Hands-On 4 – Spring Core: Load Country from Spring XML Configuration

# 1. Objective

To store country information using Spring XML configuration and load it as a Spring bean to demonstrate Dependency Injection and Spring container behavior.

# 2. Implementation Steps

## Step 1: Create country.xml in src/main/resources

Define a bean for a country with ISO code and name.

<?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  
 https://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>

## Step 2: Create Country.java Class

Include:

- Private instance variables: code, name

- Default constructor with log: 'Inside Country Constructor.'

- Getters and setters with debug logs

- Overridden toString() method

## Step 3: Modify SpringLearnApplication.java

Add displayCountry() method and call it in main().

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

# 3. Output in Console

Inside Country Constructor.  
Setter called for code  
Setter called for name  
Getter called for code  
Getter called for name  
Country : Country [code=IN, name=India]

# 4. Explanation of Key Concepts

## bean tag

Defines a Spring-managed component.

## id attribute

Unique identifier to retrieve the bean from the container.

## class attribute

Fully qualified class name to instantiate the bean.

## property tag

Used to inject values into fields via setter methods.

## name attribute (in property)

Indicates which setter method to call.

## value attribute

The literal value to be passed to the setter.

## ApplicationContext & ClassPathXmlApplicationContext

- ApplicationContext is the Spring container interface.  
- ClassPathXmlApplicationContext loads the context definition from XML file(s) located in the classpath.

## What Happens When You Load XML

1. Spring reads and parses the XML configuration.  
2. Instantiates the bean class using its default constructor.  
3. Injects the values using setter methods (via property tags).  
4. Returns the bean when getBean() is called.