

XML Configuration

Spring Fundamentals

Bryan Hansen

twitter: bh5k

<http://www.linkedin.com/in/hansenbryan>



pluralsight 
hardcore developer training

Configuration



The applicationContext.xml

- Doesn't have to be named applicationContext.xml
 - More of a loose standard
- A simple view of Spring is that it is Hashmap of objects
 - Objects are name/value pairs
- Although not the intention of Spring, it can be used as a simple Registry
- XML configuration begins with a file named the applicationContext.xml
- There are namespaces that aid in configuration and validation

```
<?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
```

Xml Configuration

- **Beans**

- Essentially classes
- Defining beans replaces using the keyword “new”
- Define the class, use the interface

- **Constructor Args**

- Used to reference properties of the constructor

- **Properties**

- Getters and Setters of the POJO that we are working with

- **References**

- References to other beans that we have defined

- **Values**

- Basic primitive values that we are setting on our POJO

Beans

- **Id or Name**

- Can be used interchangeably
- Id has to be a valid XML identifier
 - Can't contain special characters, ie: "*", "/", "."
- Name can contain special characters
 - Often doesn't matter with just Spring, but when building URLs with Spring MVC can be problematic

- **Default No-Args Constructor**

- Setter Injection VS Constructor Injection

- **Class**

```
<bean name="customerRepository"  
      class="com.pluralsight.repository.HibernateCustomerRepositoryImpl">  
</bean>
```

Setter VS Constructor Injection

- Setter injection is more common

```
<bean name="customerService" class="com.pluralsight.service.CustomerServiceImpl">  
  <property name="customerRepository" ref="customerRepository"></property>  
</bean>
```

- Constructor injection guarantees the contract

```
<bean name="customerService" class="com.pluralsight.service.CustomerServiceImpl">  
  <constructor-arg index="0" ref="customerRepository" />  
</bean>
```

Autowiring

- **Spring can automatically wire beans together for you**
- **byType**
 - Allows a property to be autowired if exactly one bean of the property type exists in the container. If more than one exists, a fatal exception is thrown, which indicates that you may not use byType autowiring for that bean. If there are no matching beans, nothing happens; the property is not set.
- **byName**
 - Autowiring by property name. Spring looks for a bean with the same name as the property that needs to be autowired.
- **constructor**
 - Analogous to byType, but applies to constructor arguments. If there is not exactly one bean of the constructor argument type in the container, a fatal error is raised.
- **no**
 - No Autowiring.

Summary

- `applicationContext.xml`
- Bean definition
- Setter Injection
- Constructor Injection
- Autowiring

