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

**SME-Walkthrough**

1. **bean tag**
   * Used to define a Spring bean (an object managed by Spring container).
   * Example:
   * <bean id="country" class="com.cognizant.spring\_learn.Country">  
      <property name="code" value="IN" />  
      <property name="name" value="India" />  
     </bean>
2. **id attribute**
   * Unique name for the bean in Spring context.
   * Accessed using context.getBean("id").
3. **class attribute**
   * Fully qualified name of the Java class for the bean.
   * Must match the actual package name.
4. **property tag**
   * Used to inject values into the bean’s properties (fields) via setter methods.
5. **name attribute (of property tag)**
   * Specifies which property (field) to set in the bean.
   * Should match the setter method, e.g., setCode() for name=“code”.
6. **value attribute (of property tag)**
   * The value to inject into the property.
7. **ApplicationContext**
   * Interface for accessing Spring container.
   * Loads the beans and provides them via getBean().
8. **ClassPathXmlApplicationContext**
   * Implementation of ApplicationContext.
   * Loads Spring configuration XML from classpath.
   * Example:
   * ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
9. **What happens when context.getBean() is invoked?**
   * Spring searches the XML for a bean with the given ID.
   * It creates an object of the class (if not already created).
   * Injects all properties via setters.
   * Returns the fully initialized object.
10. **Debug Example Flow**

* Constructor Log: “Inside Country Constructor”
* Setter Logs: “Inside setCode”, “Inside setName”
* Output from LOGGER.debug(...):
* Country : Country [code=IN, name=India]