

# Web应用开发 之 JSP技术模型

赵小敏

浙江工业大学计算机科学与技术学院

# 本节内容

- 3.8 JavaBeans

## 3.8 JavaBeans

- JavaBeans是Java平台的组件技术，在Java Web开发中常用JavaBeans来存放数据、封装业务逻辑等，从而很好地实现业务逻辑和表示逻辑的分离，使系统具有更好的健壮性和灵活性。
- 对程序员来说，JavaBeans最大的好处是可以实现**代码的重用**，另外对程序的易维护性等也有很大的意义。

## 3.8.1 JavaBeans规范

- JavaBeans是用Java语言定义的类，这种类的设计需要遵循JavaBeans规范的有关约定。任何遵循下面三个规范的Java类都可以作为JavaBeans使用。

（1）JavaBeans应该是public类，并且具有无参数的public构造方法。

（2）JavaBeans类的成员变量一般称为属性（property）。对每个属性访问权限一般定义为private，而不是public。

**注意：**属性名必须以小写字母开头。

## 3.8.1 JavaBeans规范

(3) 每个属性通常定义两个public方法，一个是访问方法（getter），另一个是修改方法（setter），使用它们访问和修改JavaBeans的属性值。访问方法名应该定义为getXxx()，修改方法名应该定义为setXxx()。

## 3.8.1 JavaBeans规范

- 例如，假设JavaBeans类中有一个String类型的color属性，
- 下面是访问方法和修改方法的定义：

```
public String getColor() {  
    return this.color;  
}
```

```
public void setColor(String color) {  
    this.color = color;  
}
```

## 3.8.1 JavaBeans规范

- 除了访问方法和修改方法外，JavaBeans类中还可以定义其他的方法实现某种业务逻辑。也可以只为某个属性定义访问方法，这样的属性就是只读属性。
- 示例：Customer类使用三个private属性封装了客户信息，并提供了访问和修改这些信息的方法。

# Customer.java

```
public class Customer{

    private String custName;
    private String email;
    private String phone;

    public Customer() {}
    public Customer(String custName, String email, String phone) {
        this.custName = custName;
        this.email = email;
        this.phone = phone;
        //get和set方法
    }
```



## 3.8.1 JavaBeans规范

### ➤ 使用JavaBeans的优点

- 在JSP页面中使用JavaBeans可使代码更简洁;
- JavaBeans有助于增强代码的可重用性;
- 它们是Java语言对象, 可以充分利用该语言面向对象的特征。

## 3.8.2 使用<jsp:useBean>动作

- 在JSP页面中使用JavaBeans主要是通过三个JSP标准动作实现的，它们分别是：
  - <jsp:useBean>动作
  - <jsp:setProperty>动作
  - <jsp:getProperty>动作

## 3.8.2 使用<jsp:useBean>动作

- <jsp:useBean>动作用来在JSP页面中查找或创建一个bean实例。一般格式如下：

```
<jsp:useBean id="beanName"
    scope="page|request|session|application"
    {class="package.class" |
    type="package.class" |
    }
{ /> | >其他元素</jsp:useBean> }
```

# 1. 属性说明

- **id属性**用来唯一标识一个bean实例，该属性是必须的。
- 在JSP页面实现类中，id的值被作为Java语言的变量，因此可以在JSP页面的表达式和小脚本中使用该变量。

# 1. 属性说明

- `scope`属性指定bean实例的作用域。与隐含对象类似，JavaBeans在JSP页面中的存在和可访问性是由4个JSP作用域决定的：
  - `page`
  - `request`
  - `session`
  - `application`。
- 该属性是可选的，默认值为`page`作用域。如果`page`指令的`session`属性设置为`false`，则bean不能在JSP页面中使用`session`作用域。

# 1. 属性说明

- **class属性**指定创建bean实例的Java类。如果容器在指定的作用域中不能找到一个现存的bean实例，它将使用class属性指定的类创建一个bean实例。如果该类属于某个包，则必须指定类的全名，如 `com.demo.Customer`。
- **type属性**指定由id属性声明的变量的类型，由于该变量是在请求时指向实际的bean实例，其类型必须与bean类的类型相同或者是其超类，或者是一个bean类实现的接口。同样，如果类或接口属于某个包，需要指定其全名，如 `com.demo.Customer`。

## 2. 属性的使用

- 在<jsp:useBean>动作的属性中，id属性是必须的，scope属性是可选的。class和type至少指定一个或两个同时指定。

1) 只指定class属性的情况

```
<jsp:useBean id="customer"  
class="com.demo.Customer" />
```

## 2. 属性的使用

- 当JSP页面执行到该动作时，Web容器首先在page作用域中查找名为customer的bean实例。
- 如果找到就用customer引用指向它，如果找不到，将使用Customer类创建一个对象，并用customer引用指向它，同时将其作为属性添加到page作用域中。
- 该bean只能在它所定义的JSP页面中使用，且只能被该页面创建的请求使用。



## 2. 属性的使用

- 该动作与下面的一段代码等价：

```
Customer customer =  
(Customer)pageContext.getAttribute("customer");  
if (customer == null){  
    customer = new Customer();  
    pageContext.setAttribute(  
        "customer", customer);  
}
```

## 2. 属性的使用

- 下面的动作使用了id、class和scope属性声明一个JavaBeans:

```
<jsp:useBean id="customer"  
class="com.demo.Customer" scope="session" />
```

- 当JSP页面执行到该动作时，容器在会话（session）作用域中查找或创建bean实例，并用customer引用指向它。这个过程与下面的代码等价：

```
Customer customer =(Customer)session.getAttribute("customer");  
if (customer == null) {  
    customer = new Customer();  
    session.setAttribute("customer", customer);  
}
```

## 2. 属性的使用

### 2) 只指定type属性的情况

- 可以使用type属性代替class属性，例如：

```
<jsp:useBean id="customer"  
type="com.demo.Customer" scope="session"/>
```

- 该动作在指定作用域中查找类型为Customer的实例，如果找到用customer指向它，如果找不到产生Instantiation异常。
- 使用type属性必须保证bean实例存在。

## 2. 属性的使用

### 3) 同时使用class和type属性的情况

```
<jsp:useBean id="customer"  
class="com.demo.Customer"  
type="com.demo.Customer" scope="session"/>
```

### 3.8.3 使用<jsp:setProperty>动作

- <jsp:setProperty>动作用来给bean实例的属性赋值，它的格式如下。

```
<jsp:setProperty name="beanName"  
    { property = "propertyName"  
      value="{string | <%=expression%>}" |  
      property="propertyName" [param="paramName"] |  
      property = "*" } />
```

# 1. 属性说明

- **name属性**用来标识一个bean实例，该实例必须是前面使用<jsp:useBean>动作声明的，并且name属性值必须与<jsp:useBean>动作中指定的一个id属性值相同。该属性是必须的。
- **property属性**指定要设置值的bean实例的属性，容器将根据指定的bean的属性调用适当的setXxx()，因此该属性是必须的。

# 1. 属性说明

- `value`属性为bean的属性指定新值，该属性值可以接受请求时属性表达式。
- `param`属性指定请求参数名，如果请求中包含指定的参数，那么使用该参数值来设置bean的属性值。
- `value`属性和`param`属性都是可选的并且不能同时使用。如果这两个属性都没有指定，容器将查找与属性同名的请求参数。

## 2. 属性使用

- 假设已按下面的代码声明了一个bean实例。

```
<jsp:useBean id="customer"  
class="com.demo.Customer" />
```

### 1) 使用value属性

- 下面动作将名为customer的custName、email和phone属性值分别设置为"Mary"、"mary@163.com"和"8899123"。

```
<jsp:setProperty name="customer" property="custName"  
value="Mary"/>
```

```
<jsp:setProperty name="customer" property="email"  
value="mary@163.com"/>
```

```
<jsp:setProperty name="customer" property="phone"  
value="8899123"/>
```



## 2. 属性使用

### 2) 使用param属性

- 下面的例子中没有指定value属性的值，而是使用param属性指定请求参数名。

```
<jsp:setProperty name="customer" property="email"  
param="myEmail" />
```

```
<jsp:setProperty name="customer" property="phone"  
param="myPhone" />
```

## 2. 属性使用

### 3) 使用默认参数机制

- 如果请求参数名与bean的属性名匹配，就不必指定param属性或value属性，如下所示。

```
<jsp:setProperty name="customer" property="email" />
```

```
<jsp:setProperty name="customer" property="phone" />
```

## 2. 属性使用

4) 在一个动作中设置所有属性

- 在一个动作中设置bean的所有属性

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

- `property`的属性值指定"`*`", 它将使用请求参数的每个值为属性赋值, 这样就不用单独为bean的每个属性赋值。

## 3.8.4 使用<jsp:getProperty>动作

- <jsp:getProperty>动作检索并向输出流中打印bean的属性值，它的语法非常简单。

```
<jsp:getProperty name="beanName"  
                  property="propertyName" />
```

- 该动作只有两个属性name和property，并且都是必须的。name属性指定bean实例名，property属性指定要输出的属性名。
- 下面的动作指示容器打印customer的email和phone属性值。

```
<jsp:getProperty name="customer" property="email" />  
<jsp:getProperty name="customer" property="phone" />
```

## 3.8.5 JavaBeans应用示例

- 下面示例首先在inputCustomer.jsp中输入客户信息，然后将控制转到CustomerServlet，最后将请求转发到displayCustomer.jsp页面。

程序inputCustomer.jsp

- 在Servlet代码中创建JavaBeans类的实例，以及如何使用作用域对象共享它们。可以直接在Servlet中使用JavaBeans。并且可以在JSP页面中和Servlet中共享bean实例。

程序CustomerServlet.java

- 这个例子说明在Servlet中可以把JavaBeans对象存储到作用域对象中。这里需要注意的是会话作用域对象的访问使用了同步（synchronized）代码块，这是因为

## 3.8.5 JavaBeans应用示例

HttpSession对象不是线程安全的，其他Servlet和JSP页面可能在多个线程中同时访问或修改这些对象。

- 如果要在JSP页面中使用存储在会话作用域中的bean对象，如下声明即可。

```
<jsp:useBean id="customer"  
class="com.demo.Customer"  
scope="session" />
```

- 下面的页面在会话作用域内查找CustomerBean的一个实例并用表格的形式打印出它的属性值。

## 3.8.5 JavaBeans应用示例

程序displayCustomer.jsp

- 该页面首先在会话作用域内查找名为customer的bean实例，如果找到将输出bean实例的各属性值，如果找不到将创建一个bean实例并使用同名的请求参数为bean实例的各属性赋值，最后也输出各属性的值。

# 小 结

- JavaBeans是遵循一定规范的Java类，它在JSP页面中主要用来表示数据。JSP规范提供了下面三个标准动作：
  - <jsp:useBean>
  - <jsp:setProperty>
  - <jsp:getProperty>



# 作业

- 1、某学校的学生信息管理系统有一个录入学生信息的功能，请按如下要求编写程序。
  - (1) 编写一个名为Student的JavaBeans，包括3个属性：stuid表示学号、name表示学生姓名和major表示专业；
  - (2) 编写输入学生信息页面inputStudent.jsp，通过表单输入学生信息，将请求转发到StudentServlet；
  - (3) StudentServlet从JSP页面得到学生信息，并将学生信息通过作用域共享后转发至学生信息显示页面displayStudent.jsp；
  - (4) displayStudent.jsp显示学生的信息；