Annotation Configuration Using XML

Bryan Hansen

DIRECTOR OF SOFTWARE DEVELOPMENT

@bh5k

applicationContext.xml

```
<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">
```

Xml Declaration

```
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="...
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-
context.xsd">

<context:annotation-config/>

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

Stereotype Annotations

@Component, @Service, @Repository

Semantically the same

@Component - any POJO

@Service - business logic layer

@Repository - data layer

Autowire

Better with Annotations

Tied to location

Member Variables

Constructor

Setter

Autowired Member Variable

```
@Autowired
private CustomerRepository customerRepository;
```

Autowired Setter Injection

```java
private CustomerRepository customerRepository;

@Autowired
public void setCustomerRepository(CustomerRepository
customerRepository) {

  System.out.println("We are using setter injection");
  this.customerRepository = customerRepository;
}
```

Autowired Constructor Injection

```
@Autowired
public CustomerServiceImpl(CustomerRepository customerRepository) {

  System.out.println("We are using constructor injection");
  this.customerRepository = customerRepository;
}
```

JSR-330

Dependency Injection for Java

Beyond our scope

Why Spring?

Summary

applicationContext.xml

Component Scanner

Stereotype Annotations

Autowired

JSR-330