Session 9

Introduction to Servlets

Lecture Objectives

- Understand the foundations for client/server Web interactions
- Understand the servlet life cycle

Reading & Reference

- Reading
  - Java Servlet
    en.wikipedia.org/wiki/Java_servlet
  - Java Annotation
    en.wikipedia.org/wiki/Java_annotation
    docs.oracle.com/javase/tutorial/java/annotations/
  - Excellent tutorial, explaining how to set up a servlet in NetBeans
    www.studytonight.com/servlet/creating-servlet-in-netbeans.php
- Reference
  - Use the on-line Servlet API documentation at:
    http://docs.oracle.com/javaee/7/api/

Use the J2EE 7 Tutorial as a Reference

3

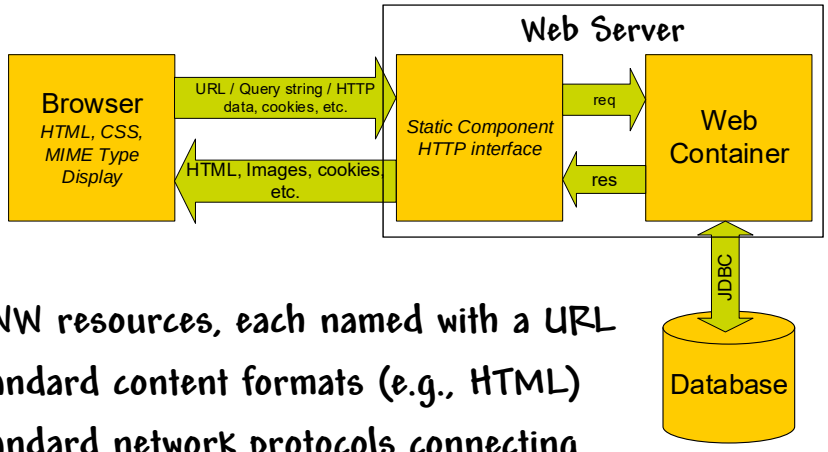
Java Servlet Releases

- Servlet 3.0 - 2009
- Servlet 3.1 - May 2013
  (consistent with Java EE 7)
- Servlet 4.0 - September 2017
  (consistent with Java EE 8 and HTTP/2)

Java EE now maintained by the Eclipse foundation, and renamed as Jakarta EE

4

Typical Client/Server Interaction



Web container is also referred to as a servlet container

- WWW resources, each named with a URL
- Standard content formats (e.g., HTML)
- Standard network protocols connecting any browser with any server

5

Server Strategies

- Generation of HTML/CSS
  - Server responds with a dynamically generated page that includes HTML, CSS, and data (inserted in the page)
  - Data insertion usually performed by a server-side scripting engine
- Web services
  - Server responds with data (no HTML and CSS)
  - Data structured based on some coordination between client and server (e.g., JSON, XML, text)

6

Servlets

- Conforms to the Java Servlet API
- Superseded by JAX-RS (Java API for RESTful Web Services) API
- A servlet:
  - Is a Java class that can be loaded dynamically to expand the capability of the Web server
  - Runs inside the Java Virtual Machine on the server (safe and portable)
  - Is able to access all Java APIs supported in the server
  - Does not have a main method

7

Servlet Implementation

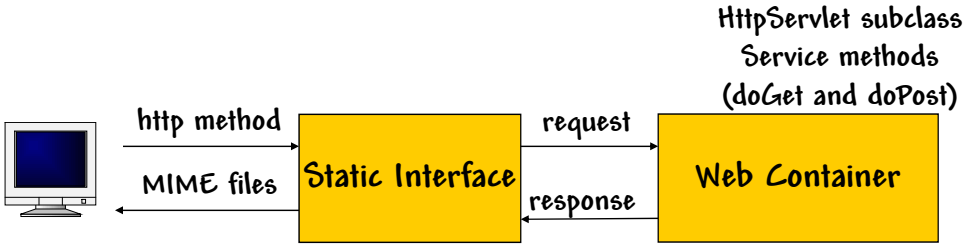
- Web Server (Web Container) Market Share[1] of active sites

Developer	March 2018	Percent
Apache	76M	43.0%
nginx	37M	21.0%
Google	12M	7.7%
Microsoft	12M	6.8%

1. Data from NetCraft

8

Servlet / Web Server Interface



The Web Server static interface primarily serves resources, and passes complex request to the Web Container

Frequently both servers are packaged together, but it is good to think of them as separate systems

9

Servlet Interface

- Objects are used to pass information to the server and to return information from the server
- Service methods:

Request data include HTTP version, URL, browser software, client MIME type preferences, data, etc.

```
protected void doGet(HttpServletRequest req,
HttpServletResponse resp)
      throws ServletException, java.io.IOException

protected void doPost( HttpServletRequest req,
HttpServletResponse resp)
      throws ServletException, java.io.IOException
```

Response data includes HTTP version, status code, MIME type of data, document size, document

10

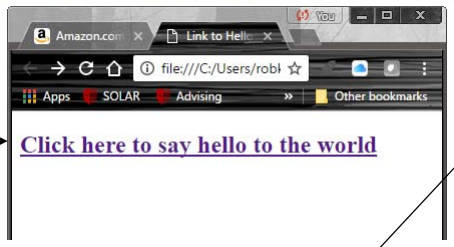
Server Generation of an HTML page

- The following HTML can be returned to the browser directly by the Web server (static file on the Web Server) - or the same html page can be generated (on the fly) by the Web Container

```
<html>
<head>
    <title>Hello World</title>
</head>
<body>
    <p>Hello World</p>
</body></html>
```


11

Invoking the Hello World Servlet



The URL in this link maps to the servlet

Port 8080 is typical for a test http server

Name of the Web application

```
<!doctype html>
<html>
<head> <title>Link to Hello World Servlet</title> </head>
<body>
<p>
<a href="http://localhost:8080/CSE336-2017/HelloWorld">
Click here to say hello to the world</a> </p>
</body> </html>
```

Verify the port number used by your test server

This is not really a file - it maps to your servlet

12

Hello World Servlet Method

```
protected void processRequest(
  HttpServletRequest request,
  HttpServletResponse response)
  throws ServletException, IOException {

  response.setContentType(
   "text/html;charset=UTF-8");
  PrintWriter out = response.getWriter();
  String docTyp= "<!DOCTYPE html >";
  out.println(docType);
  out.println("<html>");
  out.println("<head><title>"
    + "Hello World</title></head>");
  out.println("<meta charset=\"UTF-8\">");
  out.println("<body>");
  out.println("<h1>Hello World</h1>");
  out.println("</body></html>");
  out.close();  }
```

13

HelloWorld Servlet Class

```
package lectures;
import java.io.*;import java.net.*;import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
protected void processRequest(HttpServletRequest request, HttpServletResponse
   response)
  throws ServletException, IOException {
}
protected void doGet(HttpServletRequest request, HttpServletResponse response)
  throws ServletException, IOException {
    processRequest(request, response);
  }
protected void doPost(HttpServletRequest request, HttpServletResponse response)
  throws ServletException, IOException {
    processRequest(request, response);
  }
```

processRequest is a method used by convention in NetBeans

The Web container calls either doGet or doPost, which then calls processRequest

14

Web Application

- Your Web application contains all the servlets, JSPs, files, etc. associated with your application.
- Your Web application is stored in a directory (and deployed as a war file)
- Top level directory of the Web application is the document root of the application, containing JSP pages and static Web resources (or subdirectories of JSP, etc.)
- Document root contains a sub-directory called WEB-INF, containing
  - web.xml - the deployment descriptor
  - Sub-directories containing classes
  - Meta data for the Web Application

Take a look at your Web App in your NetBeans or Eclipse project pane

15

How to Specify the Servlet in Your HTML

- A URL is used to request that the container run your servlet (in an anchor tag or form tag)
- URL contains the host name, port (optional), and path
- In a servlet container, the path can be mapped (what you see is not always what you get)

`http://localhost:8080/CSE336-2017/HelloWorld`

There is no HelloWorld resource

16

How URLs Run Servlets

`http://localhost:8080/CodeCSE336/helloWorld`

Context name (Web Application or Project name)

- The servlet container evaluates the URL request to see if the first part of the path matches a context name
- If the path matches a context name, the context name is mapped to a Web application root directory (using the web.xml deployment descriptor)

17

Mapping Requests to Servlets

- Two approaches
  - Web.xml (Deployment Descriptor)
  - Annotation

18

Annotation Recap ...

- Part of Java language, starting with Java 5
- Annotations are tags that you insert into source code so that some tool can process it
  (not part of the normal execution of the program)
- Proper style places the annotation on a line preceding the statement it relates to

```
@Entity
public class Team implements Serializable {
```

Think of it as a modifier for the declaration

You can annotate classes, methods, fields, and local variables

19

... Annotation recap

- Annotations can be defined to have elements

```
@ManyToOne(cascade=CascadeType.PERSIST)
   public Team getTeam() { … }
```

  - Examples
    - Unit testing (JUnit)
    - Java Persistence API (JPA)
    - Servlets
    - Java Beans

20

Annotation Example - JPA

- Java Persistence allow you to specify the DB design in your code

```
@Entity
public class Player implements Serializable {
...
@ManyToOne(cascade=CascadeType.PERSIST)
    public Team getTeam() {
        return team;
    }
}
```


Servlet Mapping with Annotations

- New with Java EE 5
- Keeps information about mapping a URL pattern to a servlet class in one place - servlet code
- Example

```
@WebServlet(name = "HelloWorld",
     urlPatterns = {"/HelloWorld"})
public class HelloWorld extends HttpServlet {
```

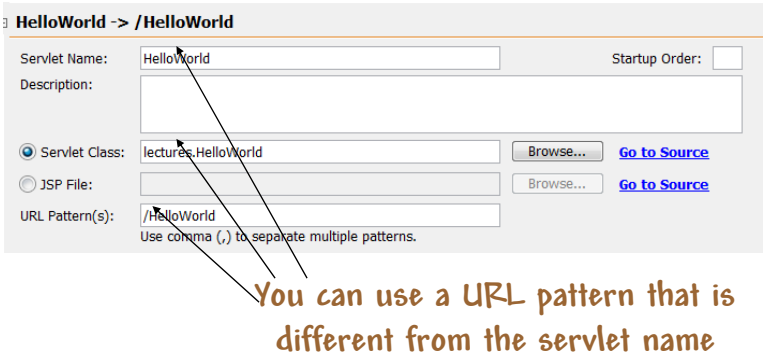
Servlet Annotations

Modifier and Type	Optional Element and Description
boolean	asyncSupported - Declares whether the servlet supports asynchronous operation mode.
String	description - The description of the servlet
String	displayName - The display name of the servlet
WebInitParam[]	initParams The init parameters of the servlet
String	largeIcon - The large-icon of the servlet
int	loadOnStartup - The load-on-startup order of the servlet
String	name - The name of the servlet
String	smallIcon - The small-icon of the servlet
String[]	urlPatterns - The URL patterns of the servlet
String[]	value - The URL patterns of the servlet

23

Deployment Descriptor (web.xml)

- Alternate (older) technique to deploy your Web application (i.e., servlets, JSPs, etc.)
- NetBeans display of Deployment Descriptor (below)



You can use a URL pattern that is different from the servlet name

24

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
 <servlet>
   <description>A Hello World servlet </description>
   <servlet-name>HelloWorld</servlet-name>
   <servlet-class>lectures.HelloWorld</servlet-class>
 </servlet>
...
 <servlet-mapping>
        <servlet-name>HelloWorld</servlet-name>
        <url-pattern>/HelloWorld</url-pattern>
 </servlet-mapping>
...
 <session-config>
        <session-timeout>
            30
        </session-timeout>
 </session-config>
</web-ap>
```

25

Are We on Track?

- Create a HelloWorld servlet that displays Hello World in your browser
- Invoke it from your IDE
- Invoke it from your browser

Be sure you have installed the Web components of your IDE

This is just to discover anomalies in your lab or personal implementation of your IDE

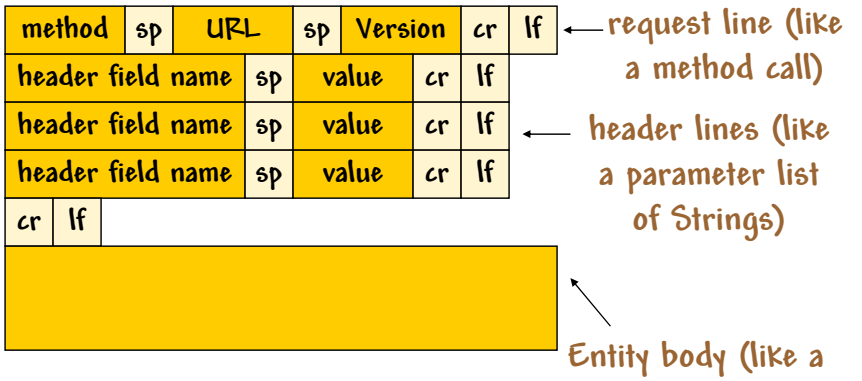
26

Were We on Track?

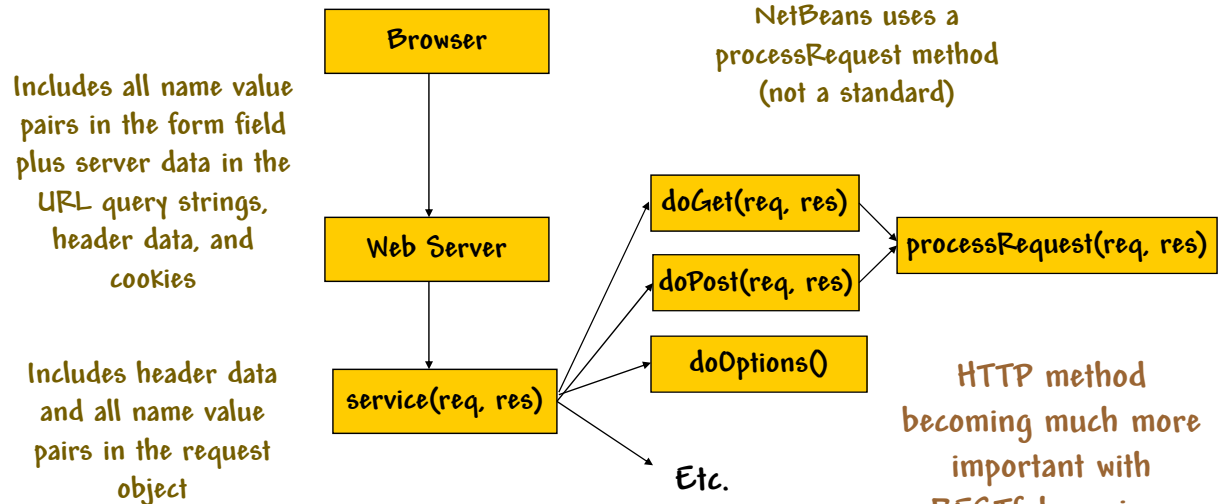
```
@WebServlet(name = "HelloWorld", urlPatterns = {"/HelloWorld"})
public class HelloWorld extends HttpServlet {
protected void processRequest(HttpServletRequest request,
  HttpServletResponse response)
  throws ServletException, IOException {
  response.setContentType("text/html;charset=UTF-8");
  try (PrintWriter out = response.getWriter()) {
    out.println("<!DOCTYPE html>");
    out.println("<html>"); out.println("<head>");
    out.println("<title>Servlet HelloWorld</title>");
    out.println("</head>"); out.println("<body>");
    out.println("<h1>Hello World</h1>");
    out.println("</body>");
    out.println("</html>"); } }
```


Request Message Format

- The http request is specified by the request line, a variable number of header fields, and the entity body



Calling Sequence



29

Some HttpServletRequest Methods

ServletRequest

- getParameter(String s)
- getScheme()
- getProtocol()
- getRemoteAddress()
- isSecure()
- getContentType()
- getRequestedSessionID()

HttpServletRequest

- getRequestURI
- isRequestedSessionValid()
- getQueryString()
- getRemoteUser
- getMethod()
- getCookies()
- getHeader()

30

HttpServletResponse

- Some methods:
  - getWriter - from ServletResponse
  - sendError(int sc)
  - addCookie(Cookie cookie)
  - sendRedirect(String location)
- Some fields:
  - SC_GONE
  - SC_INTERNAL_SERVER_ERROR
  - SC_NOT_FOUND

31

Why doGet and doPost?

- HTTP - a simple stateless protocol (Web browser makes a request and the server responds)
- The request from the browser specifies an HTTP method, along with client data
- HTTP methods - GET, POST, HEAD, PUT, DELETE, etc.
- Method called (doGet or doPost) corresponds to the HTTP method requested by the browser

32

doGet / doPost Practice

- A servlet usually does not distinguish between a GET and a POST method call
- One of the methods usually invokes the other

```
public void doPost(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
  this.doGet(request, response);
}
```

Web services takes a stricter view of the http method used

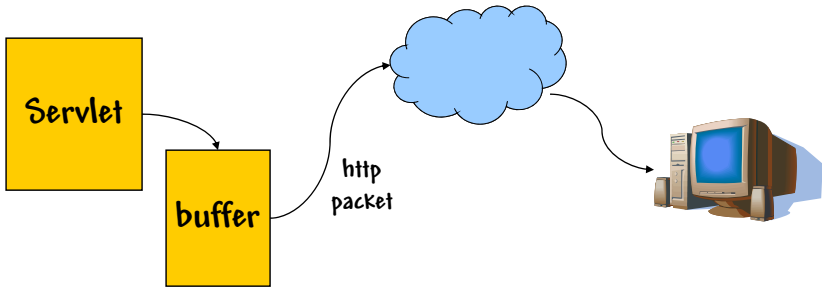
Or NetBeans generates a processRequest method

```
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
  throws ServletException, IOException {
    processRequest(request, response);
  }
```


33

Servlet Generation of HTML

- A servlet will generate 2 kinds of output
  - Information about the transmission to the browser - this is stored in the http header of the response
  - Data (e.g., HTML) that is stored in the http body of the response



34

Server Stream Caching

- Header data - headers can be set in any order and are not sent until the first buffer fills
  - Response header data includes age, cache control, language, message digest, MIME type, expiration date, last modified date, etc.
  - isCommitted method - returns boolean indicating that headers have been sent
- Buffered stream data
  - setBufferSize
  - flushBuffer

35

Have You Satisfied the Lecture Objectives?

- Understand the foundations for client/server Web interactions
- Understand the servlet life cycle

36