

CS601: Jetty. Servlets. Html Form Processing.

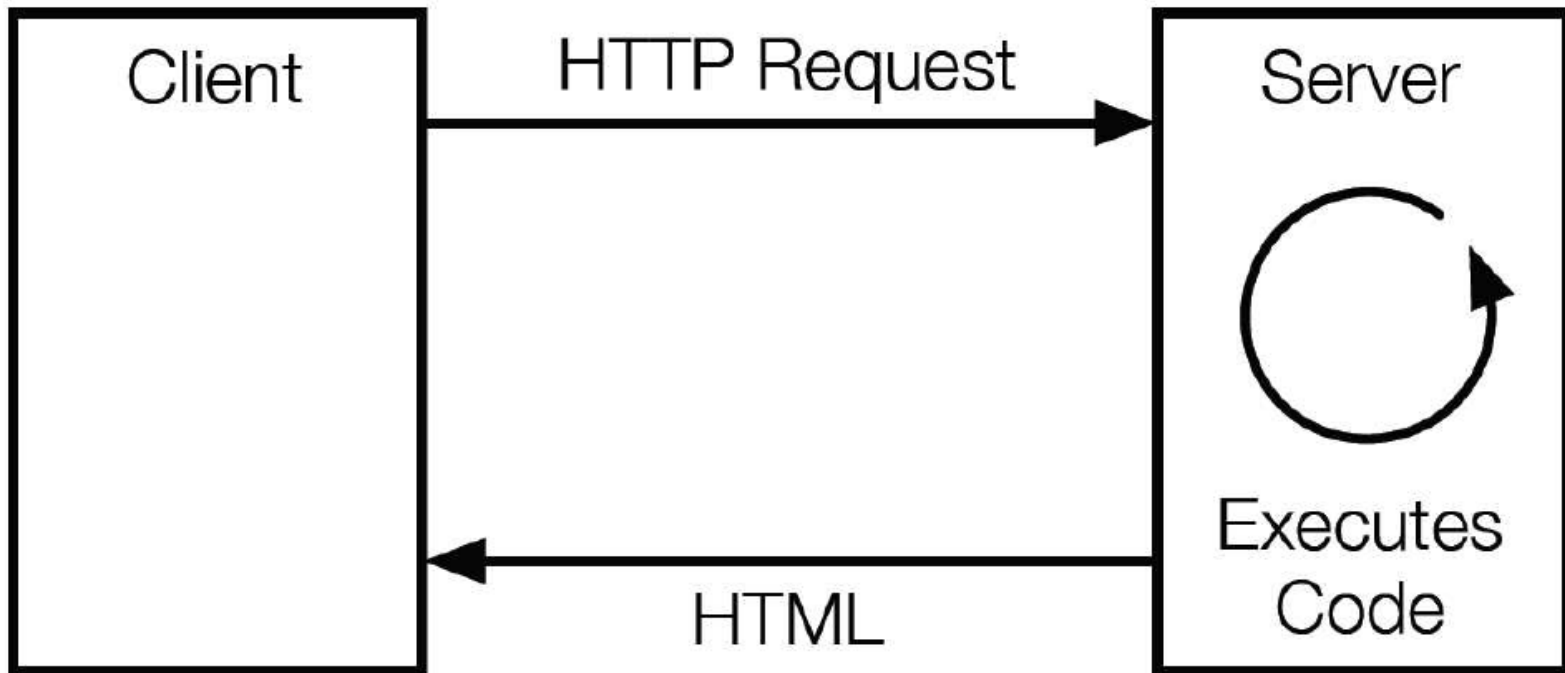
Olga Karpenko

This presentation is based on the course materials prepared by of Sophie Engle.

Announcements

- Lab 5 due on Wednesday
- Code review for all students
- Quiz on Wednesday (regex, Jetty/servlets)
- Feedback on the midterm?

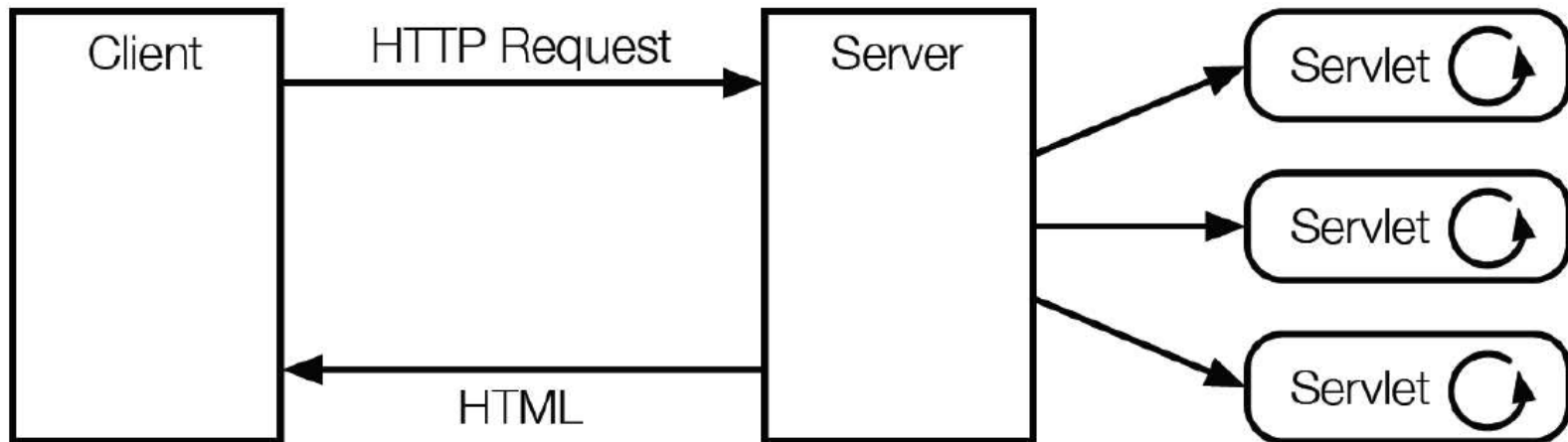
Server-Side Dynamic Webpages



Servlet

- Small Java program that runs on the webserver
- The output is a dynamic HTML page
- Servlets: Java package for servicing HTTP requests

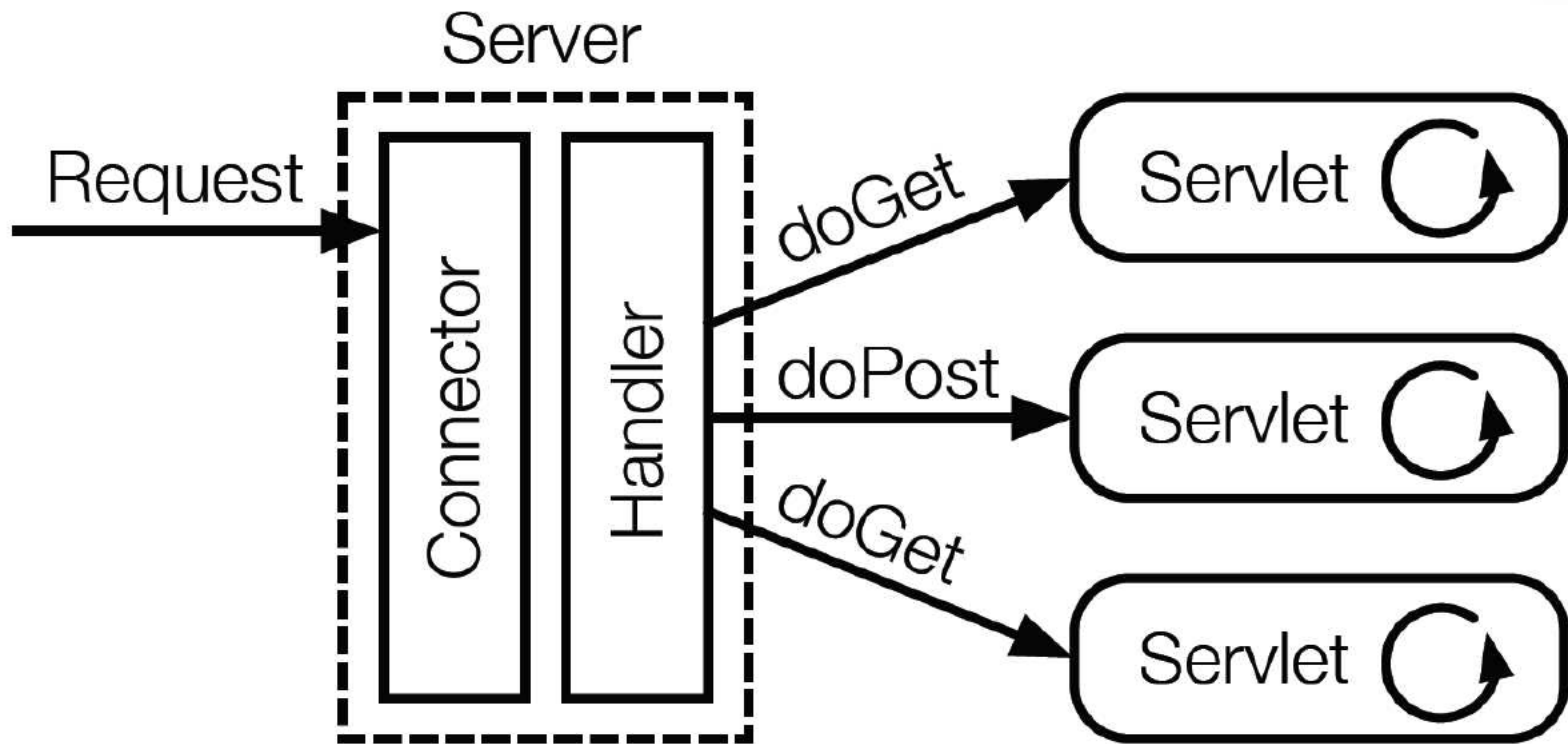
Server-Side Code Using Servlets



Jetty

- Embeddable, multithreaded web server in Java
- Includes a servlet handler

Server-Side Code Using Jetty



Servlet

- Invoked by going to a specific URL in the browser

Ex: When the user types

- `localhost:8080/hello` in the browser -> one servlet will be invoked
- `"localhost:8080/bye"` -> a different servlet will be invoked
- Overrides `doGet` or/and `doPost` methods to handle GET/POST requests

Example

- SimpleHelloServer / SimpleHelloServlet

Recap: GET Request

GET /search/?q=Zika HTTP/1.1

Host: www.cnn.com

Connection: close

Recap: GET Request

GET /courses?name1=value1&name2=value2
HTTP/1.1

Host: www.cooluniversity.edu

- In an HTTP GET request, parameters are sent as a query string in the URI

Recap: GET Request

```
GET /maps/api/place/textsearch/json?  
query=tourist%20attractions%20in  
%20Emeryville&location=37.837773,-122.2981  
42&radius=3218 HTTP/1.1
```

Host: maps.googleapis.com

Connection: close

- Key parameter removed in the example

Accessing Parameters

```
public void doGet(  
    HttpServletRequest request,  
    HttpServletResponse response)  
{  
  
    String name = request.getParameter("name");  
    // Returns the value of parameter "name"  
}
```

- Works for GET and POST methods

Escaping Characters

"bread" & "butter" needs to be replaced with:
"bread" & "butter";

- Can Use Apache Commons Lang library

XSS Attacks

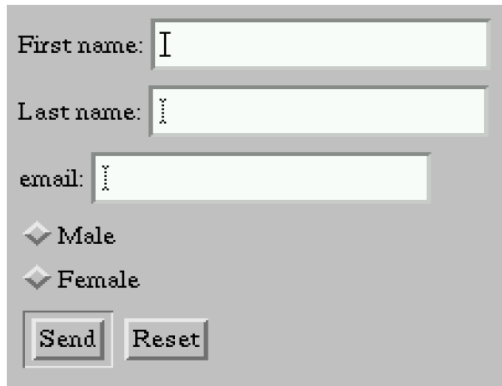
- Cross-site scripting attacks
- Malicious scripts are injected into html
- Will be run when client renders html

Example

- HelloServer/HelloServlet

Processing HTML Form Data

HTML Form Tag



First name:

Last name:

email:

☐ Male

☐ Female

- In Html: **<form params> input elements </form>**
- Input elements: text fields, checkboxes, radio-buttons, submit buttons etc.
- Allows users to submit data to the webserver

Input Elements

1) Text field:

First name: <input type="text" name="firstname">

2) Password field

Password <input type="password" name="pwd">

Attributes of Form

- action

Specifies where to send the form data once it's submitted

- method

- post or get

- Specifies the type of HTTP request when sending form data

Input Elements

3) Submit button example:

```
<form action="/register" method = "post">  
<input type="submit" value="Click here">  
</form>
```

- *doPost* method in a servlet mapped to a `/register` path will process this form

POST Requests

Recap: GET Request

```
GET /courses?name1=value1&name2=value2  
HTTP/1.1
```

```
Host: www.cooluniversity.edu
```

- In an HTTP GET request, parameters are sent as a query string in the URI

POST Request

POST /courses HTTP/1.1

Host: www.cs.usfca.edu

Content-Type: application/x-www-form-urlencoded

name1=value1

- Encodes data in the body of the request
- The parameters are **not** sent along with the URI

POST Request

POST /courses HTTP/1.1

Host: www.cs.usfca.edu

Content-Type: application/x-www-form-urlencoded

name1=value1&name2=value2

- Encodes data in the body of the request
- The parameters are **not** sent along with the URI

GET vs POST

- http://www.w3schools.com/tags/ref_httpmethods.asp

Reading Form Data in Servlets

```
public void doGet(  
    HttpServletRequest request,  
    HttpServletResponse response)  
{  
  
    String name = request.getParameter("name");  
    // Returns the value of parameter "name"  
}
```

- Works for GET and POST methods

Processing HTML Forms

Using servlets

- EchoServer / EchoServlet
- HtmlFormServer/HtmlFormServlet/
WelcomeServlet
- MessageServer/MessageServlet

Servlets

- Each Servlet is loaded and instantiated only once
- Reused every time the request URL matches the servlet's associated url-pattern.
- The servlets are shared among all requests
 - Servlet code needs to be threadsafe

Documentation

- Servlet Documentation

- <https://docs.oracle.com/javaee/7/api/javax/servlet/http/HttpServlet.html>

And other classes in javax.servlet package

- Jetty

- <http://www.eclipse.org/jetty/documentation/current/advanced-embedding.html>