

Course: Web Application Development

Lab Instructor: Vo Duy Khoi, MSc

Email: khoivd@hcmiu.edu.vn

Web Resources: <http://125.234.238.107/web/web-application-development>

Lab 3 - Introduction to Servlet Programming

Content:

- Introduction to Java Servlet Technology
- How to create a servlet and run on NetBeans IDE
- Practices and Exercises

Duration: 3 hours

Part 1: Introduction to Servlet and Servlet Life Cycle.

- What is Servlet?

- A *servlet* is a Java programming language class that is used to extend the capabilities of servers that host applications access via a request-response programming model.
- Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers. For such applications, Java Servlet technology defines HTTP-specific servlet classes.

- How Java support Servlet Programming

- The javax.servlet and javax.servlet.http packages provide interfaces and classes for writing servlets. All servlets must implement the Servlet interface, which defines servlet life-cycle methods.
- The **HttpServlet** class provides methods, such as **doGet** and **doPost**, for handling HTTP-specific services.
- When implementing a generic service, you can use or extend the **GenericServlet** class provided with the Java Servlet API.

- Tomcat Servlet/JSP container

- **Tomcat** can act as a stand-alone Web server and also as a servlet/JSP engine for other Web servers. When you download the Tomcat server, you really get a number of packages. **Catalina** and **Jasper** are the names of the servlet and JSP containers.
- Tomcat by itself is a web server. This means that you can use Tomcat to service HTTP requests for servlets, as well as static files (HTML, image files, and so on). In practice, however, since it is faster for non-servlet, non-JSP

requests, Tomcat normally is used as a module with another more robust web server, such as Apache web server or Microsoft Internet Information Server (IIS).

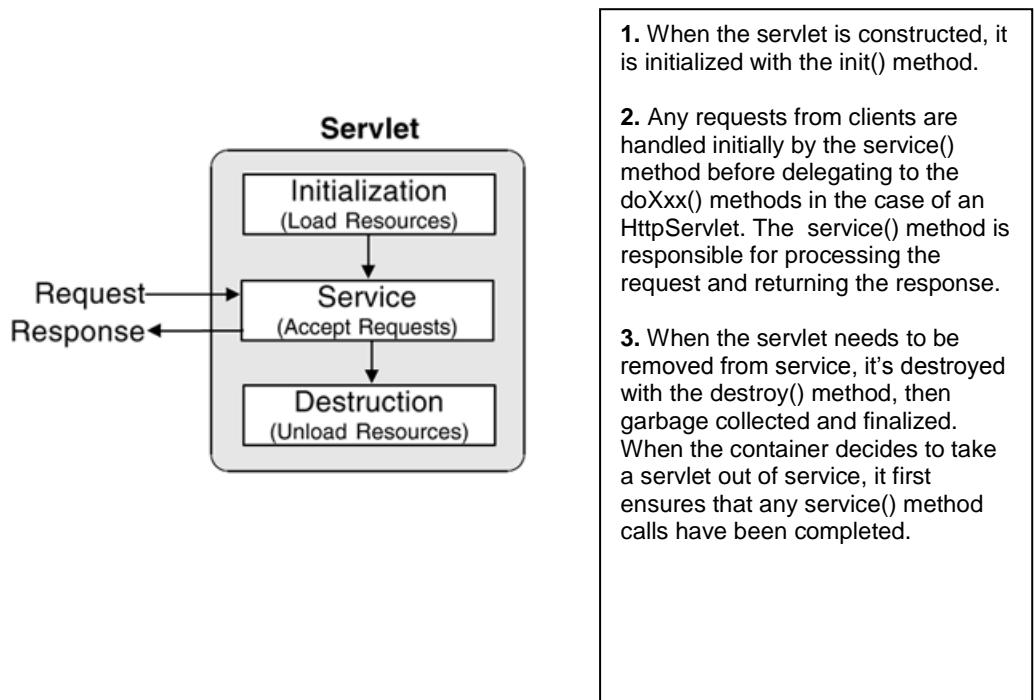
- Tomcat is not a J2EE application server. However, as J2EE app servers must themselves contain a servlet container to support the servlet/JSP APIs, J2EE app servers can embed Tomcat into their code to provide support for the Servlet and JSP APIs. One example of just such an application server is the popular open source JBoss J2EE app server (<http://www.jboss.org/>).

- **Servlet Life Cycle**

- The javax.servlet.Servlet interface defines the methods that all servlets must implement and, among others, three methods that are known as **life-cycle methods**:

```
public void init(ServletConfig config) throws ServletException  
public void service(ServletRequest req, ServletResponse res) throws ServletException,  
IOException  
public void destroy()
```

- These life-cycle methods are each called at separate times during the life span of a servlet, from the initial creation to the moment it's removed from service and destroyed. These methods are called in the following order:



- The `init` method is called by the servlet container after the servlet class has been instantiated. The servlet container calls this method exactly once to indicate to the servlet that the servlet is being placed into service. The `init` method is important also because the servlet container passes a `ServletConfig`

object, which contains the configuration values stated in the web.xml file for this application.

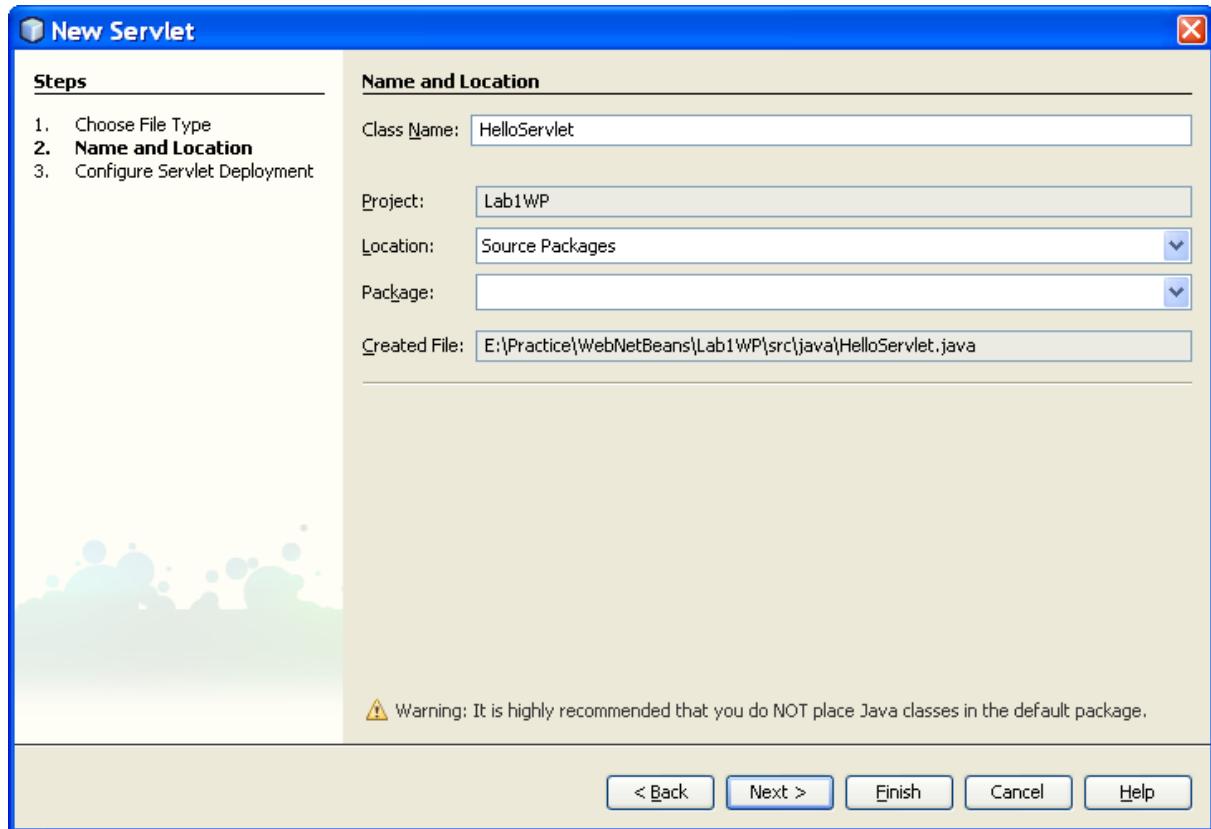
- The **service** method is called by the servlet container after the servlet's **init** method to allow the servlet to respond to a request. The servlet container passes a `ServletRequest` object and the `ServletResponse` object. The `ServletRequest` object contains the client's request and the `ServletResponse` contains the servlet's response.
- The servlet container calls the **destroy** method before removing a servlet instance from service. This normally happens when the servlet container is shut down or the servlet container needs some free memory.

(More information: refer from textbook: **Core Servlets and JavaServer Pages**)

Part 2: How to create a Servlet on NetBeans IDE.

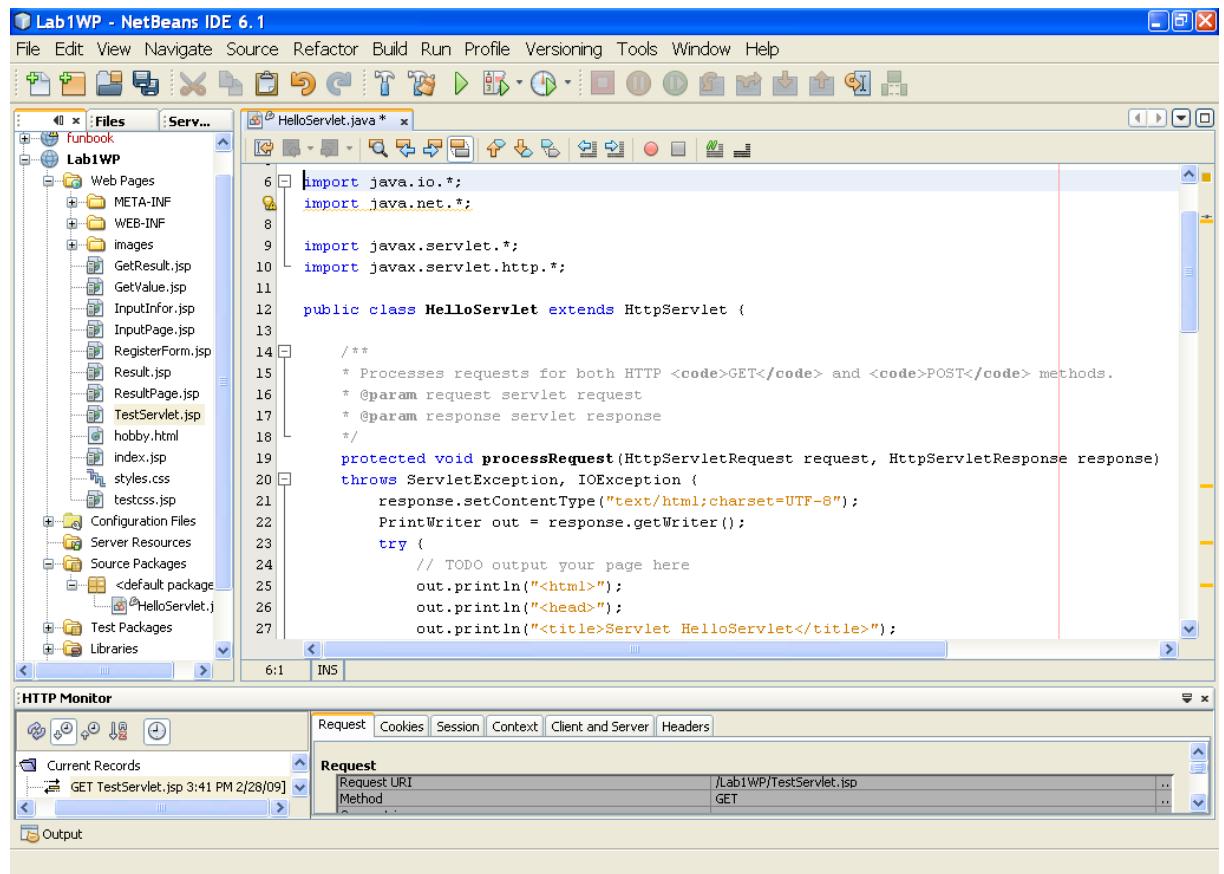
- Default servlet creation

After create a website (lab 1) on NetBeans IDE, right_click to the name of website -> New -> Servlet....

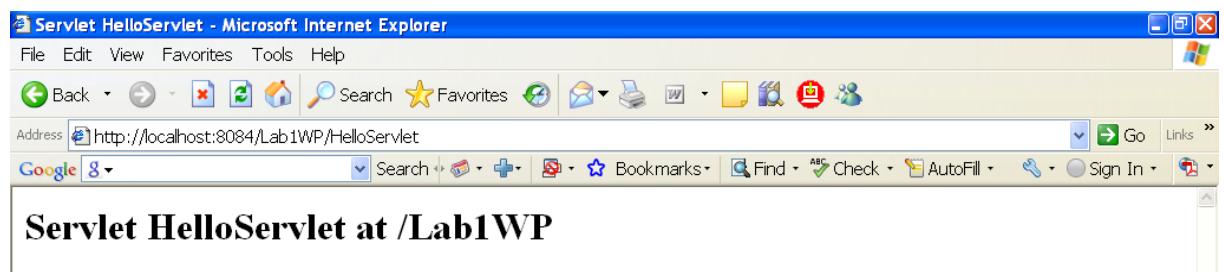


Type a name and click next, then click Finish button, the source codes generate automatically in the file HelloServlet.java.

This is source code:

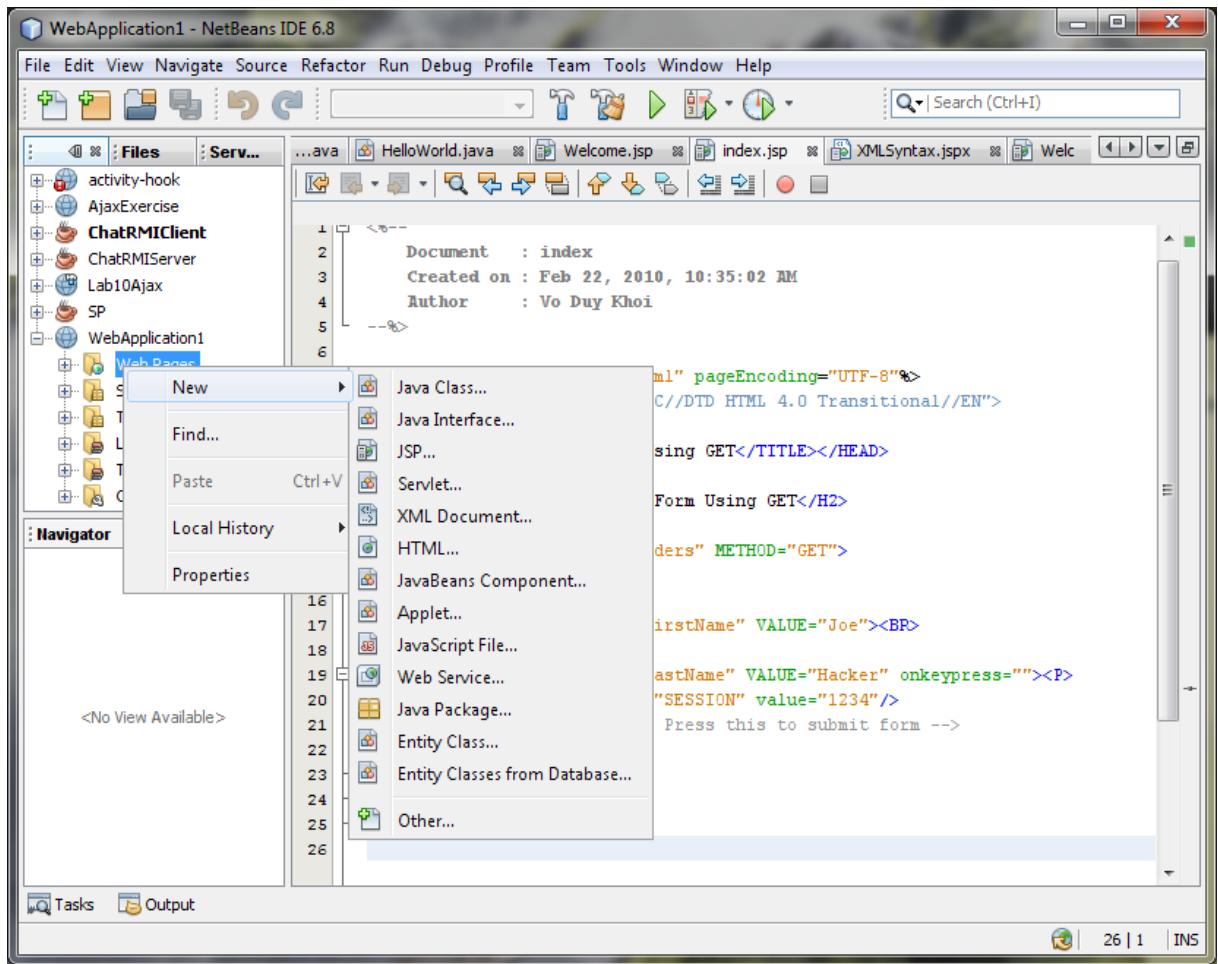


After building and running on the web server, this is the result page.



- Custom servlet creation

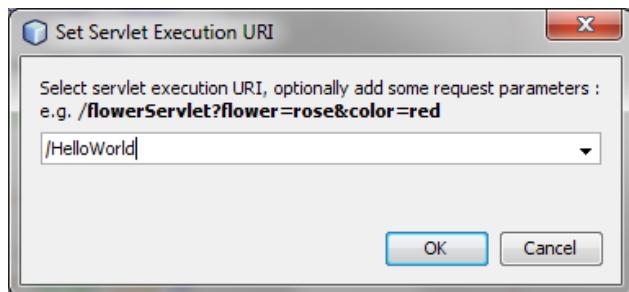
During application development, we might need to create more servlet. The steps to create new servlet in project are illustrated as figures:



Choose menu “Servlet” to create new servlet. You should not choose normal “Java Class” then write class as servlet. If you do so, you have to configure to activate servlet manually.

- How to run a servlet

A servlet can be called by name of class without extension. For example, if you create a servlet class named “HelloWorld.class” and run it under context /Lab3 then your URL to run the servlet will be “/Lab3/HelloWorld” and one message will display as follows:



Part 3: Practices and Exercises.

Exercise1: design a form (ThreeParams.JSP) as below:

JSP Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address: http://localhost:8084/TestServlet/

Google g Search Bookmarks Find Check AutoFill Sign In

Collect Three Parameters

First Parameter: John

Second Parameter: David

Third Parameter: Salar

After input three values in the textbox -> click Submit button, it call result page GetThreeParam.class (this file is built from GetThreeParam.java)

Reading Three Request Parameters - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address: http://localhost:8084/TestServlet/ThreeParams?param1=John¶m2=David¶m3=Salar

Google g Search Bookmarks Find Check AutoFill Sign In

Reading Three Request Parameters

- param1: John
- param2: David
- param3: Salar

Exercise 2: Design form Personal Information (PersonalInfor.jsp)

JSP Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address: http://localhost:8084/TestServlet/PersonalInfor.jsp

Google g Search Bookmarks Find Check AutoFill Sign In

Personal Information

ID: IT090078

Name: Nguyen Thi Kieu Linh

Email: ntkl@yahoo.com.vn

Gender: Male Female

Major: Business Administration

Interesting Field: Reading books, Swimming, etc.

The result from GetInfor.java

Reading Personal Information - Microsoft Internet Explorer	
File Edit View Favorites Tools Help	
Back Forward Stop Home Search Favorites Help	
Address	http://localhost:8084/TestServlet/GetInfor
Google	g
Search	
Bookmarks	
Find	
Check	
AutoFill	
Sign In	
Reading Personal Information	
ID	IT090078
Name	Nguyen Thi Kieu Linh
Email	ntkl@yahoo.com.vn
Gender	Female
Major	Business Administration
Interesting Field	Reading books, Swimming, etc.

Exercise 3: Design a register form

JSP Page - Microsoft Internet Explorer	
File Edit View Favorites Tools Help	
Back Forward Stop Home Search Favorites Help	
Address	http://localhost:8084/Lab1WP/RegisterForm.jsp
Google	g
School of Computer Science & Engineering	
<hr/>	
Register Form	
Full Name	<input type="text"/>
ID	<input type="text"/>
Email	<input type="text"/>
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female
Field of study	<input type="text" value="CS"/>
	<input type="checkbox"/> Principle of EE 1
	<input type="checkbox"/> Computer Network
List of subjects	<input type="checkbox"/> Web Application Development
	<input type="checkbox"/> Object Oriented Programming
	<input type="checkbox"/> Computer Graphics
Comments	<input type="text"/>
<input type="button" value="Submit"/>	<input type="button" value="Reset"/>

The result show parameter Servlet (RegisterCourse.java)

Exercise 4: Refer example 5.1 ShowRequestHeaders.java from text-book

Servlet Example: Showing Request Headers - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address: http://localhost:8084/TestServlet/RequestHeader?Submit=Submit Go Links

Google Search Bookmarks Find Check AutoFill Sign In

Servlet Example: Showing Request Headers

Request Method: GET
Request URI: /TestServlet/RequestHeader
Request Protocol: HTTP/1.1

Header Name	Header Value
accept	image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, application/x-silverlight, application/x-silverlight-2-b2, */*
referer	http://localhost:8084/TestServlet/RequestHeader.jsp
accept-language	en-us
accept-encoding	gzip, deflate
user-agent	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; GTB5; .NET CLR 1.1.4322; .NET CLR 2.0.50727)
host	localhost:8084
connection	Keep-Alive
cookie	JSESSIONID=16F575DDF1B318B9A0E7D52291160B00