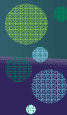


제8장

JSP- IV (자바 빈)



1. 빈(bean) 이란?

빈(Bean)은 JSP페이지코드에서 상당히 많이 접하게 되는 부분이다. 한마디로 말하면 클래스인 것이다. 반복적인 작업을 효율적으로 하기 위해 빈을 사용한다. 빈이란? JAVA언어의 데이터(속성)와 기능(메소드)으로 이루어진 클래스이다. jsp페이지를 만들고, 액션태그를 이용하여 빈을 사용한다. 그리고 빈의 내부 데이터를 처리한다. 자바를 학습할 때 클래스를 왜 만드는가? 봉어빵 틀은 왜 만드는가? 바로 봉어빵을 많이 만들기 위함이었다. 빈을 사용하는 궁극적 목적은 디자인 부분과 비즈니스 로직부분을 분리하고, 복잡한 JSP코드들을 줄이고, 프로그램의 재사용성을 증가시키기 위한 것이다.

JAVA언어를 학습 하면서 데이터 객체를 많이 만들어본 경험이 있을 것이다. 데이터 객체에는 데이터가 있어 그에 해당하는 getter와 setter가 있다. 빈을 만든다는 것은 데이터 객체를 만들기 위한 클래스를 만드는 것이다.

```
package sec01_exam;

public class Student {

    private String name;
    private int age;
    private int grade;
    private int studentNum;

    public Student() {

    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}
```

2. 빈즈의 작성 규칙

빈즈를 작성할 때는 다음 규칙을 따라야 합니다.

- ❶ 자바 클래스는 `java.io.Serializable` 인터페이스를 구현해야 합니다.
- ❷ 인수가 없는 기본 생성자가 있어야 합니다.
- ❸ 모든 멤버 변수인 프로퍼티는 `private` 접근 지정자로 설정해야 합니다.
- ❹ 모든 멤버 변수인 프로퍼티는 `getter/setter()` 메소드가 존재해야 합니다.
 - `getter()` 메소드는 멤버 변수에 저장된 값을 가져올 수 있는 메소드이고,
 - `setter()` 메소드는 멤버 변수에 값을 저장할 수 있는 메소드입니다.



3. 자바 빈즈 작성 예

[자바빈즈 작성 예]

```
package com.dao;
import java.io.Serializable;

public class MemberBean implements java.io.Serializable {
    private int id;
    private String name; } ❶ 멤버 변수(프로퍼티) 정의
    public MemberBean(){ } ❷ 기본 생성자
    public int getId(){
        return id;
    }
    public void setId(int id){
        this.id=id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```

❸ getter/setter() 메소드

4. 빈 관련 액션태그(useBean)

액션태그 중에서 bean관련한 태그가 있다. 주로 데이터를 업데이트(수정)하고, 읽어오는 역할을 한다.

업데이트(수정) : `setProperty()`, 데이터 읽어오는 역할 : `getProperty()`

useBean

특정 bean을 사용한다고 명시할 때 사용한다.

```
<jsp:useBean id="student" class="sec01_exam.Student" scope="page" />
```

빈 이름(참조변수)

클래스 경로
(패키지명 포함하여 기술)

스코프 범위

자바코드 : `Student student = new Student();`

scope

page : 생성된 페이지 내에서만 사용 가능.(기본적으로 이 값이 설정됨), 페이지가 바뀌면 빈 객체가 공유 유지가 되지 않는다.

request : request요청을 받고 처리를 완료할 때까지 생존되는 scope이다.

session : 클라이언트당 하나씩 할당되는 영역이다.클라이언트가 브라우저를 종료하기 전까지 유지된다.

application : 사이트 전체의 범위를 가지며, 서버가 종료되기 전까지 계속 유지된다.

4. 빈 관련 액션태그(useBean)

속성	설명
id	자바빈즈를 식별하기 위한 이름입니다.
class	패키지 이름을 포함한 자바빈즈 이름입니다. 자바빈즈는 인수가 없는 기존 생성자가 있어야 하며 추상 클래스를 사용할 수 없습니다.
scope	자바빈즈가 저장되는 영역을 설정합니다. page(기본 값), request, session, application 중 하나의 값을 사용합니다.

[useBean 액션 태그 사용 예]

```
<jsp:useBean id="member" class="com.dto.MemberBean" scope="page" />
```

```
MemberBean member = (MemberBean) request.getAttribute ( "member");  
if (member == null){  
    member = new MemberBean();  
    request.setAttribute("member", member);  
}
```

4. 빈 관련 액션태그 (setProperty, getProperty)

setProperty(자바에서 setter)

데이터 값을 설정 할 때 사용한다.

```
<jsp:setProperty name="student" property="name" value="홍길동">
```

빈 이름(참조변수)

속성
(멤버변수, 필드)

속성(데이터) 값

getProperty(자바에서 getter)

데이터 값을 가져올 때 사용한다.

```
<jsp:getProperty name="student" property="name" />
```

빈 이름(참조변수)

속성
(멤버변수, 필드)

4. 빈 관련 액션태그(setProperty)

속성	설명
name	useBean 태그에 id 속성 값으로 설정된 자바빈즈를 식별하기 위한 이름입니다.
property	자바빈즈의 프로퍼티 이름입니다. 만약 프로퍼티 이름에 '*'를 사용하면 모든 요청 파라미터가 자바빈즈 프로퍼티의 setter() 메소드에 전달됨을 의미합니다.
value	변경할 자바빈즈의 프로퍼티 값입니다. 만약 프로퍼티 값이 null이거나 존재하지 않는 요청 파라미터인 경우에는 setProperty 액션 태그가 무시됩니다.
param	자바빈즈의 프로퍼티 값을 전달하는 요청 파라미터의 이름입니다. param과 value를 동시에 모두 사용할 수 없으며 하나를 선택하여 사용하는 것은 가능합니다.

```
<jsp:useBean id="member" class="com.dto.MemberBean" scope="page" />
```

[setProperty 액션 태그 사용 예]

```
<jsp:setProperty name="member" property="id" value="admin" />
```

[자바빈즈의 프로퍼티 값 출력 예]

```
<% out.println("아이디 : "+member.getId()); %>
```


4. 빈 관련 액션태그(getProperty)

속성	설명
name	useBean 태그에 id 속성 값으로 설정된 자바빈즈를 식별하기 위한 이름입니다.
property	자바빈즈의 프로퍼티 이름입니다. 만약 프로퍼티 이름에 '*'를 사용하면 모든 요청 파라미터가 자바빈즈 프로퍼티의 getter() 메소드에 전달됨을 의미합니다.

[getProperty 액션 태그 사용 예]

```
<jsp:getProperty name="member" property="name" />
```

[자바빈즈의 프로퍼티 값 출력 예]

```
<% out.println(member.getName()); %>
```



감사합니다.