Module 2: Servlet Basics

Thanisa Kruawaisayawan
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang

Objectives

- What is Servlet?
- Request and Response Model
- Method GET and POST
- Servlet API Specifications
- The Servlet Life Cycle
- Examples of Servlet Programs

What is a Servlet?

- Java™ objects which extend the functionality of a HTTP server
- Dynamic contents generation
- Better alternative to CGI
  - Efficient
  - Platform and server independent
  - Session management
  - Java-based

Servlet vs. CGI

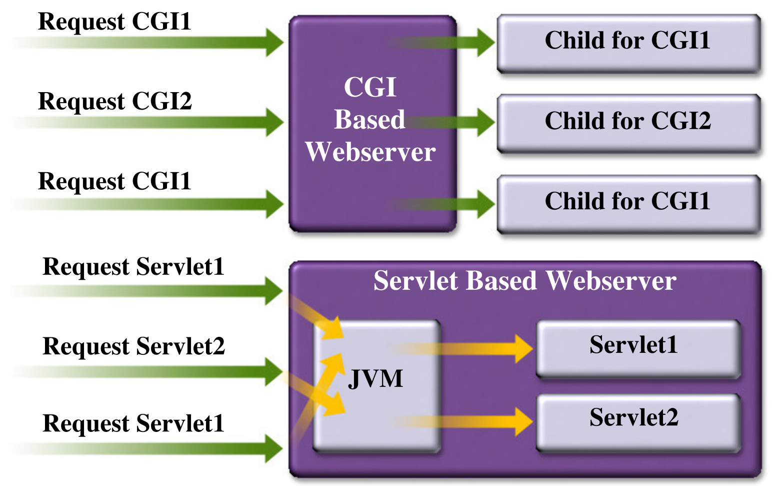
Servlet

- Requests are handled by threads.
- Only a single instance will answer all requests for the same servlet concurrently (persistent data)

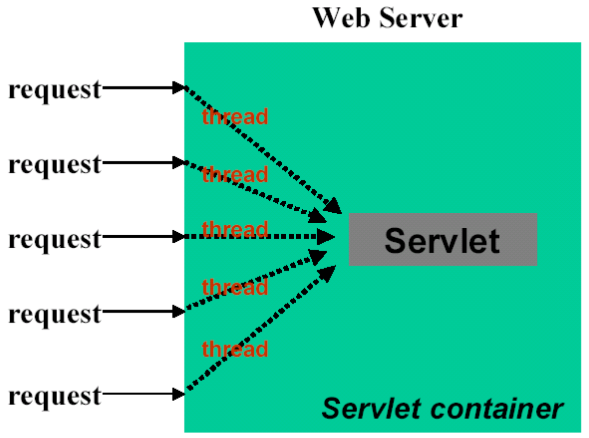
CGI

- New process is created for each request (overhead & low scalability)
- No built-in support for sessions

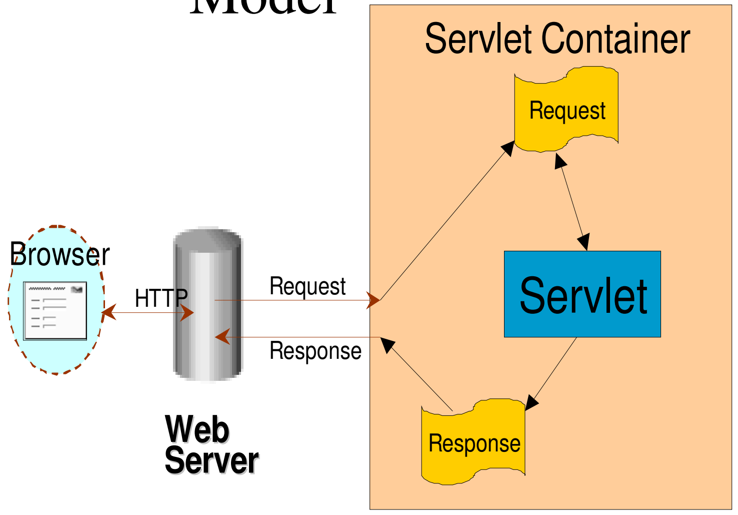
Servlet vs. CGI (cont.)



Single Instance of Servlet



Servlet Request and Response Model



What does Servlet Do?

- Receives client request (mostly in the form of HTTP request)
- Extract some information from the request
- Do content generation or business logic process (possibly by accessing database, invoking EJBs, etc)
- Create and send response to client (mostly in the form of HTTP response) or forward the request to another servlet or JSP page

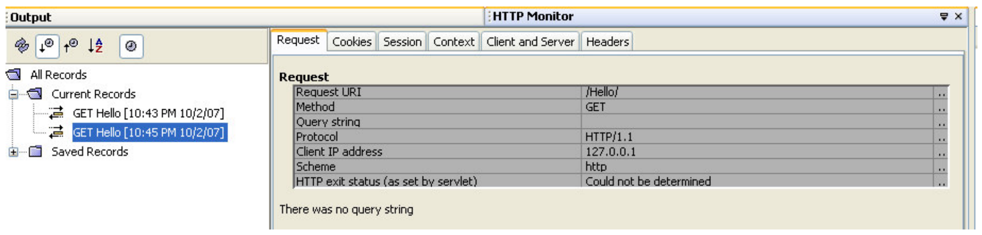
Requests and Responses

- What is a request?
  - Information that is sent from client to a server
    - Who made the request
    - Which HTTP headers are sent
    - What user-entered data is sent
- What is a response?
  - Information that is sent to client from a server
    - Text(html, plain) or binary(image) data
    - HTTP headers, cookies, etc

HTTP

- HTTP request contains
  - Header
  - Method
    - Get: Input form data is passed as part of URL
    - Post: Input form data is passed within message body
    - Put
    - Header
  - request data

HTTP Monitor in Netbeans



Request

Request URI	/Hello/
Method	GET
Query string	
Protocol	HTTP/1.1
Client IP address	127.0.0.1
Scheme	http
HTTP exit status (as set by servlet)	Could not be determined

There was no query string

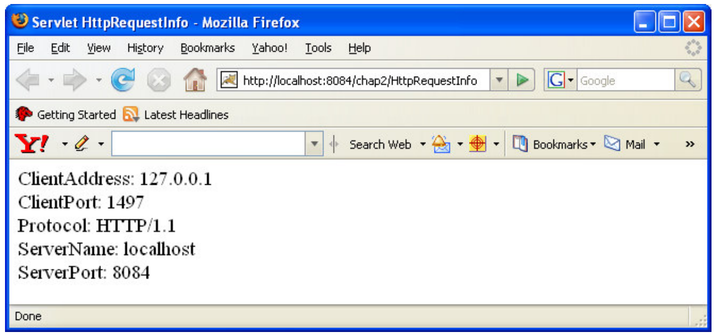
Request Methods

- `getRemoteAddr()`
  - IP address of the client machine sending this request
- `getRemotePort()`
  - Returns the port number used to sent this request
- `getProtocol()`
  - Returns the protocol and version for the request as a string of the form <protocol>/<major version>.<minor version>
- `getServerName()`
  - Name of the host server that received this request
- `getServerPort()`
  - Returns the port number used to receive this request

HttpRequestInfo.java

```
public class ServletInfo extends HttpServlet {
    :
    protected void processRequest(HttpServletRequest request,
         HttpServletResponse response)throws ServletException, IOException {
         response.setContentType("text/html");
         PrintWriter out = response.getWriter();

         out.println("ClientAddress: " + request.getRemoteAddr() + "<BR>");
         out.println("ClientPort: " + request.getRemotePort() + "<BR>");
         out.println("Protocol: " + request.getProtocol() + "<BR>");
         out.println("ServerName: " + request.getServerName() + "<BR>");
         out.println("ServerPort: " + request.getServerPort() + "<BR>");

         out.close();
    }
    :
}
```

Result


Servlet HttpRequestInfo - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://localhost:8084/chap2/HttpRequestInfo
Google
Getting Started Latest Headlines
Search Web
Bookmarks
Mail
Done

```
ClientAddress: 127.0.0.1
ClientPort: 1497
Protocol: HTTP/1.1
ServerName: localhost
ServerPort: 8084
```

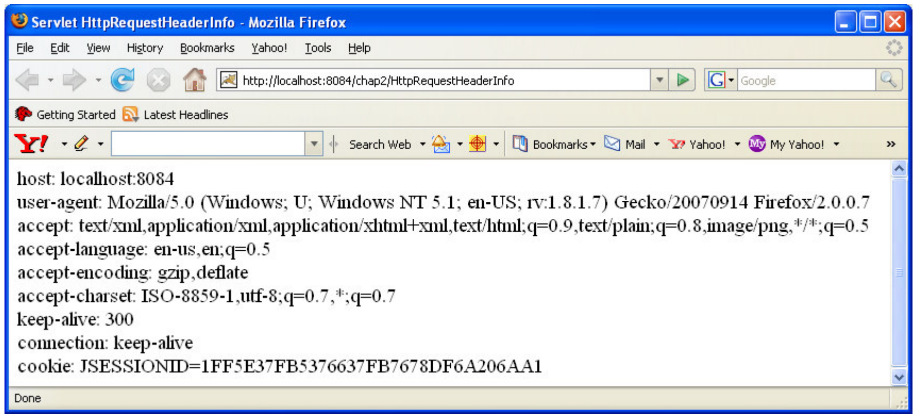
Reading Request Header

- General
  - getHeader
  - getHeaders
  - getHeaderNames
- Specialized
  - getCookies
  - getAuthType and getRemoteUser
  - getContentLength
  - getContentType
  - getDateHeader
  - getIntHeader

HttpRequestHeaderInfo.java

```
import java.util.*;

public class HttpServletInfo extends HttpServlet {
    :
    protected void processRequest(HttpServletRequest request,
       HttpServletResponse response) throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        Enumeration enum1 = request.getHeaderNames();
        while (enum1.hasMoreElements()) {
           String name = (String) enum1.nextElement();
           out.println(name + ": " + request.getHeader(name) + "<BR>");
        }
        out.close();
    }
    :
}
```

Result


Servlet HttpRequestHeaderInfo - Mozilla Firefox
http://localhost:8084/chap2/HttpRequestHeaderInfo

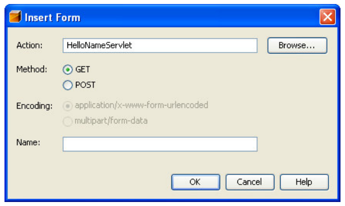
```
host: localhost:8084
user-agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.7) Gecko/20070914 Firefox/2.0.0.7
accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
accept-language: en-us,en;q=0.5
accept-encoding: gzip,deflate
accept-charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
keep-alive: 300
connection: keep-alive
cookie: JSESSIONID=1FF5E37FB5376637FB7678DF6A206AA1
```

Frequently Used Request Methods

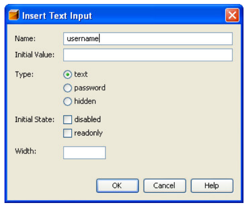
- HttpServletRequest methods
  - `getParameter()` returns value of named parameter
  - `getParameterValues()` if more than one value
  - `getParameterNames()` for names of parameters

Example: hello.html

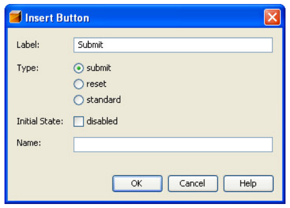
```
<HTML>
  :
  <BODY>
      <form action="HelloNameServlet">
            Name: <input type="text" name="username" />
            <input type="submit" value="submit" />
      </form>
  </BODY>
</HTML>
```



Action: HelloNameServlet
Method: [✓ GET] POST
Encoding: [✓ application/x-www-form-urlencoded] multipart/form-data
Name:



Name: username
Initial Value:
Type: [✓ text] password hidden
Initial State: disabled readonly
Width:

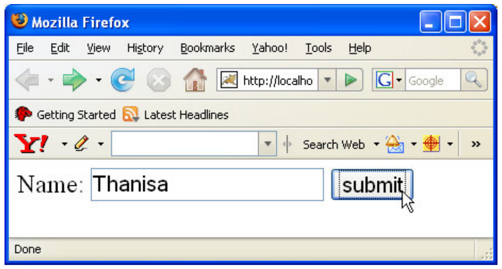


Label: Submit
Type: [✓ submit] reset standard
Initial State: disabled
Name:

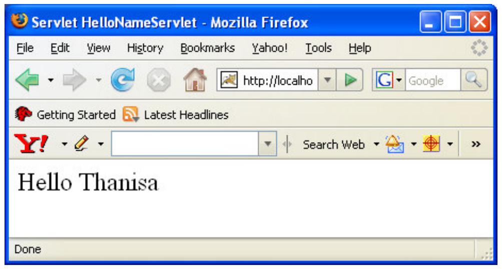
HelloNameServlet.java

```
public class HelloNameServlet extends HttpServlet {
    :
    protected void processRequest(HttpServletRequest request,
                        HttpServletResponse response)
                        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        out.println("Hello " + request.getParameter("username"));

        out.close();
    }
    :
}
```

Result


Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://localho
Getting Started Latest Headlines
Search Web
Done

Name: Thanisa submit


Servlet HelloNameServlet - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://localho
Getting Started Latest Headlines
Search Web
Done

Hello Thanisa

HTTP GET and POST

- The most common client requests
  - HTTP GET & HTTP POST
- GET requests:
  - User entered information is appended to the URL in a query string
  - Can only send limited amount of data
    - .../chap2/HelloNameServlet?username=Thanisa
- POST requests:
  - User entered information is sent as data (not appended to URL)
  - Can send any amount of data

TestServlet.java

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class TestServlet extends HttpServlet {
  public void doGet(HttpServletRequest request,
   HttpServletResponse response)
            throws ServletException, IOException {

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    out.println("<h2>Get Method</h2>");
  }
}
```

Steps of Populating HTTP Response

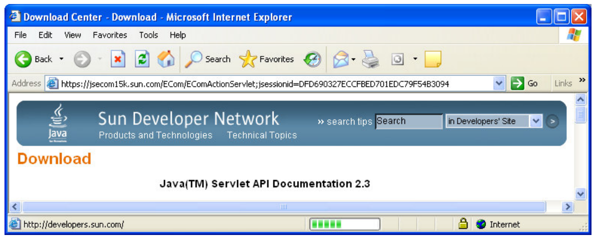
- Fill Response headers
- Get an output stream object from the response
- Write body content to the output stream

Example: Simple Response

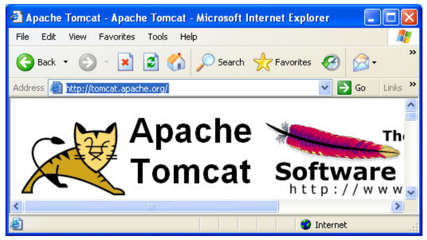
```
Public class HelloServlet extends HttpServlet {
 public void doGet(HttpServletRequest request,
  HttpServletResponse response)
                     throws ServletException, IOException {

   // Fill response headers
   response.setContentType("text/html");

   // Get an output stream object from the response
   PrintWriter out = response.getWriter();

   // Write body content to output stream
    out.println("<h2>Get Method</h2>");
  }
}
```

Servlet API Specifications

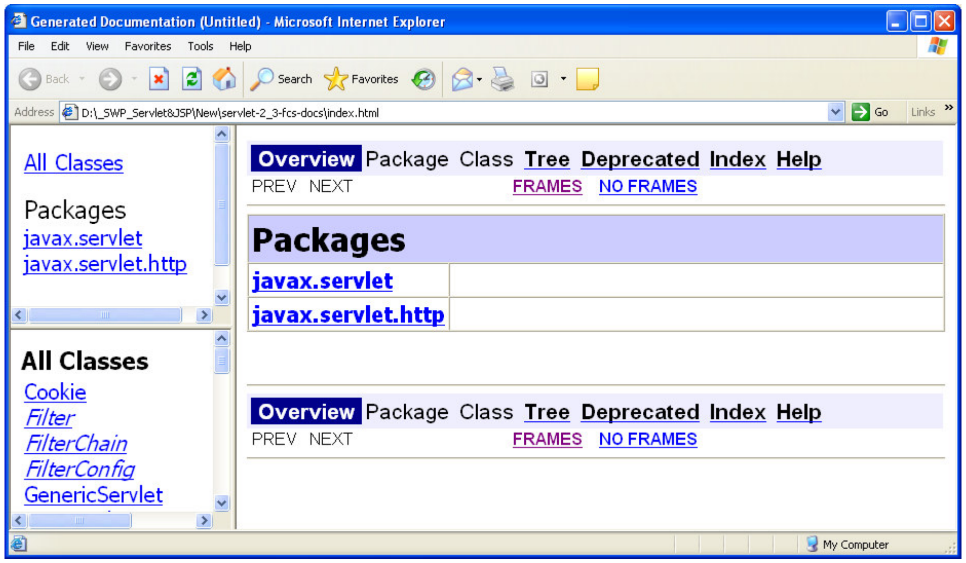
- Official Web Site: http://java.sun.com


Download Center - Download - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Sun Developer Network
Products and Technologies Technical Topics
Download
Java(TM) Servlet API Documentation 2.3
http://developers.sun.com/
Internet

- Free Servlet engines
  - Tomcat
    - http://tomcat.apache.org/


Apache Tomcat - Apache Tomcat - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://tomcat.apache.org/
Apache Tomcat
The Apache Software
http://www
Internet

Servlet API Specifications (cont.)


Generated Documentation (Untitled) - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address D:_SWP_Servlet&JSP\New\servlet-2_3-fcs-docs\index.html

All Classes

Packages
javax.servlet
javax.servlet.http

All Classes
Cookie
Filter
FilterChain
FilterConfig
GenericServlet

Overview Package Class **Tree** **Deprecated** **Index** **Help**
PREV NEXT FRAMES NO FRAMES

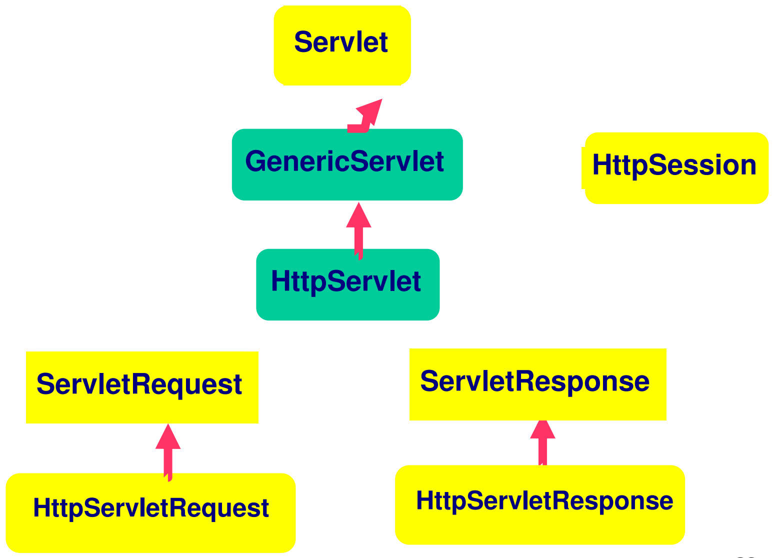
Packages

javax.servlet
javax.servlet.http

Overview Package Class **Tree** **Deprecated** **Index** **Help**
PREV NEXT FRAMES NO FRAMES

My Computer

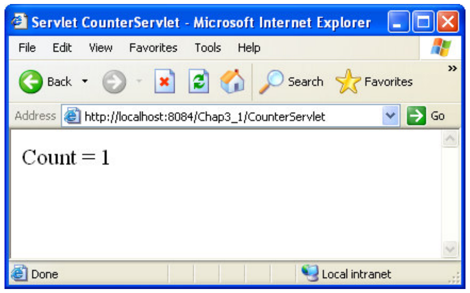
Servlet Interfaces & Classes



CounterServlet.java

```
:
public class CounterServlet extends HttpServlet {
    private int count;
          :
    protected void processRequest(HttpServletRequest request,
                 HttpServletResponse response) throws
                          ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        count++;
        out.println("Count = " + count);
        out.close();
    }
          :
}
```

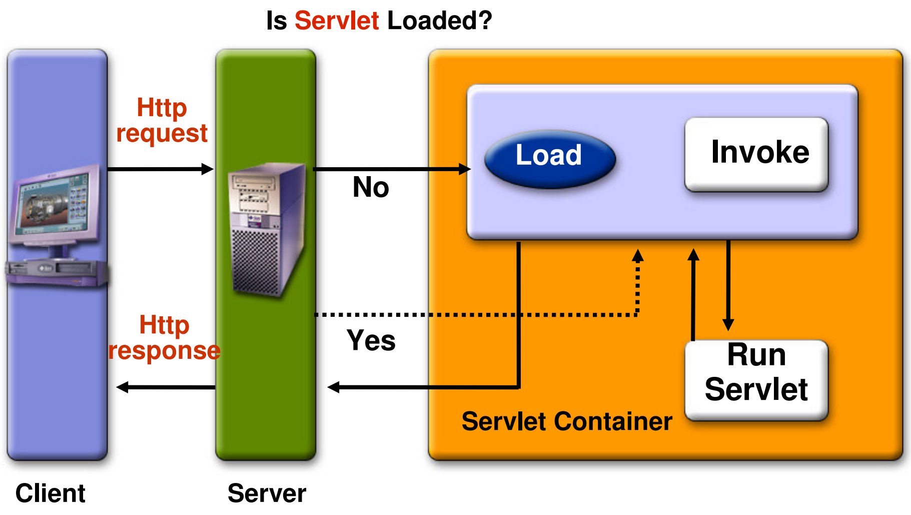
Result


Servlet CounterServlet - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Done
Local intranet

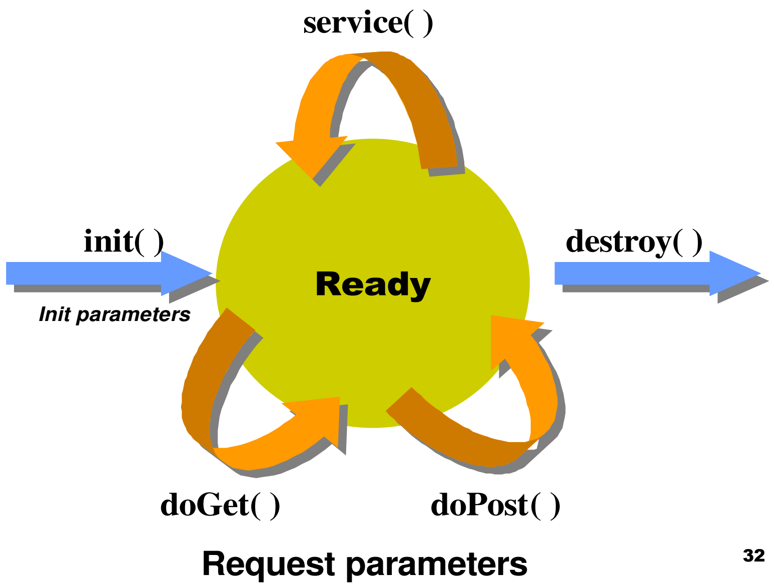
http://localhost:8084/Chap3_1/CounterServlet

```
Count = 1
```

Servlet Life-Cycle



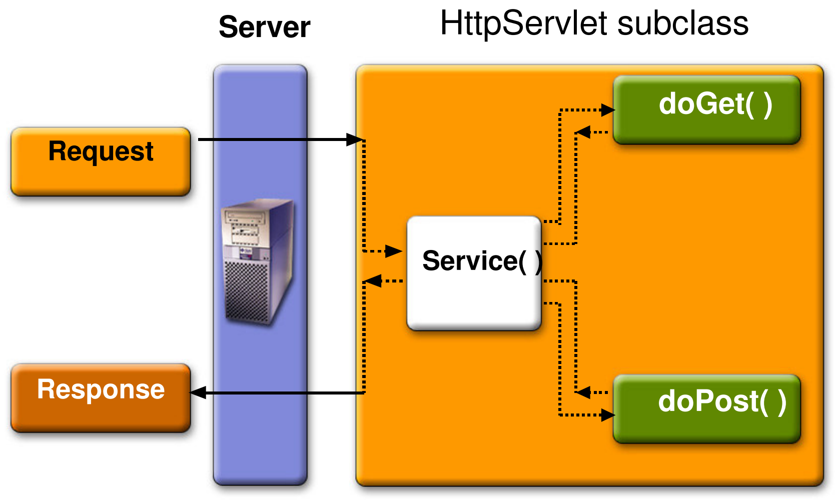
Servlet Life Cycle Methods



The Servlet Life Cycle

- init
  - executed once when the servlet is first loaded
  - Not call for each request
  - Perform any set-up in this method
    - **Setting up a database connection**
- destroy
  - called when server delete servlet instance
  - Not call after each request
  - Perform any clean-up
    - **Closing a previously created database connection**

doGet() and doPost() Methods



Key: 🟩 Implemented by **subclass**

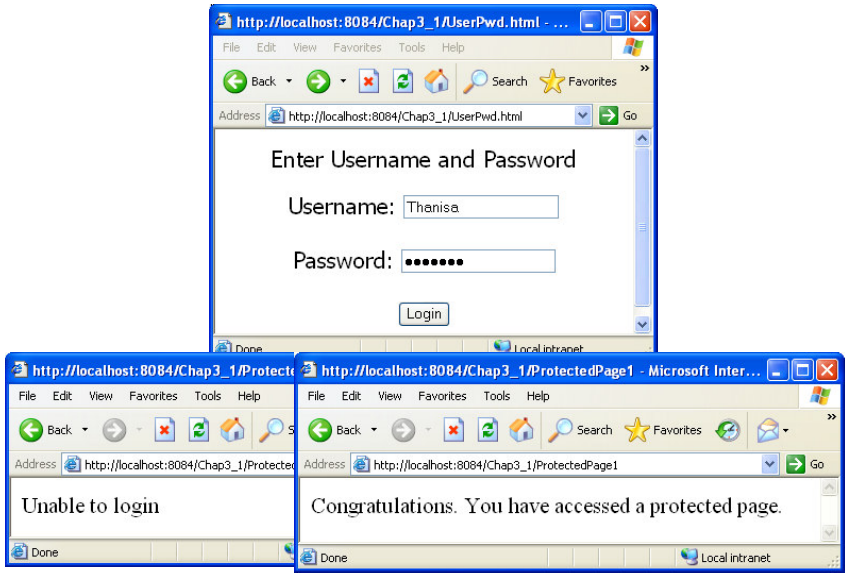
Servlet Life Cycle Methods

- Invoked by container
  - Container controls life cycle of a servlet
- Defined in
  - javax.servlet.GenericServlet class or
    - init()
    - destroy()
    - service() - this is an abstract method
  - javax.servlet.http.HttpServlet class
    - doGet(), doPost(), doXxx()
    - service() - implementation

Implementation in method service()

```
protected void service(HttpServletRequest req, HttpServletResponse
   resp)
        throws ServletException, IOException {
        String method = req.getMethod();
     if (method.equals(METHOD_GET)) {
          …
          doGet(req, resp);
          ...
        } else if (method.equals(METHOD_HEAD)) {
          ...
          doHead(req, resp); // will be forwarded to doGet(req, resp)
        } else if (method.equals(METHOD_POST)) {
          doPost(req, resp);
        } else if (method.equals(METHOD_PUT)) {
          doPut(req, resp);
        } else if (method.equals(METHOD_DELETE)) {
          doDelete(req, resp);
        } else if (method.equals(METHOD_OPTIONS)) {
          doOptions(req,resp);
        } else if (method.equals(METHOD_TRACE)) {
          doTrace(req,resp);
        } else {
          …
        }
     }
 }
```

Username and Password Example


http://localhost:8084/Chap3_1/UserPwd.html

Enter Username and Password

Username: Thanisa

Password: ••••••••

Login

http://localhost:8084/Chap3_1/ProtectedPage1

Unable to login

Congratulations. You have accessed a protected page.

Acknowledgement

Some contents are borrowed from the presentation slides of Sang Shin, Java™ Technology Evangelist, Sun Microsystems, Inc.