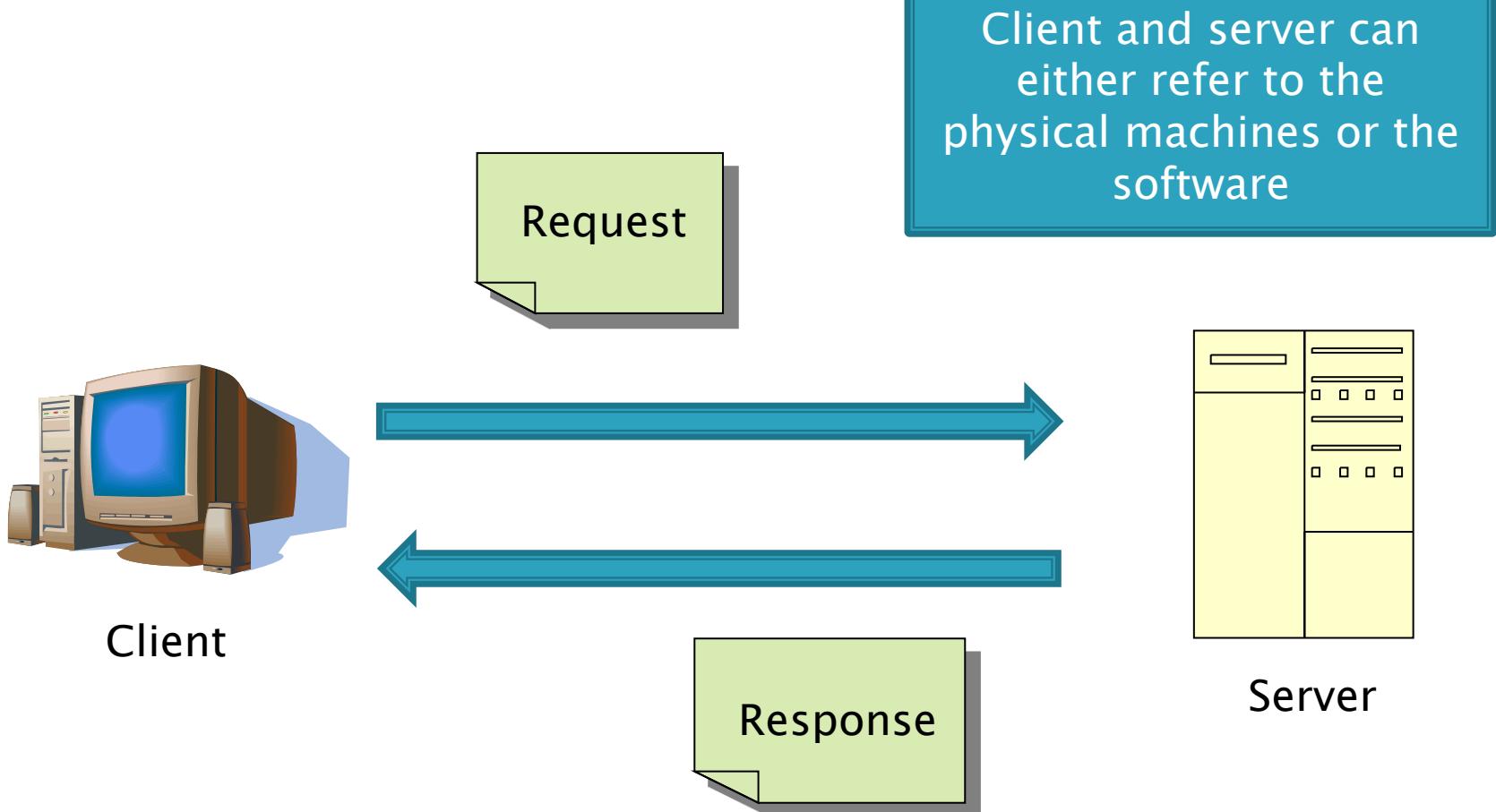


Web Applications and Servlets

EHSDI
e B u z i m a

Web servers



Web clients

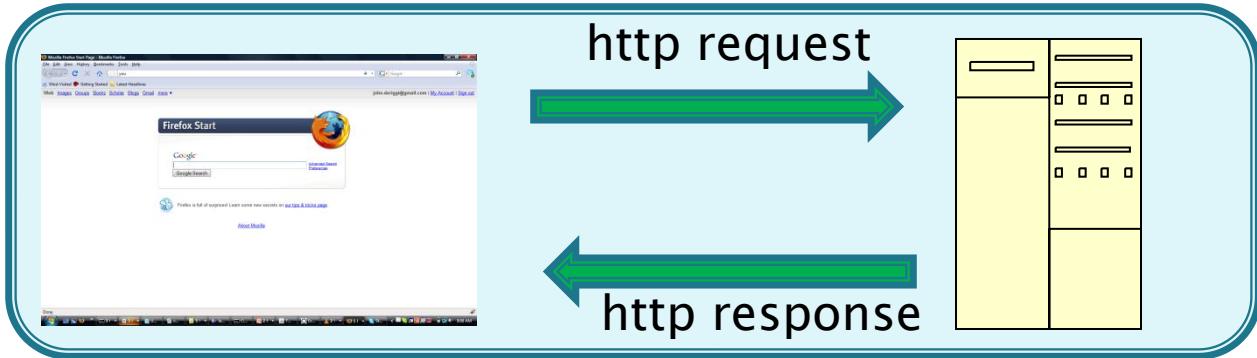
- ▶ A web client lets the user request something on the server, and shows the user the result of the request
- ▶ Usually a web client is a browser which can display images and HTML



Review: HTML and HTTP

- ▶ Hypertext Transfer Protocol (HTTP)
 - Client and server communicate using HTTP
 - Client sends a request, server replies with a response
- ▶ Hypertext Markup Language (HTML)
 - Web pages are generally composed of HTML with other resources (images etc) embedded
 - They might be static .html files
 - Or dynamic scripts that generate HTML, e.g. PHP, ASP, JSP

HTTP



- ▶ A network protocol like TCP/IP with web features
- ▶ Follows a request/response configuration

Request

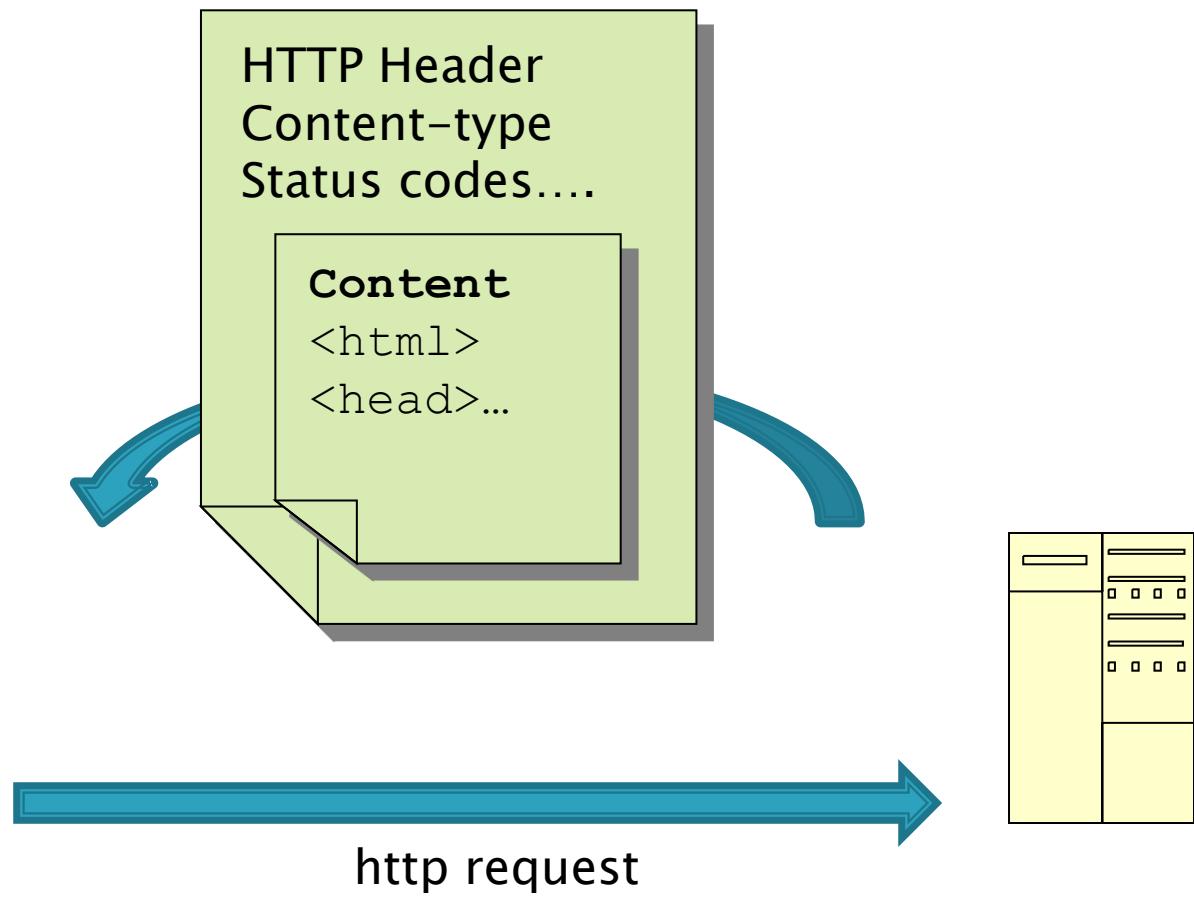
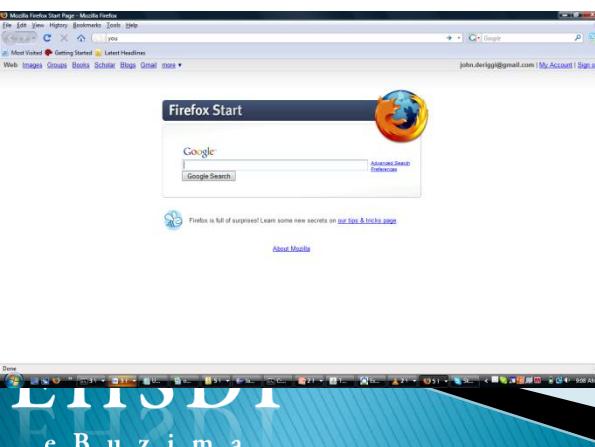
1. Method
2. Page (resource)
3. Parameters

Response

1. Status Code(success or not)
2. Content-type (text,picture)
3. Content(html, image, etc..)

HTML is Part of HTTP Response

- ▶ Browser renders the html content of the response



HTTP Request

- ▶ Request contains the request method:
- ▶ Most common request methods are:
- ▶ Get - a simple request for requesting a resource like an html page, an image, etc
- ▶ Post - can submit some data and a request

GET



http get request



- ▶ Browser makes a GET request, asking the server for a page

Post

Transfer Information

Transfer: Yes

Date of Transfer: 05/16/2009

Transfer From: Where Im From

Admission Information

Mode of Admission: VCT Program

Other:

Information de Partenaire

Partenaire: Waiting For Results

Browser sends the form data to the server



http post

GET and POST Summary

- ▶ GET – a simple request to the server for a resource
- ▶ POST – more powerful than get because you can request a resource and submit form data too

Sending Data with GET



<http://www.google.nw/search?q=john+deriggi+is+the+greatest&ie=utf-8>



Latest Headlines

- ▶ Parameters sent in a GET request is limited by an amount that is dependant on the server
- ▶ Data is appended to the URL

GET Request in FireBug

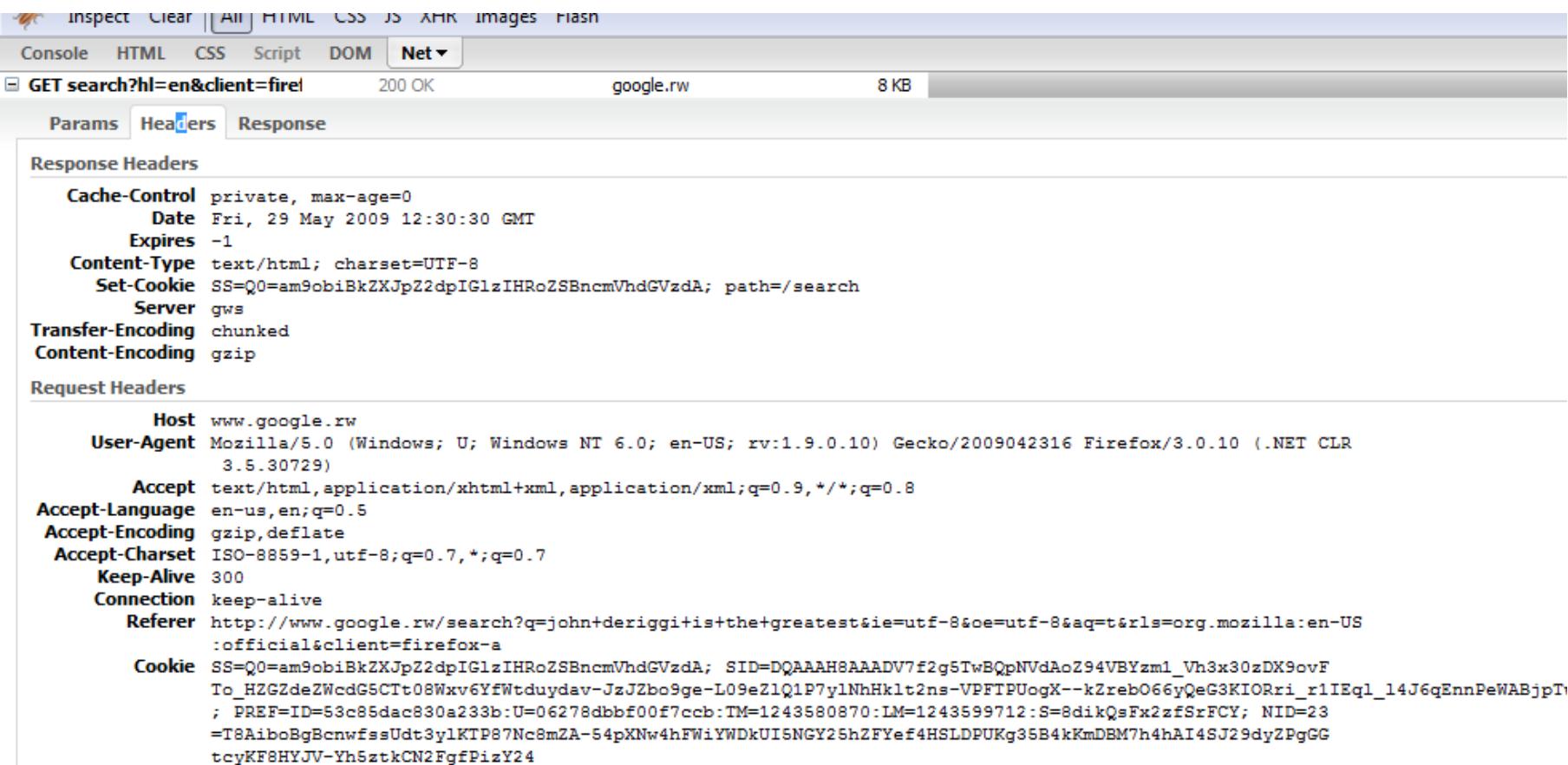
The screenshot shows the Firebug Net panel with two entries:

- GET search?hl=en&client=firef** (200 OK) from google.rw (8 KB).
Params:
 - btnG Search
 - client firefox-a
 - hl en
 - hs isg
 - q john deriggi is the greatest
 - rls org.mozilla:en-US:official
- GET csi?v=3&s=web&action=8** (204 No Content) from google.rw (8 KB).

▶ Parameters

GET Request in FireBug

► Headers!



The screenshot shows the Firebug extension's interface, specifically the 'Net' tab, displaying a captured HTTP request for a Google search. The request URL is 'GET search?hl=en&client=fire'. The status bar indicates a 200 OK response from 'google.rw' with a size of 8 KB. The 'Headers' tab is selected, showing both the Response Headers and Request Headers.

Response Headers:

```
Cache-Control: private, max-age=0
Date: Fri, 29 May 2009 12:30:30 GMT
Expires: -1
Content-Type: text/html; charset=UTF-8
Set-Cookie: SS=Q0=am9obiBkZXJpZ2dpIGlzIHRoZSBncmVhdGVzdA; path=/search
Server: gws
Transfer-Encoding: chunked
Content-Encoding: gzip
```

Request Headers:

```
Host: www.google.rw
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US; rv:1.9.0.10) Gecko/2009042316 Firefox/3.0.10 (.NET CLR 3.5.30729)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
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
Referer: http://www.google.rw/search?q=john+deriggiti+is+the+greatest&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:en-US:official&client=firefox-a
Cookie: SS=Q0=am9obiBkZXJpZ2dpIGlzIHRoZSBncmVhdGVzdA; SID=DQAAAH0AAADV7f2g5TwBQpNVdAoZ94VBYzm1_Vh3x30zDX9ovFTo_HZGZdeZWcdG5CTt08Wxv6YfWtduydav-JzJZbo9ge-L09e21Q1P7y1NhHklt2ns-VPFTPUogX--kZrebO66yQeG3KIORri_r1IEql_14J6qEnnPeWABjpT; PREF=ID=53c85dac830a233b:U=06278dbbf00f7ccb:TM=1243580870:LM=1243599712:S=8dikQsFx2zfSrFCY; NID=23=T8AiboBgBcnwfssUdt3ylKTP87Nc8m2A-54pXNw4hFWiYWDkUI5NGY25h2FYef4HSLDPUKg35B4kKmDBM7h4hAI4SJ29dyZPgGGtcyKF8HYJV-Yh5ztkCN2FgfPizY24
```

GET Request in FireBug - Response!

Inspect Clear All HTML CSS JS XHR Images Flash

Console HTML CSS Script DOM Net ▾

GET search?hl=en&client=fire 200 OK google.rw 8 KB

Params Headers Response

```
<!doctype html><head><title>john deriggi is the greatest - Google Search</title><script>window.google ={kEI:"ZtUfSsObHIGRjAfl6cnGBg",kEXPI:"17259,19592,20572,20665",kHL:"en"}; window.google.sn="web";window.google.timers={load:{t:{start:(new Date).getTime()}}};try{window.google.pt=window.gtbExternal&&window.gtbExternal.pageT()||window.external&&window.external.pageT}catch(b){} window.google.jsrt_kill=1; </script><style>body{background:#fff; color:#000; margin:3px 8px}#gbar{height:22px; padding-left:0px}.gbh,.gbd{border-top:1px solid #c9d7f1; font-size:1px}.gbh{height:0; position:absolute; top:24px; width:100%}#gbi,#gbs{background:#fff; left:0; position:absolute; top:24px; visibility:hidden; z-index:1000}#gbi{border:1px solid; border-color:#c9d7f1 #36c #36c #a2bae7; z-index:1001}#guser{padding-bottom:7px !important; text-align:right}#gbar, #guser{font-size:13px; padding-top:1px !important}@media all{.gb1,.gb3{height:22px; margin-right:.5em; vertical-align:top}#gbar{float:left}}.gb2{display:block; padding:.2em .5em}a.gb1,a.gb2,a.gb3{color:#00c !important}.gb2,.gb3{text-decoration:none}a.gb2:hover{background:#36c; color:#fff !important}a.gb1,a.gb2,a.gb3,.link{color:#20c !important}.ts{border-collapse:collapse}.ts td{padding:0}.ti,.bl,form,#res h3{display:inline}.ti{display:inline-table}.fl:link,.gl a:link{color:#77c}a:link,.w,#prs a:visited, #prs a:active,.q:active,.q:visited{color:#20c}.mblink:visited,a:visited{color:#551a8b}a:active{color:red}.cur{color:#a90a08;font-weight:bold}.b{font-weight:bold}.j{width:42em; font-size:82%}.s{max-width:42em}.sl{font-size:82%}#gb{text-align:right; padding:1px 0 7px; margin:0}.hd{position: absolute; width:1px; height:1px; top:-1000em; overflow:hidden}.f,.m,.c h2,#mbEnd h2{color:#676767}.a,cite,.cite:link{color:green; font-style:normal}#mbEnd{float:right}h1,ol{margin:0; padding:0}li,g,body,html,.std,.c h2,#mbEnd h2,h1{font-size:small; font-family:arial,sans-serif}.c h2,#mbEnd h2,h1{font-weight:normal}#ssb,.clr{clear:both}#nav a,#nav a:visited,.blk a{color:#000}#nav a{display:block}#nav .b a,#nav .b a:visited{color:#20c}#nav .i{color:#a90a08;font-weight:bold}.csb,.ss{background:url(/images/nav_logo4.png) no-repeat 0 0; height:26px; display:block}.ss{background-position:0 -87px; position: absolute; left:0; top:0}.cps{height:18px; overflow:hidden; width:114px}.mbi{width:12px; height:12px; background-position:-114px -78px; margin-right:2px}#nav td{padding:0; text-align:center}#logo{display:block; overflow:hidden; position: relative; width:150px; height:52px; margin:14px 0 7px}#logo img{border:none; position: absolute; left:-0px; top:-26px}.w10,.w11,.w20,.w21,.w24,.wci,.wpb{background:url(/images/nav_logo4.png) no-repeat; border:0; cursor:pointer; margin-left:8px; height:16px; vertical-align:bottom; width:16px}#10{background-position:-153px 0}#11
```

POST Request in FireBug

Console HTML CSS Script DOM Net ▾

POST pageone.form?patientId 200 OK localhost:8080 9 KB

Params Headers Post Response

Response Headers

```
Server: Apache-Coyote/1.1
Pragma: No-cache
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Cache-Control: no-cache, no-store
Content-Type: text/html; charset=ISO-8859-1
Content-Language: en-US
Transfer-Encoding: chunked
Date: Fri, 29 May 2009 12:45:13 GMT
```

Request Headers

```
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US; rv:1.9.0.10) Gecko/2009042316 Firefox/3.0.10 (.NET CLR 3.5.30729)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
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
Referer: http://localhost:8080/openmrs/module/tracplus/pageone.form?patientId=237
Cookie: dashboardTab-503=TracPlusTab; JSESSIONID=7DAD6BB97B23701B7629E2AC0FCA77E9
```

GET DWRArtService.js 200 OK localhost:8080 379 B

Done

A screenshot of a Windows taskbar showing various open applications including a browser, file explorer, and system tray icons.

POST Request in FireBug

The screenshot shows the Firebug extension for a web browser. The title bar indicates the browser is running on localhost:8080. The Firebug interface has tabs for Inspect, Clear, All, HTML, CSS, JS, XHR, Images, and Flash. The Net tab is selected, showing a list of requests. One request is expanded, showing the following parameters:

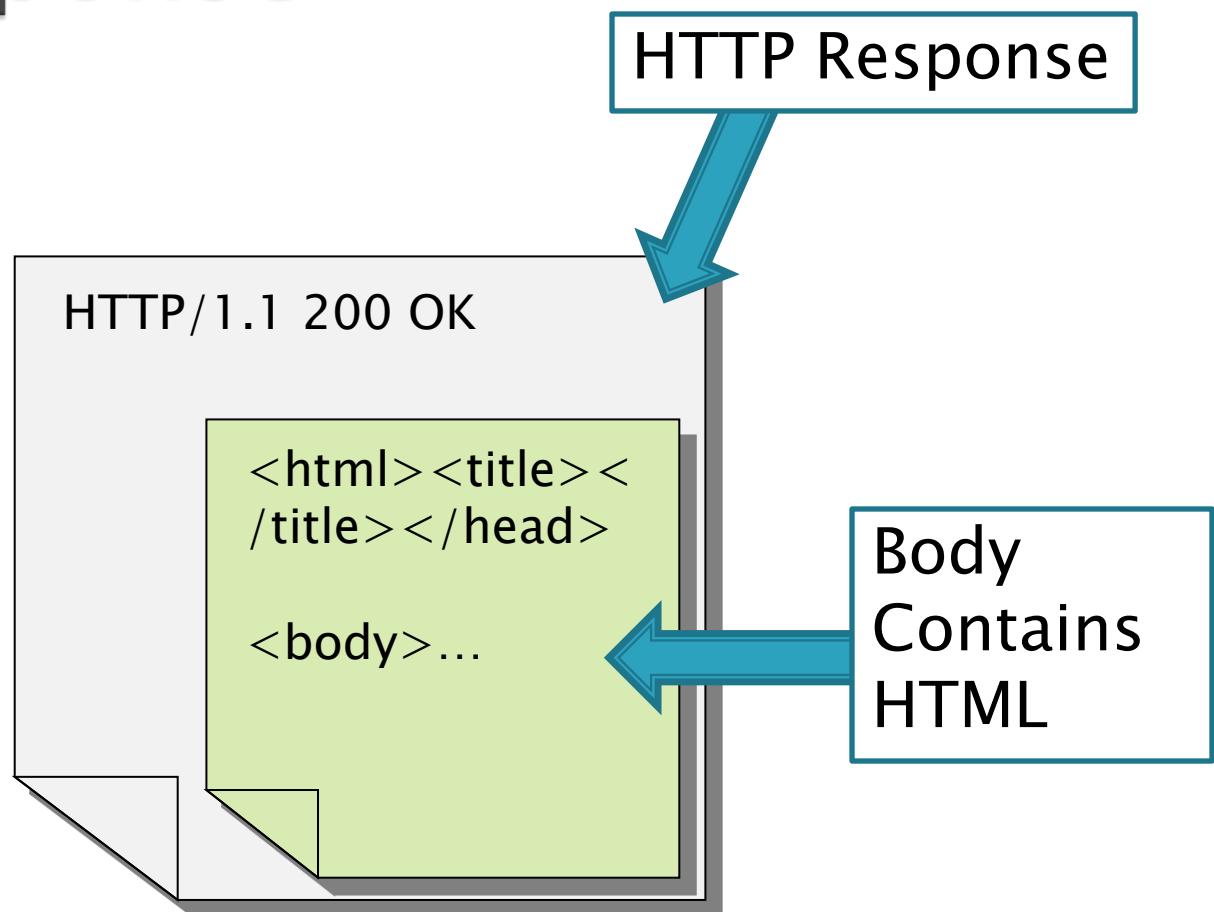
Param	Value
_programIds	1
coded_1054	-1
coded_1293	-1
coded_13041	-1
coded_1480	-1
coded_1607	-1
coded_1650	-1
coded_2531	-1
coded_2536	-1
coded_3082	-1
coded_3495	-1
coded_5959	-1
dateOfEncounter	05/07/2009
date_1428	
date_1837	05/04/2009
date_5096	05/14/2009
date_5596	
date_968	
dayOfBirth	01/18/1965
gender	F
intakeLocationId	3
intakeProviderId	503
numeric_1053	--
numeric_1294	--

At the bottom of the Firebug window, there are icons for taskbar items like Windows, Firefox, and Microsoft Word, along with file and folder icons.

POST Request

- ▶ The parameters are in the body of the message so they are not limited in size!

HTTP Response



HTTP Response

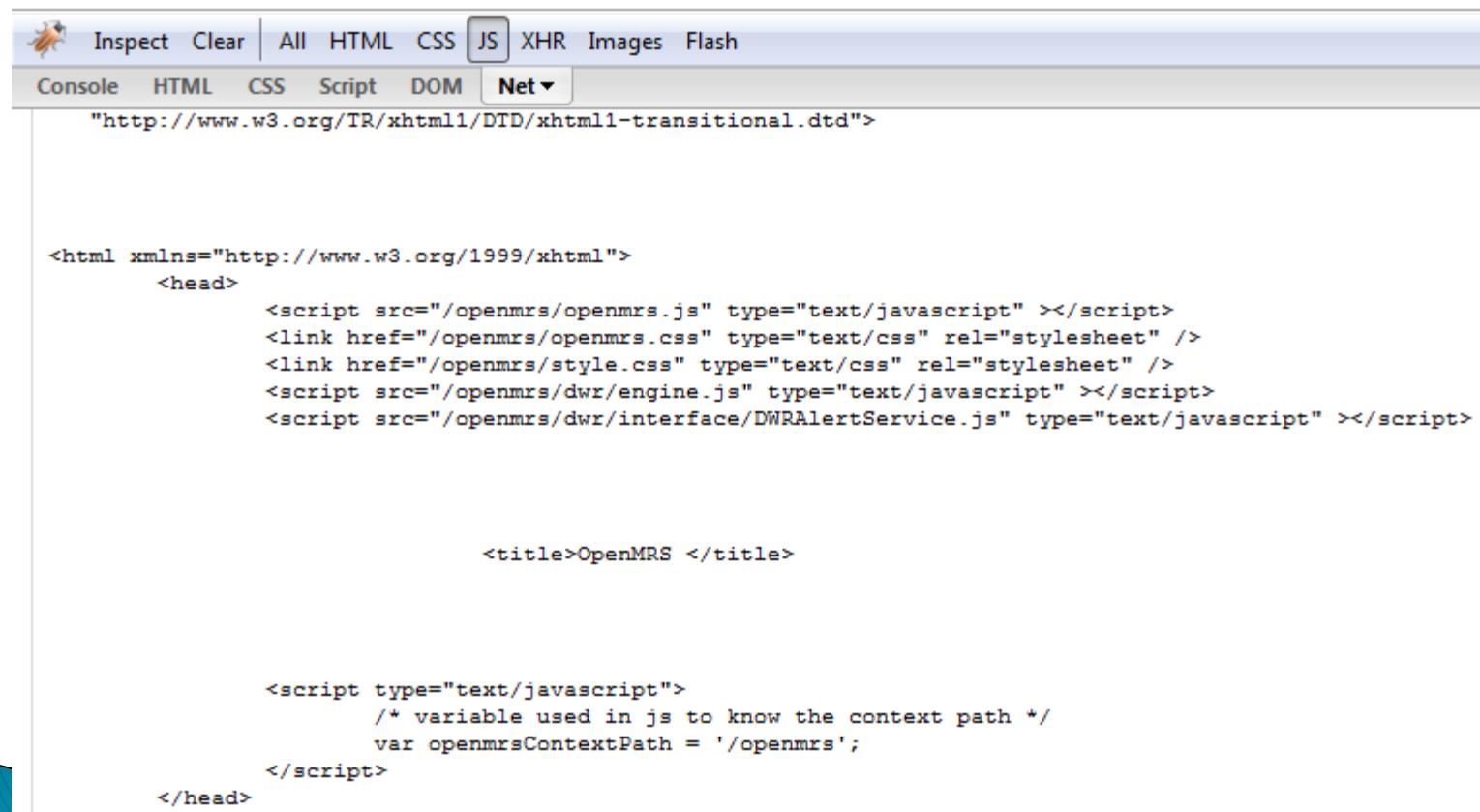
▶ Response Headers

Params Headers Post Response

Response Headers

Server	Apache-Coyote/1.1
Pragma	No-cache
Expires	Thu, 01 Jan 1970 00:00:00 GMT
Cache-Control	no-cache, no-store
Content-Type	text/html; charset=ISO-8859-1
Content-Language	en-US
Transfer-Encoding	chunked
Date	Fri, 29 May 2009 12:45:13 GMT
..	

HTTP Response Payload



The screenshot shows a browser's developer tools interface with the Network tab selected. The request URL is "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd". The response content is an XML document representing the XHTML DTD.

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <script src="/openmrs/openmrs.js" type="text/javascript" ></script>
    <link href="/openmrs/openmrs.css" type="text/css" rel="stylesheet" />
    <link href="/openmrs/style.css" type="text/css" rel="stylesheet" />
    <script src="/openmrs/dwr/engine.js" type="text/javascript" ></script>
    <script src="/openmrs/dwr/interface/DWRAjaxService.js" type="text/javascript" ></script>

    <title>OpenMRS </title>

    <script type="text/javascript">
      /* variable used in js to know the context path */
      var openmrsContextPath = '/openmrs';
    </script>
  </head>
```

```
// Send data  
URL url = new URL(fromWhere);  
URLConnection conn = url.openConnection();  
conn.setDoOutput(true);
```

```
// Get the response
BufferedReader rd = new BufferedReader(new InputStreamReader(conn.getInputStream()));
String line;
while ((line = rd.readLine()) != null) {
    System.out.println("server response: " + line);
}
wr.close();
rd.close();
```

And By the Way, You Can Try This With Java

```
// Send data
URL url = new URL(fromWhere);
URLConnection conn = url.openConnection();
conn.setDoOutput(true);
OutputStreamWriter wr = new OutputStreamWriter(conn.getOutputStream());

wr.write("POST / HTTP/1.1\r\n");
wr.write("Host: www.google.com\r\n");
wr.write("User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/20100101 Firefox/4.0.1\r\n");
wr.write("Accept: text/html, application/xhtml+xml, */*\r\n");
wr.write("Accept-Language: en-us,en;q=0.5\r\n");
wr.write("Accept-Encoding: gzip, deflate\r\n");
wr.write("Connection: keep-alive\r\n");
wr.write("\r\n");

wr.flush();

// Get the response
BufferedReader rd = new BufferedReader(new InputStreamReader(conn.getInputStream()));
String line;
while ((line = rd.readLine()) != null) {
    System.out.println("server response: " + line);
}
wr.close();
rd.close();
```

URL

OpenMRS - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://staging.pih-emr.org:8080/openmrs/login.htm

Most Visited Getting Started Latest Headlines

Yahoo! Mail (j_deriggi) OpenMRS

EMR 2.0 for the P Not logged in | Log in | Help


EMR 2.0
Welcome to Openmrs. Please login to proceed.

Username:
Password:

[I forgot my password](#)

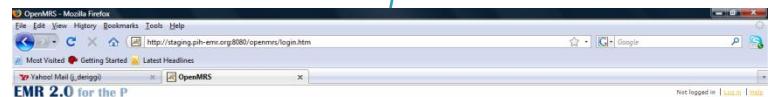
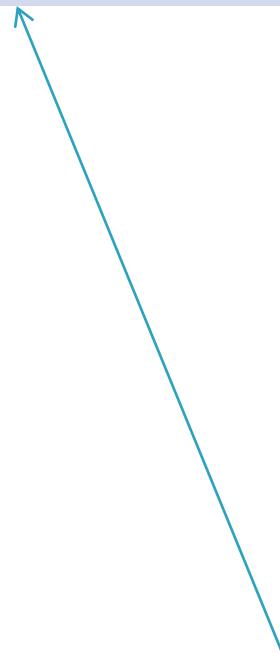
English (United Kingdom) | English (United States) | français Last Build: Mar 25 2009 03:36 PM Version: 1.4.0.23 RC Build 7163 Database Version: 1.4.0.23

Done

E Buzima

3 W. 3 M. 2 N. open... 6 W. Java... C:\P... 2 M. To... Ext 2... 2 V... 4 F... 3:14 PM

URL - Uniform Resource Locator



Welcome to OpenMRS. Please login to proceed.
Username:
Password:
[I forgot my password](#)



URL



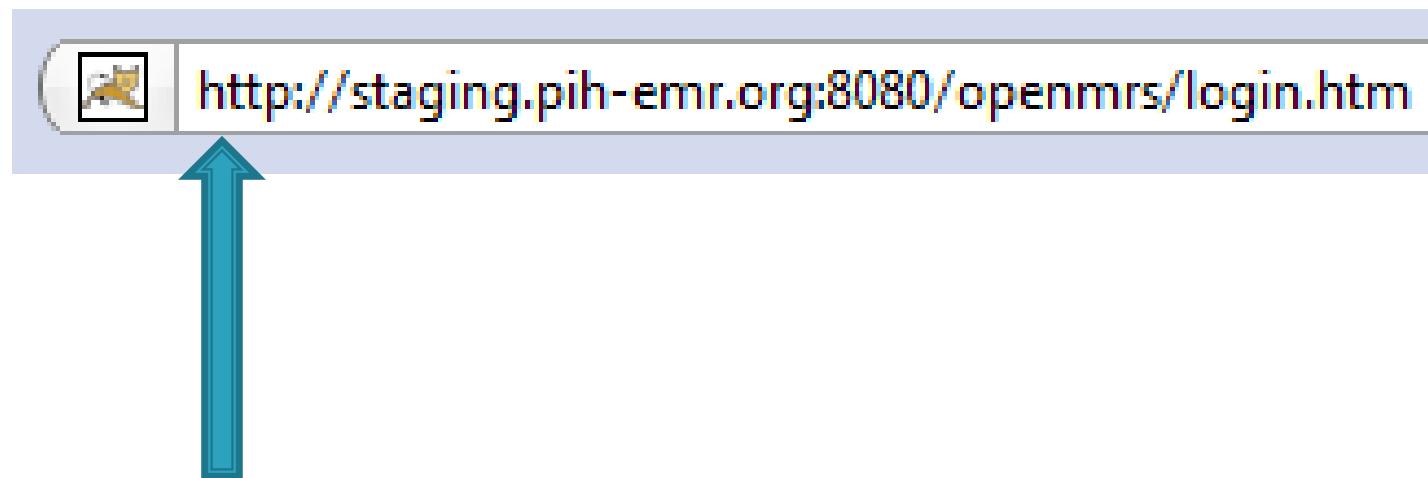
<http://staging.pih-emr.org:8080/openmrs/login.htm>

- ▶ Protocol
- ▶ Port
- ▶ Resource
- ▶ Server
- ▶ Path
- ▶ Optional Query String



Protocol

- ▶ Tells the server which communications protocol will be used



URL



<http://staging.pih-emr.org:8080/openmrs/login.htm>

- ▶ Protocol
- ▶ Server
- ▶ Port
- ▶ Resource
- ▶ Path
- ▶ Optional Query String

Server



- ▶ The unique name of the server. The name maps to a unique IP address. You can just specify the IP address if you want.

URL



<http://staging.pih-emr.org:8080/openmrs/login.htm>

- ▶ Protocol
- ▶ Server
- ▶ Port
- ▶ Resource
- ▶ Path
- ▶ Optional Query String

Port



- ▶ A server may support many ports and you can specify a server application with the port number.
- ▶ If not specified then it defaults to port 80

URL

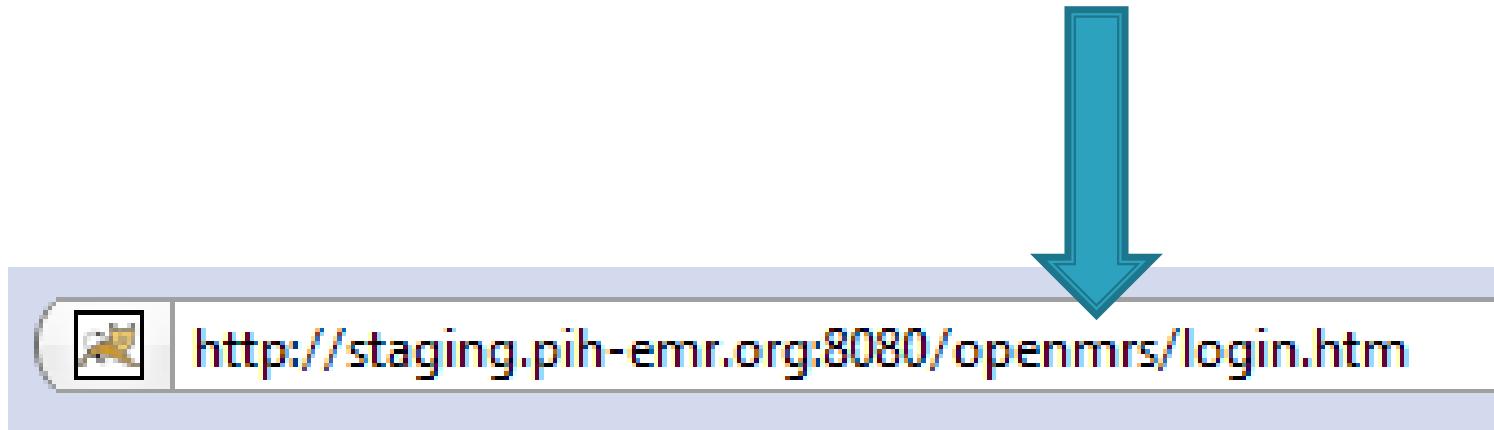


<http://staging.pih-emr.org:8080/openmrs/login.htm>

- ▶ Protocol
- ▶ Server
- ▶ Port
- ▶ Path
- ▶ Resource
- ▶ Optional Query String

Path

- ▶ The path to the location on the server. The earliest web servers were Unix machines so we use Unix syntax for file locations



URL



<http://staging.pih-emr.org:8080/openmrs/login.htm>

- ▶ Protocol
- ▶ Server
- ▶ Port
- ▶ Path

- ▶ Optional Query String

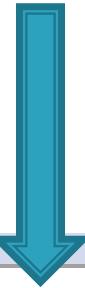
URL



<http://staging.pih-emr.org:8080/openmrs/login.htm>

- ▶ Protocol
- ▶ Server
- ▶ Port
- ▶ Path
- ▶ Resource
- ▶ Optional Query String

Resource



- ▶ The name of the requested content. HTML, PDF, a video, etc

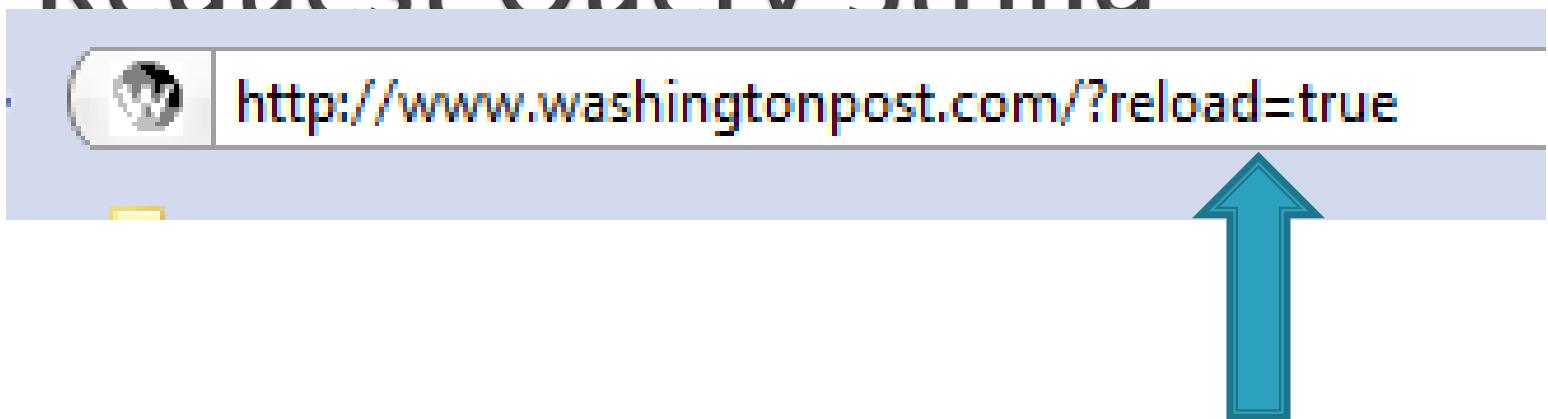
URL



<http://staging.pih-emr.org:8080/openmrs/login.htm>

- ▶ Protocol
- ▶ Server
- ▶ Port
- ▶ Path
- ▶ Resource
- ▶ Optional Query String

GET Request Query String



- ▶ In a GET Request there may be parameters on the end of the URL

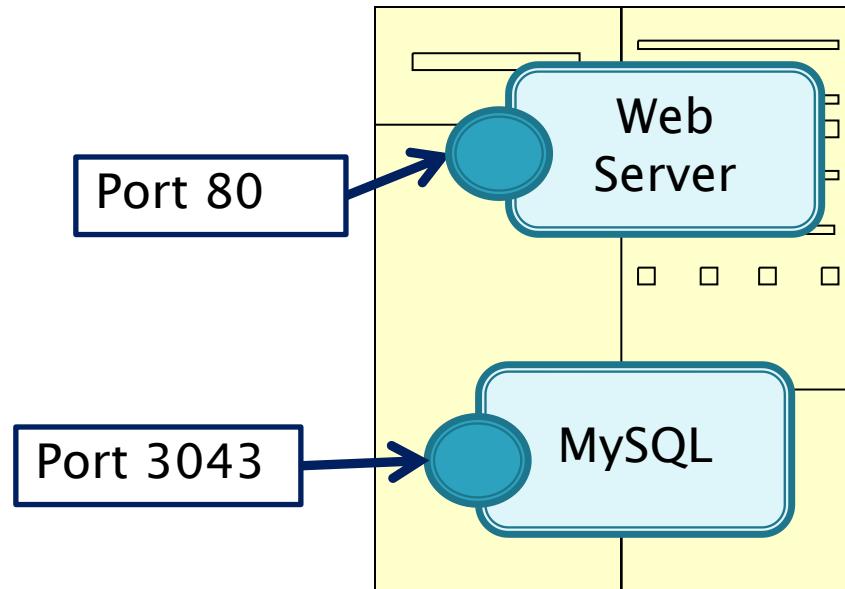
TCP Port

- ▶ A 16-bit number that identifies a program on the server
 - Examples:
 - HTTP Web servers run on port 80
 - FTP servers run on port 20
 - HTTPS runs on port 443
 - MySQL defaults to port 3043
 - TCP Port numbers range from 0 – 65536
 - Generally speaking, ports from 0 – 1023 are reserved for administrative applications



TCP Port

- ▶ Ports allow the server to determine which applications should be connected to the client

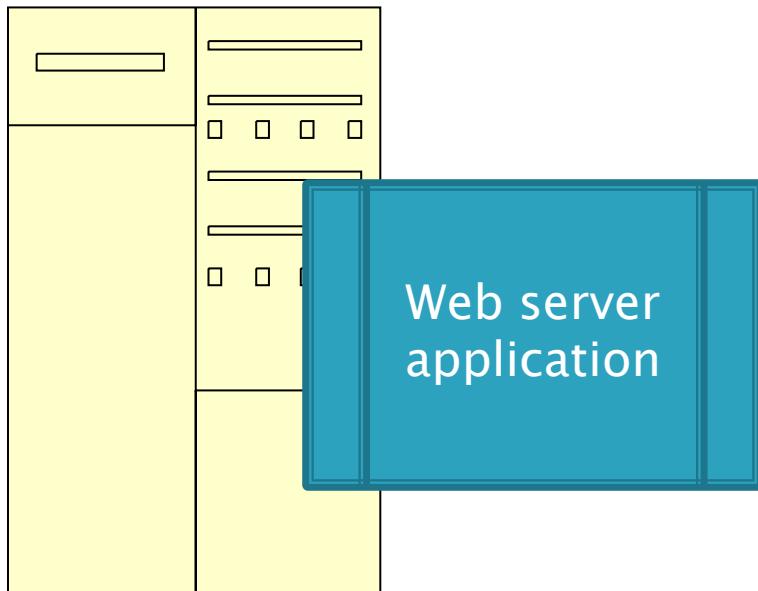


- ▶ One server application per port!
BindException ← look it up!

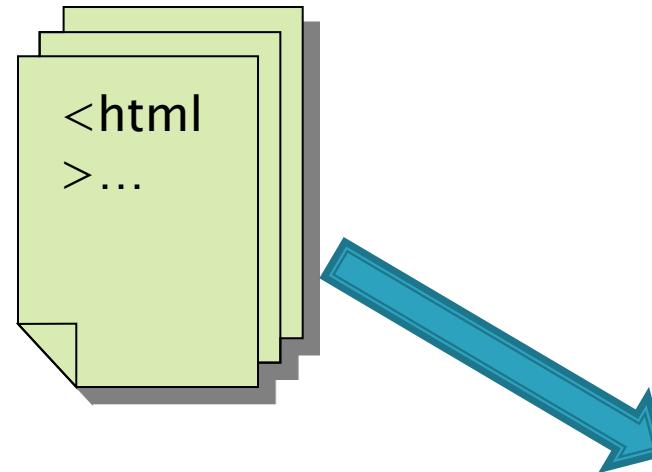
Apache Web Server?

Web Servers and Static Web Pages

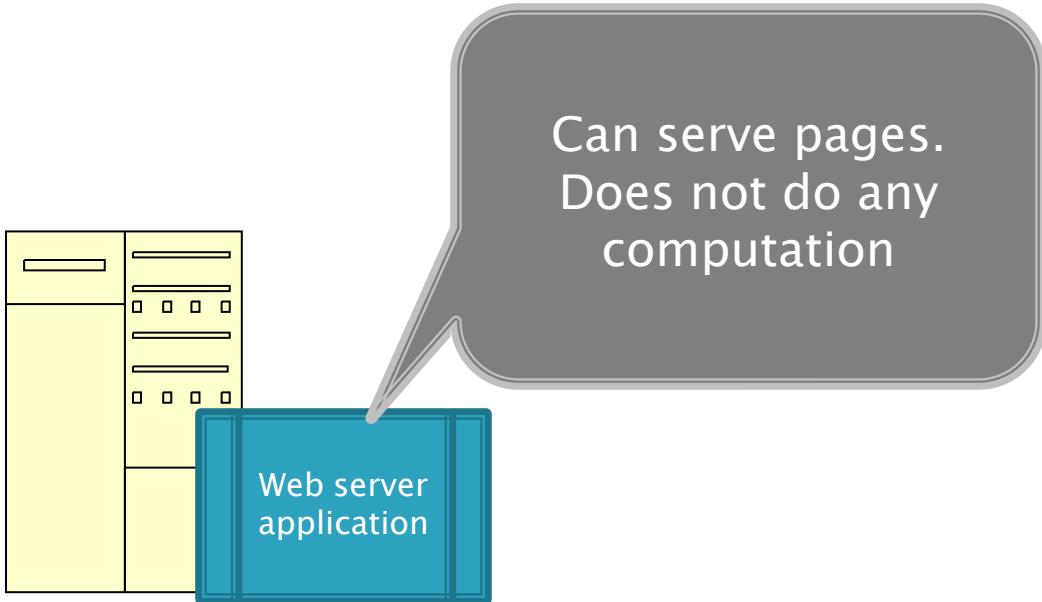
- ▶ HTML Content is delivered to the client exactly like it is on the server



web server machine



Web Servers and Dynamic Pages



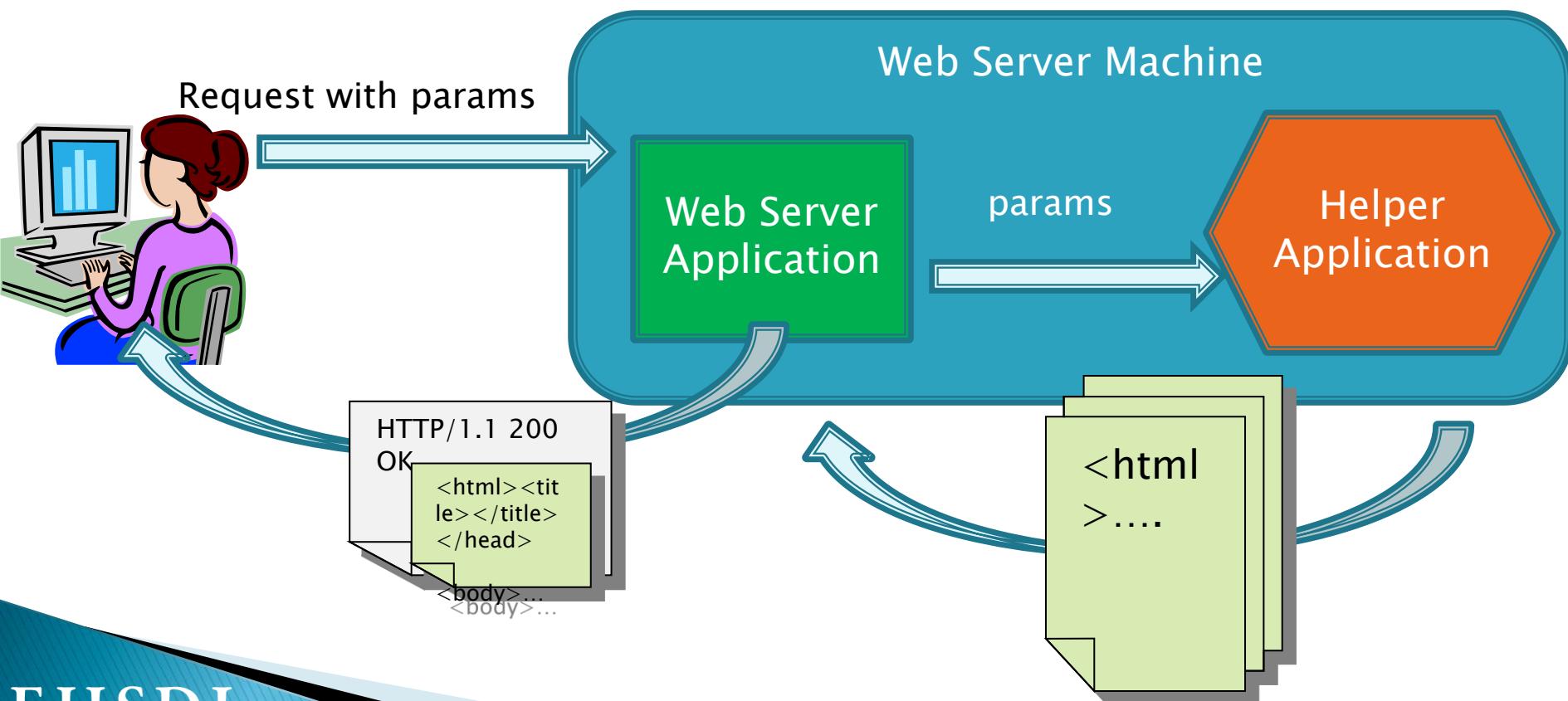
Web Server Alone Cannot...

- ▶ Serve Dynamic Content
 - A web server needs to communicate with another application on the server to assemble
- ▶ Saving Data
 - When a web server receives data from a form or an uploaded file for example, it must hand it off to another application to save it to a file

CGI – Common Gateway Interface

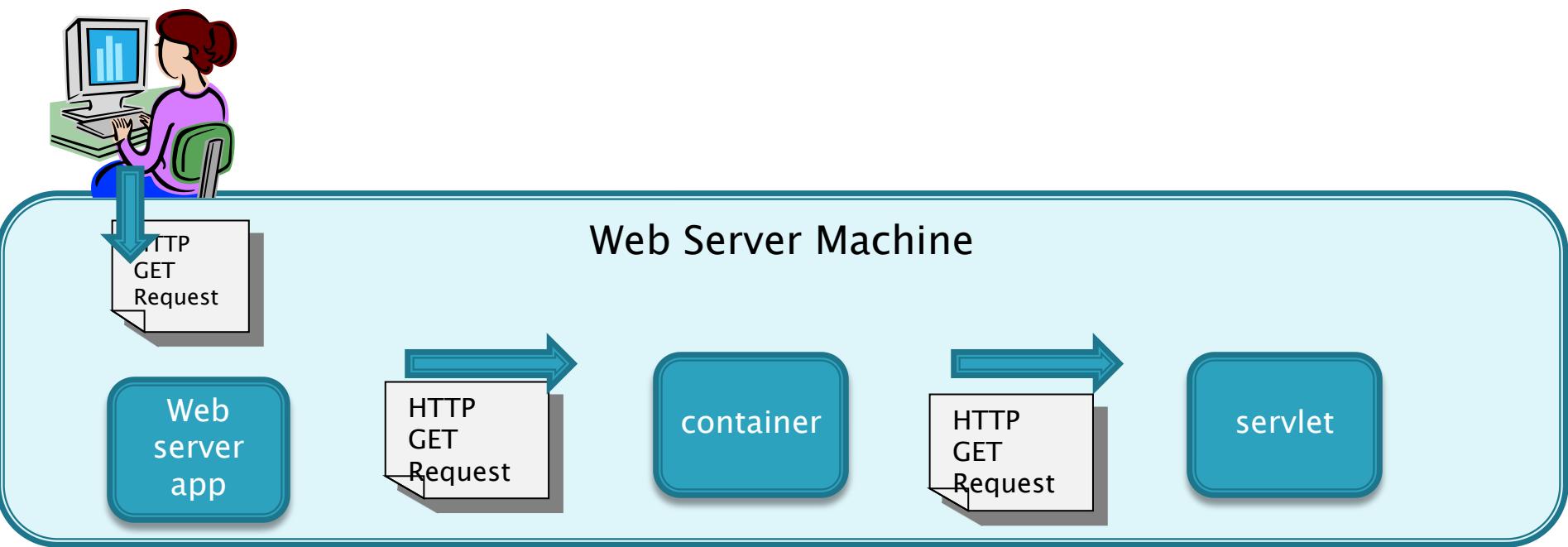
- ▶ Can be written in many languages.
- ▶ Common ones are:
 - Perl
 - Python
 - C
 - PHP

Web Server and Helper Application



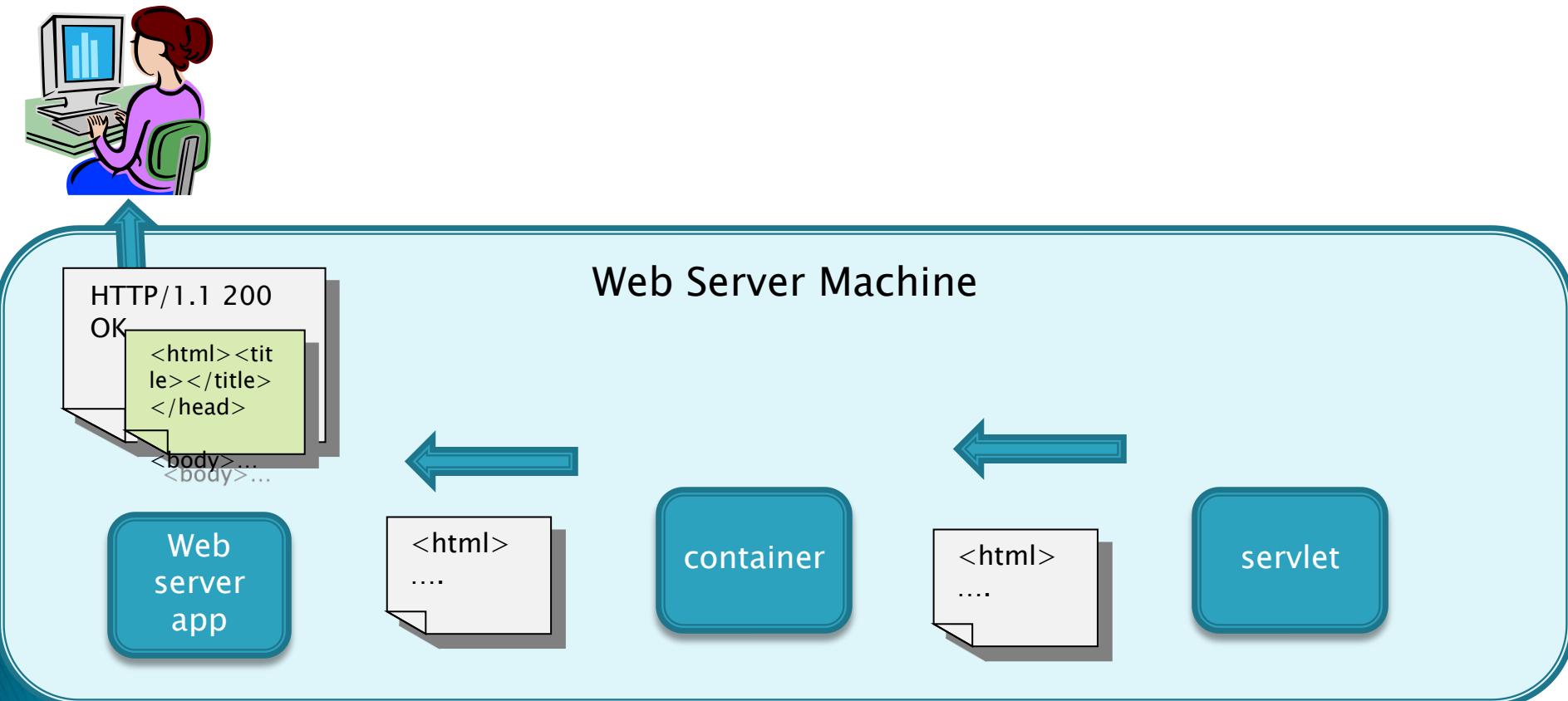
Web App Architecture

- ▶ Servlets have not a main() method
- ▶ Servlets are controlled by a Java Application called a Servlet Container



Web App Architecture

- ▶ Servlets have not a main() method
- ▶ Servlets are controlled by a Java Application called a Servlet Container

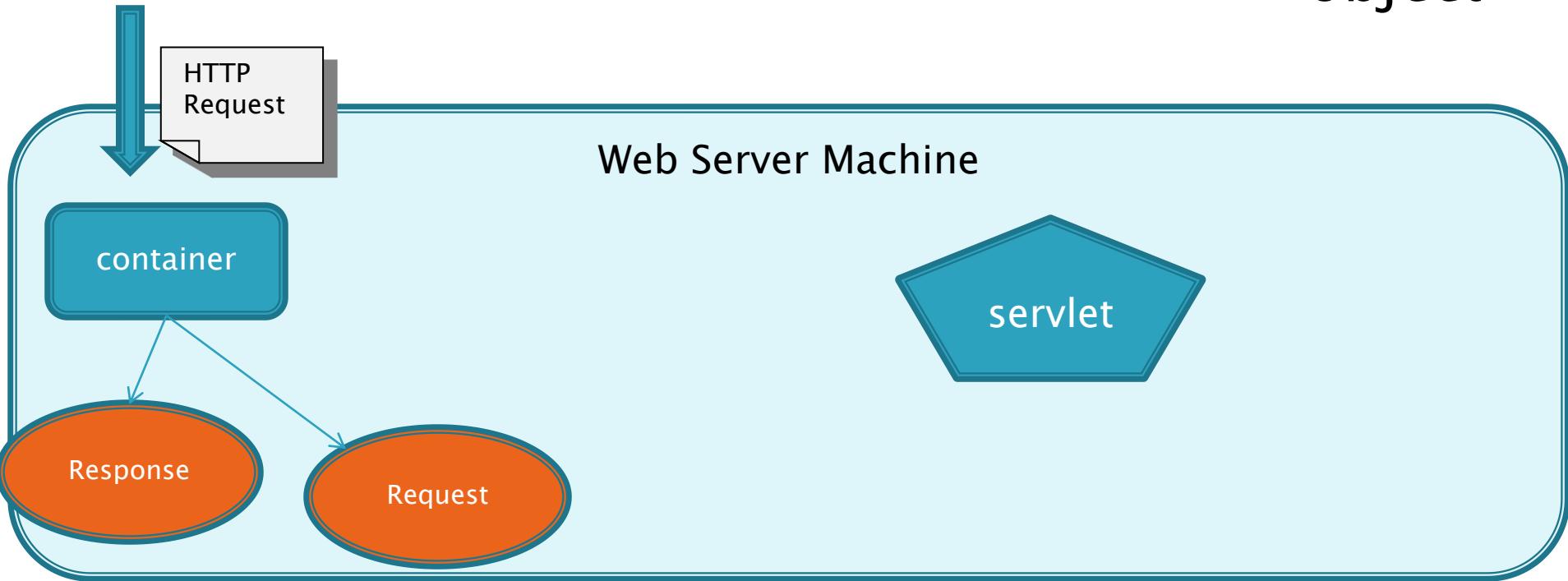


The Container Does Many Things

- ▶ **Communications Support** – It handles the communication between the servlet container and the web server
- ▶ **Lifecycle Management** – Handles the loading and instantiating of servlets, invokes(calls) the servlet methods and making the objects available for garbage collection.
- ▶ **Multihtreading Support** – New threads are created for each servlet request
- ▶ **Declarative Security** – Configurable Security Management without modifying Java source files
- ▶ **JSP Support** – Converts a JSP into Java

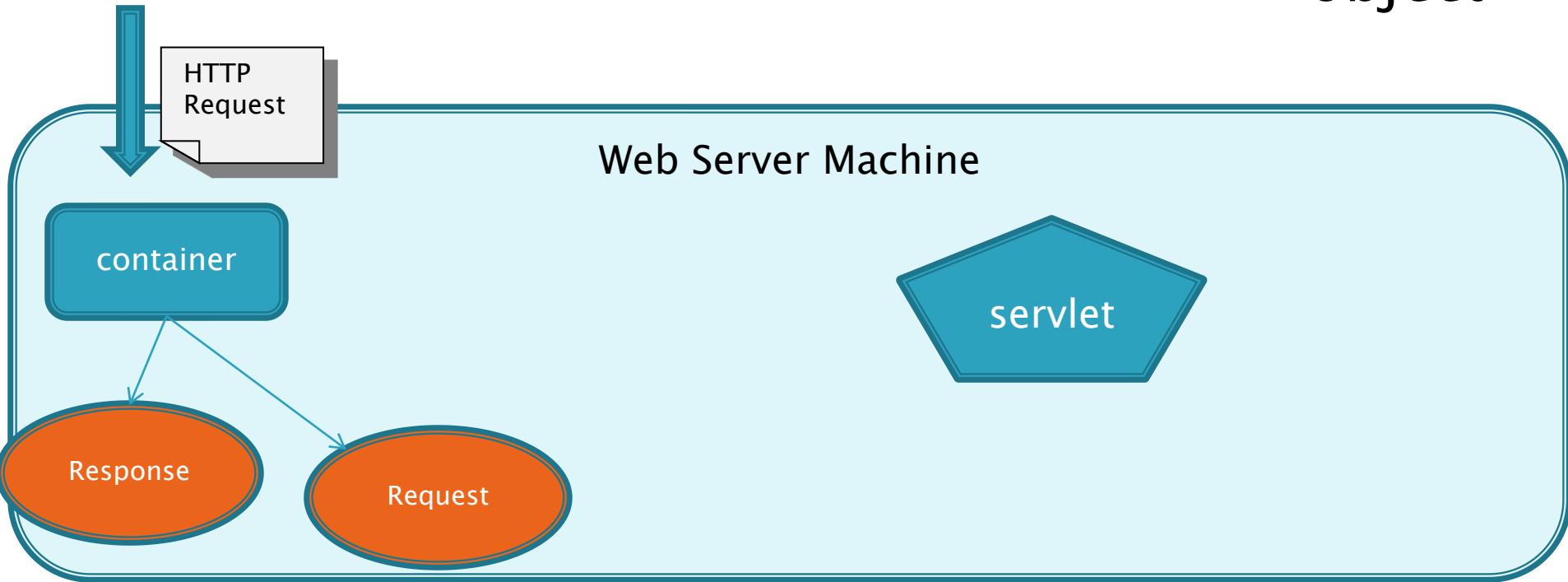
Container Request

- ▶ Container create a request and response object



Container Request

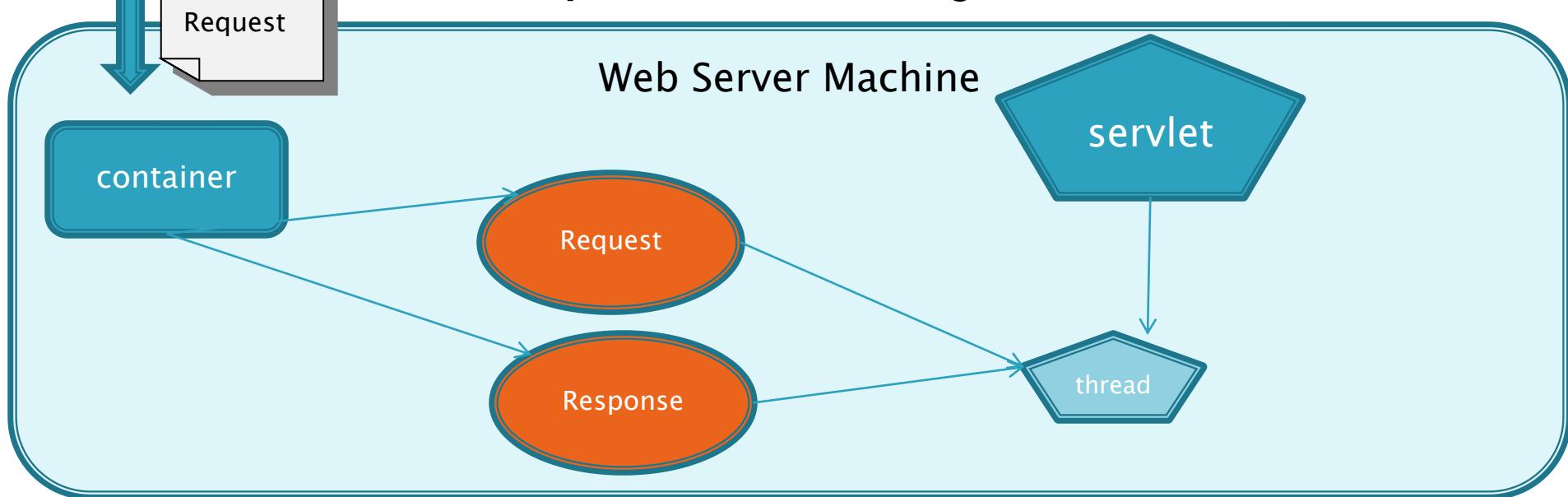
- ▶ Container create a request and response object



Container Request



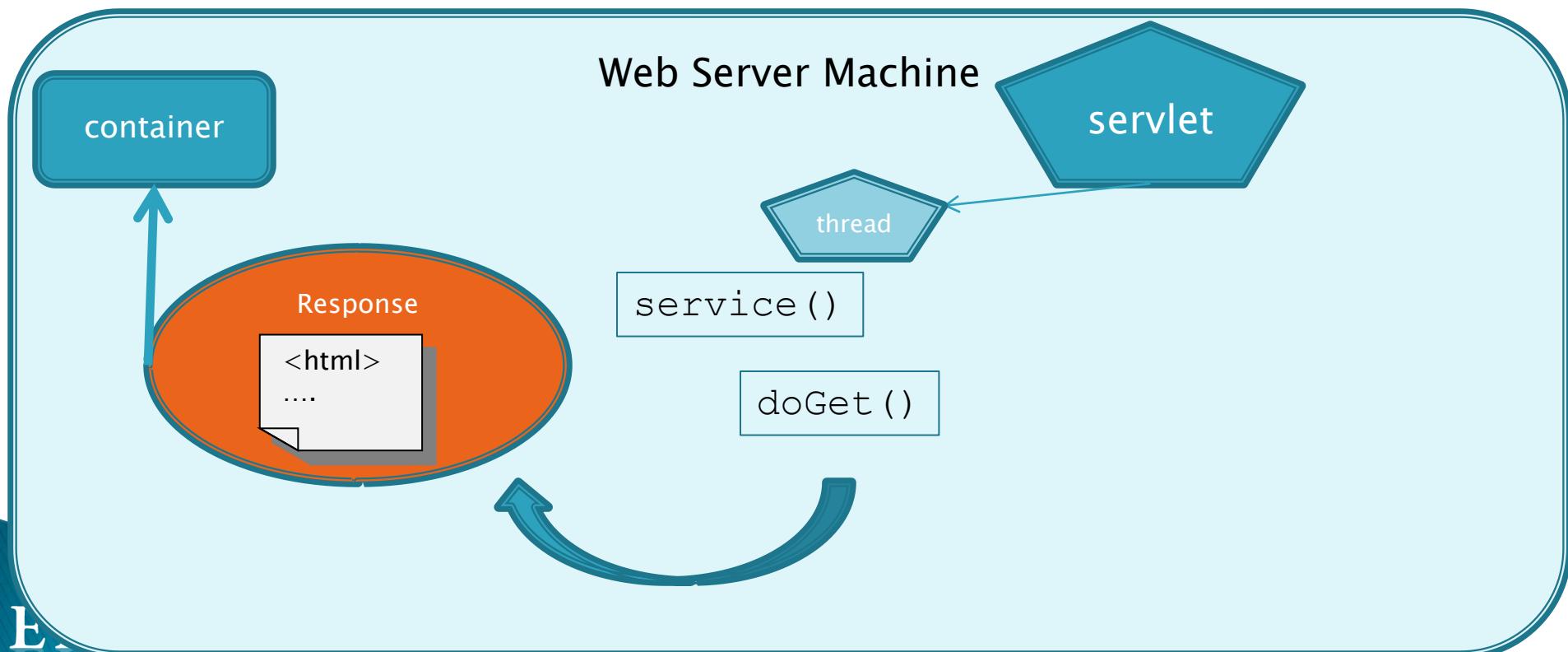
▶ Container finds the servlet corresponding the the url, spawns a thread, passes the objects to the thread



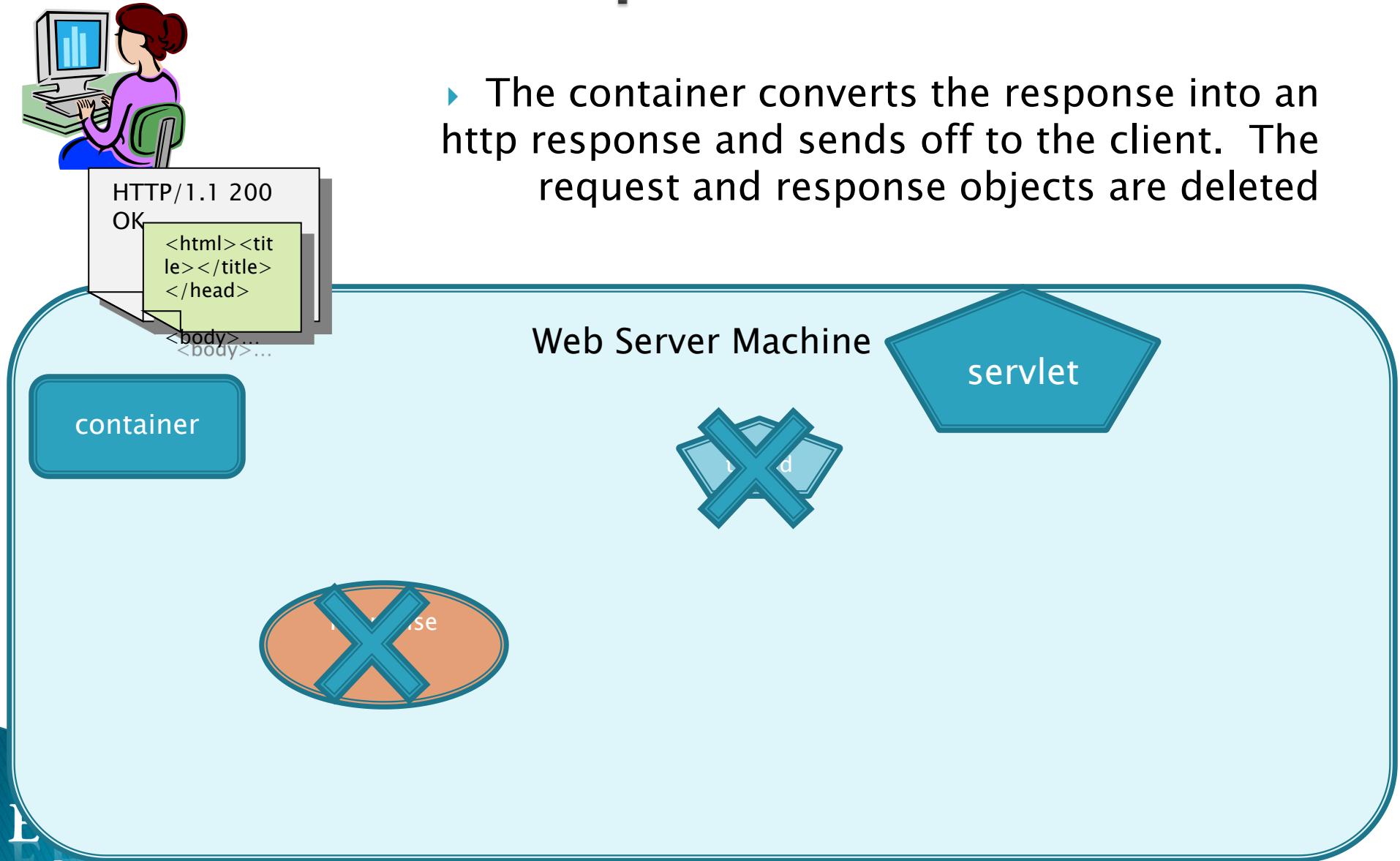
Container Request



- ▶ The Container calls the service method which then calls either `doGet()` or `doPost()`



Container Request



Servlets

- ▶ A basic servlet exercise
- ▶ The Deployment Descriptor
- ▶ Deploying the web application

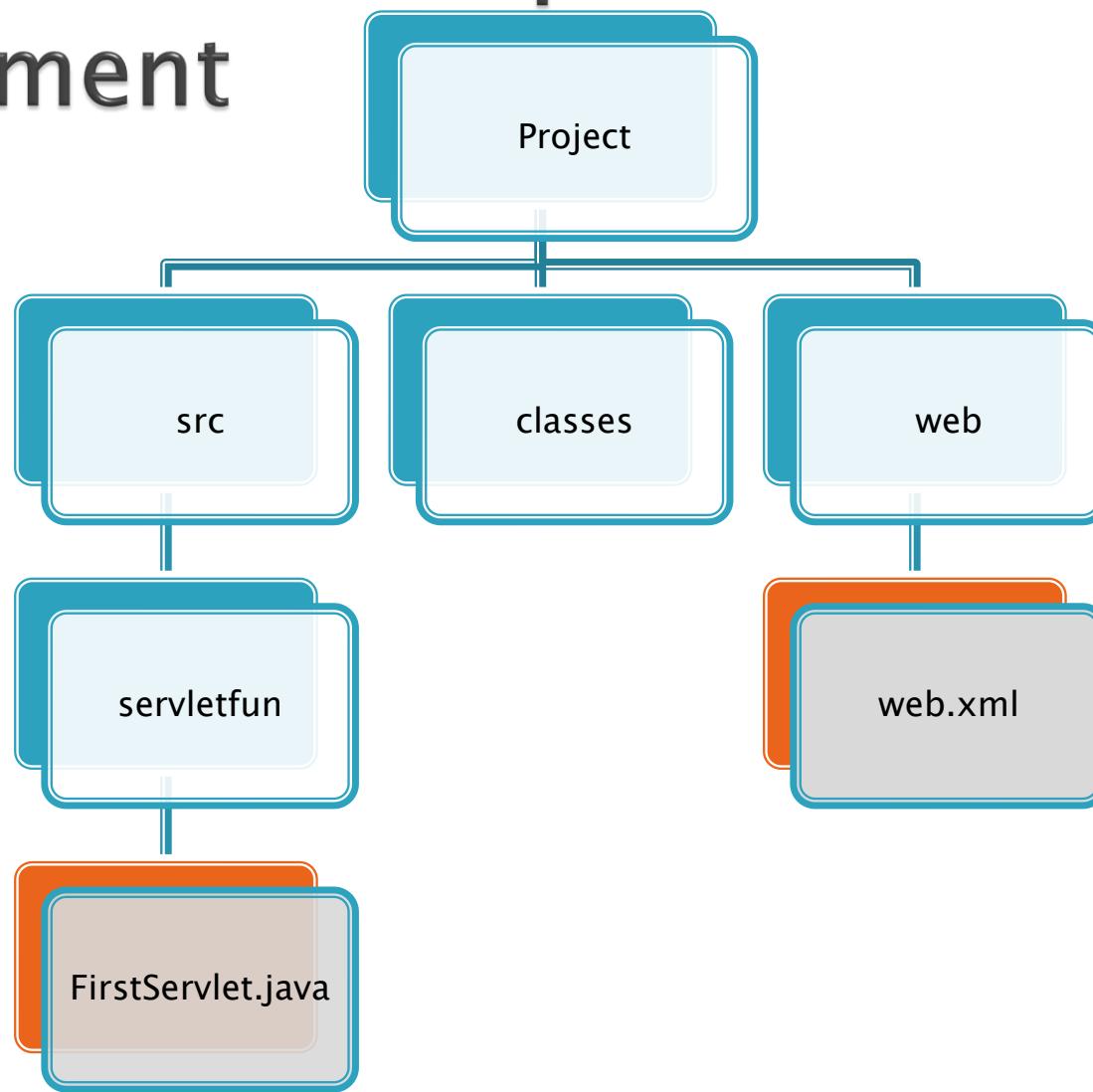
Exercise

- ▶ Build our development environment
- ▶ Write the servlet
- ▶ Create the deployment descriptor
- ▶ Deploy the app

Servlet Exercise

- ▶ This exercise is describe between slides 60–64. You may do it at your own pace

Create This Development Environment



servletfun.FirstServlet – The Servlet

```
① /**
 * Servlet implementation class FirstServlet
 */
public class FirstServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    ②     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        out.println("<html><body>Today's date is " + new Date() + "</body></html>");
    }

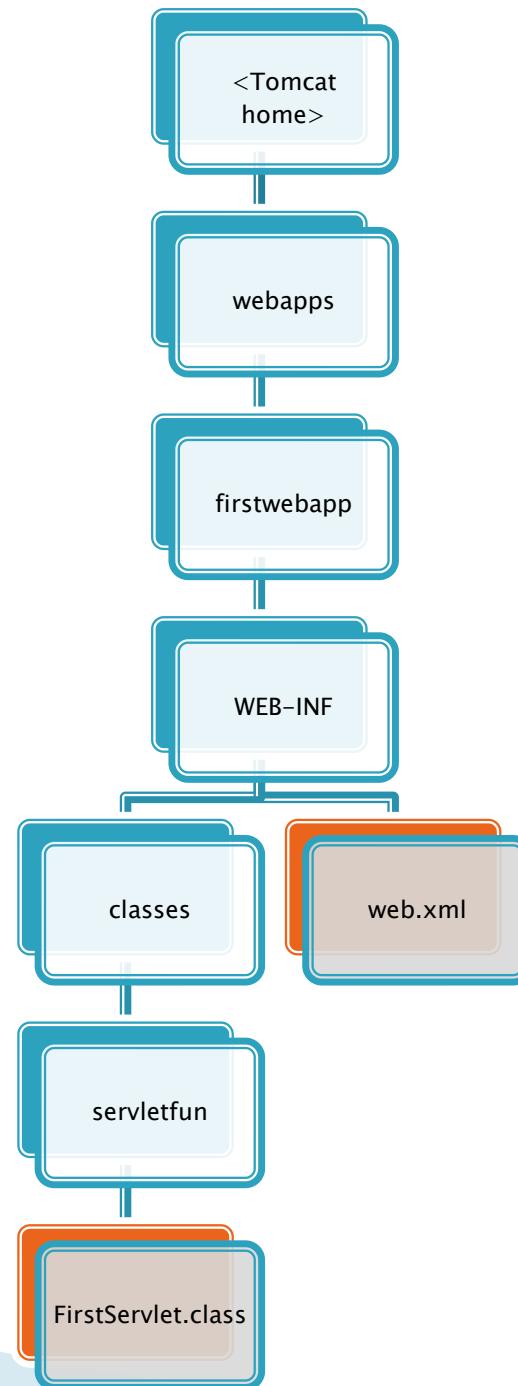
    ③     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    }

}
```

web.xml – The Deployment Descriptor

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID"
    version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <display-name>
        ServletFun</display-name>
    <servlet>
        <description>
        </description>
        <display-name>
            FirstServlet</display-name>
        <servlet-name>FirstServlet</servlet-name>
        <servlet-class>
            servletfun.FirstServlet</servlet-class>
        </servlet>
        <servlet-mapping>
            <servlet-name>FirstServlet</servlet-name>
            <url-pattern>/FirstServlet</url-pattern>
        </servlet-mapping>
        <welcome-file-list>
            <welcome-file>index.html</welcome-file>
            <welcome-file>index.htm</welcome-file>
            <welcome-file>index.jsp</welcome-file>
            <welcome-file>default.html</welcome-file>
            <welcome-file>default.htm</welcome-file>
            <welcome-file>default.jsp</welcome-file>
        </welcome-file-list>
    </web-app>
```

Build This Deployment Structure



Start Tomcat and View the Web App

- ▶ <tomcat home>/bin/startup.sh
- ▶ <http://localhost:8080/firstwebapp/FirstServlet>
- ▶ Shutdown tomcat:
 - <tomcat home>/bin/shutdown.sh

So What Did We Just Do?

The request and response references from the container

```
① /**
 * Servlet implementation class FirstServlet
 */
public class FirstServlet extends HttpServlet
    private static final long serialVersionUID = 1L;

②     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        out.println("<html><body>Today's date is " + new Date() + "</body></html>");
    }

③     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    }

④ }
```

We will almost always
override `doGet()` or
`doPost()`

The Container Must Find the Servlet

- ▶ The URL from the request is **mapped** to a specific servlet (does everyone know what it means to create a mapping)
- ▶ Think HashMap

The Deployment Descriptor

- ▶ An XML document to tell the container how to run your servlets and JSPs
- ▶ You can do many things with the deployment descriptor
- ▶ For now, let's look at the servlet mapping

The Deployment Descriptor

- ▶ < servlet >
maps internal name to a fully qualified class name
- ▶ < servlet-mapping >
maps the internal name to the public URL name

The Deployment Descriptor

```
<!-- -->
<servlet>
    <description>
    </description>
    <display-name>
        FirstServlet</display-name>
    <servlet-name>FirstServlet</servlet-name>
    <servlet-class>
        servletfun.FirstServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>FirstServlet</servlet-name>
        <url-pattern>/FirstServlet</url-pattern>
    </servlet-mapping>
```

Public name

Used only
within the DD

Used only
within the DD

Deployment Descriptor

```
<servlet> 2) the container uses the  
    <des... name...>  
        </description>  
        <display-name>  
            FirstServlet</display-name>  
        <servlet-name>FirstServlet</servlet-name>  
        <servlet-class>  
            servletfun.FirstServlet</servlet-class>  
    </servlet>  
    <servlet-mapping>  
        <servlet-name>FirstServlet</servlet-name>  
        <url-pattern>/FirstServlet</url-pattern>  
    </servlet-mapping>
```

1) When a request comes in for this URL...

3) and invokes the corresponding servlet

A Few Other Things You Can Do With the Deployment Descriptor

- ▶ Can adjust security roles
- ▶ Configure welcome pages
- ▶ Configure error pages

Exercises

- ▶ URLEncoder get or post request
- ▶ Servlet to render html
- ▶ Servlet to process form entry
- ▶ A basic web app

A Bit About Development Environments

- ▶ To stay organized, we create separate places for:
 - Source files (.java files)
 - Class files (.class files)
 - lib files (.jar files)
 - Configuration files(.xml, .properties)

Let's Build a Web Application

- ▶ Examine the view design (what's displayed in the browser)
- ▶ Create the development environment
- ▶ Create the deployment environment
- ▶ Perform behavior validation during design

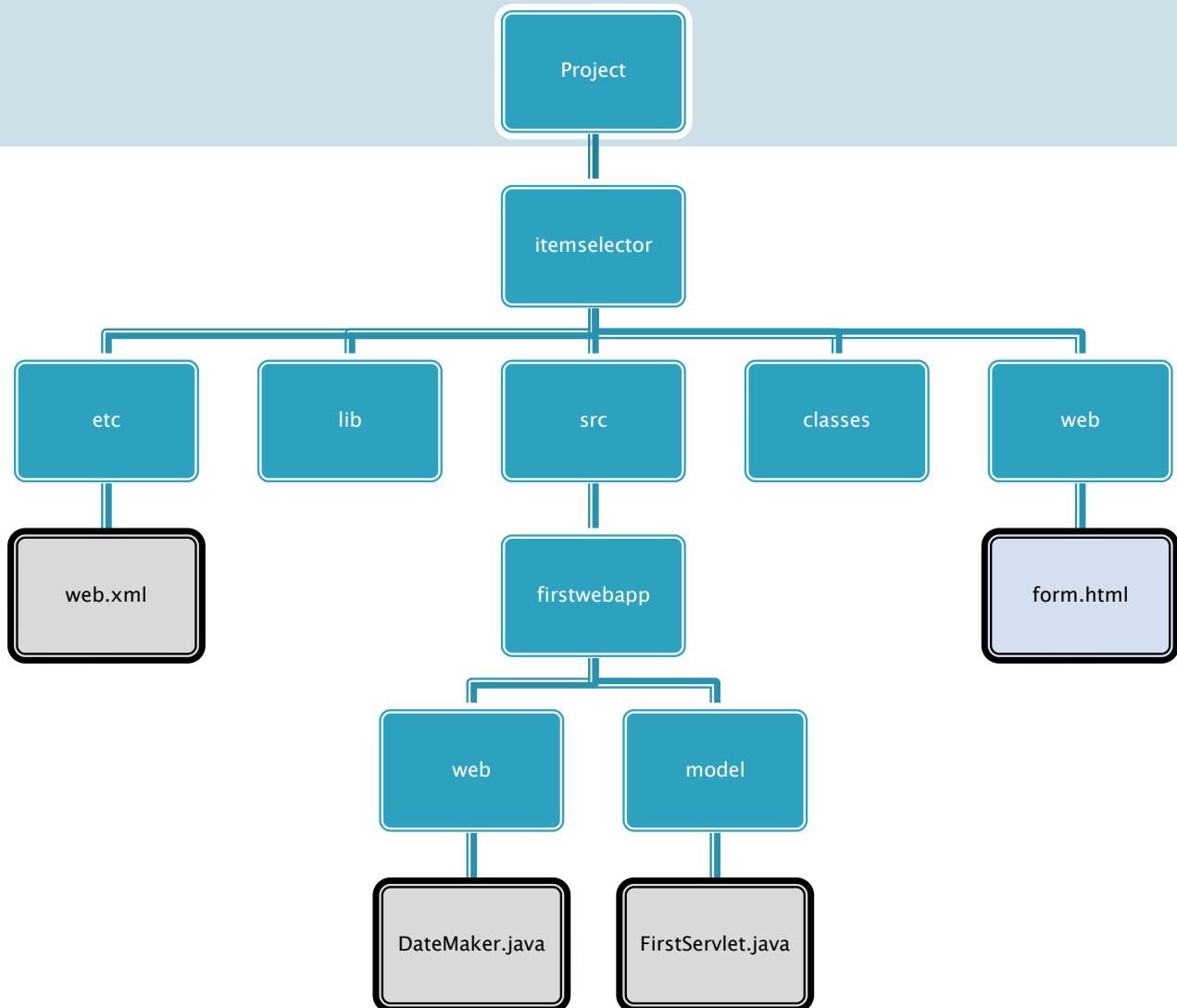
Application Behavior

- ▶ Read a parameter from the user's form
- ▶ Render html with the parameter and the current date and time

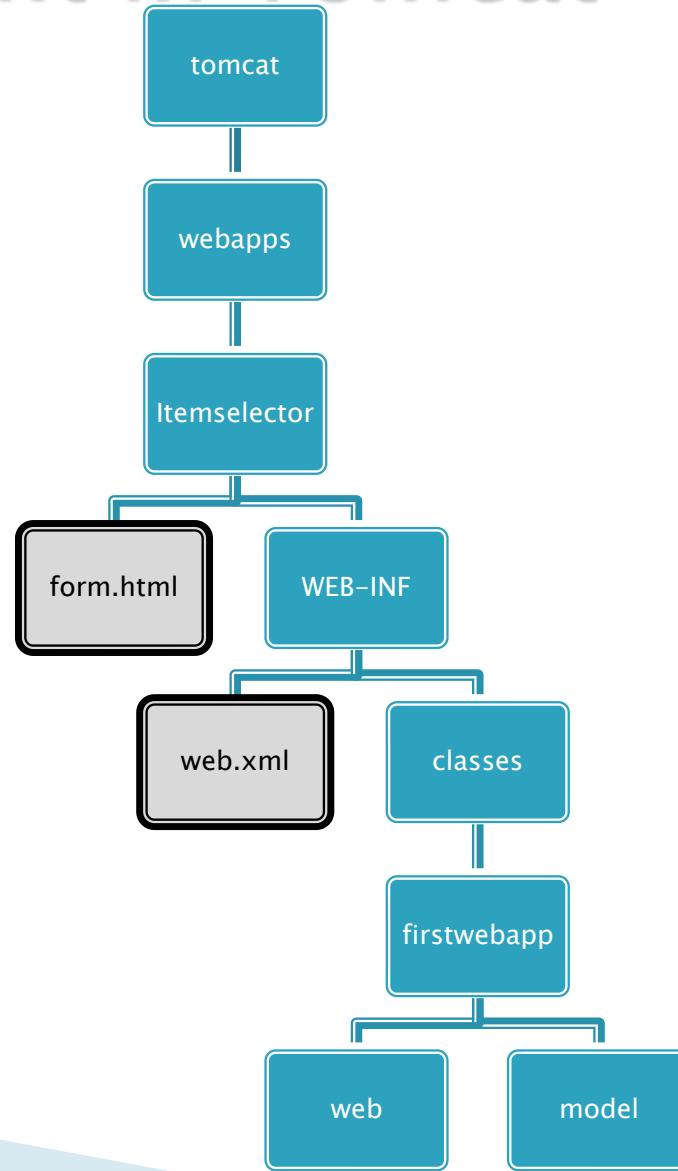


Create Development Environment

Project Root

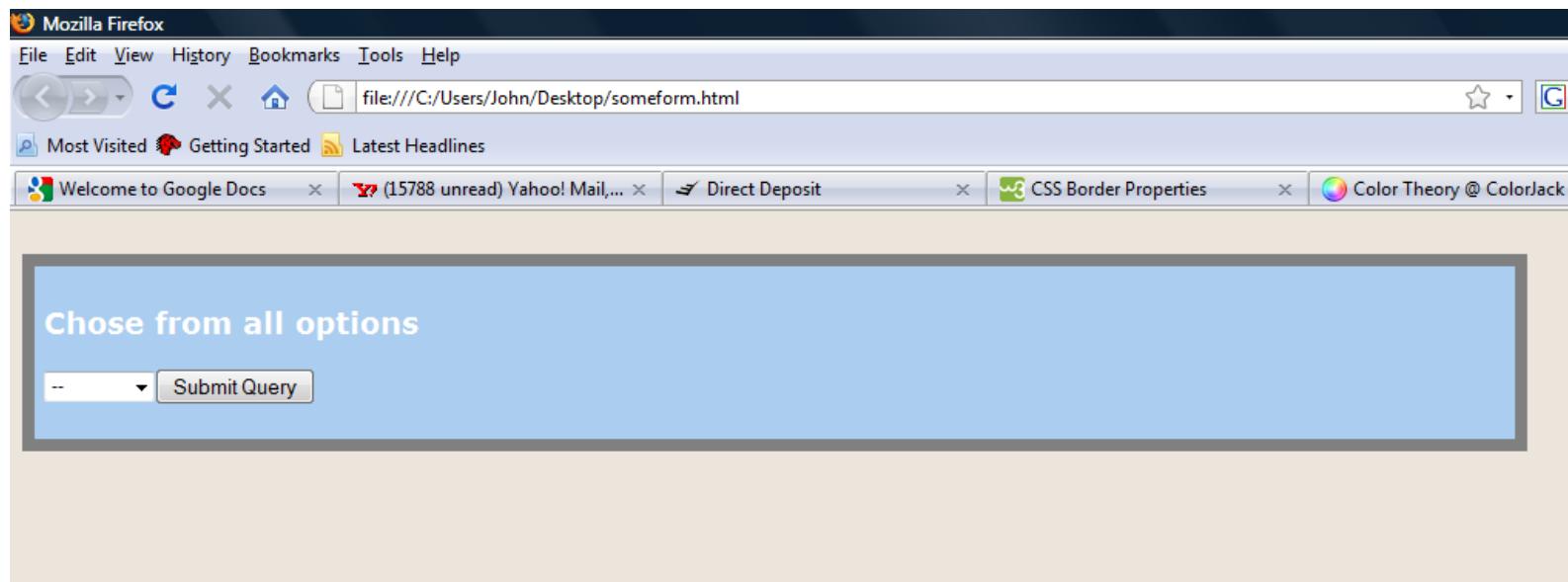


Create the Deployment Environment in Tomcat

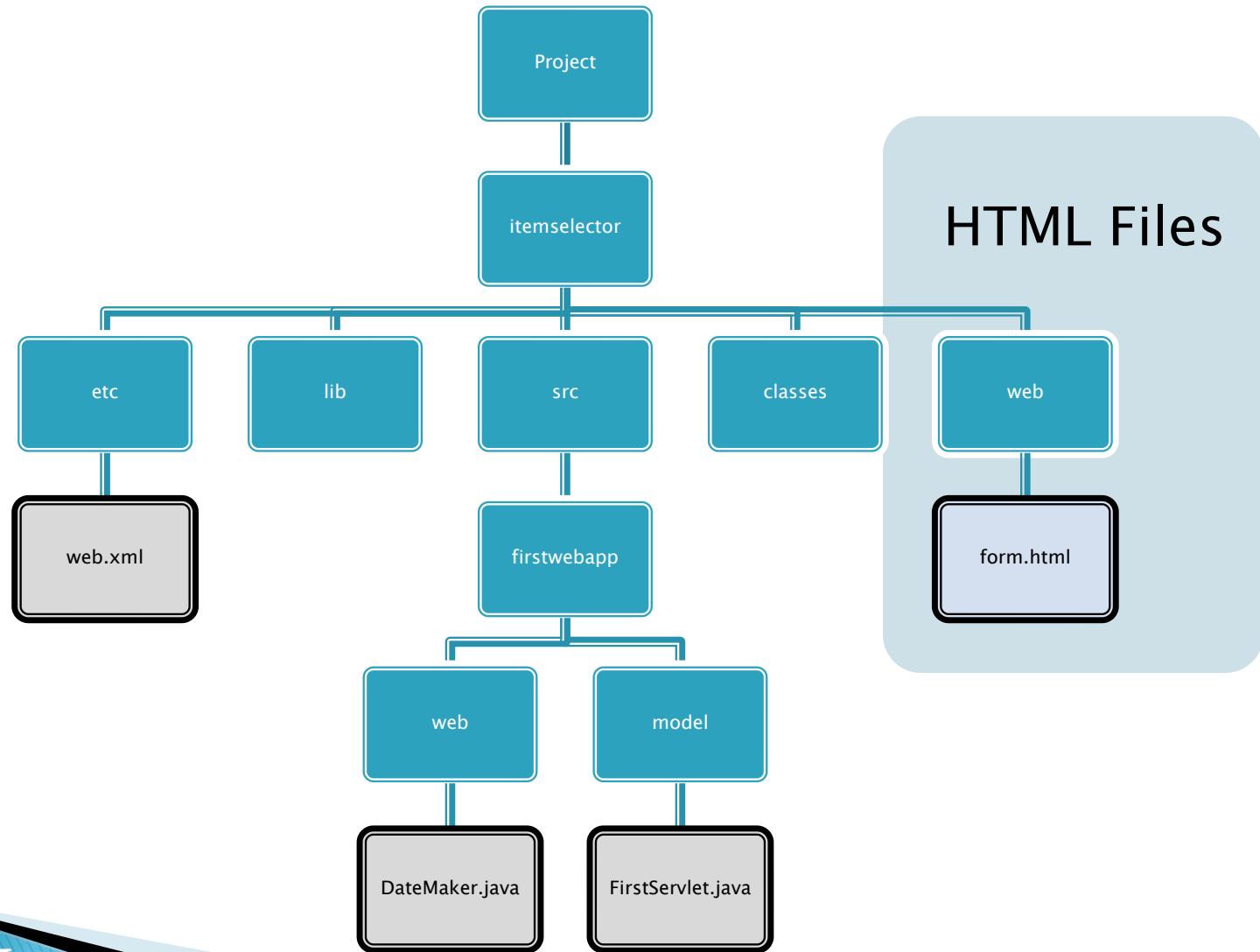


Views - form.html

- ▶ A simple form where the user makes a selection

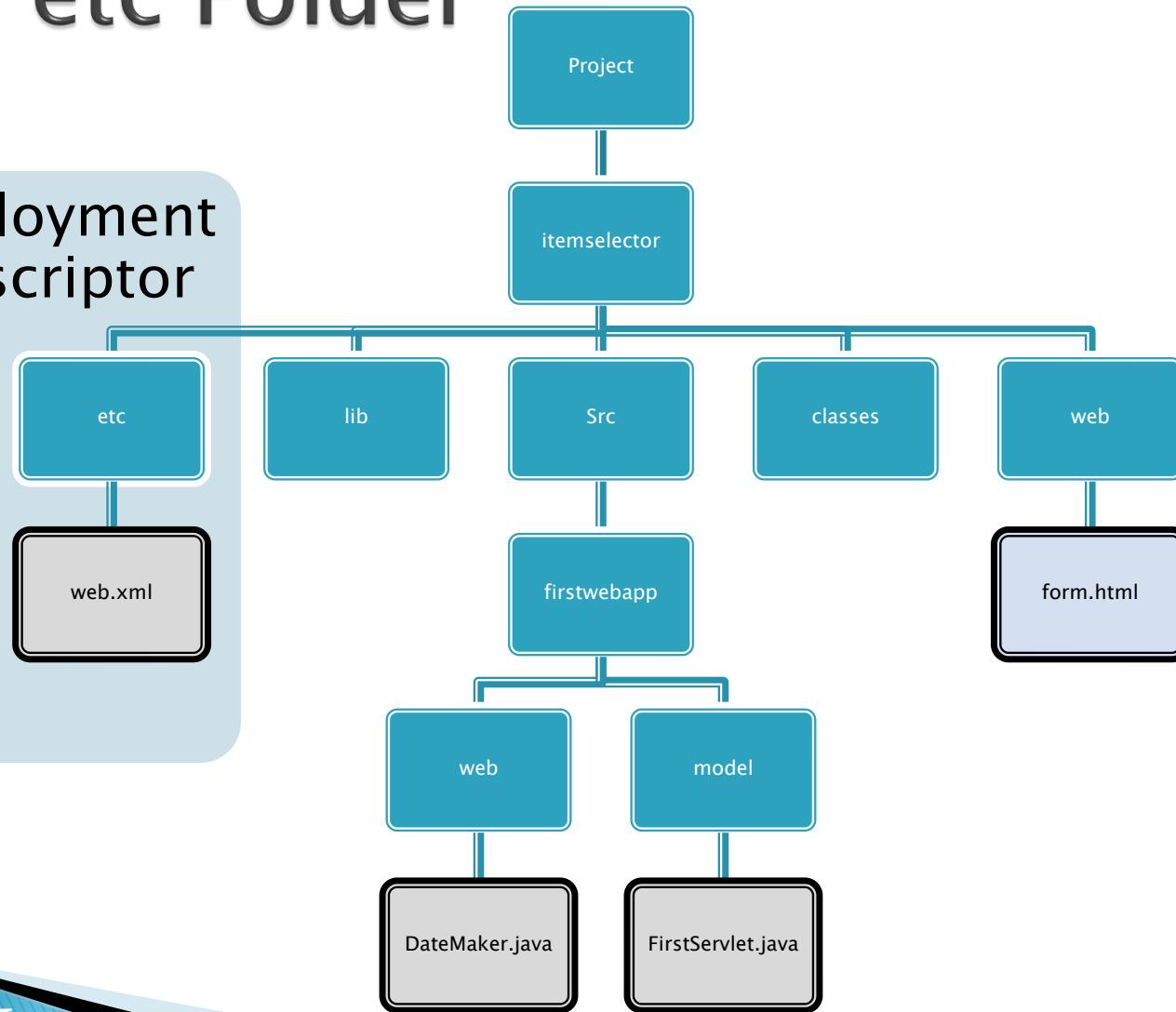


Create the HTML

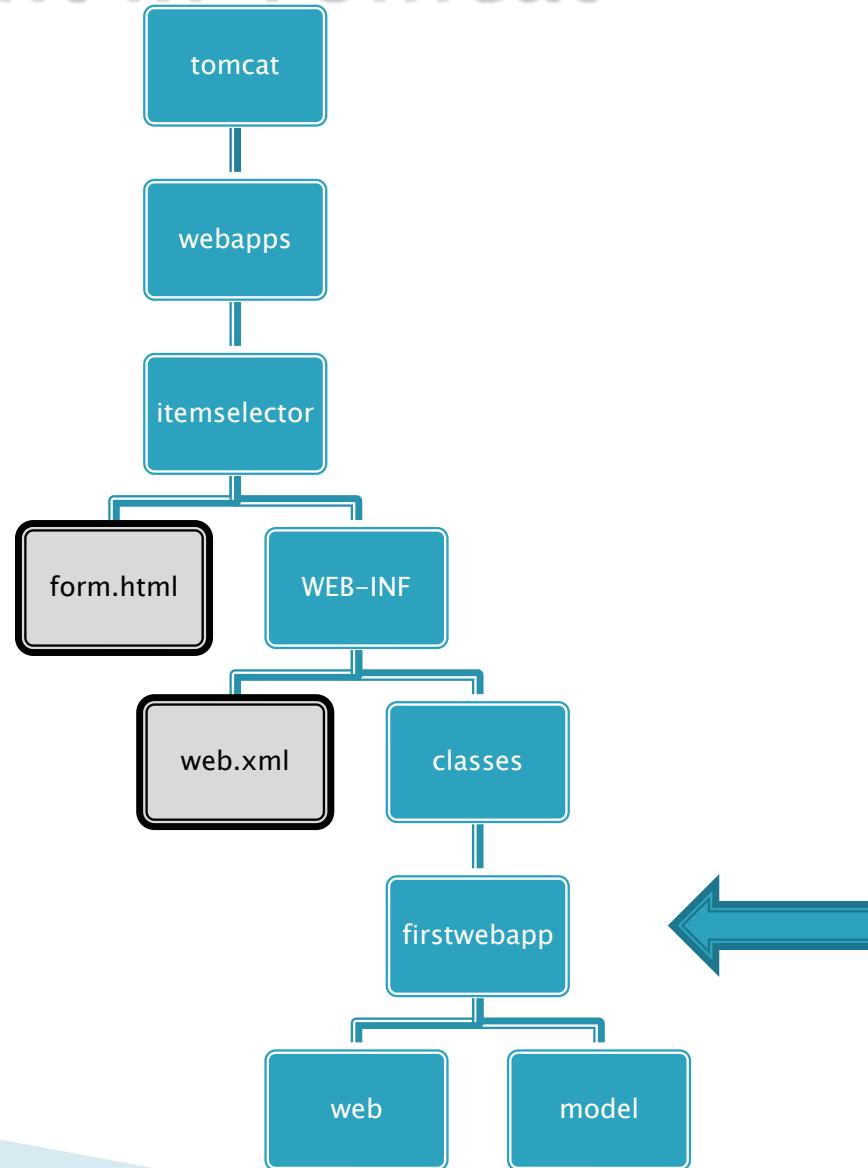


Put the Deployment Descriptor In the etc Folder

Deployment
Descriptor

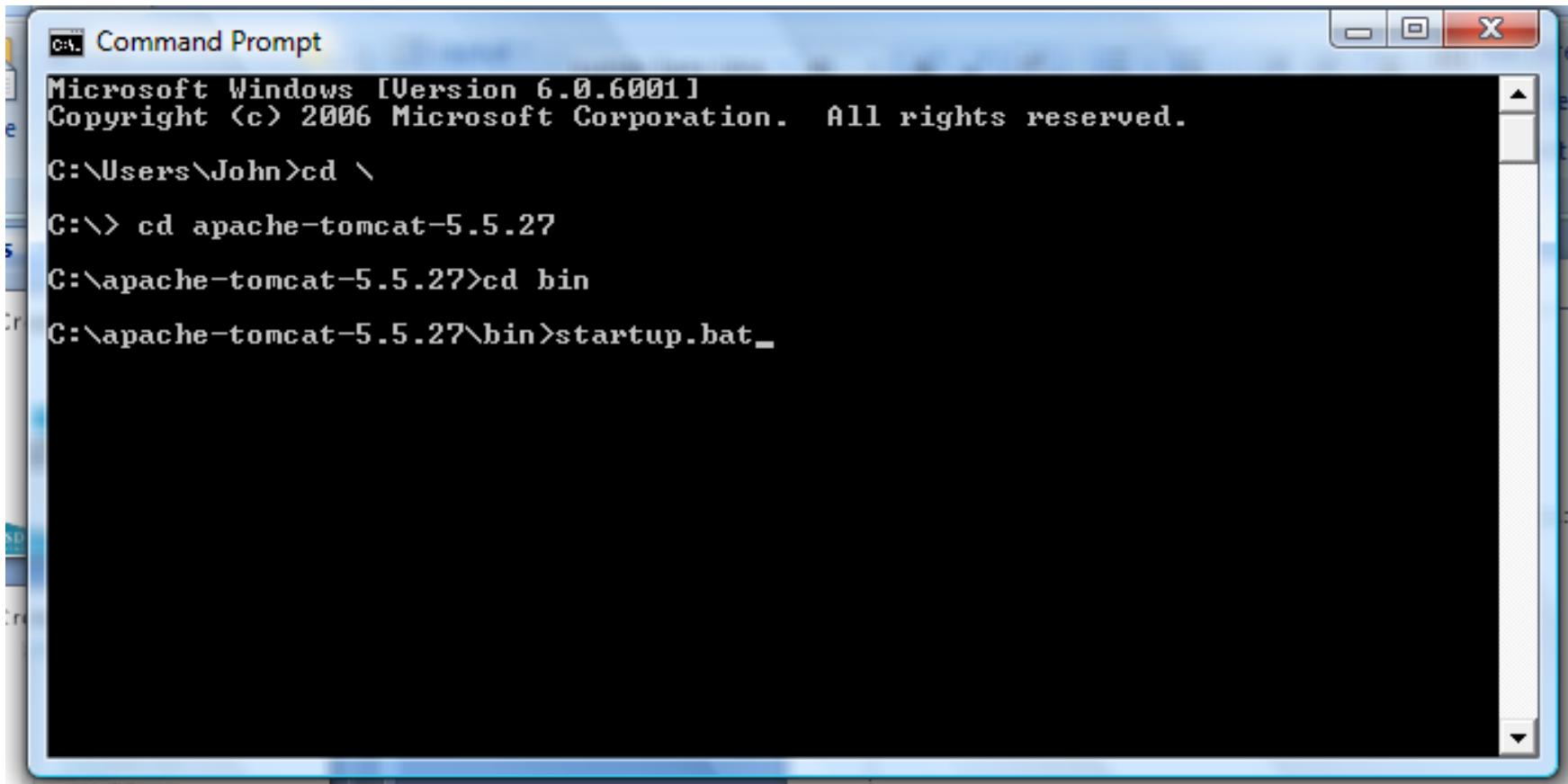


Create the Deployment Environment in Tomcat



Class files
Go here

Start Tomcat



A screenshot of a Microsoft Windows Command Prompt window titled "Command Prompt". The window shows the following text:

```
Microsoft Windows [Version 6.0.6001]
Copyright <c> 2006 Microsoft Corporation. All rights reserved.

C:\Users\John>cd \
C:\> cd apache-tomcat-5.5.27
C:\apache-tomcat-5.5.27>cd bin
C:\apache-tomcat-5.5.27\bin>startup.bat
```

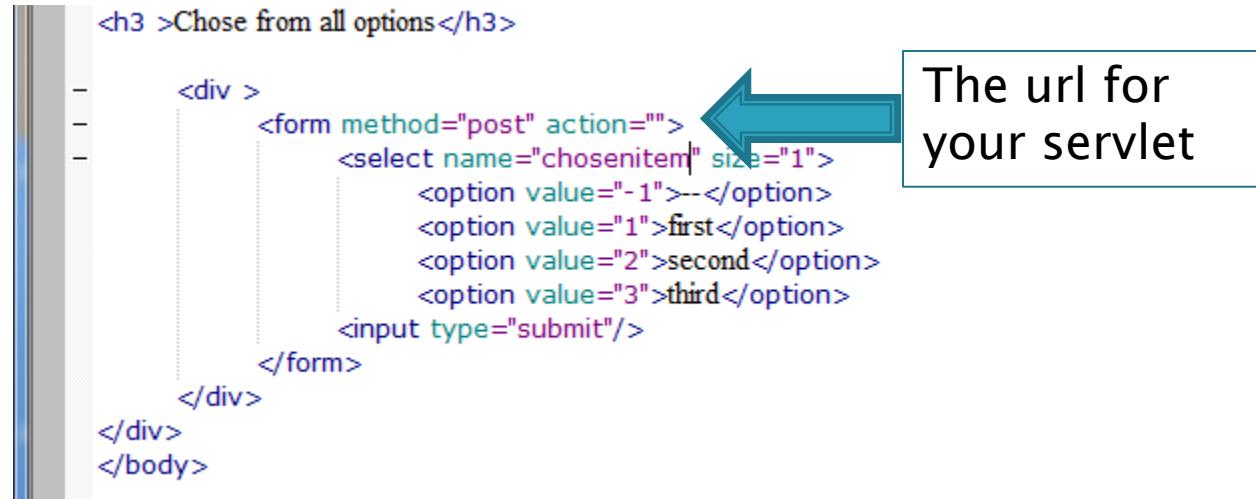
Find the html page in your browser

<http://localhost:8080/firstwebapp/form.html>

Create a Servlet

▶ Reads a request parameter

- `request.getParameter("chosenitem");`



▶ Returns the selected value and the current date and time

Find the HTML Page in Your Browser

- ▶ <http://localhost:8080/firstwebapp/form.html>
- ▶ Submit the form and view the response from your servlet

EHSDI
e B u z i m a