private String capital;

    public Country(String name, String capital) {
        this.name = name;
        this.capital = capital;
    }

    public String getName() {
        return name;
    }

    public String getCapital() {
        return capital;
    }
}

```

CountryController.java

```

package com.example.countryservice;

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

import java.util.HashMap;
import java.util.Map;

@RestController
@RequestMapping("/country")
public class CountryController {

    private static final Map<String, Country> countryMap = new HashMap<>();

```

```

static {
    countryMap.put("IN", new Country("India", "New Delhi"));
    countryMap.put("US", new Country("United States", "Washington, D.C."));
    countryMap.put("JP", new Country("Japan", "Tokyo"));
    countryMap.put("FR", new Country("France", "Paris"));
}

@GetMapping("/{code}")
public Country getCountryByCode(@PathVariable String code) {
    Country country = countryMap.get(code.toUpperCase());
    if (country == null) {
        throw new CountryNotFoundException("Country code '" + code + "' not found.");
    }
    return country;
}

@ResponseStatus(code = org.springframework.http.HttpStatus.NOT_FOUND)
static class CountryNotFoundException extends RuntimeException {
    public CountryNotFoundException(String message) {
        super(message);
    }
}
}

```

CountryRestApplication.java

```

package com.example.countryservice;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class CountryRestApplication {
    public static void main(String[] args) {
        SpringApplication.run(CountryRestApplication.class, args);
    }
}

```

6.Create authentication service that returns JWT

pom.xml

```

<project ...>
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.example</groupId>
    <artifactId>jwt-auth-service</artifactId>
    <version>1.0</version>

```

```

<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>3.1.4</version>
</parent>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </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>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
</dependencies>
</project>

```

JwtAuthApplication.java

```
package com.example.jwt;
```

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
public class JwtAuthApplication {
    public static void main(String[] args) {
        SpringApplication.run(JwtAuthApplication.class, args);
    }
}

```

AuthRequest.java

```
package com.example.jwt.model;

public class AuthRequest {
    private String username;
    private String password;

    // Getters and setters
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
}
```

JwtService.java

```
package com.example.jwt.service;

import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;
import io.jsonwebtoken.security.Keys;
import org.springframework.stereotype.Service;

import java.security.Key;
import java.util.Date;

@Service
public class JwtService {

    private final Key key = Keys.secretKeyFor(SignatureAlgorithm.HS256);

    public String generateToken(String username) {
        return Jwts.builder()
            .setSubject(username)
            .setIssuer("example-app")
            .setIssuedAt(new Date())
            .setExpiration(new Date(System.currentTimeMillis() + 1000 * 60 * 60)) // 1 hour
            .signWith(key)
            .compact();
    }
}
```

```

    }

    public Key getKey() {
        return key;
    }
}

```

AuthController.java

```

package com.example.jwt.controller;

import com.example.jwt.model.AuthRequest;
import com.example.jwt.model.AuthResponse;
import com.example.jwt.service.JwtService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

@RestController
public class AuthController {

    @Autowired
    private JwtService jwtService;

    // Hardcoded user credentials
    @PostMapping("/authenticate")
    public ResponseEntity<AuthResponse> authenticate(@RequestBody AuthRequest request) {
        if ("admin".equals(request.getUsername()) && "admin".equals(request.getPassword())) {
            String token = jwtService.generateToken(request.getUsername());
            return ResponseEntity.ok(new AuthResponse(token));
        } else {
            return ResponseEntity.status(401).build();
        }
    }
}

```

SecurityConfig.java

```

package com.example.jwt.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.Customizer;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
public class SecurityConfig {

    @Bean

```

```
public SecurityFilterChain
filterChain(org.springframework.security.config.annotation.web.builders.HttpSecurity http) throws
Exception {
    http
        .csrf(csrf -> csrf.disable())
        .authorizeHttpRequests(auth -> auth
            .requestMatchers("/authenticate").permitAll()
            .anyRequest().authenticated()
        )
        .httpBasic(Customizer.withDefaults());
    return http.build();
}
}
```