

## Objectives

- Review: HTML Forms
- Intro to Java Server-side Web Technology

April 29, 2016

Sprengle - CS335

1

## Review: HTML Forms

- What attribute is required in a **form** tag?
  - What attribute is optional?
- What attribute do we use to create different types of **input**?
- How do we distinguish between input data?
- How do we “group” radio buttons and checkbox buttons?

April 25, 2013

Sprengle - CS335

2

## INTRODUCTION TO SERVER-SIDE PROGRAMMING

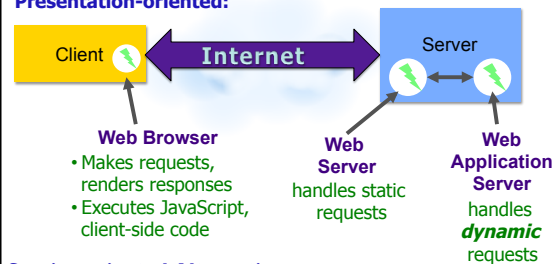
April 29, 2016

Sprengle - CS335

3

## Architecture of the Web

### Presentation-oriented:



### Service-oriented Alternative:

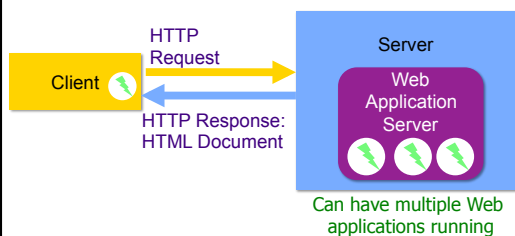
Client is another server/non-user computer

April 29, 2016

Sprengle - CS335

4

## Java-based Web Application Server



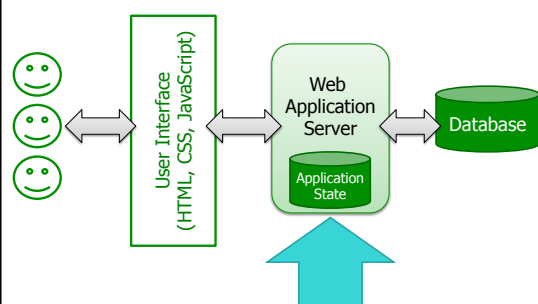
- Web Application Server
  - **Container** to run the Java-based web applications
  - Typically listens to port 8080 (rather than 80)

April 29, 2016

Sprengle - CS335

5

## Web Application Architecture

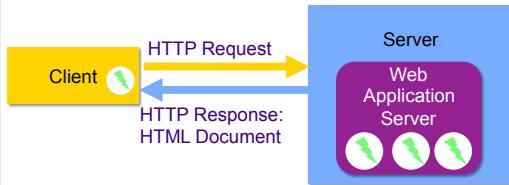


April 29, 2016

Sprengle - CS335

6

## Architecture of the Web



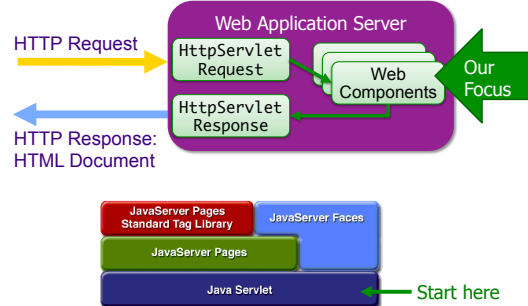
- **Web Application Server**
  - Parses request, including data
  - Executes request
  - Returns response (often an HTML document)
    - May do other things, like send email, ...

April 29, 2016

Sprengle - CS335

7

## Request Handling in Java



April 29, 2016

Sprengle - CS335

8

## Servlets

- A Java class that extends the functionality of web servers
  - Processes requests on server
  - Sends results (typically as an HTML file) back to client
- In `javax.servlet.*` packages
  - Part of Java Enterprise Edition (EE), as a separate download
  - Eclipse for EE development (Web Tools Platform)
- Java's answer to CGI (Common Gateway Interface)
- Portable, more secure (no buffer overflows)
- Supported by many major Web servers
  - E.g., Apache with Tomcat, Resin, etc.

April 29, 2016

Sprengle - CS335

9

## The Servlet Interface

Review from CSCI209: What is an *interface*?

- `javax.servlet.Servlet`
- All servlets implement the **Servlet** interface
  - `HttpServlet`, `GenericServlet`, `FacesServlet`
  - Web application server invokes many methods of **Servlet** automatically

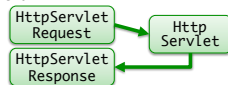
April 29, 2016

Sprengle - CS335

10

## The HttpServlet Class

- Web-based servlets typically extend **HttpServlet**
  - Implements **Servlet** interface
- **HttpServlet** implements the **service** method
  - **service** calls the respective method (e.g., `doGet` or `doPost`) in response to a HTTP GET or POST request
  - Parameters
    - **HttpServletRequest** - from the client
    - **HttpServletResponse** - to the client
- Recall:
  - **GET** - data encoded in URL
    - Request a resource (file) or retrieve data
  - **POST** - data encoded in body of message
    - Upload data; processing; hide data from URL



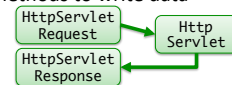
April 29, 2016

Sprengle - CS335

11

## HttpServletResponse

- Provides output streams and methods to write data back to the client



- **Methods:**
  - `ServletOutputStream` `getOutputStream()`
  - `PrintWriter` `getWriter()`
  - `void setContentType(String)`
    - Specifies the MIME type of the response
    - Browser knows what it received and how to format it
    - `"text/html"` specifies an HTML document

April 29, 2016

Sprengle - CS335

12

## HttpServletResponse methods

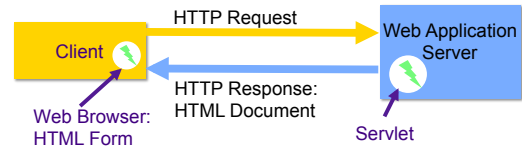
- **ServletOutputStream getOutputStream()**
  - Obtains a byte output stream that enables the servlet to send *binary data* to the client
- **PrintWriter getWriter()**
  - Obtains a text writer that enables the servlet to send *character data* (text) to the client
- **void setContentType(String)**
  - Specifies the MIME type of the response
    - Browser knows what it received and how to format it
  - "text/html" specifies an HTML document

April 29, 2016

Sprengle - CS335

13

## Example Communication with Servlet



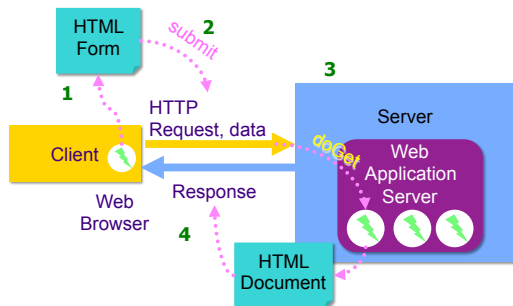
- HTML page with a basic form containing a submit button
  - To trigger client's request
- When the button is pressed, browser passes the servlet a basic GET request

April 29, 2016

Sprengle - CS335

14

## Example Servlet Flow



April 29, 2016

Sprengle - CS335

15

## To Generate a Response

- **doGet** method needs to
  - Obtain an output writer to write back to the client
  - Generate/write the HTML page to the client using the writer
  - Close the writer

How can we implement these steps?

```
void doGet(HttpServletRequest request,
            HttpServletResponse response)
```

April 29, 2016

Sprengle - CS335

16

## Accepting Certain Types of Requests

- We can design the servlet to **only** accept/handle GET requests
  - Override the **doGet** method
  - Could return an error inside of **doPost**

April 29, 2016

Sprengle - CS335

17

## HTTP Response Errors

- **HttpServletResponse** has a method for returning errors and fields that define errors codes

```
void sendError(int statusCode [,
                String msg])
```

- Example status code fields:
  - SC\_HTTP\_VERSION\_NOT\_SUPPORTED
  - SC\_METHOD\_NOT\_ALLOWED
  - SC\_NOT\_IMPLEMENTED

April 29, 2016

Sprengle - CS335

18

Tomcat  
Eclipse/Tomcat  
Using Eclipse  
HTML, CSS  
Start servlets

## IN-CLASS WORK

April 29, 2016 Sprengle - CS335 19

## Handling Data

- So far, we haven't done anything with data that comes with the request

**Requests for a digital publication library:**  
GET dspace/search?search=xxx&sort=date&title=Title

- Data is stored in a hash table

```

graph LR
    title["title"] --> Title["Title"]
    sort["sort"] --> date["date"]
    search["search"] --> xxx["xxx"]
  
```

April 29, 2016 Sprengle - CS335 20

## HttpServletRequest

```

graph TD
    Request[HttpServletRequest] --> Servlet[Http Servlet]
    Servlet --> Response[HttpServletResponse]
  
```

- Provides input streams and methods to read data from the client
- Methods:
  - > **String** `getParameter(String paramname)`
    - Returns the value of a request parameter
    - > null if parameter doesn't exist
  - > **String[]** `getParameterValues(String paramname)`
    - Returns an array of Strings containing the values for a specific request parameter
  - > **Enumeration<String>** `getParameterNames()`
    - Returns the names of all of the parameters passed in the request

April 29, 2016 Sprengle - CS335 21

## HttpServletRequest Methods

**Requests for a digital publication library:**  
GET dspace/simple-search?search=xxx&sort=date&title=Title

- `getParameter("title")`
  - > Returns "Title"
- `getParameterValues("sort")`
  - > Returns [ "date" ]
- `Enumeration<String> getParameterNames()`
  - > Returns the names of all of the parameters passed to the servlet

Note these are Strings

April 29, 2016 Sprengle - CS335 22

## A More Complex Example: Survey

- An HTML form that asks the user for their favorite type of pet
- After user submits the form, the server sends back the current results of the survey
- Uses object serialization to write to/read from file

April 29, 2016 Sprengle - CS335 23

## Deployment: WAR files

- Web Archives
  - > Analog to JAR files
  - > Bundles together all the code, files for the web application
- Copy into webapps directory of web application server
  - > Server will automatically extract files and run
  - > Procedure for Apache/Tomcat and Resin
- Can export WAR files from Eclipse

April 29, 2016 Sprengle - CS335 24

## TODO

- Complete Lab 3
  - Due tonight at midnight
- Static Mockups due Monday at midnight
  - Pick 3 “key” pages to mock up
    - Basis for other pages
- Read “Quality Attributes of Web Software Applications”
  - See Course Schedule for link
  - Writeup on Sakai
  - Due Wednesday @ midnight

April 29, 2016

Sprenkle - CS335

25