

1. 개념

가. 가정: 오브젝트 A 오브젝트 B를 의존

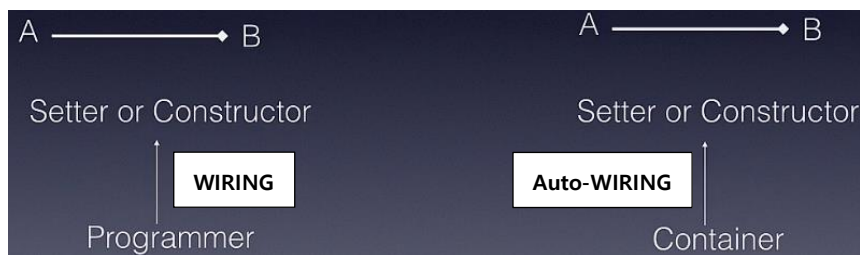
나. B: dependency(종속) 오브젝트, A: dependent(의존적인) 오브젝트

다. 기존: 세터주입, 생성자주입 => ref 태그 사용

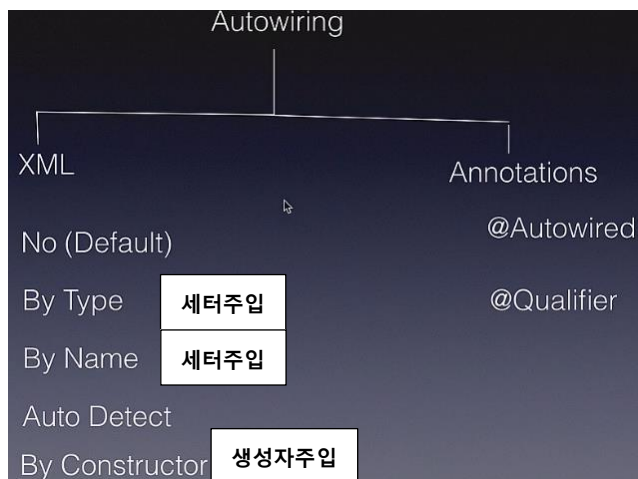
라. 빈 와이어링(Wiring of beans): dependent와 dependency를 연결해주는 프로세스

마. 와이어링 주체: 프로그래머

바. 와이어링 구분



사. Autowiring 방법



2. 프로젝트 만들기



The screenshot displays an IDE interface with the 'pom.xml' file for the 'springcoreadvanced' project open in the editor. The project structure is visible in the left sidebar, and the XML content is shown in the main editor area.

Project Structure (Left Sidebar):

- springcoreadvanced
 - Spring Elements
 - src/main/java
 - com.springcoreadvanced** (highlighted with a red box)
 - JRE System Library [JavaSE-11]
 - Maven Dependencies
 - src
 - target
 - pom.xml

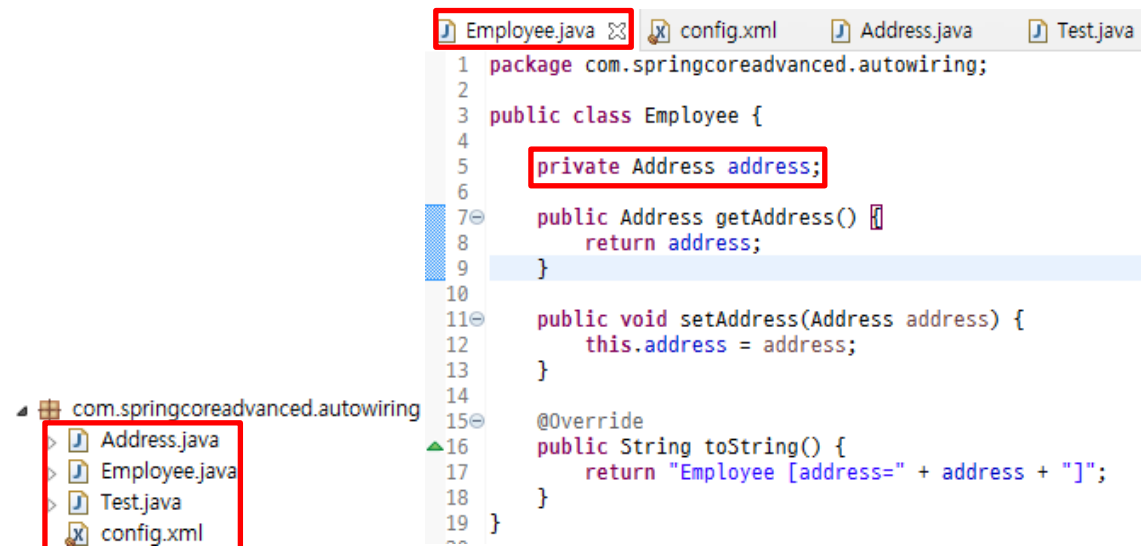
pom.xml Content (Main Editor):

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apa
5   <modelVersion>4.0.0</modelVersion>
6
7   <groupId>com</groupId>
8   <artifactId>springcoreadvanced</artifactId>
9   <version>0.0.1-SNAPSHOT</version>
10  <packaging>jar</packaging>
11
12  <name>springcore</name>
13  <url>http://maven.apache.org</url>
14
15  <properties>
16    <springframework.version>4.3.6.RELEASE</springframework.version>
17  </properties>
18
19  <dependencies>
20    <dependency>
21      <groupId>org.springframework</groupId>
22      <artifactId>spring-core</artifactId>
23      <version>${springframework.version}</version>
24    </dependency>
25    <dependency>
26      <groupId>org.springframework</groupId>
27      <artifactId>spring-context</artifactId>
28      <version>${springframework.version}</version>
29    </dependency>
30
31    <dependency>
32      <groupId>javax.annotation</groupId>
33      <artifactId>javax.annotation-api</artifactId>
34      <version>1.3.2</version>
35    </dependency>
36  </dependencies>
37  <build>
38    <pluginManagement>
39      <plugins>
40        <plugin>
41          <groupId>org.apache.maven.plugins</groupId>
42          <artifactId>maven-compiler-plugin</artifactId>
43          <version>3.8.0</version>
44          <configuration>
45            <source>11</source>
46            <target>11</target>
47          </configuration>
48        </plugin>
49      </plugins>
50    </pluginManagement>
51  </build>
52 </project>
```

Red boxes highlight the 'com.springcoreadvanced' package in the project structure and the Maven compiler plugin configuration in the pom.xml file.

3. 타입을 이용한 자동 와이어링(Auto-Wiring)

- 두 개의 오브젝트 타입 필요
- Autowiring은 레퍼런스 타입에 대해 작동



```
1 package com.springcoreadvanced.autowiring;
2
3 public class Employee {
4     private Address address;
5
6     public Address getAddress() {
7         return address;
8     }
9
10    public void setAddress(Address address) {
11        this.address = address;
12    }
13
14    @Override
15    public String toString() {
16        return "Employee [address=" + address + "]";
17    }
18 }
19
20 package com.springcoreadvanced.autowiring;
21
22 public class Address {
23     private int hno;
24     private String street;
25     private String city;
26
27     public int getHno() {
28         return hno;
29     }
30
31     public void setHno(int hno) {
32         this.hno = hno;
33     }
34
35     public String getStreet() {
36         return street;
37     }
38
39     public void setStreet(String street) {
40         this.street = street;
41     }
42
43     public String getCity() {
44         return city;
45     }
46
47     public void setCity(String city) {
48         this.city = city;
49     }
50
51     @Override
52     public String toString() {
53         return "Address [hno=" + hno + ", "
54             + "street=" + street + ", city=" + city + "]";
55     }
56 }
```

Employee 클래스내의 Address 타입 필드존재
byType으로 검색



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:context="http://www.springframework.org/schema/context"
5     xmlns:p="http://www.springframework.org/schema/p"
6     xmlns:c="http://www.springframework.org/schema/c"
7     xsi:schemaLocation="http://www.springframework.org/schema/beans
8         http://www.springframework.org/schema/beans/spring-beans.xsd
9         http://www.springframework.org/schema/context
10         http://www.springframework.org/schema/context/spring-context.xsd">
11
12     <bean class="com.springcoreadvanced.autowiring.Address"
13         name="address" p:hno="123" p:street="sadangro" p:city="seoul" />
14
15     <bean class="com.springcoreadvanced.autowiring.Employee"
16         name="employee" autowire="byType" />
17
18 </beans>
19
20 package com.springcoreadvanced.autowiring;
21
22 import org.springframework.context.ApplicationContext;
23
24 public class Test {
25     private static ApplicationContext context;
26
27     public static void main(String[] args) {
28         context = new ClassPathXmlApplicationContext(
29             "com/springcoreadvanced/autowiring/config.xml");
30         Employee employee = (Employee) context.getBean("employee");
31         System.out.println(employee);
32     }
33 }
```

4월 08, 2023 7:22:33 오전 org.springframework.context.support.ClassPathXmlApplicationContext: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext
4월 08, 2023 7:22:33 오전 org.springframework.beans.factory.xml.XmlBeanDefinitionReader: Loading XML bean definitions from class path resource [com/springcoreadvanced/autowiring/config.xml]
Employee [address=Address [hno=123, street=sadangro, city=seoul]]

4. byType에 의한 auto-wiring관한 중요한 것
- 가. 스프링 컨테이너가 타입 매치되는 것을 찾으려 시도, 없으면 null 값을 주입
5. 이름을 이용한 오토 와이어링(auto-wiring)
6. 생성자를 이용한 오토 와이어링(auto-wiring)

7. Autowired 애노테이션을 이용한 오토 와이어링(auto-wiring)

```

Address.java
1 package com.springcoreadvanced.autowiring;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 public class Employee {
6
7     @Autowired
8     private Address address;
9
10    public Employee(Address address)
11    {
12        this.address = address;
13    }
14
15    public Address getAddress() {
16        return address;
17    }
18
19    public void setAddress(Address address) {
20        this.address = address;
21    }
22
23    @Override
24    public String toString() {
25        return "Employee [address=" + address + "]";
26    }
27 }

Employee.java
1 package com.springcoreadvanced.autowiring;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 public class Employee {
6
7     @Autowired
8     private Address address;
9
10    public Employee(Address address)
11    {
12        this.address = address;
13    }
14
15    public Address getAddress() {
16        return address;
17    }
18
19    public void setAddress(Address address) {
20        this.address = address;
21    }
22
23    @Override
24    public String toString() {
25        return "Employee [address=" + address + "]";
26    }
27 }

Test.java
1 package com.springcoreadvanced.autowiring.annotations;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5
6 public class Test {
7
8     private static ApplicationContext context;
9
10    public static void main(String[] args) {
11        context = new ClassPathXmlApplicationContext(
12            "com/springcoreadvanced/autowiring/annotations/config.xml");
13        Employee employee = (Employee) context.getBean("employee");
14        System.out.println(employee);
15    }
16 }

config.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:context="http://www.springframework.org/schema/context"
5       xmlns:p="http://www.springframework.org/schema/p"
6       xmlns:c="http://www.springframework.org/schema/c"
7       xsi:schemaLocation="http://www.springframework.org/schema/beans
8         http://www.springframework.org/schema/beans/spring-beans.xsd
9         http://www.springframework.org/schema/context
10        http://www.springframework.org/schema/context/spring-context.xsd">
11
12
13
14     <bean class="com.springcoreadvanced.autowiring.annotations.Address"
15         name="address" p:hno="123" p:street="sadangro" p:city="seoul" />
16
17     <bean class="com.springcoreadvanced.autowiring.annotations.Employee"
18         name="employee" />
19
20 </beans>
  
```

org.springframework.beans.factory.BeanCreationException

org.springframework.beans.factory.BeanCreationException

Employee 빈 생성에 실패: 내부에 필드로 존재하는 Address를 주입하는데 실패
기본적으로 스프링은 autowire 기능이 disable 되어 있어, enable 시켜주어야 함.

```
<context:annotation-config />
```

```
<bean class="com.springcoreadvanced.autowiring.annotations.Address"
      name="address" p:hno="123" p:street="sadangro" p:city="seoul" />
```

```
<bean class="com.springcoreadvanced.autowiring.annotations.Employee"
      name="employee" />
```

8. Qualifier 애노테이션을 이용한 오토 와이어링

```
@Qualifier("address123")  
NoSuchBeanException
```

9. 요약

Container	Reference Types
XML	Annotations
byType	@Autowired
byName	@Qualifier
byConstructor	

10. 과제

Autowiring	Hotel Management
Customer	Reservation
String name	int id
	String time