

Case study:

Create Product POJO class - product name, product description , Create Beans.xml file and configure that in App.java

Code:

//product.java

**package** com.example.spring\_product;

**public class** Product {

**private** String name;

**private** String description;

**public** Product() {

    }

**public** String getName() {

**return** name;

    }

**public void** setName(String name) {

**this**.name = name;

    }

**public** String getDescription() {

**return** description;

    }

**public void** setDescription(String description) {

**this**.description = description;

    }

```

        public void display() {
            System.out.println("Product Name: " + name);
            System.out.println("Product Description: " + description);
        }
    }

// SpringProductApplication
package com.example.spring_product;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringProductApplication {

    public static void main(String[] args) {
        ApplicationContext context = new
ClassPathXmlApplicationContext("beans.xml");

        Product product = (Product) context.getBean("productBean");
        product.display();

    }

}

//beans.xml
<?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

```

*<http://www.springframework.org/schema/beans/spring-beans.xsd>*>

<!-- Define Product Bean -->

<bean id="productBean" class="com.example.spring\_product.Product">

    <property name="name" value="Android Phone"/>

    <property name="description" value="Android phone with long battery life and fast performance."/>

</bean>

</beans>

Output:

Product Name: Android Phone

Product Description: Android phone with long battery life and fast performance.