

Hazırlayan: Barış Dere
www.barisdere.com @BarisDere
baris.dere@gmail.com

Spring Framework Eđitimi

Ders 2: Temel XML konfigürasyonu kullanımı



[Http://www.youtube.com/barisdere](http://www.youtube.com/barisdere)

Ders içeriđi

- Spring IoC container tanıtımı
- Spring IoC container çalışma şekli
- Metadata konfigürasyon seçenekleri
- XML konfigürasyonu tanıtımı
- Temel XML konfigürasyonu kullanımı

Spring IoC container tanıtımı

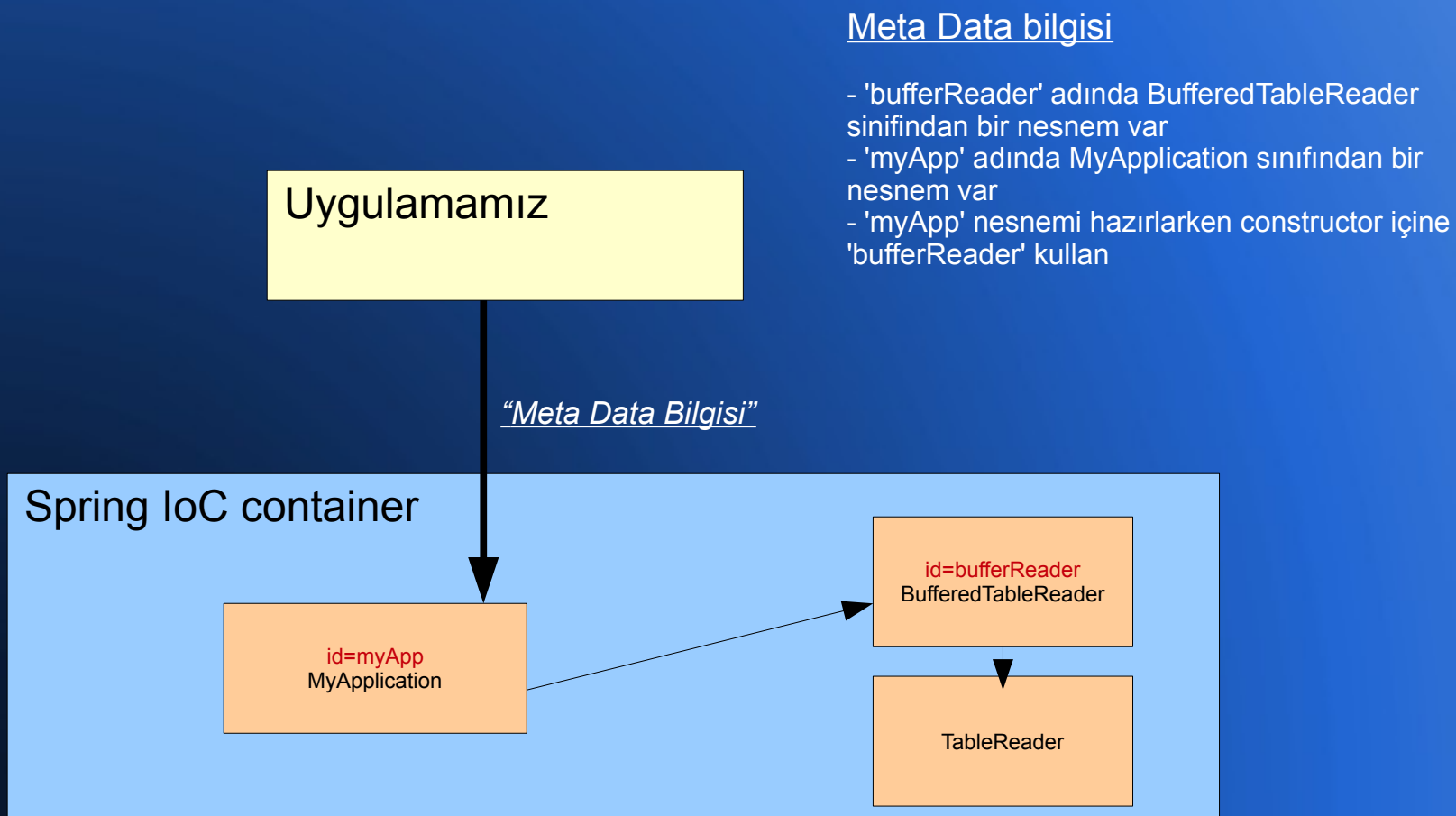
- BEAN – Spring tarafından yönetilen tüm nesnelere verilen isim
- `org.springframework.context.ApplicationContext`
 - Farklı uygulama sınıfları
 - `ClassPathXmlApplicationContext`
 - `FileSystemXmlApplicationContext`
- Metadata konfigürasyonu

Spring IoC container çalışma şekli

```
public interface TableReader {  
    void read();  
}  
  
class BufferedTableReader implements TableReader {  
    public void read() {  
    }  
}  
  
class MyApplication {  
    private TableReader tableReader;  
  
    public MyApplication(TableReader tableReader) {  
        this.tableReader = tableReader;  
    }  
  
    public void doSomethingWithTableReader() {  
    }  
}
```

```
public static void main(String[] args) {  
    TableReader reader = new BufferedTableReader();  
    MyApplication application = new  
        MyApplication(reader);  
    application.doSomethingWithTableReader();  
}  
}
```

Spring IoC container çalışma şekli (3)



Metadata konfigürasyon seçenekleri

- XML
 - Sınıf dosyaları değişmiyor
- Java
 - Sınıf dosyaları değişmiyor
 - Ek konfigürasyon sınıf(ları)
- Anotasyon
 - Sınıf dosyaları değişiyor

XML konfigürasyonu tanıtımı

my-spring-config.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-3.0.xsd">

  <bean id="..." class="...">
</bean>

  <bean id="..." class="...">
</bean>

</beans>
```

- Bir çok bean eklenebilir
- Bean id aynı xml dosyasında bir den fazla kere kullanılamaz
- Tüm bean ler default olarak Singleton olarak oluşturulur

XML konfigürasyonu tanıtımı (2)

```
<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-3.0.xsd">

  <bean id="myCustomer" class="com.spring.training.Customer">
  </bean>

</beans>
```

```
package com.spring.training;

public class Customer { }
```


XML konfigürasyonu tanıtımı (3)

```
<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-3.0.xsd">

  <bean id="myCustomer" class="com.spring.training.Customer">
  </bean>

</beans>
```

spring-config.xml

```
public static void main(String[] args) {

  // create and configure beans
  ApplicationContext context = new ClassPathXmlApplicationContext("spring-config.xml");

  // retrieve configured instance
  Customer customer = context.getBean("myCustomer", Customer.class);

}
```

Temel XML konfigürasyonu kullanımı

- Spring IoC nesne oluşturma teknikleri
 - Constructor kullanarak
 - Static Factory metodu
 - Instance Factory metodu

Temel XML konfigürasyonu kullanımı

Spring IoC nesne oluşturma teknikleri

Constructor kullanarak nesne oluşturma

```
<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-3.0.xsd">

  <bean id="myCustomer" class="com.spring.training.Customer">
  </bean>

</beans>
```

```
package com.spring.training;

public class Customer { }
```

```
=> Customer myCustomer = new Customer();
```

Temel XML konfigürasyonu kullanımı

Spring IoC nesne oluşturma teknikleri (2)

Static Factory metodu

```
<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-3.0.xsd">

  <bean id="clientService"
    class="examples.ClientService"
    factory-method="createInstance"/>

</beans>
```

```
public class ClientService {

  private static ClientService clientService = new ClientService();
  private ClientService() {}

  public static ClientService createInstance() {
    return clientService;
  }
}
```

```
=> ClientService clientService = ClientService.createInstance();
```

Temel XML konfigürasyonu kullanımı

Spring IoC nesne oluşturma teknikleri (3)

Instance Factory metodu

```
<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-3.0.xsd">

  <bean id="serviceLocator" class="examples.DefaultServiceLocator"></bean>

  <bean id="clientService"
    factory-bean="serviceLocator"
    factory-method="createClientServiceInstance"/>
</beans>
```

```
public class DefaultServiceLocator {

  private static ClientService clientService = new ClientServiceImpl();

  private DefaultServiceLocator() {}

  public ClientService createClientServiceInstance() {
    return clientService;
  }
}
```

```
=> ClientService clientService = DefaultServiceLocator.getInstance().createClientServiceInstance();
```

Temel XML konfigürasyonu kullanımı

- Spring IoC Dependency Injection teknikleri
 - Constructor kullanarak
 - Setter metodu yardımıyla

Temel XML konfigürasyonu kullanımı

Spring IoC dependency Injection teknikleri

Constructor kullanarak

```
<beans>
  <bean id="foo" class="x.y.Foo">
    <constructor-arg ref="bar"/>
    <constructor-arg ref="baz"/>
  </bean>

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

</beans>
```

```
package x.y;

public class Foo {

  public Foo(Bar bar, Baz baz) {

    // ...

  }
}
```

Temel XML konfigürasyonu kullanımı

Spring IoC dependency Injection teknikleri (2)

Constructor kullanarak

```
<bean id="exampleBean" class="examples.ExampleBean">
  <constructor-arg type="int" value="7500000"/>
  <constructor-arg type="java.lang.String" value="42"/>
</bean>

<bean id="exampleBean" class="examples.ExampleBean">
  <constructor-arg index="0" value="7500000"/>
  <constructor-arg index="1" value="42"/>
</bean>
```

```
public class ExampleBean {

  private int years;

  private String ultimateAnswer;

  public ExampleBean(int years, String ultimateAnswer) {
    this.years = years;
    this.ultimateAnswer = ultimateAnswer;
  }
}
```


Temel XML konfigürasyonu kullanımı

Spring IoC dependency Injection teknikleri (3)

Constructor kullanarak

```
<bean id="exampleBean" class="examples.ExampleBean">
  <constructor-arg name="years" value="7500000"/>
  <constructor-arg name="ultimateAnswer" value="42"/>
</bean>
```

```
package examples;

public class ExampleBean {

    @ConstructorProperties({"years", "ultimateAnswer"})
    public ExampleBean(int years, String ultimateAnswer) {
        this.years = years;
        this.ultimateAnswer = ultimateAnswer;
    }
}
```

Temel XML konfigürasyonu kullanımı

Spring IoC dependency Injection teknikleri (4)

Setter metodu kullanarak

```
<bean id="myCustomer" class="com.spring.training.Customer">  
  <property name="name" value="Baris"/>  
  <property name="years" value="45"/>  
</bean>
```

```
public class Customer {  
  
    private String name;  
    private int years;  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public void setYears(int years) {  
        this.years = years;  
    }  
}
```

Temel XML konfigürasyonu kullanımı

Spring IoC dependency Injection teknikleri (5)

Setter metodu kullanarak

```
<bean id="myCustomer" class="com.spring.training.Customer">  
  <property name="name" value="Baris"/>  
  <property name="years" value="45"/>  
  <property name="address" ref="myAddress"/>  
</bean>
```

```
<bean id="myAddress" class="com.spring.training.Address"/>
```

```
public class Customer {  
  
    private String name;  
    private int years;  
  
    public void setName(String name) {  
        this.name = name;  
    }  
    public void setYears(int years) {  
        this.years = years;  
    }  
    public void setAddress(Address address) {  
        this.address = address;  
    }  
}  
public class Address { }
```

Temel XML konfigürasyonu kullanımı

Dependency Injection yönetimi

```
<bean id="company" class="com.spring.training.Company">
  <constructor-arg name="customer" ref="myCustomer"/>
  <constructor-arg name="since" value="1945"/>
</bean>

<bean id="myCustomer" class="com.spring.training.Customer">
  <property name="name" value="Baris"/>
  <property name="years" value="45"/>
  <property name="address" ref="myAddress"/>
</bean>

<bean id="myAddress" class="com.spring.training.Address"/>
```

Customer
Address
Company constructor 2 arguments

Company constructor 2 arguments
Customer
Address

```
public class Customer {

    private String name;
    private int years;

    public void setName(String name)
    public void setYears(int years)
    public void setAddress(Address address)
    //default constructor 'Customer'
}

public class Address {
    // default constructor 'Address'
}

public class Company {
    private Customer customer;
    private int since;

    public Company(Customer customer, int since) {
        //print 'Company constructor 2 arguments'
        this.customer = customer;
        this.since = since;
    }
}
```

Temel XML konfigürasyonu kullanımı

Dependency Injection yönetimi (2)

```
<bean id="company" class="com.spring.training.Company">
  <constructor-arg name="customer" ref="myCustomer"/>
  <constructor-arg name="since" value="1945"/>
</bean>

<bean id="myCustomer" class="com.spring.training.Customer">
  <property name="name" value="Baris"/>
  <property name="years" value="45"/>
  <property name="address" ref="myAddress"/>
</bean>

<bean id="myAddress" class="com.spring.training.Address"/>
```

DOĞRU
 Customer
 Address
 Company constructor 2 arguments

YANLIŞ
 Company constructor 2 arguments
 Customer
 Address

```
public class Customer {

    private String name;
    private int years;

    public void setName(String name)
    public void setYears(int years)
    public void setAddress(Address address)
    //default constructor 'Customer'
}

public class Address {
    // default constructor 'Address'
}

public class Company {
    private Customer customer;
    private int since;

    public Company(Customer customer, int since) {
        //print 'Company constructor 2 arguments'
        this.customer = customer;
        this.since = since;
    }
    public Company() {
        //print 'Company default constructor'
    }
}
```

Temel XML konfigürasyonu kullanımı

Değişkenleri data tiplerine bağlama

Temel data tipleri için otomatik değer çevirme mevcut

```
<bean id="example" class="com.spring.training.ExampleBean">
  <property name="intValue" value="123"/>
  <property name="longValue" value="123"/>
  <property name="stringValue" value="123"/>
  <property name="floatValue" value="123"/>
  <property name="boolValue" value="false"/>
  <property name="doubleValue" value="123"/>
</bean>
```

```
public class ExampleBean {
  private int intValue;
  private long longValue;
  private String stringValue;
  private float floatValue;
  private boolean boolValue;
  private double doubleValue;

  //setter methods
}
```

Temel XML konfigürasyonu kullanımı

p-namespace kullanımı

Temel data tipleri için otomatik değer çevirme mevcut

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://www.springframework.org/schema/p"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

<bean id="example2" class="com.spring.training.ExampleBean"
  p:intValue="123"
  p:longValue="123"
  p:stringValue="123"
  p:floatValue="123"
  p:booleanValue="false"
  p:doubleValue="123"/>
```

```
public class ExampleBean {
  private int intValue;
  private long longValue;
  private String stringValue;
  private float floatValue;
  private boolean booleanValue;
  private double doubleValue;

  //setter methods
}
```

Temel XML konfigürasyonu kullanımı

p-namespace kullanımı (2)

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://www.springframework.org/schema/p"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

<bean id="example2" class="com.spring.training.ExampleBean"
  p:address-ref="myAddress"/>

<bean id="myAddress" class="com.spring.training.Address"/>
```


Temel XML konfigürasyonu kullanımı

c-namespace kullanımı

Temel data tipleri için otomatik değer çevirme mevcut

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:c="http://www.springframework.org/schema/c"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

  <bean id="foo" class="x.y.Foo">
    <constructor-arg ref="bar"/>
    <constructor-arg ref="baz"/>
    <constructor-arg value="foo@bar.com"/>
  </bean>

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

```

```
<bean id="foo" class="x.y.Foo" c:bar-ref="bar" c:baz-ref="baz" c:email="foo@bar.com"/>

<bean id="foo" class="x.y.Foo" c:_0-ref="bar" c:_1-ref="baz"/>

```

Temel XML konfigürasyonu kullanımı

Properties data tipi

```
<bean id="mappings" class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">  
  
  <!-- typed as a java.util.Properties →  
  <property name="properties">  
    <value>  
      jdbc.driver.className=com.mysql.jdbc.Driver  
      jdbc.url=jdbc:mysql://localhost:3306/mydb  
    </value>  
  </property>  
  
</bean>
```

Temel XML konfigürasyonu kullanımı

Java Collections kullanımı

`<list>` `<set>` `<map>` `<props>` içerisinde List, Set, Map ve Properties data tiplerini kullanabilirsiniz.

```
<bean id="moreComplexObject" class="example.ComplexObject">
  <!-- results in a setSomeList(java.util.List) call →
  <property name="someList">
    <list>
      <value>a list element followed by a reference</value>
      <ref bean="myDataSource" />
    </list>
  </property>

  <!-- results in a setSomeMap(java.util.Map) call →
  <property name="someMap">
    <map>
      <entry key="an entry" value="just some string"/>
      <entry key="a ref" value-ref="myDataSource"/>
    </map>
  </property>

  <!-- results in a setSomeSet(java.util.Set) call →
  <property name="someSet">
    <set>
      <value>just some string</value>
      <ref bean="myDataSource" />
    </set>
  </property>
</bean>
```

Temel XML konfigürasyonu kullanımı

Java Collections kullanımı (2)

```
<beans>
  <bean id="foo" class="x.y.Foo">
    <property name="accounts">
      <map>
        <entry key="one" value="9.99"/>
        <entry key="two" value="2.75"/>
        <entry key="six" value="3.99"/>
      </map>
    </property>
  </bean>
</beans>
```

```
class Foo {

  private Map<String, Float> accounts;

  public void setAccounts(Map<String, Float> accounts) {
    this.accounts = accounts;
  }
}
```

Temel XML konfigürasyonu kullanımı

Null ve Empty kullanımı

```
<bean class="ExampleBean">  
  <property name="email" value=""/>  
</bean>
```

```
<bean class="ExampleBean">  
  <property name="email"><null/></property>  
</bean>
```

Temel XML konfigürasyonu kullanımı

Lazy kullanımı

Gerçekten kullanılina kadar nesne oluşturulmaz

```
<bean id="lazy" class="com.foo.ExpensiveToCreateBean" lazy-init="true"/>
```

```
<bean name="not.lazy" class="com.foo.AnotherBean"/>
```

Ders notları

- Presentasyon dosyalarını download etmek için:
www.barisdere.com
- Tüm videolar için:
<http://www.youtube.com/barisdere>
- Takip etmek için: @BarisDere
- Bana ulaşmak için: baris.dere@gmail.com