**SME Walkthrough Points — explained**

**bean tag**

xml

<bean id="country" class="com.cognizant.springlearn.Country">

...

</bean>

* **bean tag**: Defines an **object managed by Spring IoC container**.
* **id attribute**: Unique name to identify the bean (country).
* **class attribute**: Fully qualified class name to instantiate (Country).

**property tag**

xml

CopyEdit

<property name="code" value="IN" />

* **property tag**: Tells Spring which **setter method** to call.
* **name attribute**: Which property to set (Spring calls setCode()).
* **value attribute**: The value to set (Spring passes "IN").

**ApplicationContext & ClassPathXmlApplicationContext**

* **ApplicationContext**: Spring’s **IoC container interface** — it holds beans and their configurations.
* **ClassPathXmlApplicationContext**: An implementation of ApplicationContext that loads beans from an **XML file in the classpath**.

So:

{ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");}

means *“Hey Spring, look for country.xml in my src/main/resources and load all beans defined there.”*

**What happens on context.getBean("country")?**

This is **crucial** for interviews & understanding Spring:

1] ClassPathXmlApplicationContext reads the XML.  
2] It **scans the <bean> tags**.  
3] It **creates an instance** of Country using the **default constructor** (Inside Country Constructor).  
4] It **injects properties** using setters:

* Calls setCode("IN")
* Calls setName("India")

5] When you call context.getBean("country"):

* It **retrieves the ready-made bean** from the container.
* If it’s a singleton (default), it gives you **the same instance** every time.