

KoreaIT Spring01

root-context.xml 에서의 Bean 생성

기본 생성자를 사용하고 setterMethod를 사용하는 방식

```
<!-- 기존 방식 -->
<!-- PersonVO p1 = new PersonVO() 기본 생성자를 만드는 방식 -->
<bean id="p1" class="vo.PersonVO">
  <!-- 기존 방식
    p1.setName("홍길동");
    p1.setAge(30);
    p1.setTel("010-111-1111"); -->
    <property name="name" value="홍길동"/>
    <property name="age" value="30"/>
    <property name="tel" value="010-111-1111"/>
  </bean>
```

생성자에 파라미터를 넣어서 생성하는 방법

```
<!-- PersonVO p2 = new PersonVO("김길동", "010-2222-2222", 32); -->
<bean id="p2" class="vo.PersonVO">
  <constructor-arg value="박길동"/>
  <constructor-arg value="010-2222-2222"/>
  <constructor-arg value="32"/>
</bean>
```

파라미터 순서를 써줘야 하고 아니라면 아래와 같이 인덱스를 적어줘야 한다.

```
<constructor-arg value="박길동" index="1"></constructor-arg>
<constructor-arg value="010-2222-2222" index="2"></constructor-arg>
<constructor-arg value="32" index="3"></constructor-arg>
```

Service의 역할

중간다리 역할

controller - service - DAO - DB

DB의 값을 controller에 가져가기 위해 사용.

의존성 주입

```
<!-- BoardDAOImpl dao = new BoardDAOImpl -->
<bean id="dao" class="dao.BoardDAOImpl"/>

<bean id="serviceBean" class="service.BoardServiceImpl">
    <constructor-arg ref="dao"></constructor-arg>
</bean>
```

컨트롤러에 대한 생성자 생성은

root-context.xml 이 아닌 servlet-context.xml 에서 생성한다.

```
<!-- 자동생성 패키지에 있는 controller 자동으로 실행해 준다. -->
<context:component-scan base-package="com.korea.testdi" />
<!-- 수동생성 BoardController 수동 생성 -->
<!-- 자동생성을 하는 경우 생성자 파라미터나 setter 메서드를 호출하는 것이
불가능 하다 생성자나 setter로 받아야 할 데이터가 있으면 자동이 아닌 수동으로
생성해야 한다. -->
<beans:bean class="com.korea.testdi.BoardController">
    <beans:constructor-arg ref="serviceBean"/>
</beans:bean>

<!-- <beans:constructor-arg ref="serviceBean"/> -->
<beans:property name="service" ref="serviceBean"></beans:property>

setter인젝션을 통해 생성
```

여기에서 serviceBean은 root-context.xml 생성한 녀석이다.

Controller에서 사용

```
//사용자가 요청한 url을 접수
//사용자가 list.do를 요청하면 해당 메서드가 실행이 된다.
//jsp의 url매핑 형태라고 생각하면 된다.
@RequestMapping("/list.do")
public String list() {
    //service를 통해서 dao의 selectList()를 호출할 수 있다.
    List list = service.selectList();
    return "board_list";
}
```

http://localhost:9090/testdi/list.do 이동시

실행이 된다면 rootContextPath/board_list.jsp 로 실행이된다.

```
<beans:bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <beans:property name="prefix" value="/WEB-INF/views/" />
    <beans:property name="suffix" value=".jsp" />
</beans:bean>
```

servlet-context.xml에 이 부분에 의해 prefix와 suffix가 조정된다.

Controller에서는 모델 객체로 포워딩한다.

```
@RequestMapping("/list.do")
public String list(Model model) {
    //service를 통해서 dao의 selectList()를 호출할 수 있다.
    List list = service.selectList();
    //Model의 객체로 바인딩하여 포워딩한다.
    model.addAttribute("list",list);
    return "board_list";
}
```

DAO -> 추상메서드(CRUD) -> DAOImpl(메서드 구체화) -> Service(interface) ->

ServiceImpl에서 메서드 구체화 -> Controller 서비스 객체 의존주입 후 바인딩 -> 포워딩