

Java Tabanlı Container Konfigürasyonu



Java Tabanlı Container Konfigürasyonu

- Spring **bean tanımlarının** Java sınıflarında yapılmasını sağlar
- XML tabanlı konfigürasyona **birebir karşılık** gelmektedir
- Avantajı “**type safety**” dir
- Konfigürasyon metadata'nın yazıldığı Java sınıfları **@Configuration** anotasyonu ile işaretlenmelidir

XML vs Java Konfigürasyonları

appContextConfig.xml

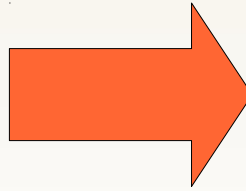
```
<beans...>
```

```
<bean id="foo" class="x.y.Foo">
  <property name="bar">
    <ref bean="bar"/>
  </property>
  <property name="baz"
ref="baz"/>
</bean>
```

```
<bean id="bar"
class="x.y.Bar"/>
```

```
<bean id="baz"
class="x.y.Baz"/>
```

```
</beans>
```



@Configuration

```
public class AppConfig {
```

@Bean

```
public Foo foo() {
    Foo foo = new Foo();
    foo.setBar(bar());
    foo.setBaz(baz());
    return foo;
}
```

@Bean

```
public Bar bar() {
    return new Bar();
}
```

@Bean

```
public Baz baz() {
    return new Baz();
}
```

```
}
```

Bağımlılıkların Enjekte Edilmesi

```
@Configuration
public class AConfig {

    @Autowired
    private Bar bar;

    @Bean
    public Foo foo() {
        Foo foo = new Foo();
        foo.setBar(bar);
        return foo;
    }
}
```

```
@Configuration
public class BConfig {

    @Bean
    public Bar bar() {
        return new Bar();
    }
}
```

Bağımlılıkların Enjekte Edilmesi

```
@Configuration  
public class AConfig {
```

```
    @Autowired
```

```
    private BConfig bConfig;
```

```
    @Bean
```

```
    public Foo foo() {  
        Foo foo = new Foo();  
        foo.setBar(bConfig.bar());  
        return foo;  
    }
```

```
}
```

```
@Configuration
```

```
public class BConfig {
```

```
    @Bean
```

```
    public Bar bar() {  
        return new Bar();  
    }
```

```
}
```

Bağımlılıkların Enjekte Edilmesi

```
@Configuration
public class AConfig {

    @Bean
    @Autowired
    public Foo foo() {
        Foo foo = new Foo();
        return foo;
    }
}
```

```
public class Foo {
    @Autowired
    private Bar bar;
}
```

```
//...
}
```

```
@Configuration
public class BConfig {

    @Bean
    public Bar bar() {
        return new Bar();
    }
}
```

Bağımlılıkların Enjekte Edilmesi

```
@Configuration
public class AConfig {

    @Bean
    public Foo foo(Bar bar) {
        Foo foo = new Foo();
        foo.setBar(bar);
        return foo;
    }
}
```

```
@Configuration
public class BConfig {

    @Bean
    public Bar bar() {
        return new Bar();
    }
}
```

ApplicationContext Yaratılması

```
AnnotationConfigApplicationContext applicationContext =  
new AnnotationConfigApplicationContext();  
  
applicationContext.register(AConfig.class, BConfig.class);  
  
applicationContext.refresh();  
  
Foo foo = applicationContext.getBean(Foo.class);
```


@ImportResource ve @Import

@ImportResource herhangi bir XML spring bean definition dosyasının yüklenmesini sağlar

```
@Configuration
@ImportResource("classpath:/appcontext/beans-config.xml")
public class AConfig {

}
```

```
@Configuration
@Import(AConfig.class)
public class BConfig {

}
```

@Import herhangi bir başka configuration sınıfının diğer bir configuration sınıfı tarafından yüklenmesini sağlar

@ComponentScan

@Component ve türevi
anotasyonların scan edileceği
paketleri tanımlar

```
@Configuration
@ComponentScan("com.javaegitimleri.petclinic")
public class AConfig {

}
```

@PropertySource

@PropertySource anotasyonu ile belirtilen resource'lar Environment'a PropertySource olarak eklenirler


```
@Configuration
@PropertySource("classpath:/application.properties")
public class AppConfig {

    @Autowired
    Environment env;

    @Bean
    public Foo foo() {
        Foo foo = new Foo();
        foo.setName(env.getProperty("foo.name"));
        return foo;
    }
}
```

@PropertySource

targetPlatform değişkeni halihazırda bu aşamaya kadar Environment'e register olmuş PropertySource nesneleri arasından resolve edilmeye çalışılır.



```
@Configuration
@PropertySource("classpath:/application_${targetPlatform}.properties")
public class AppConfig {

    @Autowired
    private Environment env;

    @Bean
    public Foo foo() {
        Foo foo = new Foo();
        foo.setName(env.getProperty("foo.name"));
        return foo;
    }
}
```

@PropertySource ve Placeholder Kullanımı

- **PropertySource** kullanırken **<bean>** tanımlarındaki veya **@Value** annotasyonundaki **\${..}** placeholder'ların resolve edilmesi de mümkündür
- Bunun için bir **PropertySourcesPlaceholderConfigurer** bean'ının register edilmesi gerekir
- **<context:property-placeholder>** annotasyonu bunu otomatik yapar
- Configuration sınıfında ise bu bean **@Bean** annotasyonu ile **statik bir factory metot içerisinde** oluşturulmalıdır

Java Tabanlı Konfigürasyon ve Bean Profil Kabiliyeti

Konfigürasyon sınıflarının hangi profillerde yükleneceği sınıf düzeyinde @Profile anotasyonu ile tanımlanır

```
@Configuration
```

```
@Profile("dev")
```

```
public class DevConfig {
```

```
    @Bean
```

```
    public DataSource dataSource() {
```

```
        return new EmbeddedDatabaseBuilder()
```

```
            .setType(EmbeddedDatabaseType.HSQL)
```

```
            .addScript("classpath:/schema.sql")
```

```
            .addScript("classpath:/data.sql")
```

```
            .build();
```

```
    }
```

```
}
```

```
@Configuration
```

```
@Profile("prod")
```

```
public class ProdConfig {
```

```
    @Bean
```

```
    public DataSource dataSource() throws
```

```
        Exception {
```

```
        Context ctx = new InitialContext();
```

```
        return (DataSource)
```

```
            ctx.lookup("java:comp/env/jdbc/DS");
```

```
    }
```

```
}
```

```
AnnotationConfigApplicationContext applicationContext = new  
AnnotationConfigApplicationContext();
```

```
applicationContext.getEnvironment().setActiveProfiles("dev");
```

```
applicationContext.scan("com.javaegitimleri");
```

```
applicationContext.refresh();
```

ApplicationContext yaratılırken aktif profillerin hangileri olacağı belirtilir

İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- info@java-egitimleri.com

