

자바 빈이란

- 회원 가입을 할 때 회원 가입페이지에 받은 정보를 데이터베이스에 전달을 해야 한다.
- 이때 두 가지 방법으로 서버에 전송된다.
 - 이름, 아이디, 비밀번호 등 개별적으로 전송하는 방법
 - 회원정보를 하나로 묶어서 전송하는 방법
- 이중에서 하나로 묶어서 전송하는 방법이 보다 효율적일 것이다.
- 이렇게 관련 있는 정보를 하나로 묶어 관리 하기 위해 나온 매커니즘이 바로 자바 빈(java bean)이다.

자바 빈

● 자바 빈(Java Bean)

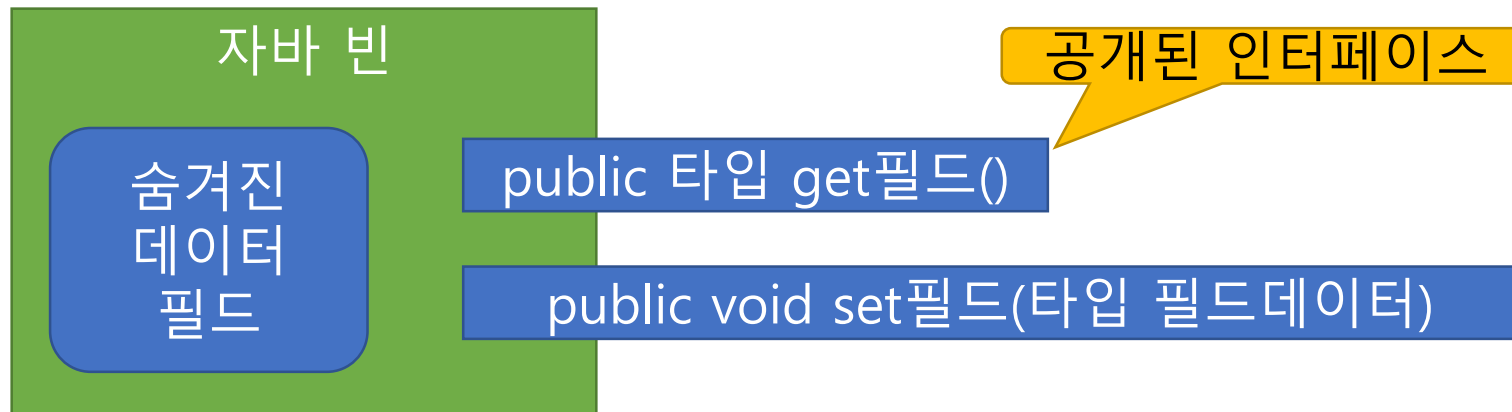
- 자바의 재활용 가능한 컴포넌트 모델을 의미함,
- 웹 개발에만 국한된 개념이 아니며 POJO라고 하는 단순한 구조를 가짐
 - POJO(Plain Old Java Object): 특정 기술이나 프레임워크에 종속하지 않고 기본 생성자와 멤버 변수에 대한 getter/setter 메서드를 제공하고 직렬화할 수 있는 자바 클래스를 의미함
- 예) 엔티티 클래스 혹은 DO(Data Object)는 기본적으로 테이블 칼럼에 해당하는 private 멤버 변수와 getter/setter 메서드로 구성할 수 있음

● 자바 빈 구조의 특징

- 인자가 없는 생성자(기본 생성자)로 구성됨
- 파일 혹은 네트워크를 통해 객체를 주고받을 수 있는 직렬화 구조가 가능함
- getter, setter 메서드를 통해 멤버 변수(속성)에 접근함

자바 빈이란

- 자바 빈은 정보의 덩어리로 필요한 정보를 객체로 구성해서 한꺼번에 접근해서 사용 가능하다.
- 단순 정보의 저장뿐 아니라 데이터 은닉 개념을 사용한다.



자바 빈이란

- 자바 빈도 클래스이다.
- 당연히 필드와 메소드 내용이 구성된다.
 - 자바 빈 클래스 정의하기
 - 패키지 선언방법
 - (도메인이름.폴더이름.)형태로 선언한다.
 - 필드 선언하기
 - 자바빈 프로퍼티
- 예1) MemberBean.java

자바 빈

- 회원 관리를 위한 Member 클래스를 자바 빈 구조

- getter/setter 메서드의 이름: 'get/set + 대문자로 시작되는 변수명'의 형태
- 인자를 받아 현재 멤버 변수에 값을 대입하거나 현재 값을 리턴하는 구조
 - 물론 메서드 내에서 추가적으로 필요한 기능을 구현할 수 있으며 이러한 구조는 객체지향에서 캡슐화의 한 형태이기도 함

```
class Member {                                // 클래스 선언
    private int id;
    private String name;
    private String email;
    ...
    public void setId(int id) { // setter 메서드 선언
        this.id = id
    }
    public int getId() {          // getter 메서드 선언
        return id;
    }
    ...
}
```

자바 빈이란

- 어떻게 자바빈을 JSP에서 활용하는지 알아보자
- JSP에서 자바빈을 활용하는 액션태그는 3가지로 구성된다.

액션태그	설명
<jsp:useBean>	자바 빈을 생성한다.
<jsp:getProperty> >	자바 빈에서 정보를 얻어 온다
<jsp:setProperty> >	자바 빈에 정보를 저장한다.

자바 빈이란

- JSP에서 자바 빈을 생성하자
- 액션 태그 없이 생성하는 법

```
<%@ page import="패키지명.자바빈클래스" %>  
<%  
MemberBean member = new MemberBean();  
%>
```

- 임포트 없이 자바빈 객체를 생성하는 법

```
<%  
패키지명.MemberBean member = new 패키지명. MemberBean();  
%>
```

자바 빈이란

- JSP에서 자바 빈 객체에서 프로퍼티 값을 얻어오기 위해 getter메소드를 호출한다.

```
<%=member.getName()%>  
<%=member.getUserid()%>
```

- JSP에서 자바 빈 객체에 새로운 프로퍼티 값을 설정하기 위해 setter 메소드를 호출한다.

```
<%  
member.setName("홍길동")  
member.setUserid("hong")  
%>
```


useBean 액션

- useBean 액션

- JSP에서 자바 빈 객체를 생성하거나 참조하기 위한 액션
- 매우 유용하지만 JSP를 단순히 뷰 역할로만 사용한다면 사용할 일은 없음

- useBean의 기본적인 동작 방식

- ① useBean을 이용해 만든 객체의 범위는 지정하는 속성인 scope에 주어진 id의 객체가 있는지 확인한다.
- ② 객체가 없다면 새로 객체를 생성하고 해당 scope에 저장한다.

useBean 액션

- useBean 액션의 기본 구문과 사용 예

```
<jsp:useBean id="instanceName" scope="page | request | session | application"
              class="packageName.className" type="packageName.className"
              beanName="packageName.className" >
</jsp:useBean>
```

- **id**: 자바 빈을 특정 scope에 저장하거나 가져올 때 사용하는 이름
 - 현재 페이지에서는 해당 인스턴스를 참조하기 위한 변수명이 됨
- **scope**: 해당 클래스 타입의 객체를 저장하거나 가지고 오는 범위로 내장객체의 일부
- **class**: 생성하거나 참조하려는 객체의 클래스명
 - 반드시 패키지명까지 명시해야 하며, 추상클래스나 인터페이스는 사용할 수 없음
- **type**: 특정 타입의 클래스를 명시할 때 사용함
 - 추상 클래스나 인터페이스, 일반 클래스가 될 수 있으며 class 속성의 클래스에서 상속 혹은 구현이 이루어져야 함
- **beanName**: type과 beanName 사용을 통해 class 속성을 대체할 수 있음

useBean의 활용

- useBean을 주로 활용하는 경우

- HTML 폼에서 입력한 값을 자바 객체로 연동할 때 주로 활용함
- 예) 회원 가입 페이지에서 아이디, 이름, 전화번호, 주소 등 여러 정보를 입력하고 가입하는 경우
 - 입력값을 받아 Member 객체에 넣고 이를 데이터베이스에 저장하기 위한 메서드 호출에 인자로 전달해야 함

```
<jsp:useBean id="m" class="com.my.Member" />
<jsp:setProperty name="m" property="*" />

<%
    MemberDAO dao = new MemberDAO();
    dao.insertDB(m);
%>
```

예제 2 useBeanDemo.jsp

자바 빈이란

- JSP에서 자바 빈 객체를 액션 태그로 가져오자

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

- 위 액션 태그는 다음처럼 자바 빈 객체를 생성하는 것과 같다.

```
<%  
패키지명.MemberBean member = new 패키지명. MemberBean();  
pageContext.setAttribute("member",member);  
%>
```

자바 빈이란

- scope속성은 자바빈 태그에 의해 생성된 객체가 사용될 수 있는 유효 범위를 결정한다.

scope 속성 값	의미
page	자바 빈은 생성된 페이지 내에서만 사용 할 수 있다
request	자바 빈은 생성된 요청을 수행하는 페이지들 에서만 사용 할 수 있다
session	자바 빈은 생성된 세션에서 요청을 처리하는 페이지들에서 사용될 수 있다
application	자바 빈은 생성된 응용 프로그램에 포함된 모든 페이지들에서 사용 될 수 있다.

자바 빈이란

- 마찬가지로 액션태그를 통해서 프로퍼티에 값을 얻거나 저장할 수 있다.

```
<jsp:getProperty name="빈 이름" property="프로퍼티 이름" />
```

```
<jsp:setProperty name="빈 이름" property="프로퍼티 이름" value="프로퍼티 값"/>
```

- 기존 표현식에서 어떻게 연결되는 지 확인한다

```
<%=member.getName()%>
```

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

Diagram: Two arrows originate from the `member` and `getName()` parts of the expression `<%=member.getName()%>`. One arrow points to the `name="member"` attribute of the `<jsp:getProperty>` tag, and the other points to the `property="name"` attribute.

```
<%=member.setName("홍길동")%>
```

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

Diagram: Three arrows originate from the `member`, `setName()`, and `"홍길동"` parts of the expression `<%=member.setName("홍길동")%>`. One arrow points to the `name="member"` attribute of the `<jsp:setProperty>` tag, another points to the `property="name"` attribute, and the third points to the `value="홍길동"` attribute.

- 예제 3 propertyDemo.jsp

자바 빈이란

- <jsp:setProperty>는 request.getProperty와 같은 역할을 하므로

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

```
Member.setName('request.getParameter("name")')
```

- 위 두 예제는 같다.

```
<jsp:setProperty name="member" property="*" />
```

- 위와 같은 형태로 한꺼번에 데이터를 가져와서 저장하는 것도 가능하다.

자바 빈이란

- 폼 양식을 통해 입력한 내용을 자바빈으로 처리 해보자
- MemberBean.java, addMemberForm.jsp, addMember.jsp

회원의 정보 입력 폼

이름

아이디

별명

비밀번호

이메일

전화번호

입력 완료된 회원 정보

이름 이정은

아이디 berrytop

별명 teacher

비밀번호 1234

이메일 berrytop@naver.com

전화번호 01062487440