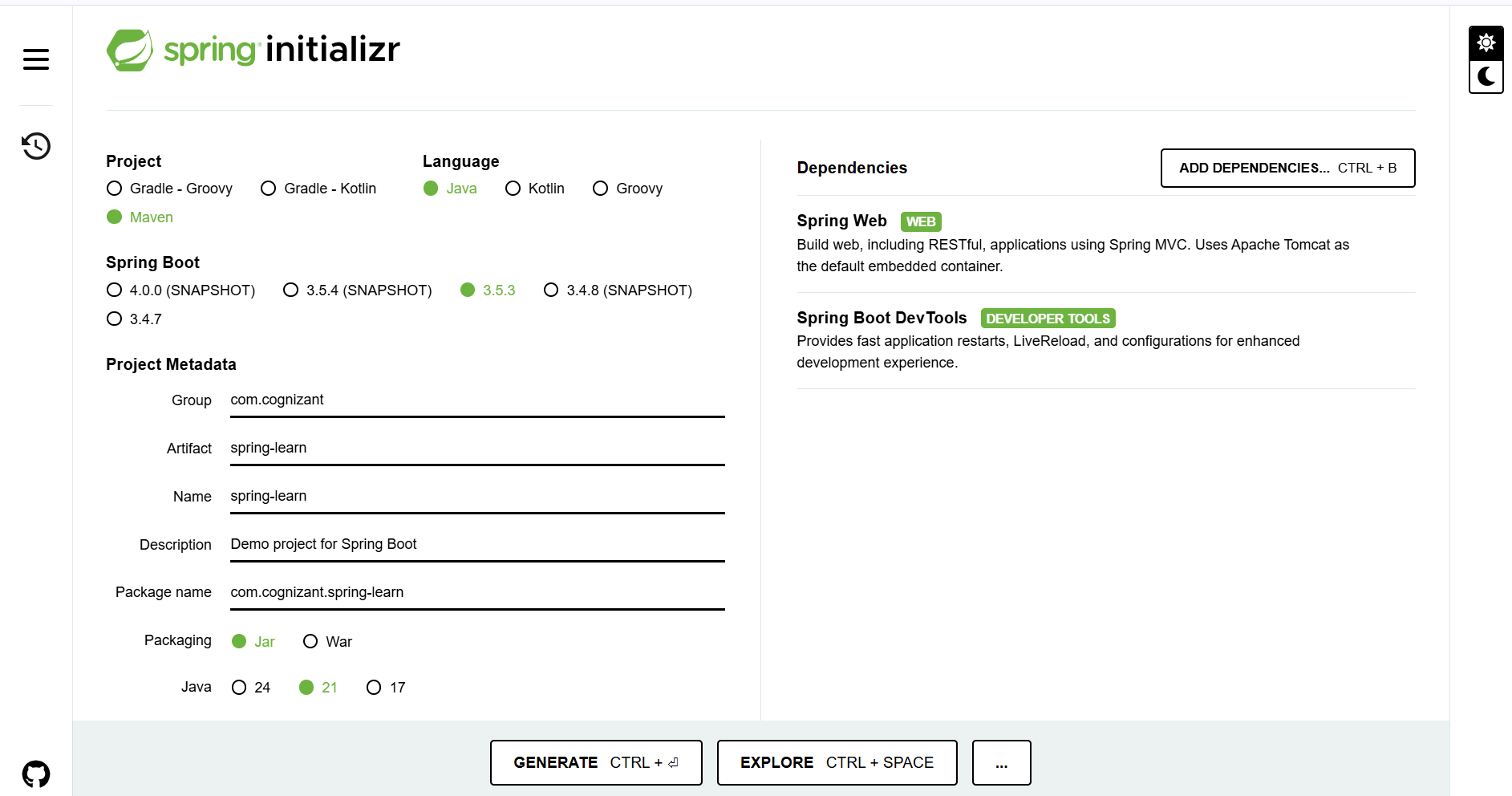
**Week 4 Spring REST using Spring Boot Mandatory HandsOn Solutions**

**Hands on 1**

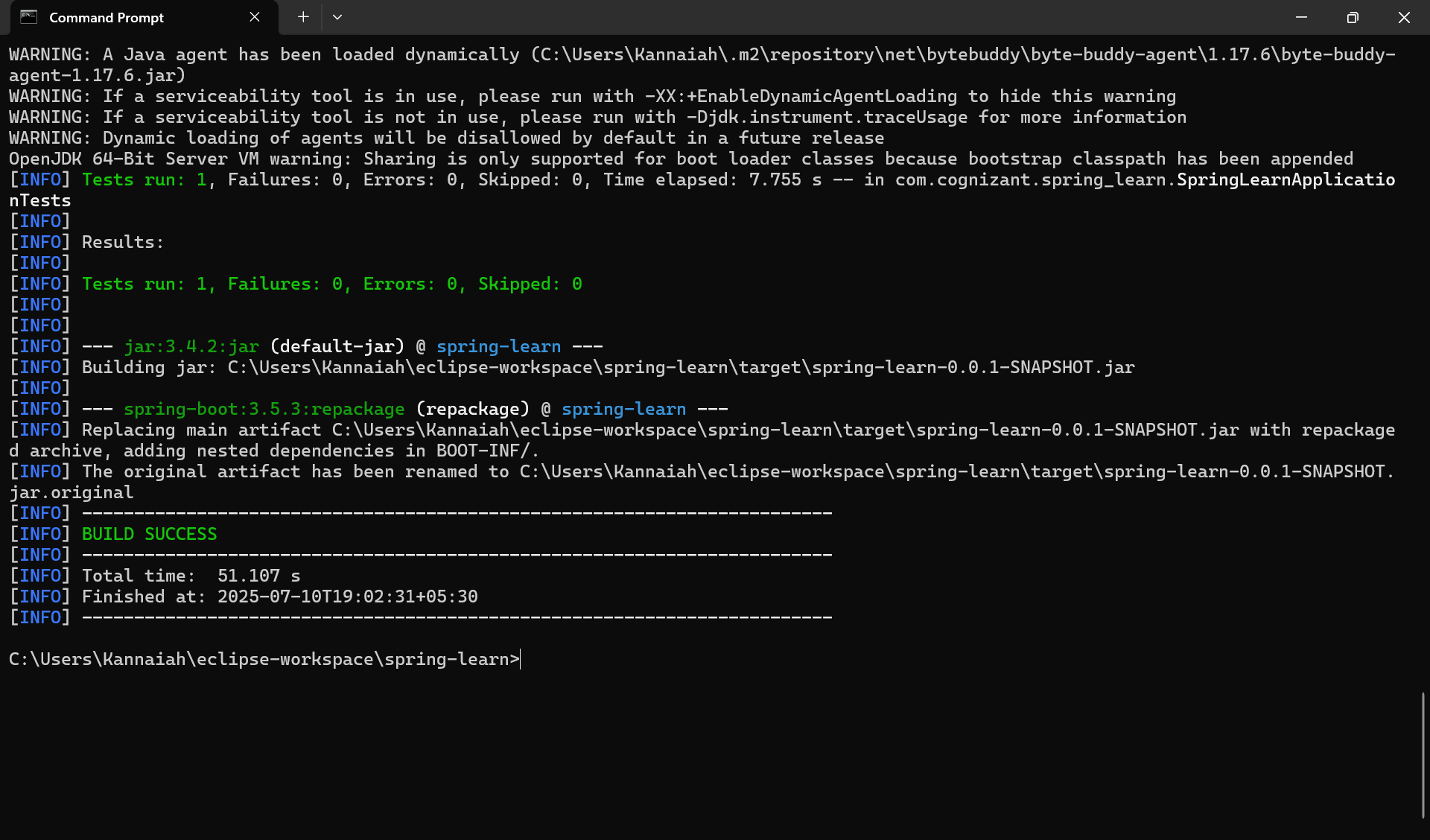
**Create a Spring Web Project using Maven**   
**Steps:**

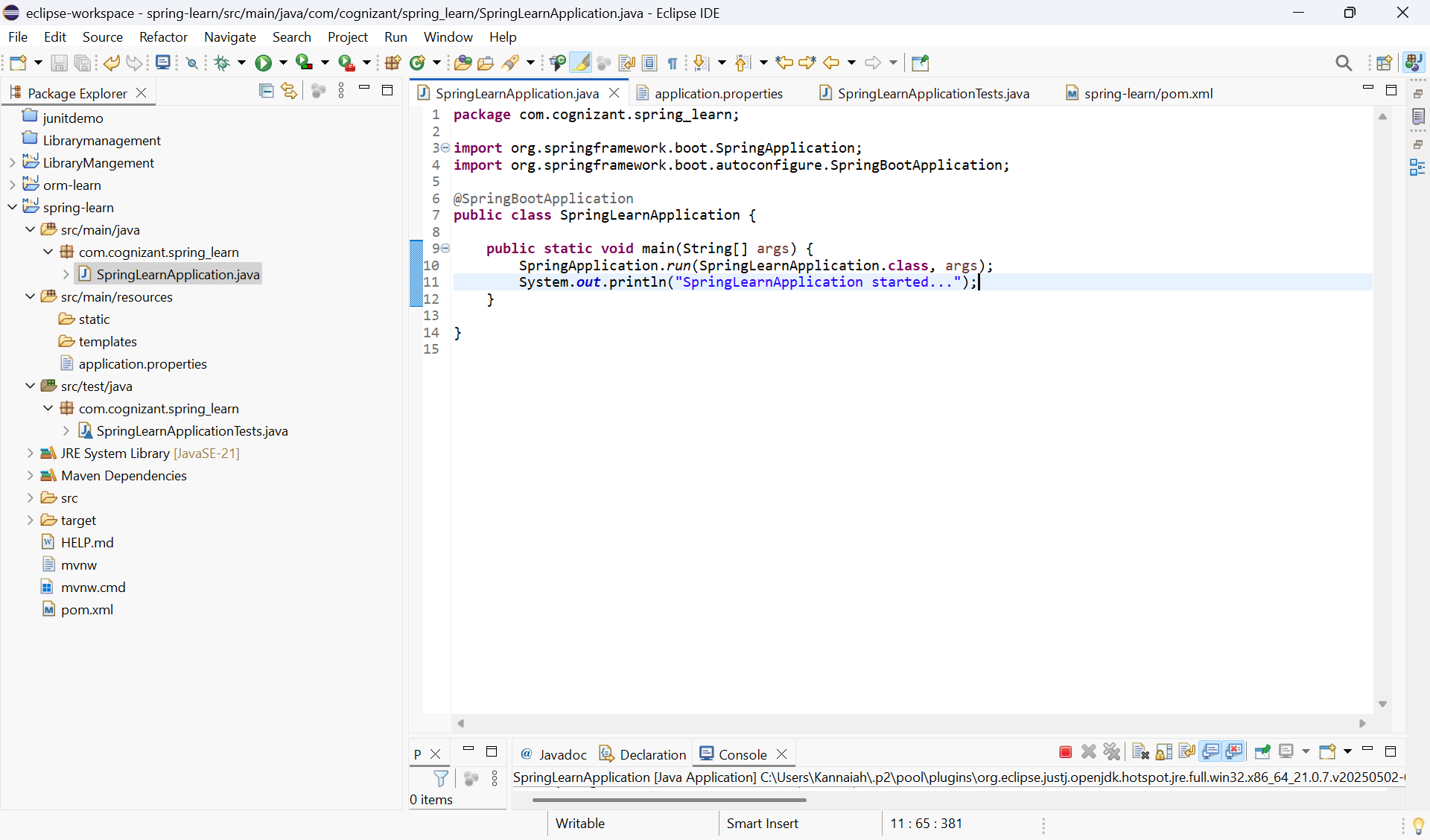
1. Visit <https://start.spring.io>, set Group as com.cognizant, Artifact as spring-learn, and add dependencies: **Spring Web**, **Spring Boot DevTools**.



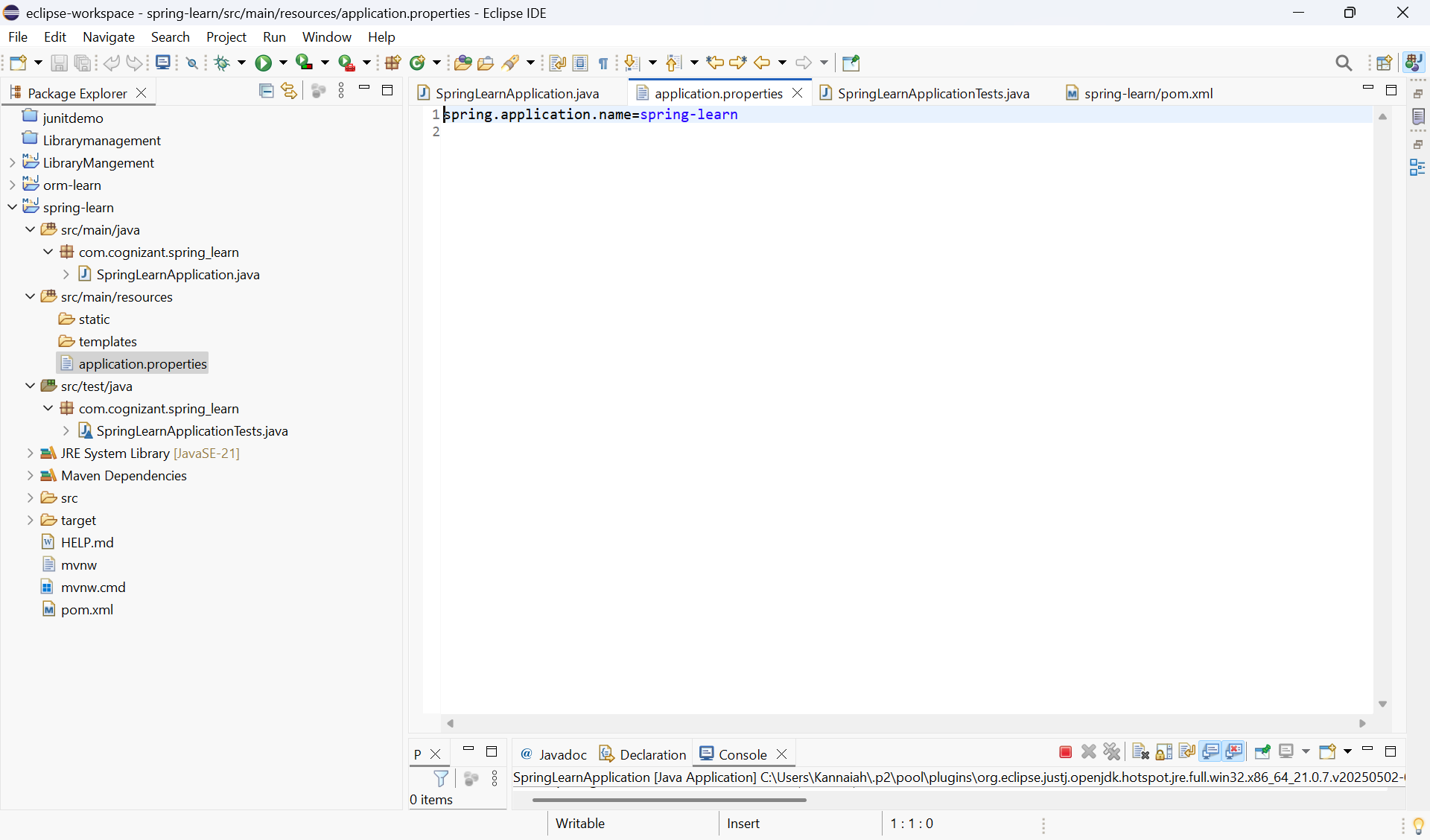
1. Click **Generate**, download and extract the ZIP to your Eclipse workspace.
2. Build the project via command line using:

mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456

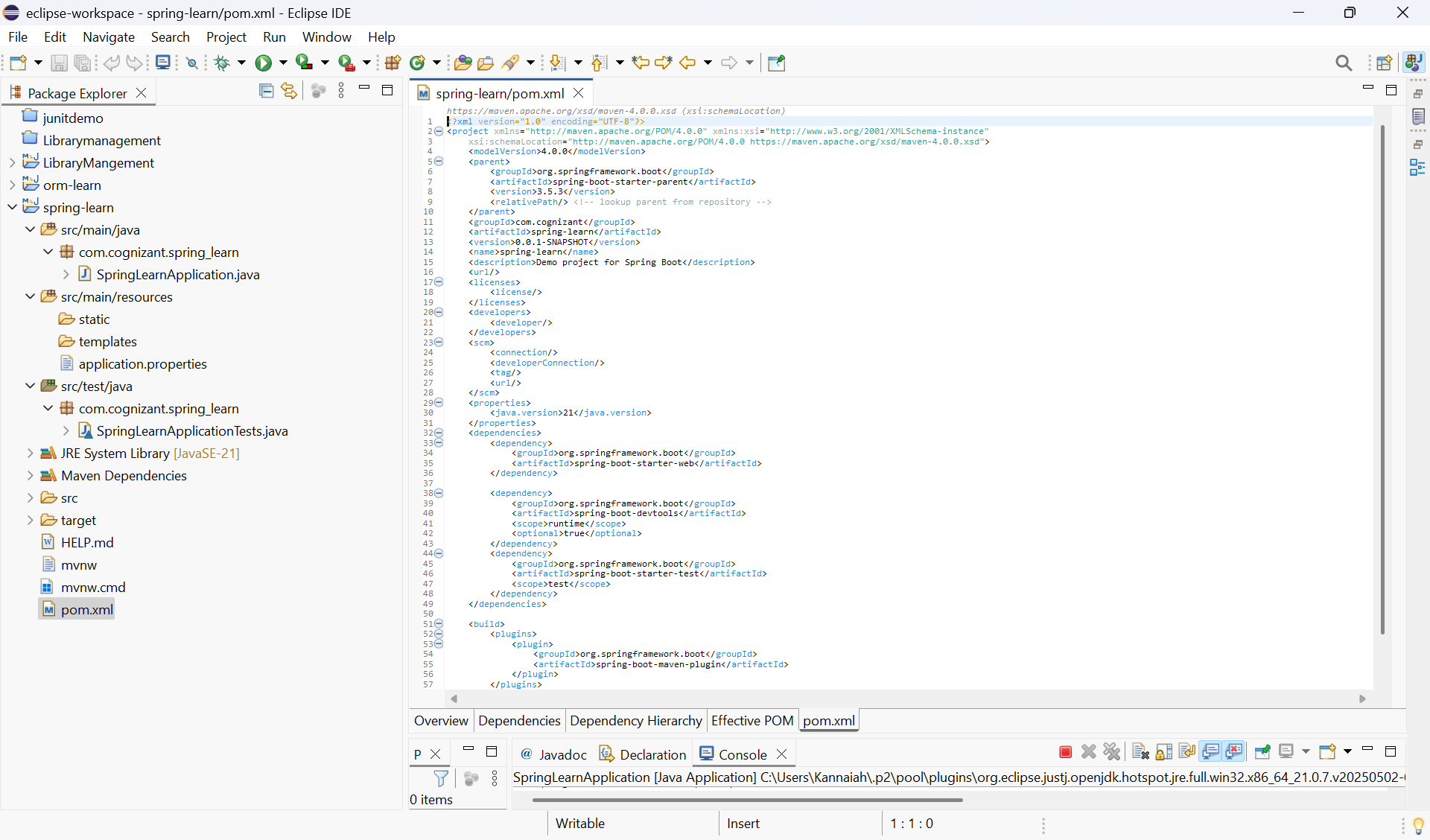


1. Import into Eclipse via File > Import > Maven > Existing Maven Projects.
2. Open **SpringLearnApplication.java**

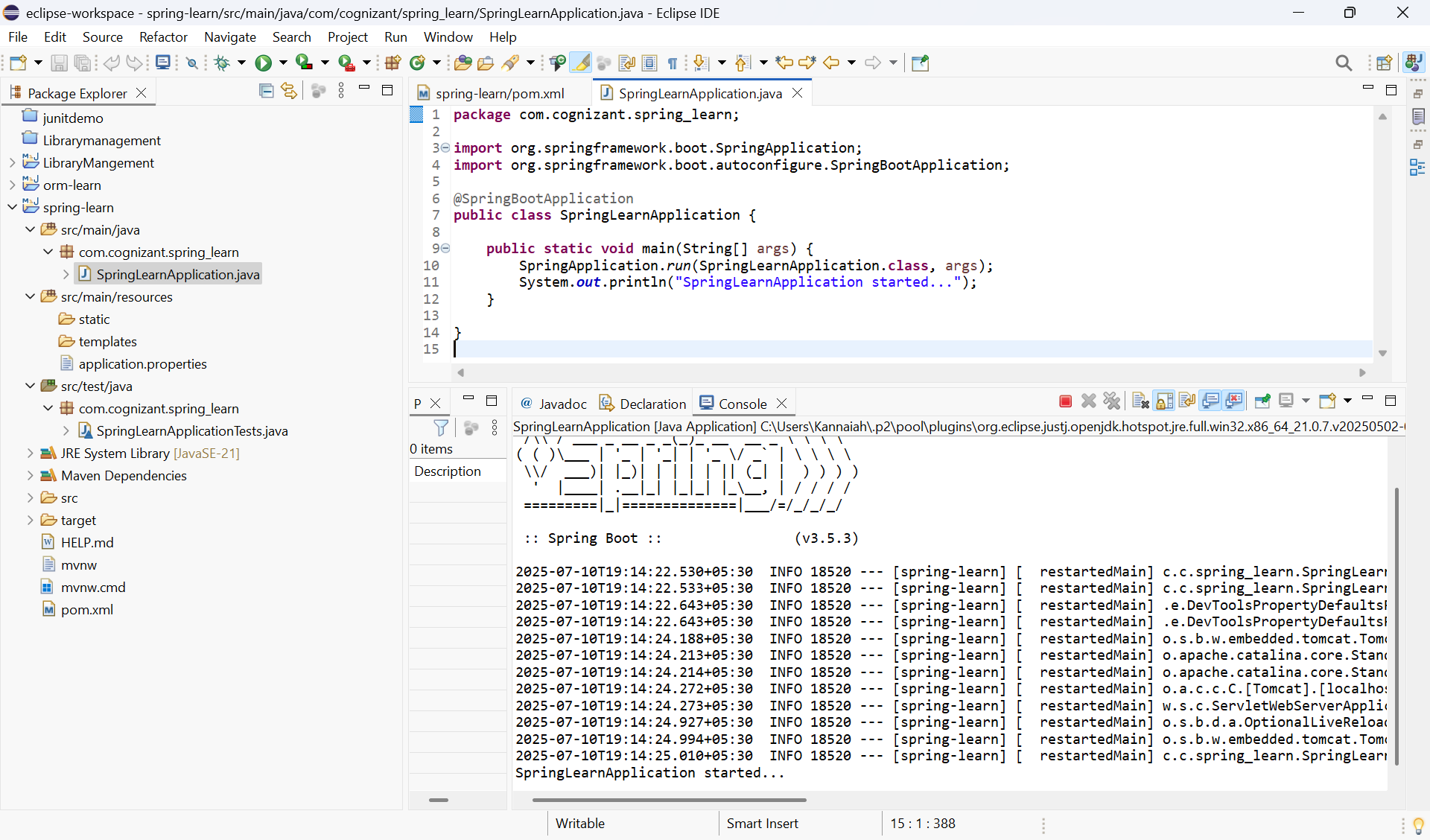
**Application.properties**



**Pom.xml**

****

**Output:**



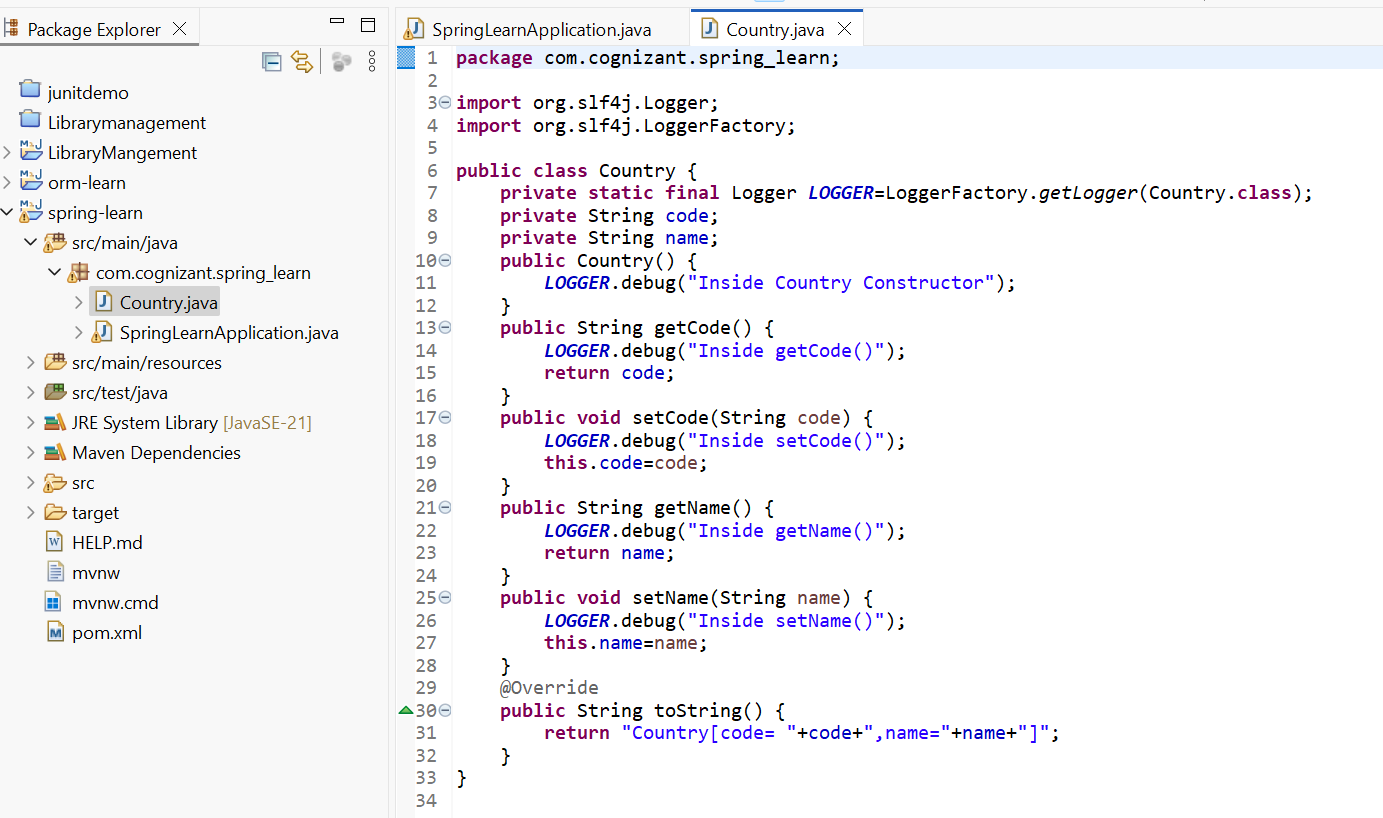
**Hands on 4**

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

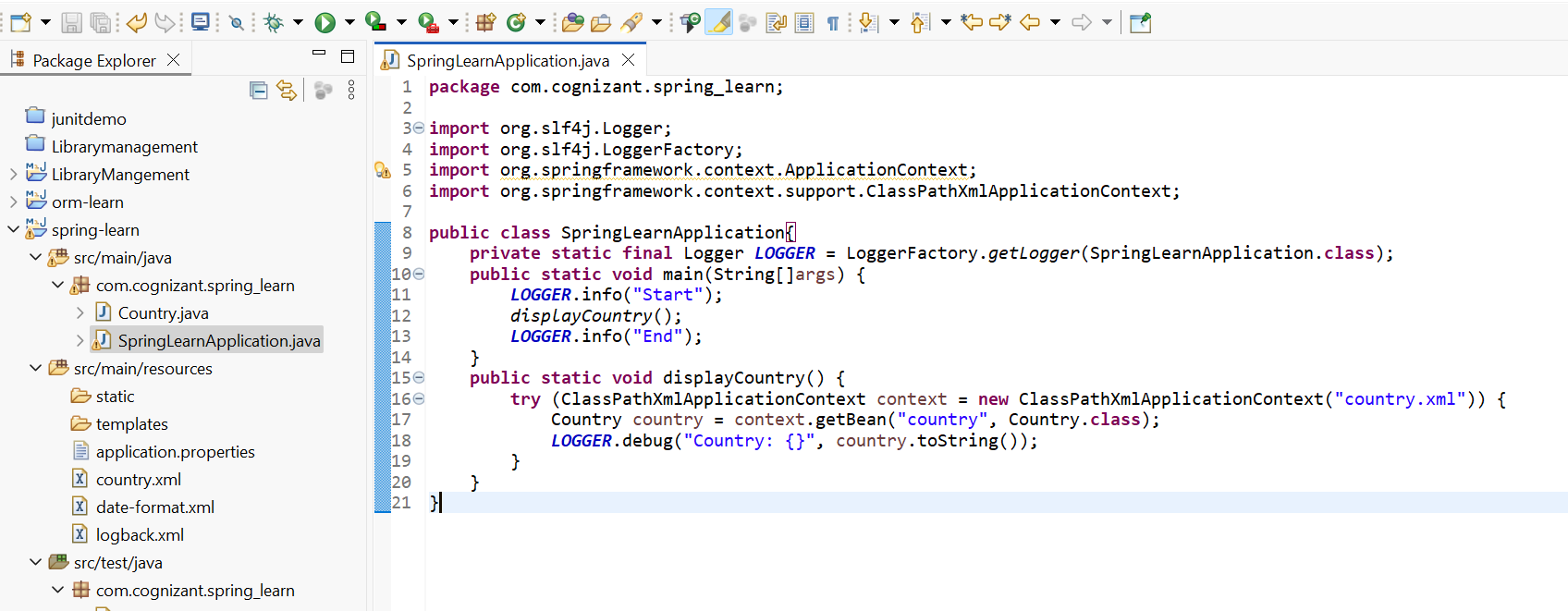
**Country.xml**

****

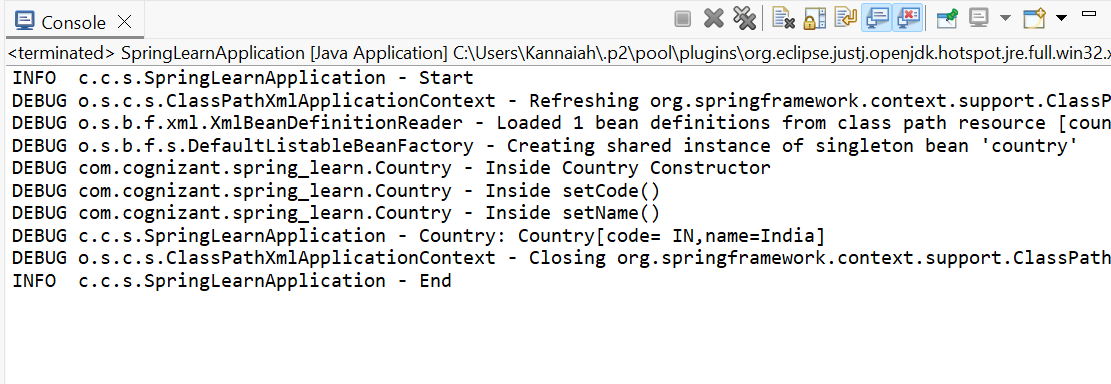
**Country.java**

****

**SpringLearnApplication.java**

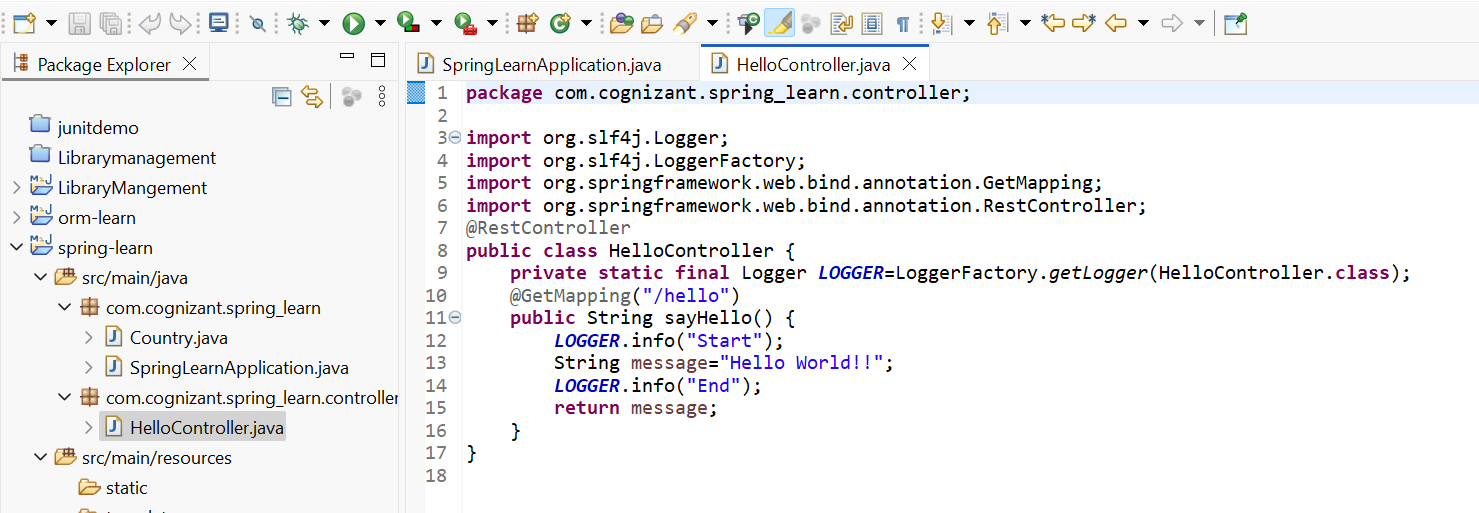
****

**Output:**

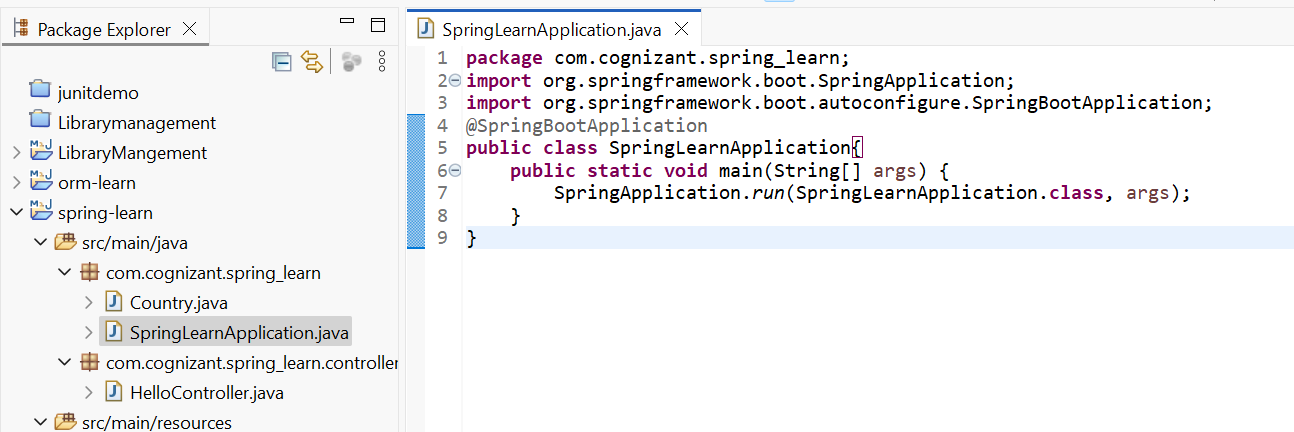
****

**Hello World RESTful Web Service**   
  
Write a REST service in the spring learn application created earlier, that returns the text "Hello World!!" using Spring Web Framework. Refer details below:  
  
**Method:** GET  
**URL:** /hello  
**Controller:** com.cognizant.spring-learn.controller.HelloController  
**Method Signature:** public String sayHello()  
**Method Implementation:** return hard coded string "Hello World!!"  
**Sample Request**: http://localhost:8083/hello  
**Sample Response:** Hello World!!

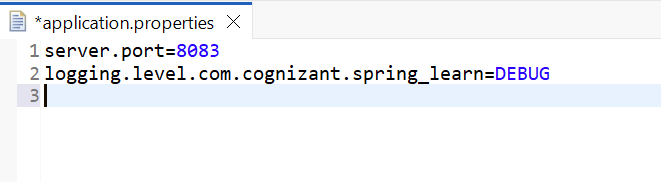
**HelloController.java**

****

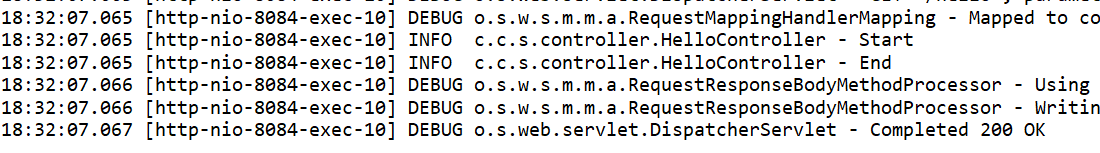
**SpringLearnApplication.java**

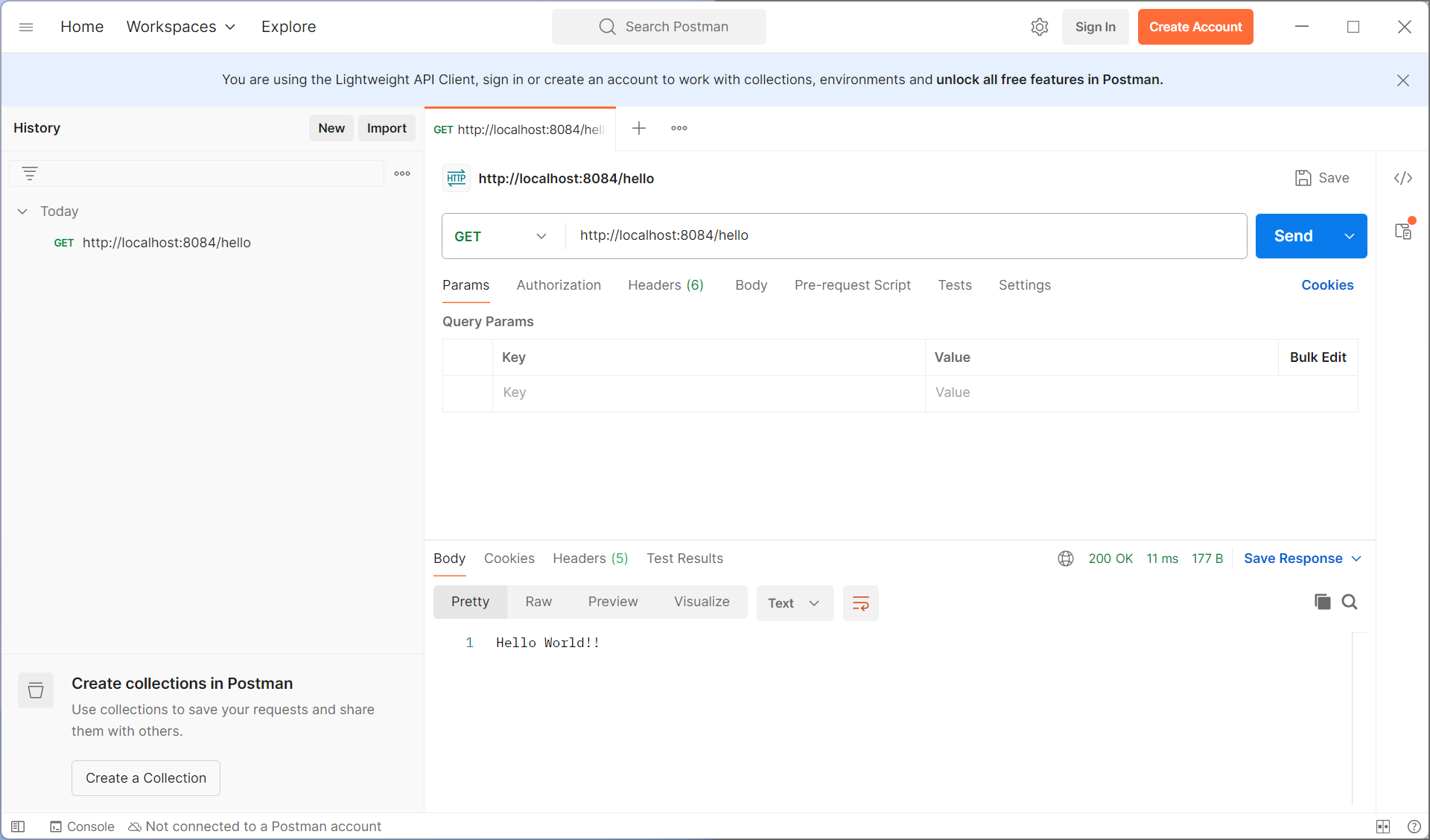
****

**application.properties**

****

**Output**:





**REST - Country Web Service**

Write a REST service that returns India country details in the earlier created spring learn application.  
  
**URL**: /country  
**Controller**: com.cognizant.spring-learn.controller.CountryController  
**Method Annotation**: @RequestMapping  
**Method Name**: getCountryIndia()  
**Method Implementation**: Load India bean from spring xml configuration and return  
**Sample Request**: http://localhost:8083/country  
**Sample Response**:

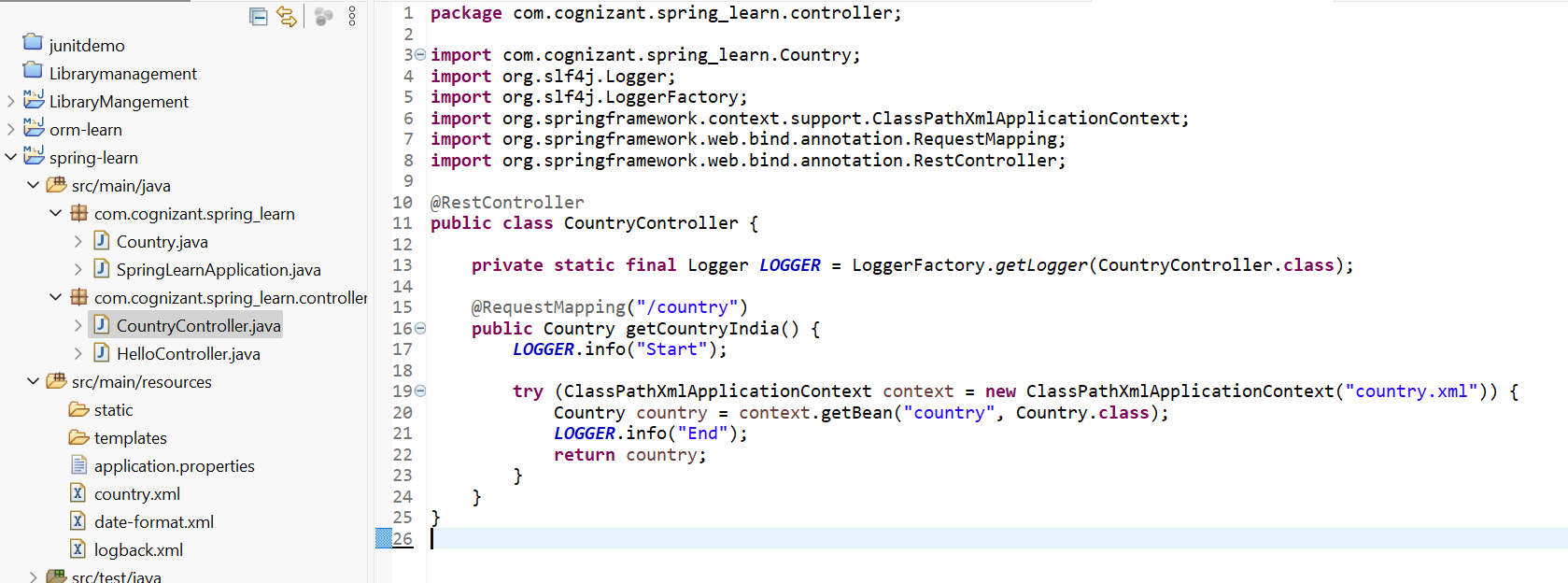
{

  "code": "IN",

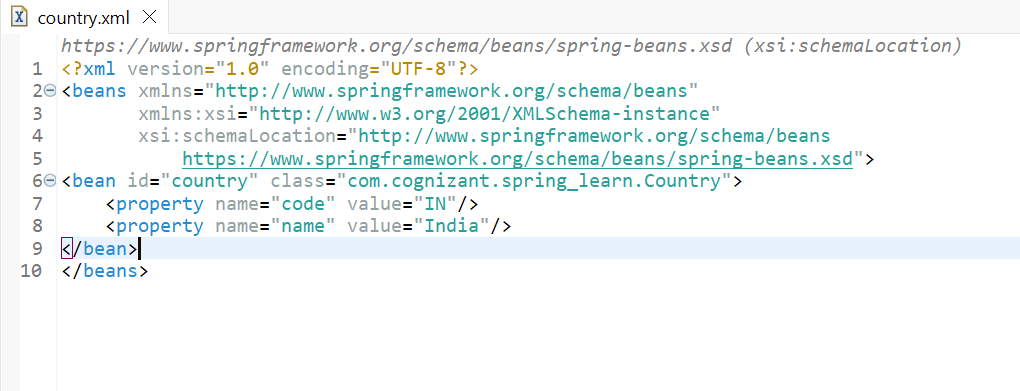
  "name": "India"

}

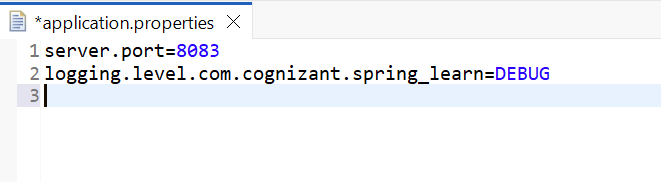
**CountryController.java**

****

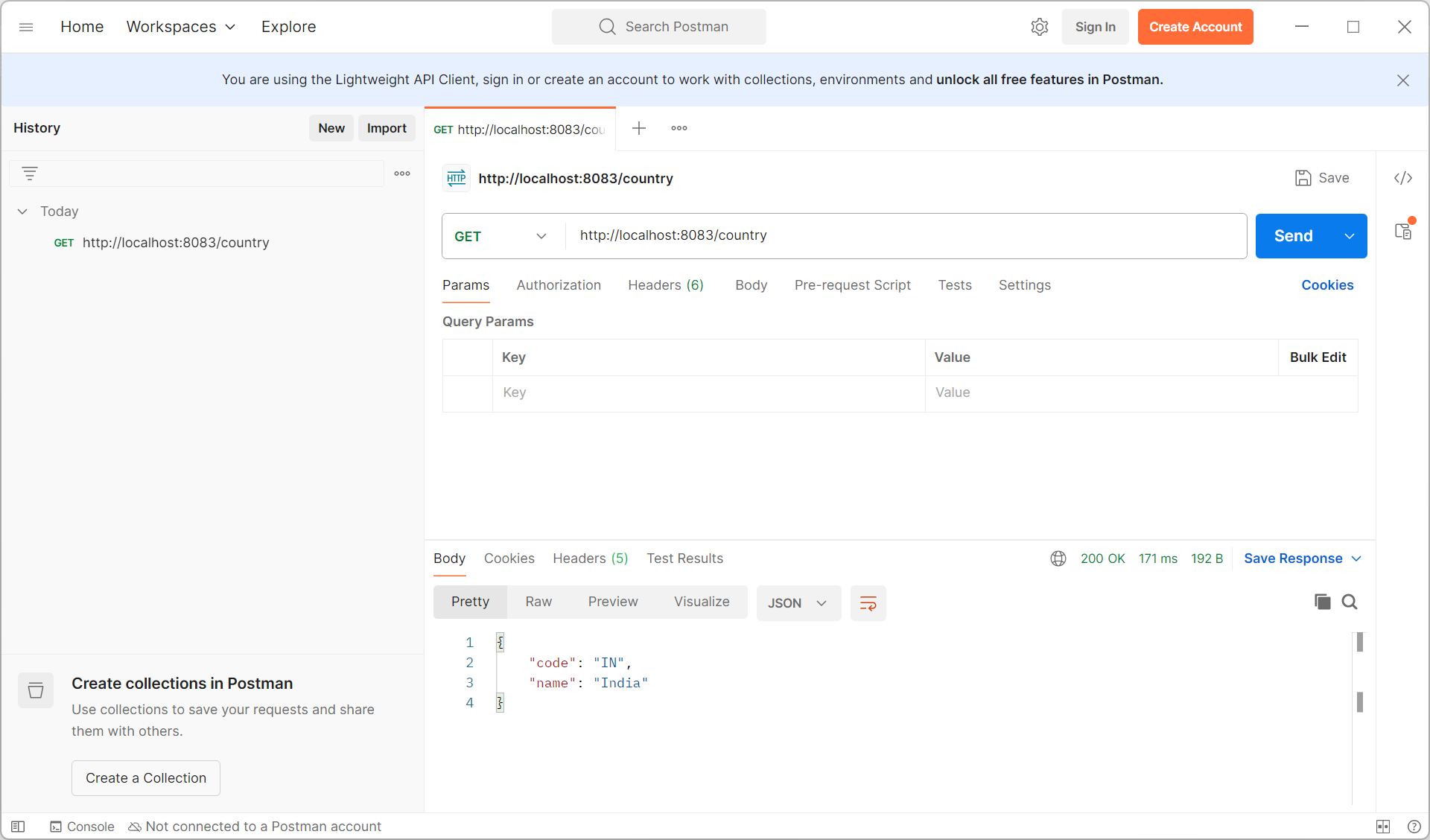
**Country.xml**

****

**Application. Properties**

****

**Output:**

****

**REST - Get country based on country code**   
  
Write a REST service that returns a specific country based on country code. The country code should be case insensitive.  
  
**Controller**: com.cognizant.spring-learn.controller.CountryController  
**Method Annotation:** @GetMapping("/countries/{code}")  
**Method Name**: getCountry(String code)  
**Method Implementation**: Invoke countryService.getCountry(code)   
**Service Method:**com.cognizant.spring-learn.service.CountryService.getCountry(String code)  
  
**Service Method Implementation**:

* Get the country code using @PathVariable
* Get country list from country.xml
* Iterate through the country list
* Make a case insensitive matching of country code and return the country.
* Lambda expression can also be used instead of iterating the country list

**Sample Request**: http://localhost:8083/country/in  
  
**Sample Response**:

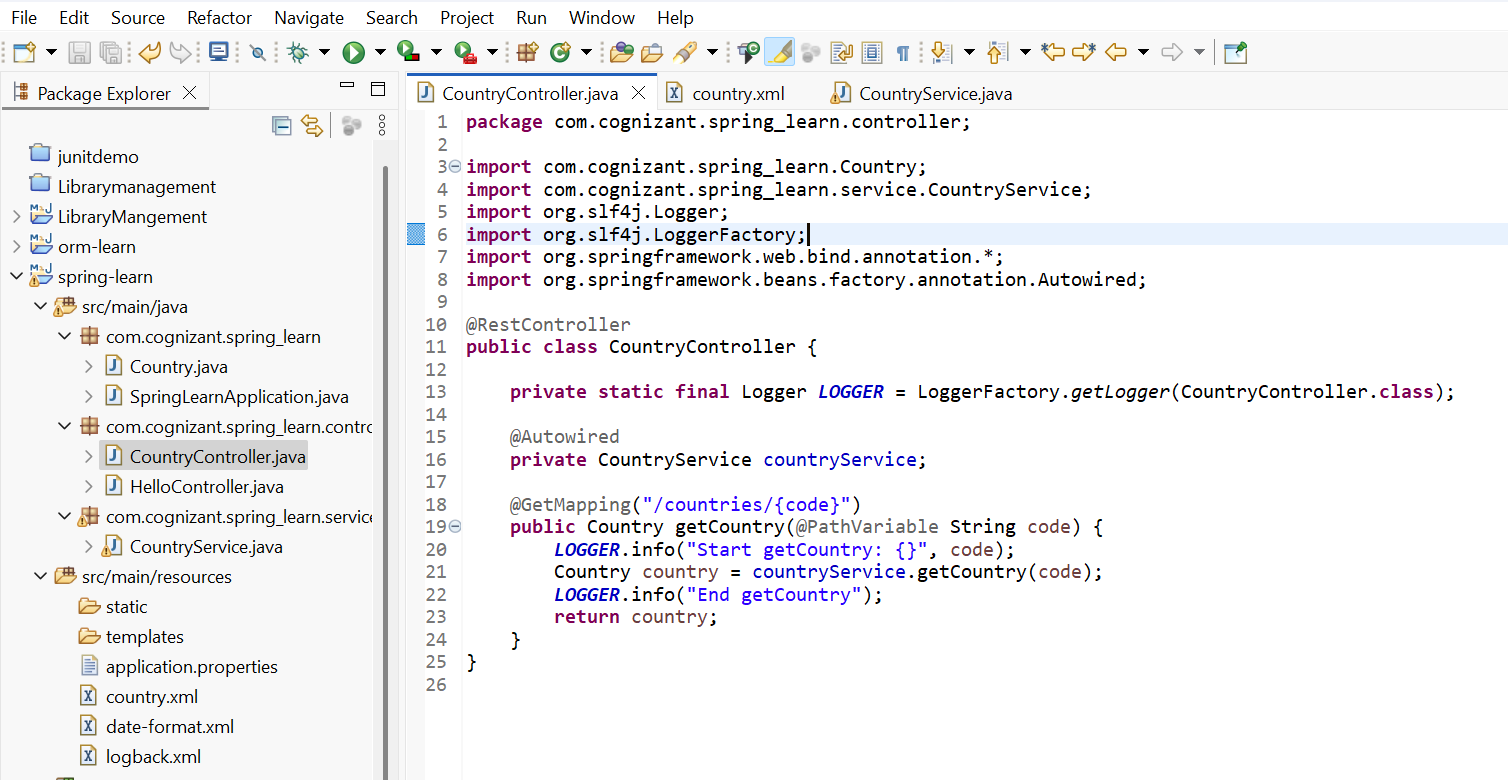
{

  "code": "IN",

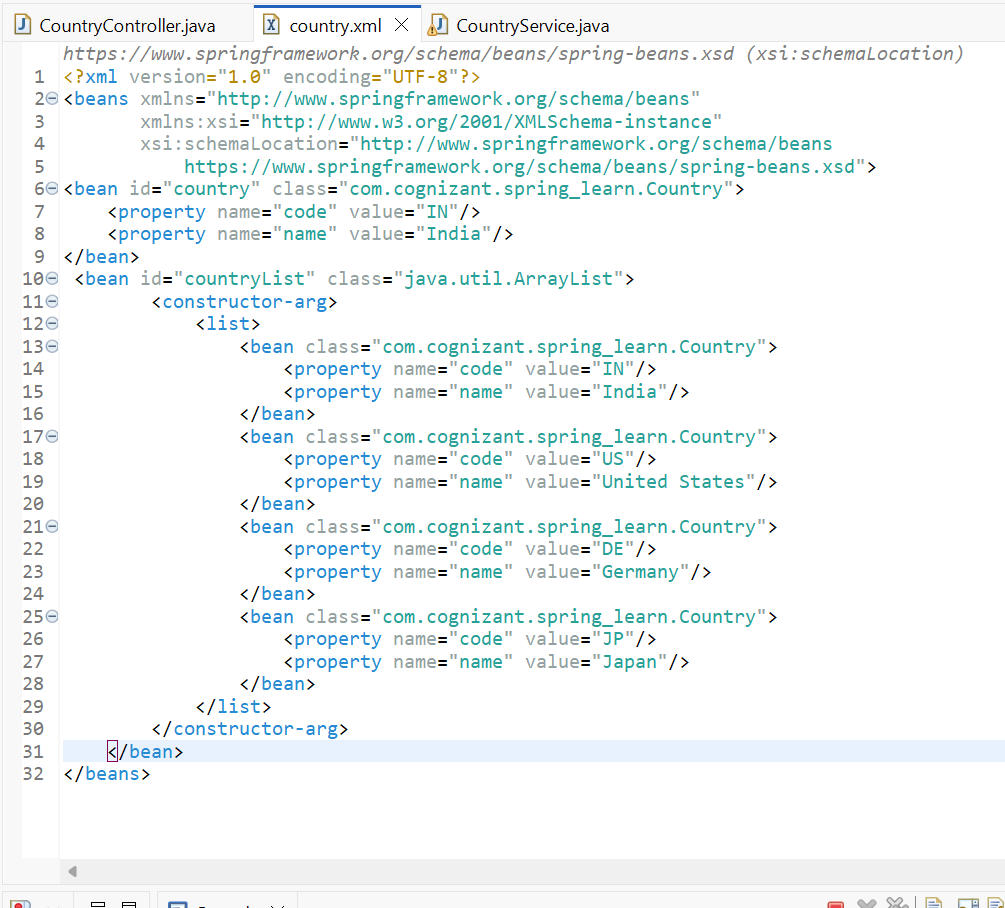
  "name": "India"

}

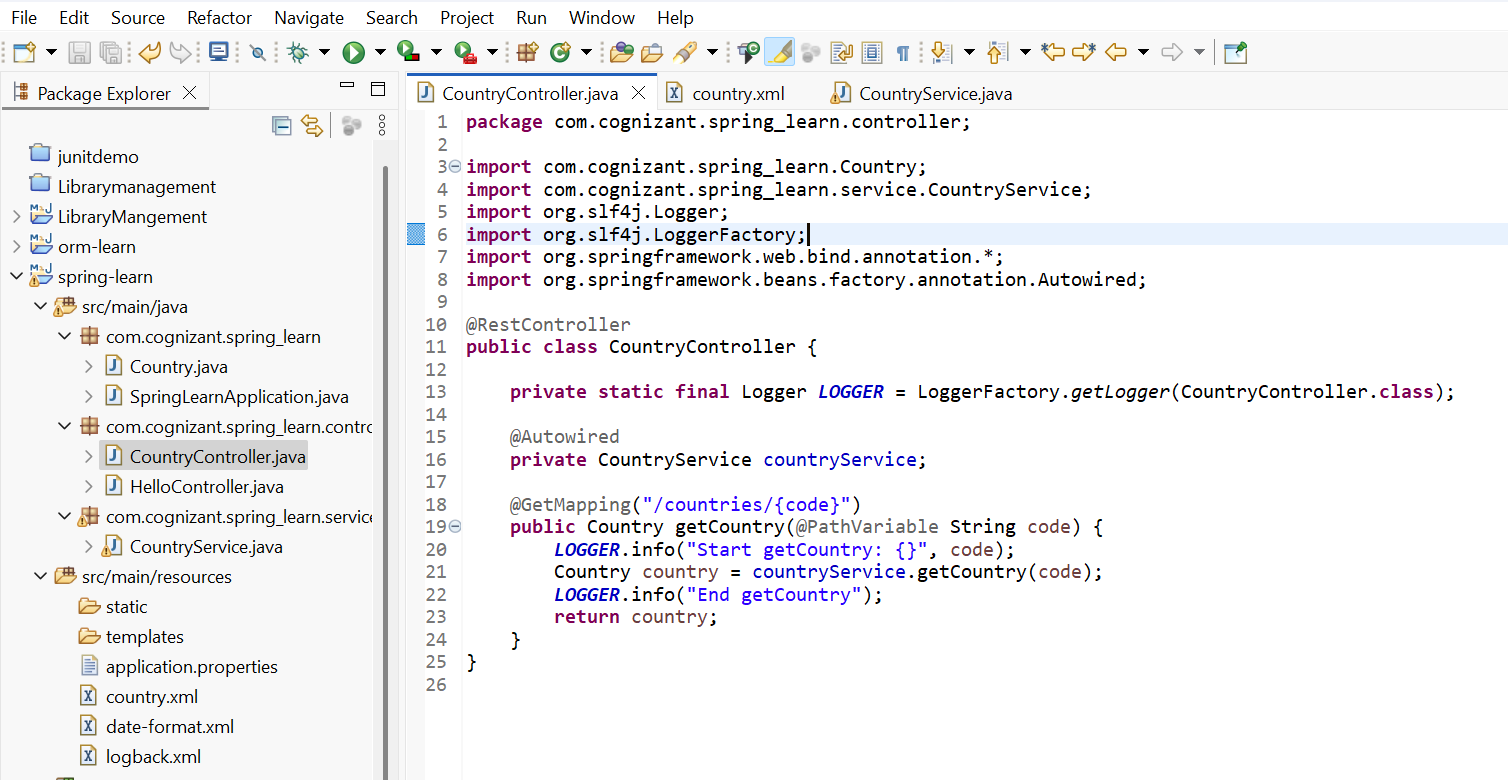
**CountryController.java**

****

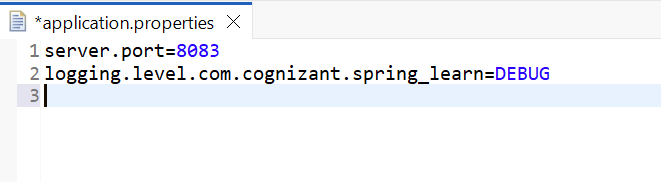
**Country.xml**

****

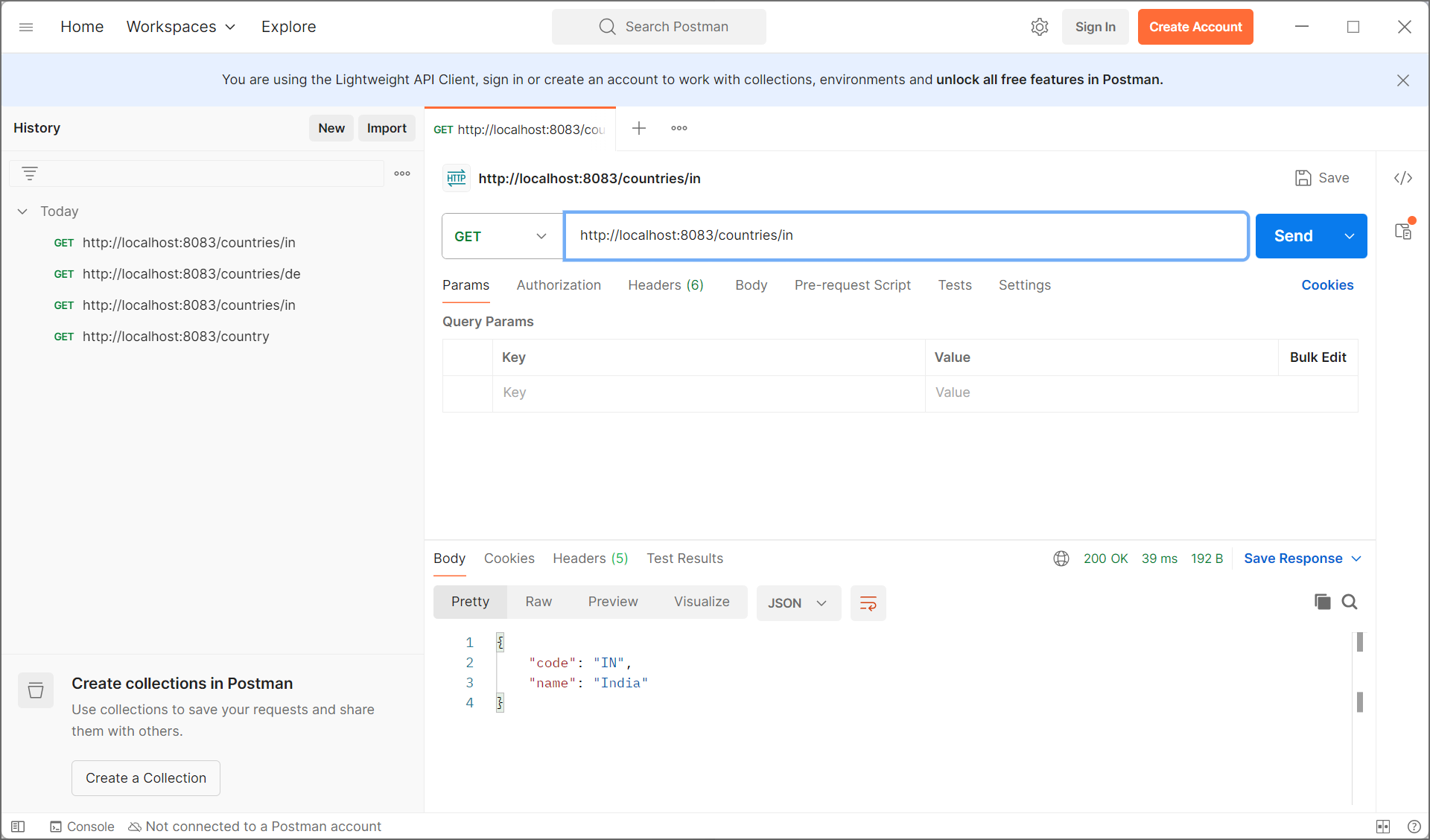
**CountryController.java**

****

**Application. Properties**

****

**Output:**

****

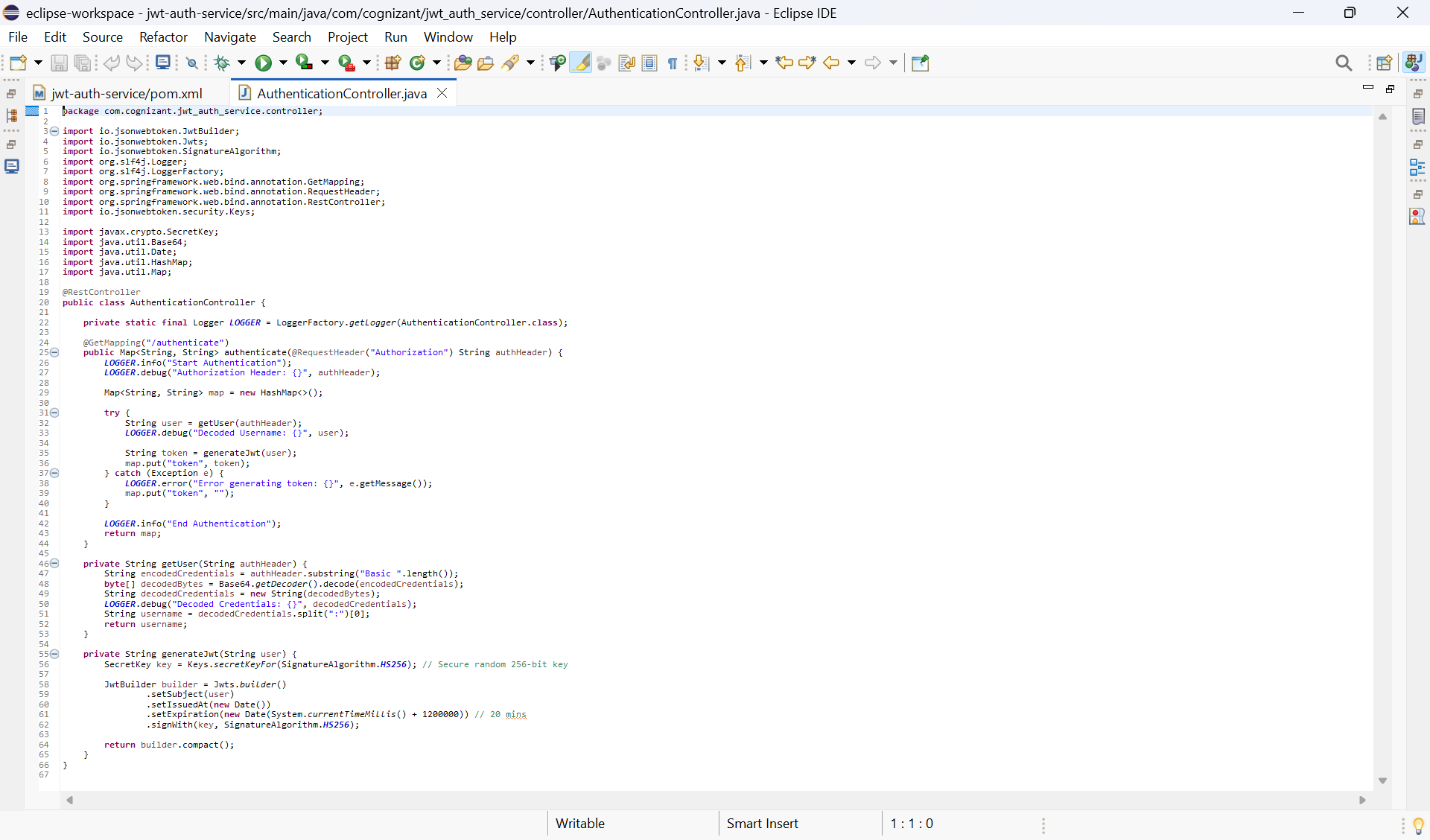
**Create authentication service that returns JWT**   
  
As part of first step of JWT process, the user credentials needs to be sent to authentication service request that generates and returns the JWT.  
  
Ideally when the below curl command is executed that calls the new authentication service, the token should be responded. Kindly note that the credentials are passed using -u option.  
  
**Request**

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

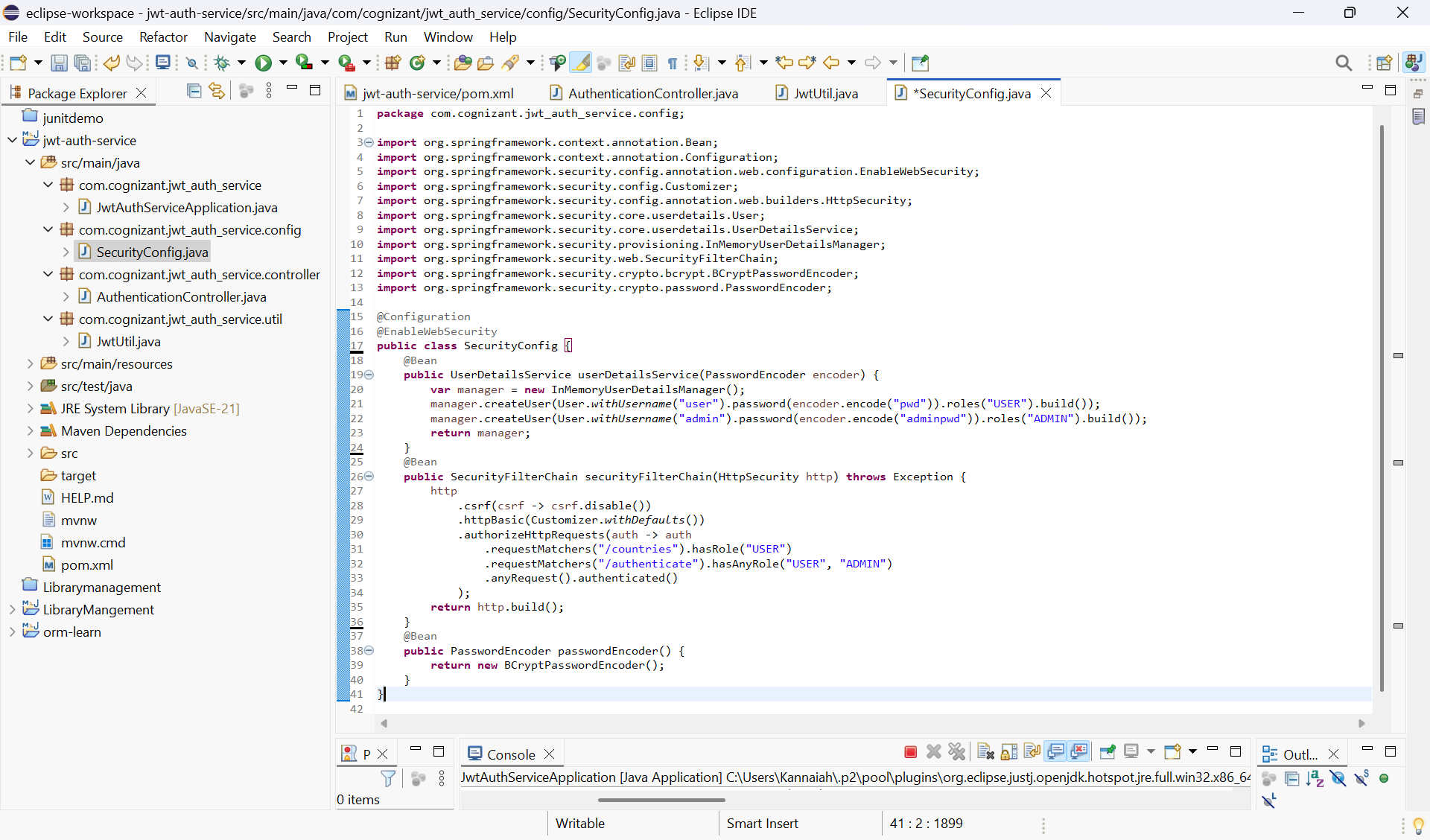
**Response**

{"token":"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNTcwMzc5NDc0LCJleHAiOjE1NzAzODA2NzR9.t3LRvlCV-hwKfoqZYlaVQqEUiBloWcWn0ft3tgv0dL0"}

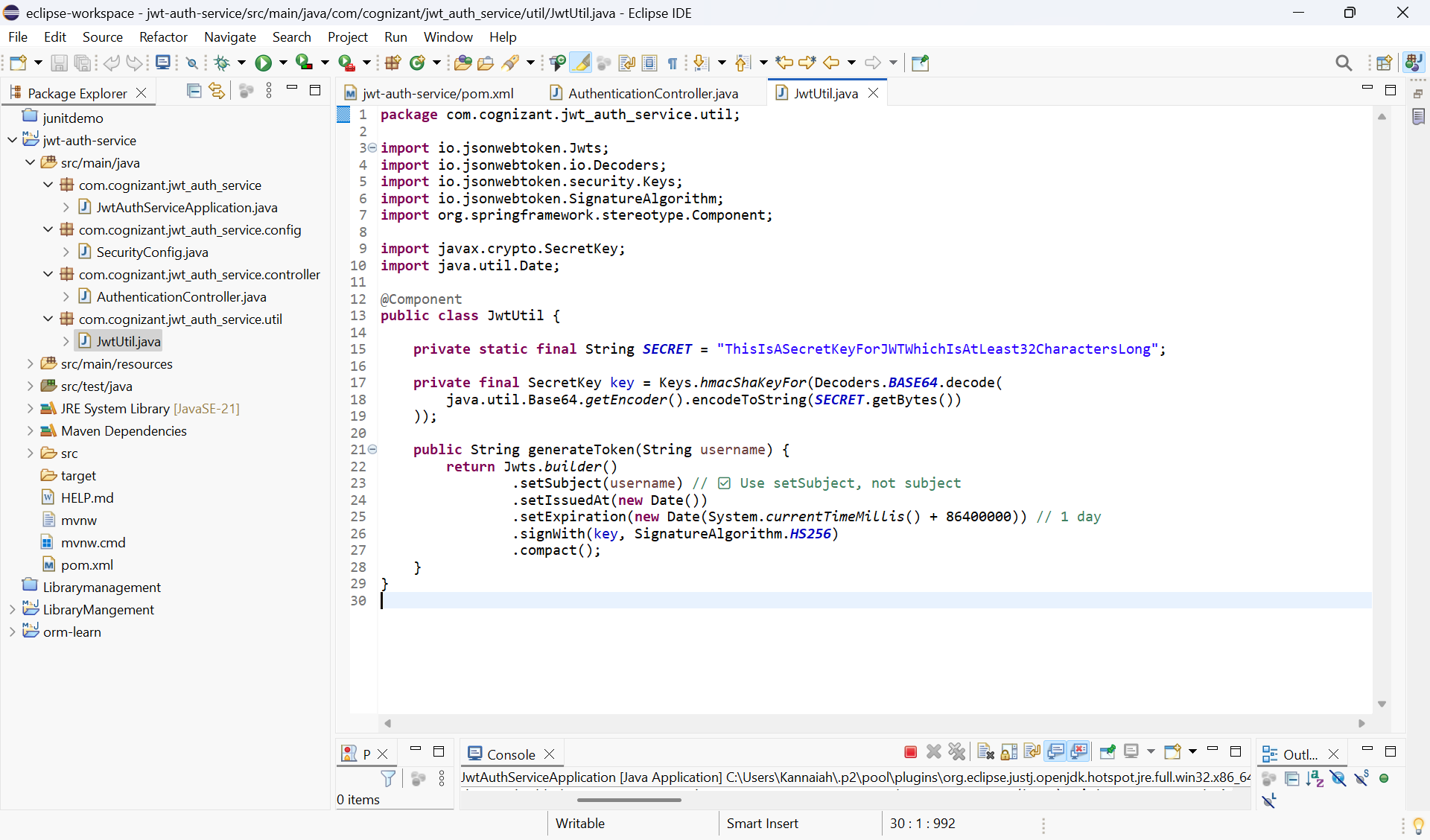
**AuthenticationController.java**

****

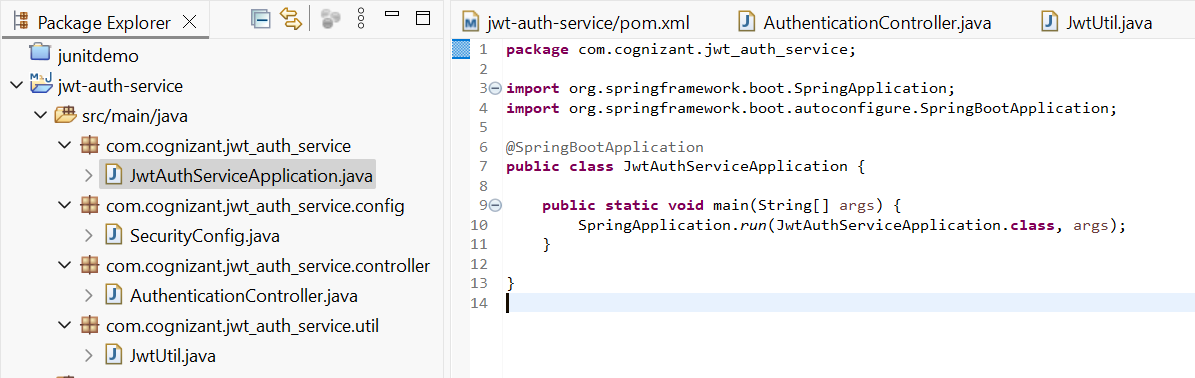
**SecurityConfig.java**

****

**Jwutil.java**



**JwtAuthServiceApplication.java**



**Output**:

