**Create a Spring Web Project using Maven**

**Code:**

package com.cognizant.spring\_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@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.info("STARTED SpringLearnApplication");

    }

}

package com.cognizant.spring\_learn;

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

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

@RestController

public class HelloController {

    @GetMapping("/")

    public String hello() {

        return "Hello from Spring Boot!";

    }

}

**Output:**

A black text on a white background

AI-generated content may be incorrect.

**Spring Core – Load SimpleDateFormat from Spring Configuration XML**

**Code:**

package com.cognizant.spring\_learn;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

    public static void main(String[] args) {

        displayDate(); // Call the method

    }

    public static void displayDate() {

        ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");

        SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

        try {

            Date date = format.parse("31/12/2018");

            System.out.println("Parsed Date: " + date);

        } catch (Exception e) {

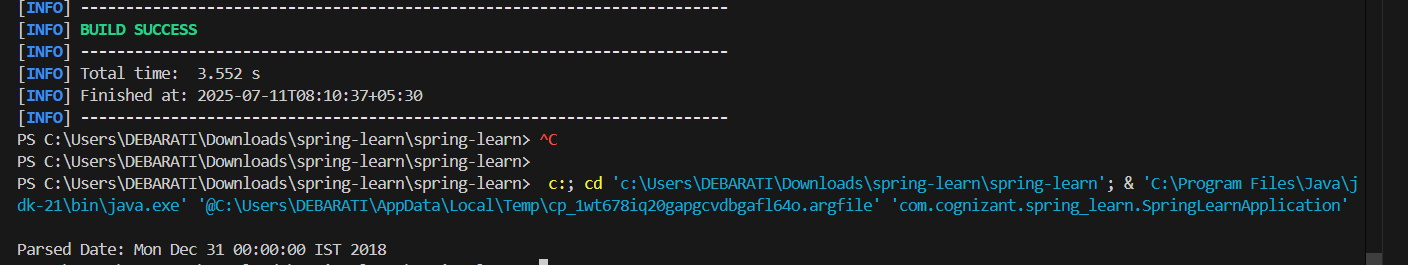
            e.printStackTrace();

        }

    }

}

**Output:**



**Hello World RESTful Web Service**

**Code:**

package com.cognizant.spring\_learn.controller;

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 message = "Hello World!!";

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

        return message;

    }

}

package com.cognizant.spring\_learn;

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);

    }

}

**Output:**

A black and white text

AI-generated content may be incorrect.

**REST - Country Web Service**

**Code:**

package com.cognizant.spring\_learn.controller;

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;

import com.cognizant.spring\_learn.model.Country;

@RestController

public class CountryController {

    @RequestMapping("/country")

    public Country getCountryIndia() {

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

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

        return country;

    }

}

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

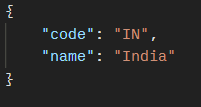
    public static void main(String[] args) {

        SpringApplication.run(SpringLearnApplication.class, args);

    }

}

**Output:**



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

**Code:**

package com.cognizant.spring\_learn.controller;

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

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.model.Country;

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

@RestController

public class CountryController {

    @Autowired

    private CountryService countryService;

    @GetMapping("/countries/{code}")

    public Country getCountry(@PathVariable String code) {

        return countryService.getCountry(code);

    }

}

package com.cognizant.spring\_learn.service;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.cognizant.spring\_learn.model.Country;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

@Service

public class CountryService {

    public Country getCountry(String code) {

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

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

        Optional<Country> country = countries.stream()

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

            .findFirst();

        if (country.isPresent()) {

            return country.get();

        } else {

            throw new CountryNotFoundException("Country not found for code: " + code);

        }

    }

}

package com.cognizant.spring\_learn.service;

public class CountryNotFoundException extends RuntimeException {

    public CountryNotFoundException(String message) {

        super(message);

    }

}

**Output:**

A black background with white text

AI-generated content may be incorrect.

**Create authentication service that returns JWT**

**Code:**

package com.example.jwt\_auth.config;

import org.springframework.context.annotation.\*;

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

import org.springframework.security.core.userdetails.\*;

import org.springframework.security.provisioning.InMemoryUserDetailsManager;

import org.springframework.security.web.SecurityFilterChain;

@Configuration

public class SecurityConfig {

    @Bean

    public InMemoryUserDetailsManager userDetailsService() {

        UserDetails user = User.withUsername("user")

                               .password("{noop}pwd")

                               .roles("USER")

                               .build();

        return new InMemoryUserDetailsManager(user);

    }

    @Bean

    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

        http

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

            .authorizeHttpRequests(auth -> auth

                .requestMatchers("/authenticate").authenticated()

                .anyRequest().permitAll())

            .httpBasic();

        return http.build();

    }

}

package com.example.jwt\_auth.controller;

import com.example.jwt\_auth.service.JwtUtil;

import org.springframework.http.ResponseEntity;

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

import java.security.Principal;

@RestController

public class AuthenticationController {

    private final JwtUtil jwtUtil;

    public AuthenticationController(JwtUtil jwtUtil) {

        this.jwtUtil = jwtUtil;

    }

    @GetMapping("/authenticate")

    public ResponseEntity<?> authenticate(Principal principal) {

        String username = principal.getName();

        String token = jwtUtil.generateToken(username);

        return ResponseEntity.ok().body(new JwtResponse(token));

    }

    static class JwtResponse {

        private String token;

        public JwtResponse(String token) {

            this.token = token;

        }

        public String getToken() {

            return token;

        }

    }

}

package com.example.jwt\_auth.service;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import org.springframework.stereotype.Service;

import java.util.Date;

@Service

public class JwtUtil {

    private final String SECRET\_KEY = "mySecretKey123456789012345678901234567890";

    public String generateToken(String username) {

        return Jwts.builder()

                   .setSubject(username)

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

                   .setExpiration(new Date(System.currentTimeMillis() + 1000 \* 60 \* 10))

                   .signWith(SignatureAlgorithm.HS256, SECRET\_KEY.getBytes())

                   .compact();

    }

}

package com.example.jwt\_auth;

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);

    }

}