**JWT-Create JWT Token**

package com.example.jwtDemo.controller;

import java.util.Date;

import java.util.Base64;

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

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

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

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

@RestController

public class AuthenticationController {

private final String secretKey = "A#123vasanth%&\*hjgf";

@GetMapping("/authenticate")

public String authenticate(@RequestHeader(value = "Authorization", required = false) String authHeader) {

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

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

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

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

String username = credentials[0];

String password = credentials[1];

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

String token = Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date(0))

.setExpiration(new Date(600000L \* 1000L))

.signWith(SignatureAlgorithm.HS256, secretKey)

.compact();

return token;

}

}

return "unauthorized";

}

}

A computer screen shot of a computer screen

AI-generated content may be incorrect.

**Create a spring web project using maven**

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

<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <parent>

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

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

        <version>3.5.3</version>

        <relativePath/> <!-- lookup parent from repository -->

    </parent>

    <groupId>com.cognizant</groupId>

    <artifactId>spring-learn</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <name>spring-learn</name>

    <description>Demo project for Spring Boot</description>

    <url/>

    <licenses>

        <license/>

    </licenses>

    <developers>

        <developer/>

    </developers>

    <scm>

        <connection/>

        <developerConnection/>

        <tag/>

        <url/>

    </scm>

    <properties>

        <java.version>17</java.version>

    </properties>

    <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>

            <optional>true</optional>

        </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>

**Load Country from spring configuration 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.spring\_learn.Country">

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

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

    </bean>

    <bean id="country1" class="com.cognizant.spring\_learn.Country">

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

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

    </bean>

    <bean id="country2" class="com.cognizant.spring\_learn.Country">

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

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

    </bean>

    <bean id="country3" class="com.cognizant.spring\_learn.Country">

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

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

    </bean>

</beans>

-------------------------

package com.cognizant.spring\_learn;

public class Country {

    private String code;

    private String name;

    public Country() {

    }

    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;

    }

    @Override

    public String toString() {

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

    }

}

------------------

package com.cognizant.spring\_learn;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

@Configuration

public class AppConfig {

@Bean

public Country country() {

Country country = new Country();

country.setCode("IN");

country.setName("India");

return country;

}

@Bean

public Country country1() {

Country country = new Country();

country.setCode("US");

country.setName("United States");

return country;

}

}

A black background with many small white text

AI-generated content may be incorrect.

**Hello World RESTful Web Service**   
  
package com.cognizant.spring\_learn.controllers;

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

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

import com.cognizant.spring\_learn.SpringLearnApplication;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@RestController

public class HelloController {

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

    @GetMapping("/hello")

    public String hello(){

        LOGGER.info("Started request");

        LOGGER.info("ended request");

        return "Hello World";

    }

}

A screenshot of a chat

AI-generated content may be incorrect.

**REST - Country Web Service**   
package com.cognizant.spring\_learn.controllers;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

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

import com.cognizant.spring\_learn.Country;

@RestController

public class CountryController {

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

    @GetMapping("/country")

    public Country getCountryIndia(){

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

            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="country" class="com.cognizant.spring\_learn.Country">

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

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

    </bean>

</beans>

A screenshot of a computer

AI-generated content may be incorrect.

**REST - Get country based on country code**

package com.cognizant.spring\_learn.controllers;

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

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

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

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

import com.cognizant.spring\_learn.Country;

import com.cognizant.spring\_learn.Service.CountryService;

@RestController

public class CountryController {

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

    @GetMapping("/country")

    public Country getCountryIndia(){

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

            return country;

        }

     @Autowired

    private CountryService countryService;

    @GetMapping("/countries/{code}")

    public Country getCountry(@PathVariable String code) {

        Country country = countryService.getCountry(code);

        return country;

    }

}

<?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.spring\_learn.Country">

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

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

    </bean>

    <bean id="country1" class="com.cognizant.spring\_learn.Country">

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

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

    </bean>

    <bean id="country2" class="com.cognizant.spring\_learn.Country">

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

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

    </bean>

    <bean id="country3" class="com.cognizant.spring\_learn.Country">

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

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

    </bean>

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

        <constructor-arg>

            <list>

                <ref bean="country"/>

                <ref bean="country1"/>

                <ref bean="country2"/>

                <ref bean="country3"/>

            </list>

        </constructor-arg>

    </bean>

</beans>

package com.cognizant.spring\_learn.Service;

import com.cognizant.spring\_learn.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.stereotype.Service;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.List;

@Service

public class CountryService {

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

    public Country getCountry(String code) {

        LOGGER.info("START: getCountry() in Service");

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

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

        Country result = countryList.stream()

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

                .findFirst()

                .orElse(null);

        LOGGER.info("END: getCountry() in Service");

        return result;

    }

}A screenshot of a computer

AI-generated content may be incorrect.