



Using JavaBeans in JSPs Pages Introduction



Agenda

1

Introduction to Java Beans

Objectives

At the end of this module, you will be able to:

- Explain the JavaBean with its properties

Introduction to Java Beans

- A Java Bean is a reusable software component that works with Java
- Any Java class that follows certain design conventions can be a JavaBeans component
- It must follow certain conventions about method naming, construction, and behavior
 - Must have a public default constructor
 - Properties must be declared private
 - Properties must be accessible using *get*, *set*, and other methods (accessor and mutator methods) obeying a standard naming convention
- JSP technology directly supports using JavaBeans components with JSP language elements
- You can easily create and initialize beans and get and set the values of their properties

Identifying the property of a bean

```
public class A1 {  
    private int id;  
    private String username;  
    public String getName(){  
        return username;  
    }  
    public void setName(String x){  
        username=x;  
    }  
    public void setEmpID(int y){  
        id=y;  
    }  
    public int getEmpID(){  
        return id;  
    }  
}
```

What are the properties of this bean?

name

empID

Actions: *The jsp:useBean Action*

- Lets you load in a JavaBean to be used in the JSP page
- The simplest syntax for specifying that a bean should be used is:

```
<jsp:useBean id="name" class="package.class" />
```
- This usually means "instantiate an object of the class specified by class, and bind it to a variable with the name specified by id."

Actions: The *jsp:useBean* Action (Contd.).

```
package p1;  
  
public class Example1 {  
    private int age;  
    public int getAge(){  
    }  
    public void setAge(int i){  
    }  
}
```

If we have to instantiate the given class in java, we will use the following syntax :

p1.Example1 obj1 = new p1.Example1();

In jsp, the same result is achieved using jsp:useBean :

<jsp:useBean id="obj1" class="p1.Example1" />

<jsp:getProperty>

- Converts property names following the bean standards

- Has two attributes:

name="beanInstanceName"

- The name of the Bean instance as declared in a <jsp:useBean> tag

property="propertyName"

- The name of the Bean property whose value you want to display

Example on jsp:getProperty

```
package jspbean;

import java.util.Calendar;

public class TimeBean{
    private int hour,minute,second;

    public TimeBean(){
        Calendar cal = Calendar.getInstance();

        hour = cal.get(Calendar.HOUR);
        minute = cal.get(Calendar.MINUTE);
        second = cal.get(Calendar.SECOND);
    }

    public int getHour()    { return hour; }
    public int getMinute() { return minute; }
    public int getSecond() { return second; }
    public String getMessage(){
        return "Time: "+hour+" "+minute+" "+second;
    }
}
```

Example on jsp:getProperty (Contd.).

```
<HTML>
```

```
<HEAD><TITLE>Time Bean </TITLE></HEAD>
```

```
<BODY>
```

```
<jsp:useBean id="b2" class="jspbean.TimeBean" />
```

```
Hour=<jsp:getProperty name="b2" property="hour" /> <BR>
```

```
Minute=<jsp:getProperty name="b2" property="minute"/><BR>
```

```
Second<jsp:getProperty name="b2" property="second"/><BR>
```

```
Message= <jsp:getProperty name="b2" property="message"/>
```

```
</BODY>
```

```
</HTML>
```

<jsp:setProperty>

- Sets the value of one or more properties in a JavaBean component

Example:

```
<jsp:useBean id="person" class="p1.A1" />
<jsp:setProperty name="person" property="empID" value="999"/>
<jsp:setProperty name="person" property="empID" value = "<%=
    request.getParameter("Employee-Id") %>"/>
<jsp:setProperty name="person" property="name" param="username"
/>
<jsp:setProperty name="person" property="empID" />
<jsp:setProperty name="person" property="*" />
```

Properties of JSP:setProperty

1. **name:** This required attribute designates the bean whose property will be set. The `jsp:useBean` element must appear before the `jsp:setProperty` element.
2. **property:** This required attribute indicates the property you want to set. However, there is one special case: a value of "*" means that all request parameters whose names match bean property names will be passed to the appropriate setter methods.
3. **value:** This optional attribute specifies the value for the property. String values are automatically converted to numbers, boolean to Boolean, byte to Byte, char to Character via the standard `valueOf` method in the target or wrapper class. For example, a value of "true" for a boolean or Boolean property will be converted via `Boolean.valueOf`, and a value of "42" for an int or Integer property will be converted via `Integer.valueOf`.

Example on jsp:setProperty

CircleBean.java

```
package jspbean;

public class CircleBean{
    private double radius;
    public void setRadius(double r) { radius = r; }
    public double getArea() { return Math.PI*radius*radius; }
}
```

Circle.jsp

```
<BODY>

    <jsp:useBean id="cb" class="jspbean.CircleBean" />
    <jsp:setProperty name="cb" property="radius" value="2.0" />
    Reduced Form: <jsp:getProperty name="cb" property="area" />

</BODY>
```

Quiz

1. Identify the implicit objects in JSP
a. *request* b. *response* c. *out* d. *session* e. *application*
2. Is JSP page extensible?(TRUE/FALSE)
3. JSP handles runtime errors using _____ attribute in page directive.
2. How do I use comments within a JSP page?

Summary

In this module, you were able to:

- Describe the Software Component Assembly Model
- Explain Java's approach to developing software components
- Develop a simple Bean
- Distinguish JSP architecture vis-à-vis servlets
- Define and use the basic JSP Elements
- Create and use Java Beans

References

1. Hans Bergsten (2012). *Java Server Pages*, Retrieved on April 12, 2012, from, <http://shop.oreilly.com/product/9781565927469.do>
2. Stephanie Bodoff (2012). *JavaServer Pages Technology*. Retrieved on April 12, 2012, from, http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/JSPIntro.html
3. coe.neu.edu (2012). *How JSP Works*. Retrieved on April 24 , 2012, from, <http://www1.coe.neu.edu/~yozbek/isyg250/jsp.pdf>
4. Marty Hall (1999). *Java Server Pages (JSP) 1.0*. Retrieved April 24, 2012, from, <http://users.polytech.unice.fr/~buffa/cours/internet/POLYS/servlets/Servlet-Tutorial-JSP.html>
5. Apl.jhu.edu (1999). *Java Server Pages(JSP) 1.0*. Retrieved April 24, 2012, from, <http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/Servlet-Tutorial-JSP.html>
6. Paul Wheaton (2012). *Java - JSP*. Retrieved April 24, 2012, from, <http://www.coderanch.com/t/281980/JSP/java/Hyperlinks>



Thank You