

# JSP - JavaBeans

A JavaBean is a specially constructed Java class written in the Java and coded according to the JavaBeans API specifications.

Following are the unique characteristics that distinguish a JavaBean from other Java classes –

- It provides a default, no-argument constructor.
- It should be serializable and that which can implement the **Serializable** interface.
- It may have a number of properties which can be read or written.
- It may have a number of "**getter**" and "**setter**" methods for the properties.

## JavaBeans Properties

A JavaBean property is a named attribute that can be accessed by the user of the object. The attribute can be of any Java data type, including the classes that you define.

A JavaBean property may be **read**, **write**, **read only**, or **write only**. JavaBean properties are accessed through two methods in the JavaBean's implementation class –

S.No.	Method & Description
1	<b>getPropertyName()</b> For example, if property name is firstName, your method name would be <b>getFirstName()</b> to read that property. This method is called accessor.
2	<b>setPropertyName()</b> For example, if property name is firstName, your method name would be <b>setFirstName()</b> to write that property. This method is called mutator.

A read-only attribute will have only a **getPropertyName()** method, and a write-only attribute will have only a **setPropertyName()** method.

## JavaBeans Example

Consider a student class with few properties –

```
package com.tutorialspoint;

public class StudentsBean implements java.io.Serializable {
    private String firstName = null;
    private String lastName = null;
    private int age = 0;

    public StudentsBean() {
    }

    public String getFirstName(){
        return firstName;
    }

    public String getLastName(){
        return lastName;
    }

    public int getAge(){
        return age;
    }

    public void setFirstName(String firstName){
        this.firstName = firstName;
    }

    public void setLastName(String lastName){
        this.lastName = lastName;
    }

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

## Accessing JavaBeans

The **useBean** action declares a JavaBean for use in a JSP. Once declared, the bean becomes a scripting variable that can be accessed by both scripting elements and other custom tags used in the JSP. The full syntax for the useBean tag is as follows –

```
<jsp:useBean id = "bean's name" scope = "bean's scope" typeSpec/>
```

Here values for the scope attribute can be a **page**, **request**, **session** or **application** **based** on your requirement. The value of the **id** attribute may be any value as long as it is a unique name among other **useBean declarations** in the same JSP.

Following example shows how to use the useBean action –

```
<html>
  <head>
    <title>useBean Example</title>
  </head>

  <body>
    <jsp:useBean id = "date" class = "java.util.Date" />
    <p>The date/time is <%= date %>
  </body>
</html>
```

You will receive the following result – –

```
The date/time is Thu Sep 30 11:18:11 GST 2010
```

## Accessing JavaBeans Properties

Along with **<jsp:useBean...>** action, you can use the **<jsp:getProperty/>** action to access the get methods and the **<jsp:setProperty/>** action to access the set methods. Here is the full syntax –

```
<jsp:useBean id = "id" class = "bean's class" scope = "bean's scope">
  <jsp:setProperty name = "bean's id" property = "property name"
    value = "value"/>
  <jsp:getProperty name = "bean's id" property = "property name"/>
  .....
</jsp:useBean>
```

The name attribute references the id of a JavaBean previously introduced to the JSP by the useBean action. The property attribute is the name of the **get** or the **set** methods that should be invoked.

Following example shows how to access the data using the above syntax –

```

<html>
  <head>
    <title>get and set properties Example</title>
  </head>

  <body>
    <jsp:useBean id = "students" class = "com.tutorialspoint.StudentsBean">
      <jsp:setProperty name = "students" property = "firstName" value =
"Zara"/>
      <jsp:setProperty name = "students" property = "lastName" value =
"Ali"/>
      <jsp:setProperty name = "students" property = "age" value = "10"/>
    </jsp:useBean>

```


[Chapters](#) ▾

[Categories](#) ≡

```

    <jsp:getProperty name = "students" property = "firstName"/>
  </p>

  <p>Student Last Name:
    <jsp:getProperty name = "students" property = "lastName"/>
  </p>

  <p>Student Age:
    <jsp:getProperty name = "students" property = "age"/>
  </p>

</body>
</html>

```

Let us make the **StudentsBean.class** available in CLASSPATH. Access the above JSP. the following result will be displayed –

Student First Name: Zara

Student Last Name: Ali

Student Age: 10

## TOP TUTORIALS

Python Tutorial  
Java Tutorial  
C++ Tutorial  
C Programming Tutorial  
C# Tutorial  
PHP Tutorial  
R Tutorial  
HTML Tutorial  
CSS Tutorial  
JavaScript Tutorial  
SQL Tutorial

## TRENDING TECHNOLOGIES

Cloud Computing Tutorial  
Amazon Web Services Tutorial  
Microsoft Azure Tutorial  
Git Tutorial  
Ethical Hacking Tutorial  
Docker Tutorial  
Kubernetes Tutorial  
DSA Tutorial  
Spring Boot Tutorial  
SDLC Tutorial  
Unix Tutorial

## CERTIFICATIONS

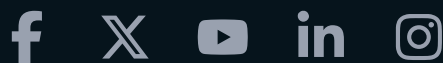
Business Analytics Certification  
Java & Spring Boot Advanced Certification  
Data Science Advanced Certification  
Cloud Computing And DevOps  
Advanced Certification In Business Analytics  
Artificial Intelligence And Machine Learning  
DevOps Certification

Game Development Certification  
Front-End Developer Certification  
AWS Certification Training  
Python Programming Certification

## COMPILERS & EDITORS

Online Java Compiler  
Online Python Compiler  
Online Go Compiler  
Online C Compiler  
Online C++ Compiler  
Online C# Compiler  
Online PHP Compiler  
Online MATLAB Compiler  
Online Bash Terminal  
Online SQL Compiler  
Online Html Editor

[ABOUT US](#) | [OUR TEAM](#) | [CAREERS](#) | [JOBS](#) | [CONTACT US](#) | [TERMS OF USE](#) |  
[PRIVACY POLICY](#) | [REFUND POLICY](#) | [COOKIES POLICY](#) | [FAQ'S](#)



Tutorials Point is a leading Ed Tech company striving to provide the best learning material on technical and non-technical subjects.

© Copyright 2025. All Rights Reserved.