Spring Quick Start

Eberhard Wolff
http://ewolff.com
eberhard.wolff@gmail.com

pluralsight
hardcore developer training

Introduction

- Explain enough Spring to follow the rest of the course
- Does not cover every detail
- May skip if you know the material

- Dependency Injection
- Java Config: Configure Spring using Java classes

- Annotations to make Java classes Spring beans
- @Component, @Service, @Repository

- XML configuration

- Spring test framework

Dependency Injection

- Do not create objects
- Have them inject instead

- Easier to configure

- More flexible

- Easier to test

@Autowired for Dependency Injection

- Marks a field or setter to inject an object

```
public class SimpleService {

  @Autowired
  SimpleRepository repository;

  public DomainObject service() {
    return repository.findDomainObject();
  }

}
```

- How can Spring know what to inject?
- Inject object must be a Spring Bean

Spring Bean Annotations

- @Service or @Repository mark Spring Beans
- @Component can also be used
- All annotations the same as far as Spring is concerned
- Spring Beans are Singletons
- Bean name = class name (small caps)

```
@Service
public class SimpleService {
  @Autowired
  SimpleRepository repository;
  …
}
```

```
@Repository
public class SimpleRepository {
 …
}
```

Java Config

- @Configuration marks class as Spring configuration
- @ComponentScan enables scanning in package (and subpackages)

```
@Configuration
@ComponentScan(basePackages="com.ewolff")
public class SystemConfigurationComponentScanning {

}
```

Java Config

- @Configuration marks class as Spring configuration
- @Bean annotated methods create Spring Beans
- Spring Beans are still singletons
- Bean name = method name

```
@Configuration
public class SystemConfigurationMethods {

        @Bean
        public SimpleService simpleService() {
                return new SimpleService();
        }

        @Bean
        public SimpleRepository simpleRepository() {
                return new SimpleRepository();
        }

}
```

Spring Test Framework

- Starts dependency injection container
- Allows dependency injection into tests
- @RunWith enables Spring support
- @ContextConfiguration defines configuration to use

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(classes = SystemConfiguration.class)
public class ConfigurationTest {

  @Autowired
  private SimpleService simpleService;

  @Test
  public void dependencyInjectionShouldWork() {
    assertNotNull(simpleService);
  }

}
```

Spring XML Configuration

- Configure dependency injection using XML
- Define each bean with a <bean > element
- Default bean name = fully qualified class name
- …or use id / name attribute
- Enable @Autowired using <context:annotation-config />

```
<beans>

  <bean class="com.ewolff.repository.SimpleRepository" />
  <bean id="service" class="com.ewolff.service.SimpleService" />

  <context:annotation-config />

</beans>
```

Spring XML Configuration

- Use <context:component-scan /> to enable @Service, @Repository and @Component
- Also enables @Autowired

```
<beans>

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

</beans>
```

Test Using XML

- **Add location of XML configuration to @ContextConfiguration**

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("/beans.xml")
public class ConfigurationBeansTest {

  @Autowired
  private SimpleService simpleService;

  @Test
  public void dependencyInjectionShouldWork() {
    assertNotNull(simpleService);
  }

}
```

Summary

- **Dependency Injection**
- **@Autowired inject other Spring beans**
- **Java Config**
  - @Configuration for configuration class
  - @ComponentScan to scan a package for Spring beans
  - @Component, @Service, @Repository to mark Spring beans
  - @Bean method creates a Spring bean
- **XML configuration**
  - bean element defines Spring bean
  - annotation-config to enable @Autowired
  - component-scan element to scan a package for Spring beans
- **Spring test framework**
  - @RunWith enables Spring support
  - @ContextConfiguration points to XML configuration / Java Config