# Hands-on 2 : Spring Core – Load Country from Spring Configuration XML

This hands-on demonstrates how to configure and load a simple Spring bean using XML configuration. An airlines website supports bookings from multiple countries and requires the country data to be stored and accessed via Spring.

## Task Description:

Store the details of a country (ISO Code and Name) in an XML configuration and retrieve it in a Spring Boot application using ApplicationContext.

## Steps Implemented:

* - Picked country: IN (India)
* - Created `country.xml` in resources with a bean definition:  
  <bean id="country" class="com.cognizant.spring\_learn.Country">  
   <property name="code" value="IN" />  
   <property name="name" value="India" />  
  </bean>
* **- Created `Country.java`:**

package com.cognizant.spring\_learn;  
  
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;  
  
 // Constructor (with debug log)  
 public Country() {  
 *LOGGER*.debug("Inside Country Constructor.");  
 }  
  
 // Getters & Setters (with debug logs)  
 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;  
 }  
  
 // toString()  
 @Override  
 public String toString() {  
 return "Country [code=" + code + ", name=" + name + "]";  
 }  
}

* **SpringLearnApplication,java:**

package com.cognizant.spring\_learn;  
  
  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(SpringLearnApplication.class);  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 *displayCountry*();  
 }  
  
 public static void displayCountry() {

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
  
 // Retrieve the country bean  
 Country country = context.getBean("country", Country.class);  
  
 // Log country details  
 *LOGGER*.debug("Country : {}", country.toString());  
 }  
}

## Conclusion:

The Spring Boot application was successfully configured to load and log a simple Country bean from an XML file using ApplicationContext. This completes Hands-on 4 as per the provided instructions.

