Using factory-bean

In Spring, you can configure beans to be created using factory instance methods and specify them in the XML configuration file. This approach allows you to delegate the creation of beans to factory methods on existing objects or static factory methods.

Here's how you can achieve this:

1. Define a class with a factory method that creates instances of the bean:

```java
public class MyFactory {
    public MyBean createMyBean() {
        // Create and configure the instance of MyBean
        return new MyBean();
    }
}
```

2. In your Spring XML configuration file (e.g., `applicationContext.xml`), define the factory bean and the bean that it produces:

```xml
<bean id="myFactory" class="com.example.MyFactory" />

<bean id="myBean" factory-bean="myFactory" factory-method="createMyBean" />
```

In the above XML configuration:

- The `<bean>` element with `id="myFactory"` defines the factory bean. It specifies the class `com.example.MyFactory`, which contains the factory method.
- The `<bean>` element with `id="myBean"` specifies the bean that is created by the factory method. The `factory-bean` attribute refers to the factory bean (`myFactory`), and the `factory-method` attribute specifies the factory method (`createMyBean`).

When the application context is initialized, Spring will invoke the factory method `createMyBean()` on the `MyFactory` bean to create the instance of `MyBean`. This allows you to customize the bean creation process and potentially perform additional configuration or logic within the factory method.