

Java Configuration

Spring Fundamentals

Bryan Hansen

twitter: bh5k

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



pluralsight 
hardcore developer training

Configuration



No applicationContext.xml

- Early Spring development got some pushback for having so much XML
- Later versions included namespaces to aid in development, but developers still wanted less/no xml
- Enter Java Configuration
- As of version 3.0 most features of Spring can be configured using Java Configuration

@Configuration

- The Java file(s) that have the @Configuration annotation replace the applicationContext files
- Methods used in conjunction with the @Bean annotation are used to get instances of Spring Beans
- @Configuration is a class level annotation:

```
@Configuration  
public class AppConfig {
```

- @Bean is a method level annotation:

```
@Bean(name="customerService")  
public CustomerService getCustomerService() {
```

- Class and method names can be anything, Spring doesn't care

Setter Injection

- As simple as a method call
- The “mystery” of injection goes away
- Setter injection is simply calling a setter:

```
@Bean(name="customerService")
public CustomerService getCustomerService() {
    CustomerServiceImpl customerService = new CustomerServiceImpl();
    customerService.setCustomerRepository(getCustomerRepository());

    return customerService;
}
```

```
@Bean(name="customerRepository")
public CustomerRepository getCustomerRepository() {
    return new HibernateCustomerRepositoryImpl();
}
```

Constructor Injection

- Constructor injection is just like setter injection:

```
@Bean(name="customerService")
public CustomerService getCustomerService() {
    CustomerServiceImpl customerService = new CustomerServiceImpl(getCustomerRepository());
    return customerService;
}
```

```
@Bean(name="customerRepository")
public CustomerRepository getCustomerRepository() {
    return new HibernateCustomerRepositoryImpl();
}
```

Autowired

- We add a `@ComponentScan` annotation:

```
@ComponentScan({"com.pluralsight"})
```

- Just like XML configuration, mark whatever you want as `@Autowired`
 - `byName` uses the `@Bean` name
 - `byType` uses the Instance Type

Summary

- **AppConfig**
- **@Configuration**
- **@Bean**
- **Setter and Constructor Injection**
- **Autowired**

