

Course: Web Application Development

Lab Instructor: Vo Duy Khoi, MSc

Email: khoivd@hcmiu.edu.vn

Lab 3 - Introduction to Servlet Programming

Content:

- Introduction to Java Servlet Technology
- How to create a servlet 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).

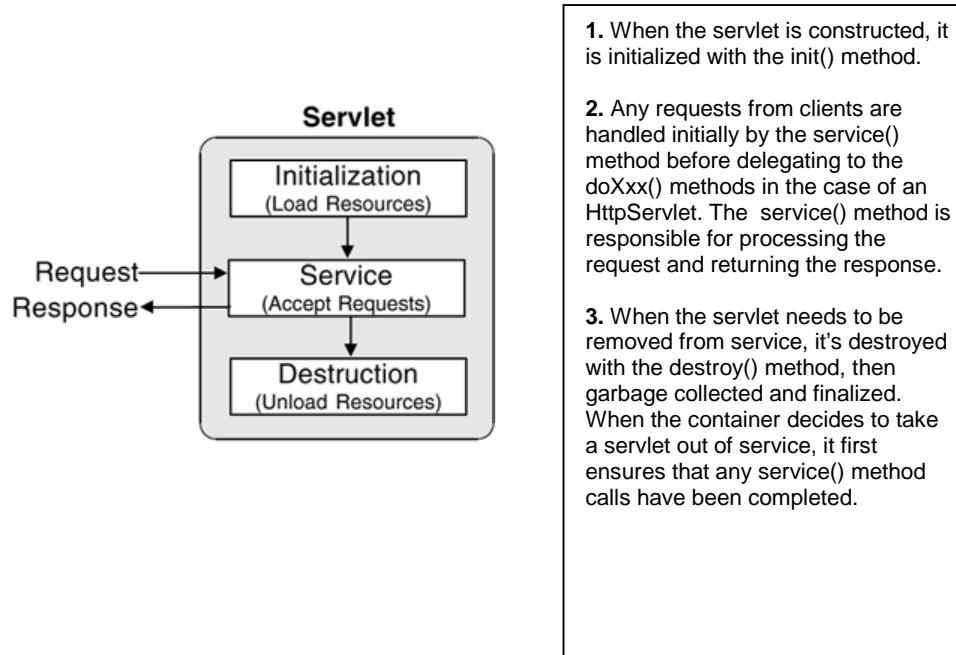
- Tomcat is not a J2EE application server. However, as J2EE app servers must themselves contain a servlet container to support the servlet/JSP APIs, J2EE app servers can embed Tomcat into their code to provide support for the Servlet and JSP APIs. One example of just such an application server is the 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:



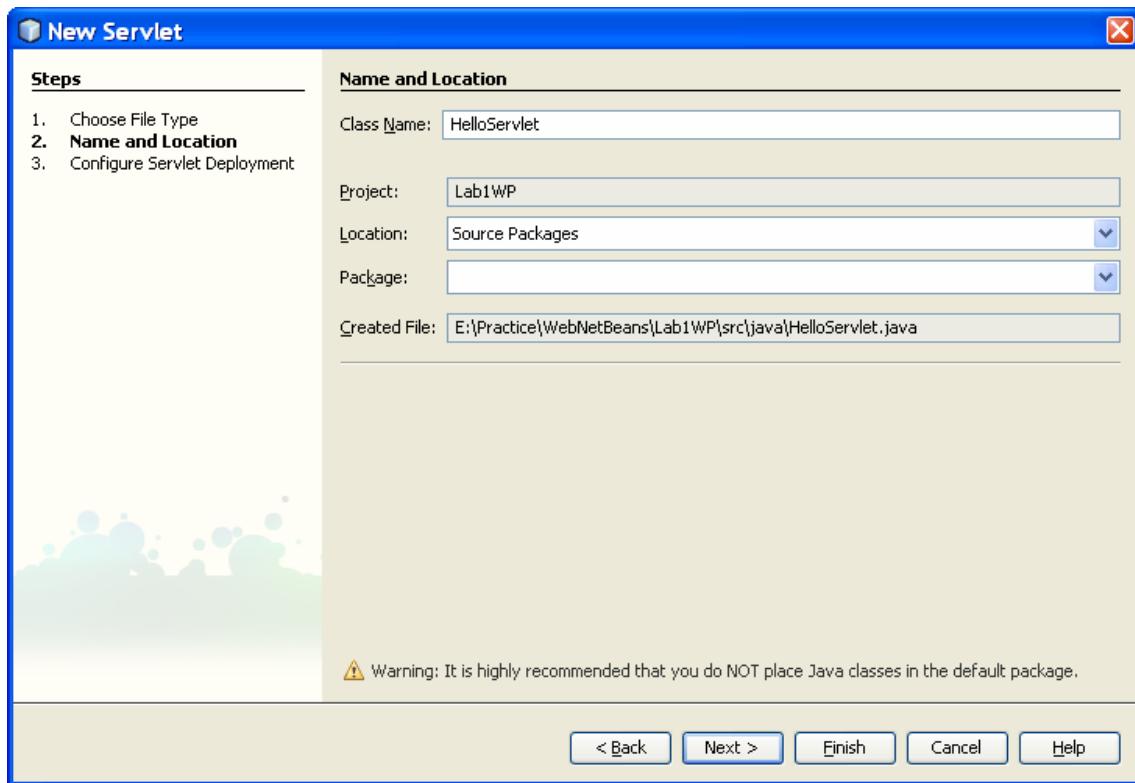
- The **init** method is called by the servlet container after the servlet class has been instantiated. The servlet container calls this method exactly once to indicate to the servlet that the servlet is being placed into service. The **init** method is important also because the servlet container passes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

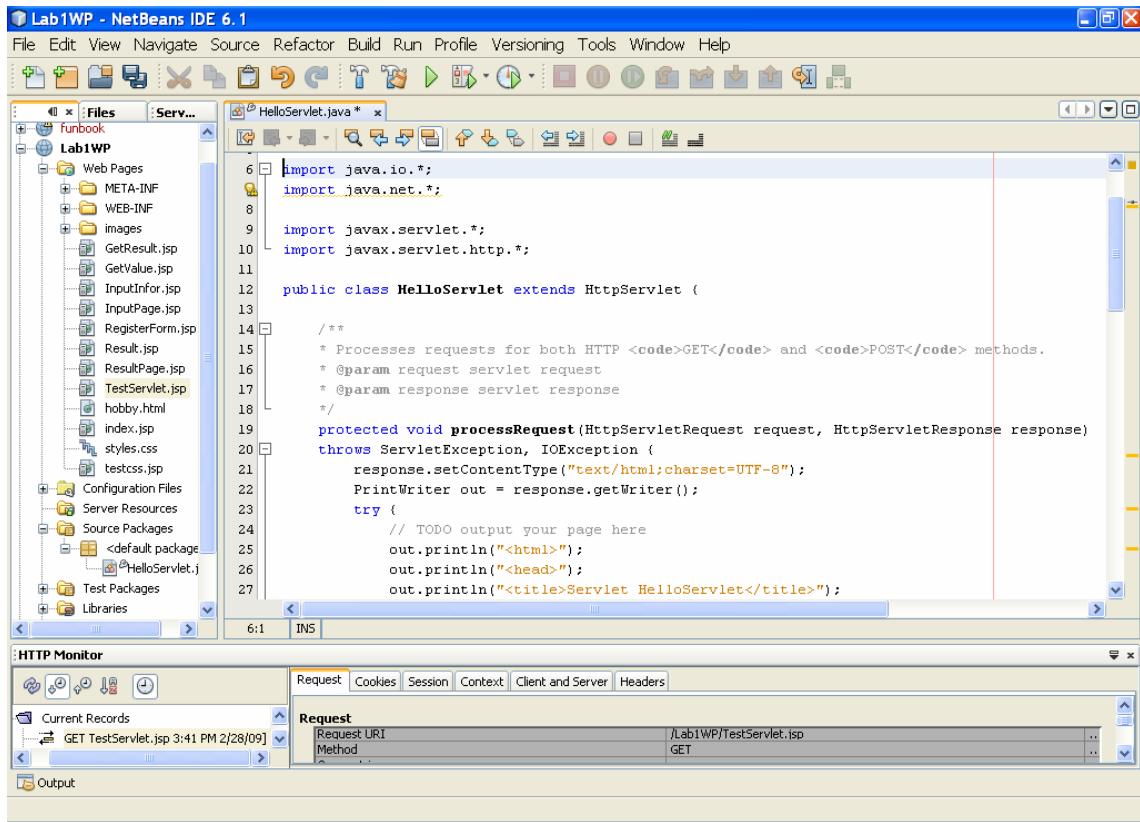
Part 2: How to create a Servlet on NetBeans IDE.

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

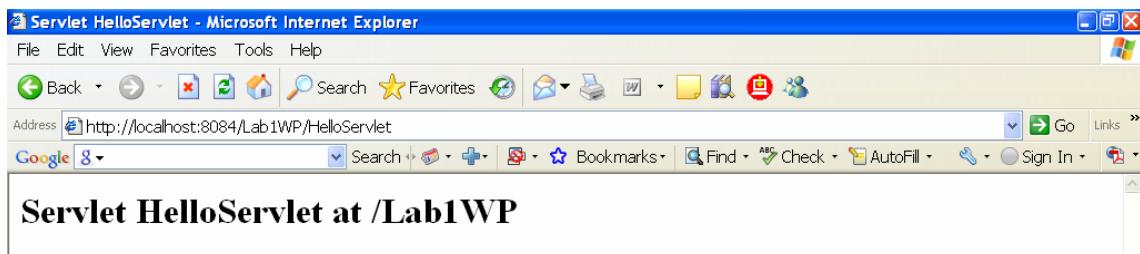


Type a name and click next, then click Finish button, the source codes generate automatically in the file HelloServlet.java.

This is source code:

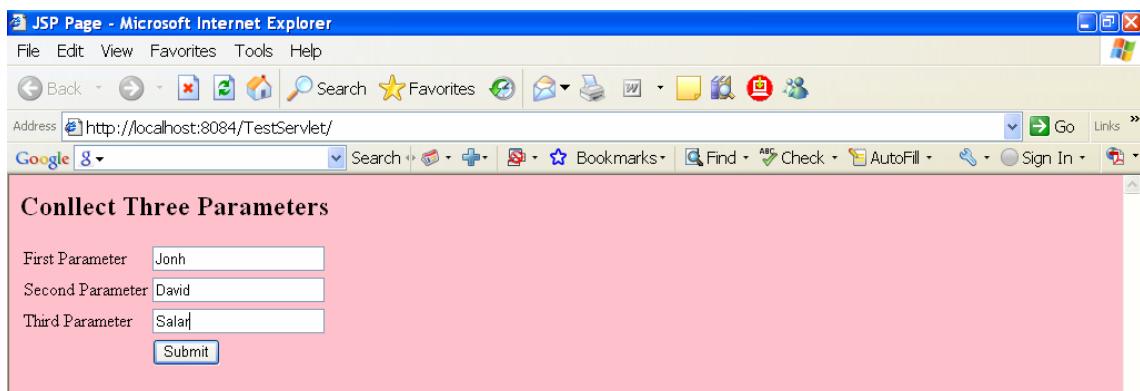


After building and running on the web server, this is the result page.

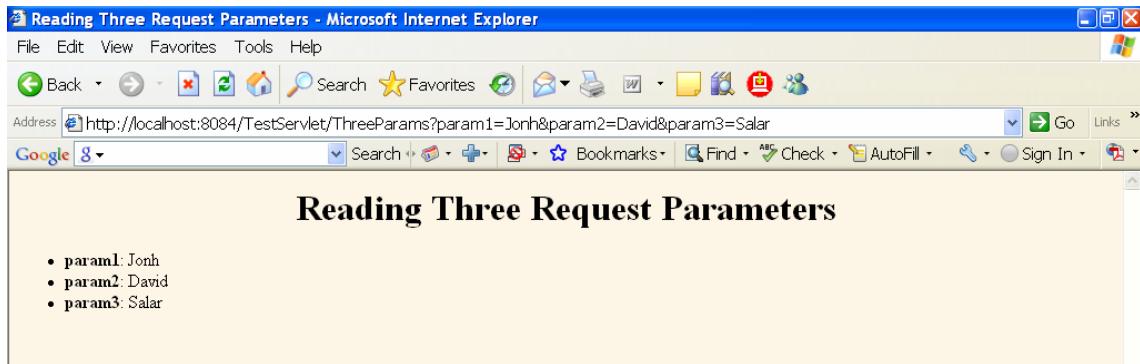


Part 3: Practices and Exercises.

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



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



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

Personal Information

ID	IT090078
Name	Nguyen Thi Kieu Linh
Email	ntkl@yahoo.com.vn
Gender	Male <input checked="" type="radio"/> Female <input type="radio"/>
Major	Business Administration
Interesting Field	Reading books, Swimming, etc.

Insert Cancel

The result from GetInfor.java

Reading Personal Information

ID	IT090078
Name	Nguyen Thi Kieu Linh
Email	ntkl@yahoo.com.vn
Gerder	Female
Major	Business Administration
Interesting Field	Reading books, Swimming, etc.

Exercise 3: Design a register form

The screenshot shows a Microsoft Internet Explorer window with the title "JSP Page - Microsoft Internet Explorer". The address bar contains "http://localhost:8084/Lab1WP/RegisterForm.jsp". The page itself has a pink header with the text "School of Computer Science & Engineering". Below this is a section titled "Register Form". The form includes fields for "Full Name" (text input), "ID" (text input), "Email" (text input), "Gender" (radio buttons for Male and Female, with Male selected), "Field of study" (dropdown menu showing "CS"), and a "List of subjects" section with checkboxes for "Principle of EE 1", "Computer Network", "Web Application Development", "Object Oriented Programming", and "Computer Graphics". There is also a "Comments" text area and two buttons at the bottom: "Submit" and "Reset".

The result show parameter Servlet (RegisterCourse.java)

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

The screenshot shows a Microsoft Internet Explorer window with the title "Servlet Example: Showing Request Headers - Microsoft Internet Explorer". The address bar contains "http://localhost:8084/TestServlet/RequestHeader?Submit=Submit". The page displays the heading "Servlet Example: Showing Request Headers". Below this, it shows the request details: "Request Method: GET", "Request URI: /TestServlet/RequestHeader", and "Request Protocol: HTTP/1.1". A table is presented with the following data:

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

Exercise 5: Design a register form with raw object (CSS will be applied next session)

The screenshot shows a Mozilla Firefox browser window with the title bar "Mozilla Firefox". The address bar displays the URL http://localhost:8080/TestServlet/application_Register.jsp. The main content area contains a form titled "Application for Dormitory". The form is organized into several sections:

- Application ID:** A text input field.
- Referee name:** A text input field.
- App Name:** A text input field.
- Student Info:** A section containing fields for "ID :" and "Name :" with a "Choose owner" button.
- Classification:** A section with a dropdown menu labeled "Select Value" and a date input field labeled "Date Of Filling".
- Notes:** A text area containing the placeholder text "{abstract}".
- Details:** A section with a file input field, a "Browse..." button, and a "Delete" checkbox.
- List of Roommates:** A scrollable list area with "More" and "Less" buttons.

At the bottom of the form are three buttons: "Register", "Cancel", and "Back".