

Spring Configuration with Java Code



Java Configuration

- Instead of configuring Spring container using XML
- Configure the Spring container with Java code

No XML!

3 Ways of Configuring Spring Container

1. Full XML Config

```
<!-- define the dependency -->
<bean id="myFortuneService"
      class="com.Luv2code.springdemo.HappyFortuneService">
</bean>

<bean id="myCoach"
      class="com.Luv2code.springdemo.TrackCoach">

    <!-- set up constructor injection -->
    <constructor-arg ref="myFortuneService" />
</bean>

<bean id="myCricketCoach"
      class="com.Luv2code.springdemo.CricketCoach">

    <!-- set up setter injection -->
    <property name="fortuneService" ref="myFortuneService" />
```

2. XML Component Scan

```
<context:component-scan base-package="com.Luv2code.springdemo" />
```

3. Java Configuration Class

```
package com.luv2code.springdemo;

import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

@Configuration
@ComponentScan("com.luv2code.springdemo")
public class SportConfig {
```

No XML!

Development Process

Step-By-Step

1. Create a Java class and annotate as **@Configuration**
2. Add component scanning support: **@ComponentScan** (optional)
3. Read Spring Java configuration class
4. Retrieve bean from Spring container

Step 1: Create a Java class and annotate as @Configuration

```
@Configuration  
public class SportConfig {  
}
```

Step 2: Add component scanning support: `@ComponentScan`

Optional

```
@Configuration  
{@ComponentScan("com.luv2code.springdemo")  
public class SportConfig {  
  
}}
```

Step 3: Read Spring Java configuration class

```
AnnotationConfigApplicationContext context =  
    new AnnotationConfigApplicationContext(SportConfig.class);
```

Step 4: Retrieve bean from Spring container

```
Coach theCoach = context.getBean("tennisCoach", Coach.class);
```