



Docs & Support

HOME > Docs & Support > NetBeans IDE 5.0

NetBeans IDE 5.0 Quick Start Guide for Web Applications

NetBeans IDE 5.5

This document takes you through the basics of using NetBeans IDE 5.0 to develop web applications. This document is designed to get you going as quickly as possible. For more information on working with NetBeans IDE, see the [Support and Docs page](#) on the NetBeans website.

[Feedback](#)

NetBeans IDE 5.0

Using NetBeans Installation Guide

You create, deploy, and execute a simple web application. The application uses a JavaServer Pages™ page to ask you to input your name. It then uses a JavaBeans™ component to persist the name during the HTTP session and repeats the name on another JavaServer Pages page.

NetBeans IDE 4.1

Setting Up a Web Application Project

Older Releases

Before you start writing code, you have to make sure you have all of the necessary software and that your project is set up correctly.

User FAQs

Installing the Software

Articles

Before you begin, you need to install the following software on your computer:

- NetBeans IDE 5.0 ([download](#)).
- Java Standard Development Kit (JDK™) version 1.4.2 ([download](#)) or 5.0 ([download](#))

Flash Demos

Books

Optionally, you can download and use the Sun Java System (SJS) Application Server ([download](#)), JBoss, or WebLogic. However, the Tomcat Web Server that is bundled with the IDE provides all the support you need for two-tier web applications such as the one described in this quick start guide. An application server (such as the SJS Application Server, JBoss, or WebLogic) is only required when you want to **develop enterprise applications**.

Learning Java

Registering the Server with the IDE

Module Developer's Resources

The bundled Tomcat Web Server is registered with the IDE automatically. However, before you can deploy to the SJS Application Server, JBoss, or WebLogic, you have to register a local instance with the IDE. If you installed the NetBeans IDE 5.0/SJS Application Server bundle, a local instance of the SJS Application Server is registered automatically. Otherwise, take the following steps:

NetBeans Support

1. Choose Tools > Server Manager from the main window.
2. Click Add Server. Select the server type and give a name to the instance. Then click Next.
3. Specify the server information, the location of the local instance of the application server, and the domain to which you want to deploy.

Creating a New Web Application Project

1. Choose File > New Project. Under Categories, select Web. Under Projects, select Web Application and click Next.
2. Under Project Name, enter `HelloWeb`. Notice that the Context Path is `/HelloWeb`.
3. Change the Project Location to any directory on your computer. From now on, this directory is referred to as `$PROJECTHOME`.
4. Select the recommendations to which your source structure will adhere, which is purely a personal preference:
 - **Java BluePrints**. For more information, see <http://java.sun.com/blueprints/code/projectconventions.html>.
 - **Jakarta**. For more information, see <http://jakarta.apache.org/tomcat/tomcat-5.0-doc/>.
5. Select the server to which you want to deploy your application. Only servers that are registered with the IDE are listed.
6. Leave the Set as Main Project checkbox selected. Click Finish.

The IDE creates the `$PROJECTHOME/HelloWeb` project folder. The project folder contains all of your sources and project metadata, such as the project's Ant build script. The HelloWeb project opens in the IDE. You can view its logical structure in the Projects window and its file structure in the Files window.

Creating and Editing Web Application Source Files

Creating and editing source files is the most important function that the IDE serves. After all, that's probably what you spend most of your day doing. The IDE provides a wide range of tools that can compliment any developer's personal style, whether you prefer to code everything by hand or want the IDE to generate large chunks of code for you.

Creating a Java Package and a Java Source File

1. Expand the Source Packages node. Note the Source Packages node only contains an empty default package node.
2. Right-click the Source Packages node and choose New > Java Class. Enter `NameHandler` in the Class Name text box and type `org.me.hello` in the Package drop-down. Click Finish.
3. In the Source Editor, declare a field by typing the following line directly below the class declaration:

```
String name;
```

4. Add the following line in the Constructor:

```
name = null;
```

Generating Getter and Setter Methods

1. Right-click the word `name` in the field declaration at the start of the class and choose Refactor > Encapsulate Fields. Click Next to run the command with its default options.
2. Click Do Refactoring. Getter and setter methods are generated for the `name` field and its access level is changed to private. The Java class should

now look similar to this:

```
package org.me.hello;

/**
 *
 * @author Administrator
 */

public class NameHandler {

    private String name;

    /** Creates a new instance of NameHandler */
    public NameHandler() {
        setName(null);
    }

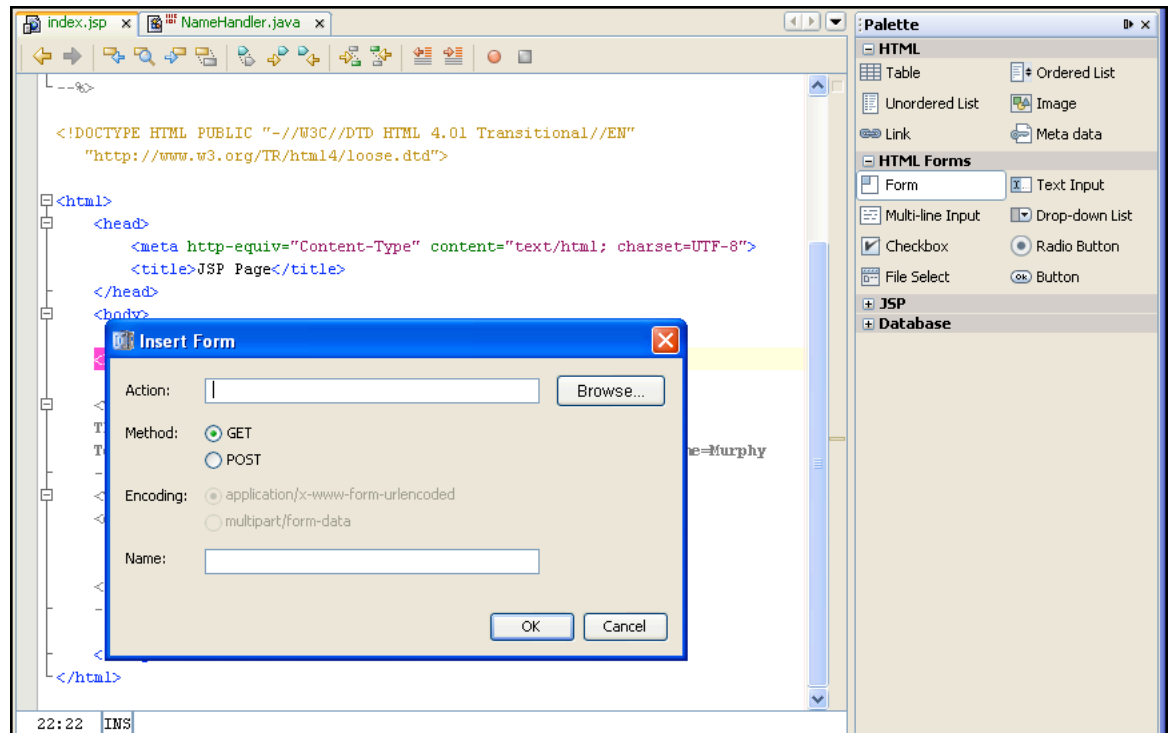
    public String getName() {
        return name;
    }

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

}
```

Editing the Default JavaServer Pages File

1. Expand the HelloWeb project node and the Web Pages node. Note that the IDE has created a default JavaServer Pages file, `index.jsp`, for you. When you create the project, the IDE opened the `index.jsp` file in the Source Editor.
2. Select the `index.jsp` Source Editor tab. The `index.jsp` file now has focus in the Source Editor.
3. In the Palette on the right side of the Source Editor, expand HTML Forms and drag a Form item below the `<h1>` tags in the Source Editor:



Set the following values:

- **Action:** response.jsp
- **Method:** GET
- **Name:** Name Input Form

Click OK. The Form is added to the `index.jsp` file.

4. Drag a Text Input item to just before the `</form>` tag.

Set the following values:

- **Name:** name
- **Type:** text

Click OK. The Text Input is added between the `<form>` tags.

5. Drag a Button item to just before the `</form>` tag.

Set the following values:

- **Label:** OK
- **Type:** submit

Click OK. The Button is added between the `<form>` tags.

6. Type `Enter your name:` in front of the `<input>` tag and change the text between the `<h1>` tags to `Entry Form`.

The tags between the `<body>` tags now look as follows:

```
<h1>Entry Form</h1><form name="Name Input Form" action="response.jsp" method="GET">
Enter your name: <input type="text" name="name" value="" /><input type="submit" value="OK" /></form>
```

Creating a JavaServer Pages File

1. Expand the HelloWeb project node and the Web Pages node.
2. Right-click the Web Pages node and choose `New > JSP`, name the JavaServer Pages file `response`, and click `Finish`.

The new `response.jsp` opens in the Source Editor.

3. In the Palette on the right side of the Source Editor, expand `JSP` and drag a `Use Bean` item right below the `<body>` tag in the Source Editor.

Set the following values:

- **ID:** mybean
- **Class:** org.me.hello.NameHandler
- **Scope:** session

Click OK. The `Use Bean` is added below the `<body>` tag.

4. Add a `Get Bean Property` item and a `Set Bean Property` item from the Palette. Then change the code so that the tags between the `<body>` tags look as follows:

```
<jsp:useBean id="mybean" scope="session" class="org.me.hello.NameHandler" />
<jsp:setProperty name="mybean" property="*" />
<h1>Hello, <jsp:getProperty name="mybean" property="name" />!</h1>
```

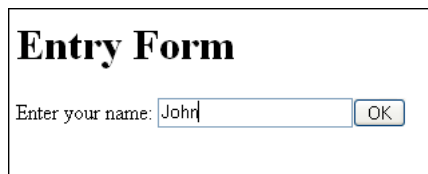
Building and Running a Web Application Project

The IDE uses an Ant build script to build and run your web applications. The IDE generates the build script based on the options you enter in the `New Project` wizard and the project's `Project Properties` dialog box.

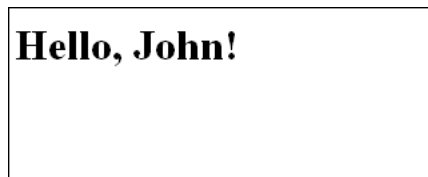
1. Choose `Run > Run Main Project (F6)` from the `Run` menu.

The IDE builds the web application and deploys it, using the server you specified when creating the project.

2. Enter your name in the text box on your deployed `index.jsp` page:



Click OK. The `response.jsp` page should open and greet you:



Next Steps

For more information about using NetBeans IDE 5.0, see the following resources:

- NetBeans IDE 5.0 Quick Start Guides:
 - [For J2SE Applications](#)
 - [For Web Applications](#)
 - [For Web Services](#)
 - [For Web Service Clients](#)
 - [For J2ME MIDP Applications](#)
- NetBeans IDE 5.0 Import Guides:
 - [For J2SE Applications](#)
 - [For Web Applications](#)
 - [For J2EE Applications](#)
 - [For J2ME MIDP Applications](#)
- NetBeans IDE 5.0 Tutorials:
 - [For Tag Libraries](#)
 - [For Applets](#)

- [NetBeans IDE Support and Docs page](#)

To send comments and suggestions, get support, and keep informed on the latest developments on the NetBeans IDE J2EE development features, join the nbj2ee@netbeans.org mailing list. For more information about upcoming J2EE development features in NetBeans IDE, see <http://j2ee.netbeans.org/>.

[MORE INFO:](#) | [HOME](#) | [SHOP](#) | [REPORT A BUG](#) | [SITE MAP](#) | [LEGAL](#) | [CONTACT](#) BY USE OF THIS WEBSITE, YOU AGREE TO THE [NETBEANS POLICIES AND TERMS OF USE](#)