1. spring-rest-handson :  
Hands on 1  
Create a Spring Web Project using Maven :  
SpringLearnApplication.java :   
package com.cognizant.springlearn;   
  
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) {  
        LOGGER.debug("START");  
        SpringApplication.run(SpringLearnApplication.class, args);  
        LOGGER.debug("END");  
    }  
}   
  
Hands on 4  
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 + "]";  
    }  
}  
Modified SpringLearnApplication.java :  
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");  
        SpringApplication.run(SpringLearnApplication.class, args);  
        displayCountry();  
        LOGGER.debug("END");  
    }   
  
    public static void displayCountry() {  
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
        Country country = (Country) context.getBean("country", Country.class);  
        LOGGER.debug("Country : {}", country.toString());  
    }  
}  
Output :  
DEBUG SpringLearnApplication: START  
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   
  
2. spring-rest-handson   
  
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()");  
        LOGGER.debug("END - sayHello()");  
        return "Hello World!!";  
    }  
}  
Output :   
  
URL: http://localhost:8083/hello   
  
Response: Hello World!!   
  
Status Code: 200 OK   
  
REST - Country Web Service and REST - Get country based on country code.  
Modified country.xml :  
<bean id="country" class="com.cognizant.springlearn.Country">  
    <property name="code" value="IN" />  
    <property name="name" value="India" />  
</bean>   
  
<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="JP" />  
                <property name="name" value="Japan" />  
            </bean>  
            <bean class="com.cognizant.springlearn.Country">  
                <property name="code" value="DE" />  
                <property name="name" value="Germany" />  
            </bean>  
        </list>  
    </constructor-arg>  
</bean>   
  
CountryController.java :  
package com.cognizant.springlearn.controller;   
  
import java.util.List;   
  
import com.cognizant.springlearn.Country;  
import com.cognizant.springlearn.service.CountryService;   
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.web.bind.annotation.\*;   
  
@RestController  
public class CountryController {   
  
    private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);   
  
    @Autowired  
    private CountryService countryService;   
  
    @RequestMapping("/country")  
    public Country getCountryIndia() {  
        LOGGER.debug("START - getCountryIndia()");  
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
        Country country = (Country) context.getBean("country", Country.class);  
        LOGGER.debug("Country : {}", country.toString());  
        LOGGER.debug("END - getCountryIndia()");  
        return country;  
    }   
  
    @GetMapping("/countries/{code}")  
    public Country getCountry(@PathVariable String code) {  
        LOGGER.debug("START - getCountry() with code: {}", code);  
        Country result = countryService.getCountry(code);  
        LOGGER.debug("Country Found: {}", result);  
        LOGGER.debug("END - getCountry()");  
        return result;  
    }  
}  
CountryService.java :  
package com.cognizant.springlearn.service;   
  
import java.util.List;   
  
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.stereotype.Service;   
  
@Service  
public class CountryService {   
  
    private static final Logger LOGGER = LoggerFactory.getLogger(CountryService.class);   
  
    public Country getCountry(String code) {  
        LOGGER.debug("START - getCountry()");   
  
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
        List<Country> countries = (List<Country>) context.getBean("countryList");   
  
        Country result = countries.stream()  
                .filter(c -> c.getCode().equalsIgnoreCase(code))  
                .findFirst()  
                .orElse(null);   
  
        LOGGER.debug("END - getCountry()");  
        return result;  
    }  
}  
Request: http://localhost:8083/country   
  
Response:   
  
{  
  "code": "IN",  
  "name": "India"  
}  
Request: http://localhost:8083/countries/in   
  
Response:   
  
{  
  "code": "IN",  
  "name": "India"  
}   
  
Request: http://localhost:8083/countries/jp   
  
Response:   
  
{  
  "code": "JP",  
  "name": "Japan"  
}   
  
5. JWT-handson  
Create authentication service that returns JWT   
  
Security Configuration :  
package com.cognizant.springlearn.config;   
  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.web.SecurityFilterChain;  
import org.springframework.context.annotation.Bean;   
  
@Configuration  
public class SecurityConfig {   
  
    @Bean  
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {  
        http  
            .csrf().disable()  
            .authorizeHttpRequests(authz -> authz  
                .requestMatchers("/authenticate").permitAll()  
                .anyRequest().authenticated()  
            )  
            .httpBasic(); // enables HTTP Basic Auth   
  
        return http.build();  
    }  
}  
JWT Utility :  
package com.cognizant.springlearn.util;   
  
import io.jsonwebtoken.Jwts;  
import io.jsonwebtoken.SignatureAlgorithm;  
import org.springframework.stereotype.Component;   
  
import java.util.Date;   
  
@Component  
public class JwtUtil {  
    private final String SECRET\_KEY = "my\_secret\_key";   
  
    public String generateToken(String username) {  
        return Jwts.builder()  
            .setSubject(username)  
            .setIssuedAt(new Date(System.currentTimeMillis()))  
            .setExpiration(new Date(System.currentTimeMillis() + 1000 \* 60 \* 10)) // 10 mins  
            .signWith(SignatureAlgorithm.HS256, SECRET\_KEY)  
            .compact();  
    }  
}  
Authentication Controller :  
package com.cognizant.springlearn.controller;   
  
import com.cognizant.springlearn.util.JwtUtil;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.http.\*;  
import org.springframework.web.bind.annotation.\*;   
  
import java.util.Base64;   
  
@RestController  
public class AuthController {   
  
    private static final Logger LOGGER = LoggerFactory.getLogger(AuthController.class);   
  
    private final JwtUtil jwtUtil;   
  
    public AuthController(JwtUtil jwtUtil) {  
        this.jwtUtil = jwtUtil;  
    }   
  
    @RequestMapping(value = "/authenticate", method = RequestMethod.GET)  
    public ResponseEntity<?> authenticate(@RequestHeader("Authorization") String authHeader) {  
        LOGGER.debug("START - authenticate()");   
  
        // Basic base64(user:pwd)  
        if (authHeader == null || !authHeader.startsWith("Basic ")) {  
            return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Missing Authorization Header");  
        }   
  
        // Decode credentials  
        String base64Credentials = authHeader.substring("Basic ".length());  
        byte[] credDecoded = Base64.getDecoder().decode(base64Credentials);  
        String credentials = new String(credDecoded);  
        String[] values = credentials.split(":", 2);   
  
        String username = values[0];  
        String password = values[1];   
  
        LOGGER.debug("Credentials - username: {}, password: {}", username, password);   
  
        // Hardcoded auth (for demo)  
        if ("user".equals(username) && "pwd".equals(password)) {  
            String token = jwtUtil.generateToken(username);  
            LOGGER.debug("Token generated: {}", token);  
            return ResponseEntity.ok().body("{\"token\":\"" + token + "\"}");  
        } else {  
            return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Invalid Credentials");  
        }  
    }  
}  
Request :  
curl -s -u user:pwd http://localhost:8090/authenticate  
Response :  
{  
  "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNzEwMTcwMDAwLCJleHAiOjE3MTAxNzA2MDB9.WQfMpJVRw2..."  
}