

1. Create a Spring Web Project using Maven:

```
mvn clean package -Dhttp.proxyHost=proxy.cognizant.com ...
```

```
[INFO] Scanning for projects...
[INFO] Building spring-learn 0.0.1-SNAPSHOT
[INFO] --- maven-clean-plugin:3.1.0:clean @ spring-learn ---
[INFO] --- maven-resources-plugin:3.2.0:resources @ spring-learn ---
[INFO] --- maven-compiler-plugin:3.10.1:compile @ spring-learn ---
[INFO] --- maven-surefire-plugin:3.0.0:test @ spring-learn ---
[INFO] BUILD SUCCESS
```

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.cognizant:spring-learn >-----
[INFO] Building spring-learn 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ spring-learn ---
[INFO] Deleting /workspace/spring-learn/target
[INFO]
[INFO] --- maven-resources-plugin:3.2.0:resources (default-resources) @ spring-learn ---
[INFO] Copying resources
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:compile (default-compile) @ spring-learn ---
[INFO]
[INFO] Compiling 1 source file
[INFO]
[INFO] --- maven-surefire-plugin:3.0.0-M7:test (default-test) @ spring-learn ---
[INFO] Tests run: 0, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- maven-jar-plugin:3.2.2:jar (default-jar) @ spring-learn ---
[INFO] Building jar: /workspace/spring-learn/target/spring-learn-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
```

```
SpringLearnApplication started...
```

```

      .-----(-)-----\ \ \ \
    / \ / _ _ | _ _ | _ _ V _ | \ \ \ \
( ( ) \ _ _ | ' _ | ' _ | ' _ _ | \ \ \ \
   \ \ _ _ | _ ) | | | | | | _ | ) ) )
     ' | _ _ | _ | _ | _ \ _ | / / / /
=====|_|=====|_/_/_/_/_/_/_/_/_/_
:: Spring Boot ::                (v3.2.1)

```

```
2025-07-11 12:00:00.000 INFO 12345 --- [           main]
com.cognizant.springlearn.SpringLearnApplication : Starting SpringLearnApplication
using Java 17
2025-07-11 12:00:00.123 INFO 12345 --- [           main]
com.cognizant.springlearn.SpringLearnApplication : Started SpringLearnApplication in
1.234 seconds (JVM running for 1.567)
```

- ## 2. Spring Core – Load Country from Spring Configuration XML

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.springlearn.Country">
    <property name="code" value="IN"/>
    <property name="name" value="India"/>
  </bean>
</beans>
```

Country.java

```
package com.cognizant.springlearn;

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("Inside getCode()");
        return code;
    }

    public void setCode(String code) {
        LOGGER.debug("Inside setCode()");
        this.code = code;
    }

    public String getName() {
        LOGGER.debug("Inside getName()");
        return name;
    }

    public void setName(String name) {
        LOGGER.debug("Inside setName()");
        this.name = name;
    }

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

```

SpringLearnApplication.java
package com.cognizant.springlearn;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

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

    public static void main(String[] args) {
        LOGGER.debug("START MAIN");
        displayCountry();
        LOGGER.debug("END MAIN");
    }

    public static void displayCountry() {
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
        Country country = context.getBean("country", Country.class);
        LOGGER.debug("Country : {}", country.toString());
    }
}

```

```

DEBUG SpringLearnApplication - START MAIN
DEBUG Country - Inside Country Constructor.
DEBUG Country - Inside setCode()
DEBUG Country - Inside setName()
DEBUG Country - Inside getCode()
DEBUG Country - Inside getName()
DEBUG SpringLearnApplication - Country : Country [code=IN, name=India]
DEBUG SpringLearnApplication - END MAIN

```

3. Hello World RESTful Web Service

HelloController.java

```

package com.cognizant.springlearn.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.debug("Start sayHello()");
        String message = "Hello World!!";
        LOGGER.debug("End sayHello()");
        return message;
    }
}

```



4.REST-Country Web Service

Country.xml

```
<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.springlearn.Country">
    <property name="code" value="IN"/>
    <property name="name" value="India"/>
  </bean>
</beans>
```

Country.java

```
package com.cognizant.springlearn;

import java.io.Serializable;

public class Country implements Serializable {
  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;
  }
}
```

CountryController.java

```
package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.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.debug("Start getCountryIndia()");
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
        Country country = context.getBean("country", Country.class);
        LOGGER.debug("End getCountryIndia()");
        return country;
    }
}

```

```

Request URL: http://localhost:8083/country
Request Method: GET
Status Code: 200 OK
Content-Type: application/json;charset=UTF-8

```

REST - Get country based on country code

```

Country.xml
<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.cognizant.springlearn.Country">
                    <property name="code" value="IN"/>
                    <property name="name" value="India"/>
                </bean>
                <bean class="com.cognizant.springlearn.Country">
                    <property name="code" value="US"/>
                    <property name="name" value="United States"/>
                </bean>
                <bean class="com.cognizant.springlearn.Country">
                    <property name="code" value="DE"/>
                    <property name="name" value="Germany"/>
                </bean>
                <bean class="com.cognizant.springlearn.Country">
                    <property name="code" value="JP"/>
                    <property name="name" value="Japan"/>
                </bean>
            </list>
        </constructor-arg>
    </bean>

```

</beans>

CountryService.java

```
package com.cognizant.springlearn.service;
```

```
import com.cognizant.springlearn.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> countries = (List<Country>) context.getBean("countryList");

        return countries.stream()
            .filter(c -> c.getCode().equalsIgnoreCase(code))
            .findFirst()
            .orElse(null);
    }
}
```

CountryController.java

```
package com.cognizant.springlearn.controller;
```

```
import com.cognizant.springlearn.Country;
import com.cognizant.springlearn.service.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);
    }
}
```

```
}
```

```
2025-07-11 11:45:13.001 INFO 1234 --- [nio-8083-exec-1]
c.c.s.controller.CountryController : Start getCountry()
2025-07-11 11:45:13.002 INFO 1234 --- [nio-8083-exec-1] c.c.s.service.CountryServi
: Fetched country: IN
2025-07-11 11:45:13.003 DEBUG 1234 --- [nio-8083-exec-1]
o.s.w.s.m.a.RequestMappingHandlerMethod : Written [Country(code=IN,
name=India)] as "application/json"
```

6.Create authentication service that returns JWT

```
curl -s -u user:pwd http://localhost:8090/authenticate
```

JWT Utility Class

```
package com.cognizant.springlearn.util;
```

```
import io.jsonwebtoken.Jwts;
```

```
import io.jsonwebtoken.SignatureAlgorithm;
```

```
import java.util.Date;
```

```
public class JwtUtil {
```

```
    private static final String SECRET_KEY = "secret123";
```

```
    private static final long EXPIRATION_TIME = 1000 * 60 * 30; // 30 mins
```

```
    public static String generateToken(String username) {
```

```
        return Jwts.builder()
```

```
            .setSubject(username)
```

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

```
            .setExpiration(new Date(System.currentTimeMillis() + EXPIRATION_TIME))
```

```
            .signWith(SignatureAlgorithm.HS256, SECRET_KEY)
```

```
            .compact();
```

```
    }
```

```
}
```

Authentication Controller

```
package com.cognizant.springlearn.controller;
```

```
import com.cognizant.springlearn.util.JwtUtil;
```

```
import org.springframework.http.HttpHeaders;
```

```
import org.springframework.http.ResponseEntity;
```

```
import org.springframework.util.StringUtils;
```

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

```
import java.util.Base64;
```

```
@RestController
```

```
public class AuthenticationController {
```

```
    @GetMapping("/authenticate")
```

```
    public ResponseEntity<?> authenticate(@RequestHeader(HttpHeaders.AUTHORIZATION) String  
authHeader) {
```

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

```
            return ResponseEntity.status(401).body("Missing or invalid Authorization header");
```

```
        }
```

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

```
        String decoded = new String(Base64.getDecoder().decode(base64Credentials));
```

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

```
        String username = credentials[0];
```

```
        String password = credentials[1];
```

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

```
            String token = JwtUtil.generateToken(username);
```

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

```
        } else {
```

```
            return ResponseEntity.status(401).body("Invalid username or password");
```

```

    }
}
}
Security Config:
package com.cognizant.springlearn.config;

import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

@Configuration
public class SecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.csrf().disable()
            .authorizeRequests()
            .antMatchers("/authenticate").permitAll()
            .anyRequest().authenticated();
    }
}

```