1. 개념

가. 가정: 오브젝트 A 오브젝트 B를 의존

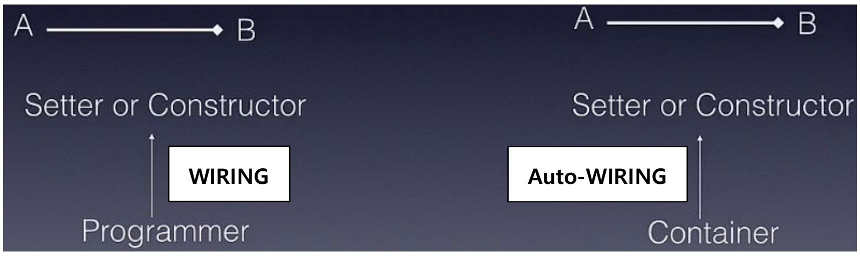
나. B: dependency(종속) 오브젝트, A: dependent(의존적인) 오브젝트

다. 기존: 세터주입, 생성자주입 => ref 태그 사용

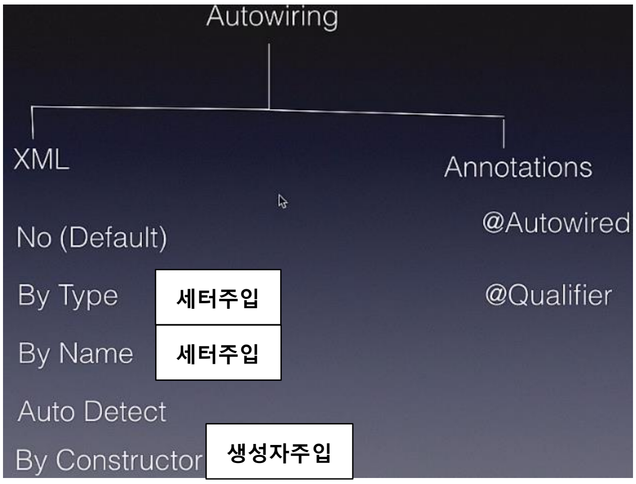
라. 빈 와이어링(Wiring of beans): dependent와 dependency를 연결해주는 프로세스

마. 와이어링 주체: 프로그래머

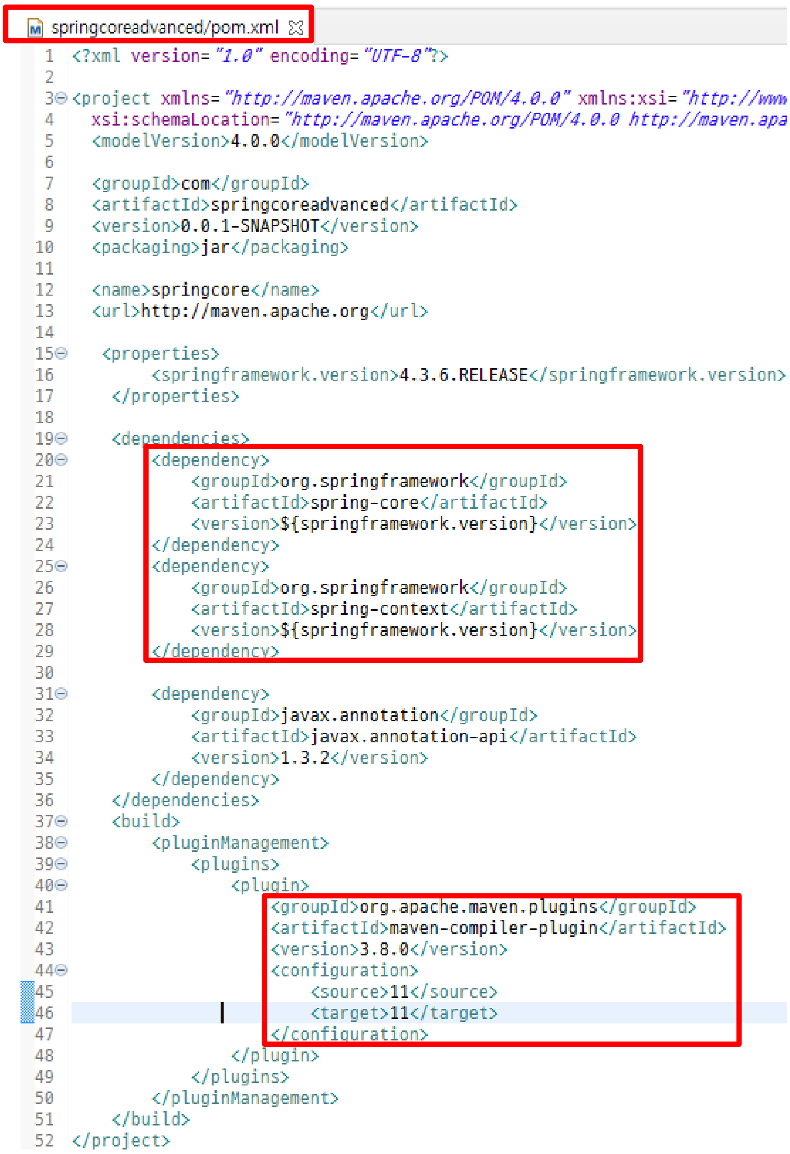
바. 와이어링 구분



사. Autowiring 방법

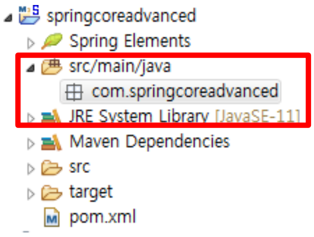


2. 프로젝트 만들기



springcoreadvanced/pom.xml

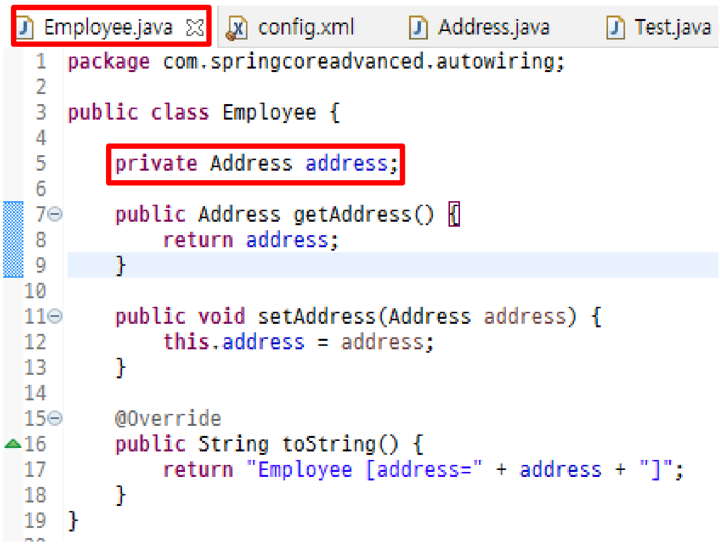
```xml
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apa
  <modelVersion>4.0.0</modelVersion>

  <groupId>com</groupId>
  <artifactId>springcoreadvanced</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>springcore</name>
  <url>http://maven.apache.org</url>

   <properties>
        <springframework.version>4.3.6.RELEASE</springframework.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-core</artifactId>
            <version>${springframework.version}</version>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>${springframework.version}</version>
        </dependency>

        <dependency>
            <groupId>javax.annotation</groupId>
            <artifactId>javax.annotation-api</artifactId>
            <version>1.3.2</version>
        </dependency>
    </dependencies>
    <build>
        <pluginManagement>
            <plugins>
                <plugin>
                    <groupId>org.apache.maven.plugins</groupId>
                    <artifactId>maven-compiler-plugin</artifactId>
                    <version>3.8.0</version>
                    <configuration>
                        <source>11</source>
                        <target>11</target>
                    </configuration>
                </plugin>
            </plugins>
        </pluginManagement>
    </build>
</project>
```



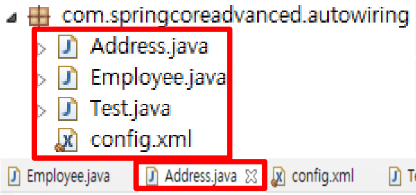
```
springcoreadvanced
  Spring Elements
  src/main/java
    com.springcoreadvanced
  JRE System Library [JavaSE-11]
  Maven Dependencies
  src
  target
  pom.xml
```

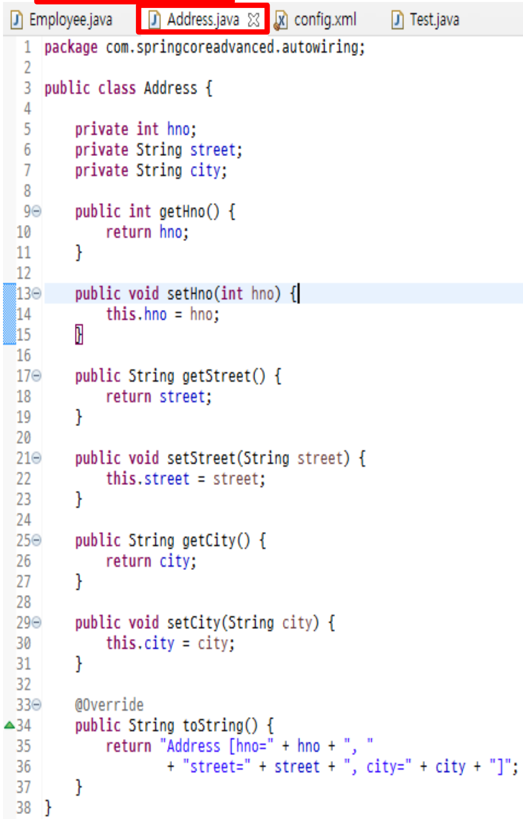
3. 타입을 이용한 자동 와이어링(Auto-Wiring)

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



```java
package com.springcoreadvanced.autowiring;

public class Employee {

    private Address address;

    public Address getAddress() {
        return address;
    }

    public void setAddress(Address address) {
        this.address = address;
    }

    @Override
    public String toString() {
        return "Employee [address=" + address + "]";
    }
}
```

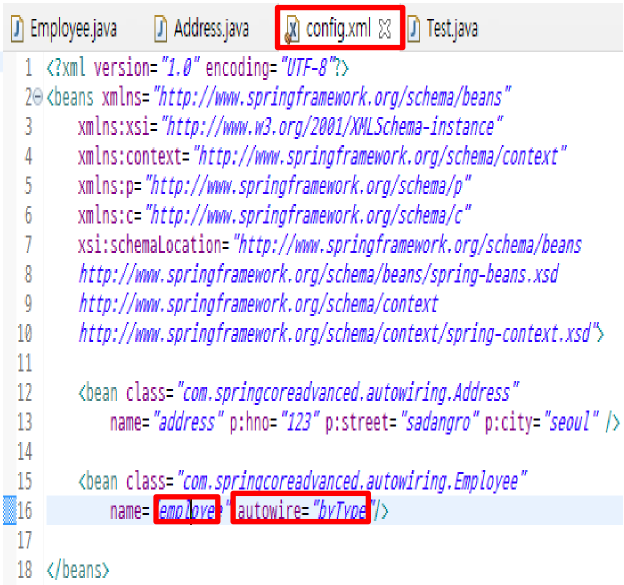




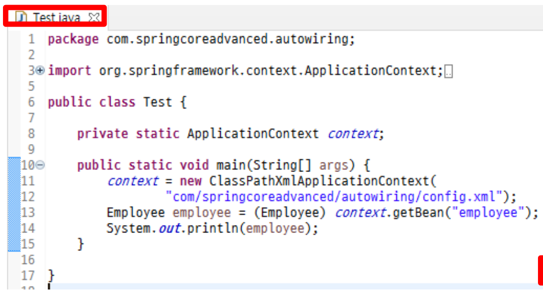
```java
package com.springcoreadvanced.autowiring;

public class Address {

    private int hno;
    private String street;
    private String city;

    public int getHno() {
        return hno;
    }

    public void setHno(int hno) {
        this.hno = hno;
    }

    public String getStreet() {
        return street;
    }

    public void setStreet(String street) {
        this.street = street;
    }

    public String getCity() {
        return city;
    }

    public void setCity(String city) {
        this.city = city;
    }

    @Override
    public String toString() {
        return "Address [hno=" + hno + ", "
                + "street=" + street + ", city=" + city + "]";
    }
}
```

Employee 클래스내의 Address 타입 필드존재

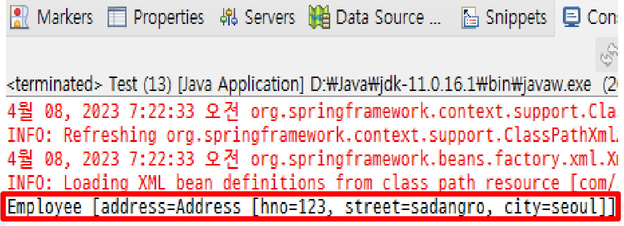
byType으로 검색



```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:c="http://www.springframework.org/schema/c"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd">

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

    <bean class="com.springcoreadvanced.autowiring.Employee"
        name="employee" autowire="byType"/>

</beans>
```



```java
package com.springcoreadvanced.autowiring;

import org.springframework.context.ApplicationContext;

public class Test {

    private static ApplicationContext context;

    public static void main(String[] args) {
        context = new ClassPathXmlApplicationContext(
                "com/springcoreadvanced/autowiring/config.xml");
        Employee employee = (Employee) context.getBean("employee");
        System.out.println(employee);
    }

}
```



```
<terminated> Test (13) [Java Application] D:₩Java₩jdk-11.0.16.1₩bin₩javaw.exe  (2
4월 08, 2023 7:22:33 오전 org.springframework.context.support.Cla
INFO: Refreshing org.springframework.context.support.ClassPathXml
4월 08, 2023 7:22:33 오전 org.springframework.beans.factory.xml.X
INFO: Loading XML bean definitions from class path resource [com/
Employee [address=Address [hno=123, street=sadangro, city=seoul]]
```

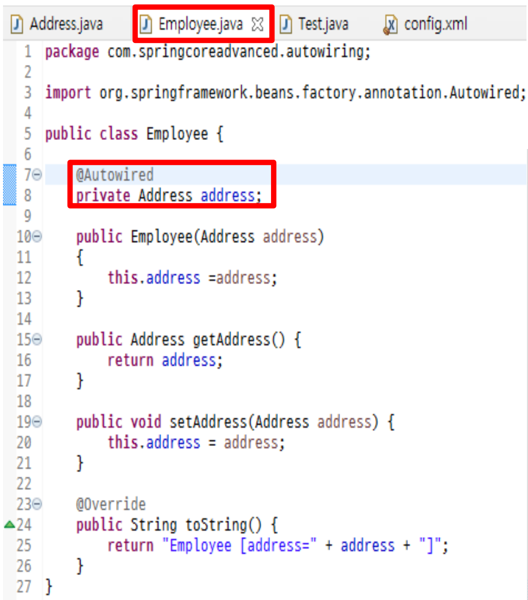
4. byType에 의한 auto-wiring관한 중요한 것

가. 스프링 컨테이너가 타입 매치되는 것을 찾으려 시도, 없으면 null 값을 주입

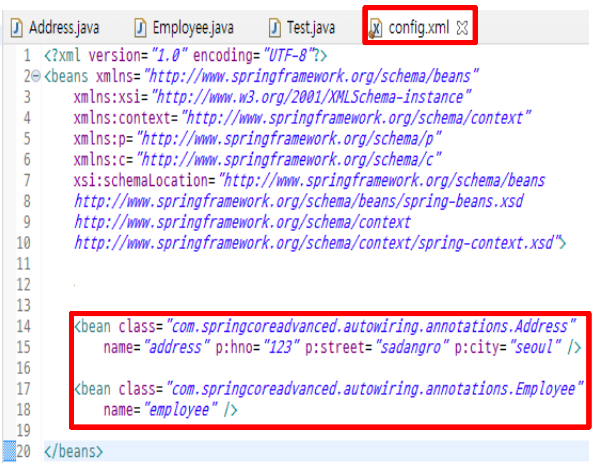
5. 이름을 이용한 오토 와이어링(auto-wiring)

6. 생성자를 이용한 오토 와이어링(auto-wiring)

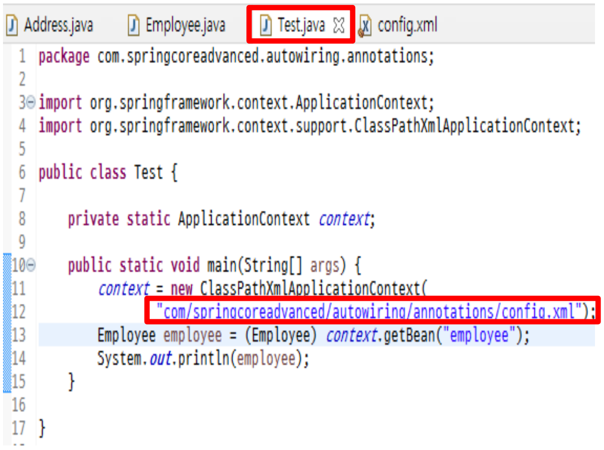
7. Autowired 애노테이션을 이용한 오토 와이어링(auto-wiring)



```java
package com.springcoreadvanced.autowiring;

import org.springframework.beans.factory.annotation.Autowired;

public class Employee {

    @Autowired
    private Address address;

    public Employee(Address address)
    {
        this.address =address;
    }

    public Address getAddress() {
        return address;
    }

    public void setAddress(Address address) {
        this.address = address;
    }

    @Override
    public String toString() {
        return "Employee [address=" + address + "]";
    }
}
```



```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:c="http://www.springframework.org/schema/c"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd">



    <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" />

</beans>
```



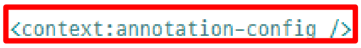
```java
package com.springcoreadvanced.autowiring.annotations;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Test {

    private static ApplicationContext context;

    public static void main(String[] args) {
        context = new ClassPathXmlApplicationContext(
            "com/springcoreadvanced/autowiring/annotations/config.xml");
        Employee employee = (Employee) context.getBean("employee");
        System.out.println(employee);
    }

}
```

org.springframework.beans.factory.BeanCreationException

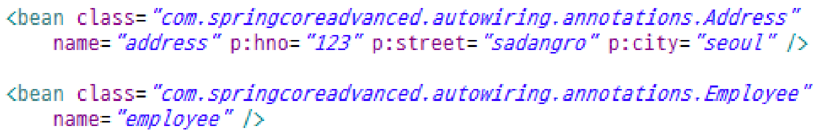
org.springframework.beans.factory.BeanCreationException

Employee 빈 생성에 실패: 내부에 필드로 존재하는 Address를 주입하는데 실패

기본적으로 스프링은 autowire 기능이 disable 되어 있어, enable 시켜주어야 함.

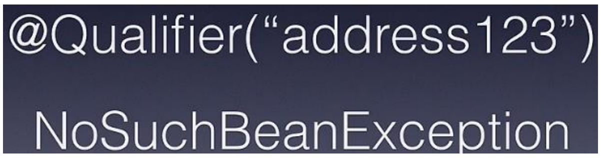


```xml
<context:annotation-config />
```



```xml
<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" />
```

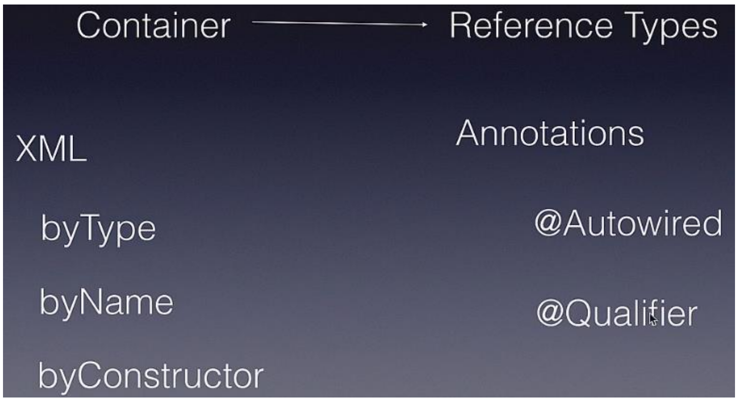
8. Qualifier 애노테이션을 이용한 오토 와이어링



@Qualifier(“address123”)

NoSuchBeanException

9. 요약



Container → Reference Types

XML

byType

byName

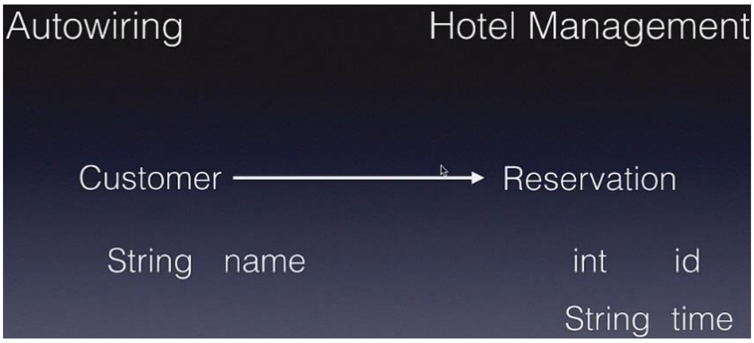
byConstructor

Annotations

@Autowired

@Qualifier

10. 과제



Autowiring

Hotel Management

Customer → Reservation

String name

int id

String time