

CMSC335

Web Application Development with JavaScript



Web Servers, Forms

Department of Computer Science

University of MD, College Park

Slides material developed by Ilchul Yoon, Nelson Padua-Perez

Web Servers

- Web server - computer program that delivers (serves up) web pages
- It is like a person that is in charge of a warehouse
- It can return static html files it already stores
- It can create HTML based on a request and return it to the browser
- It can retrieve results from a database, format results in HTML, and return the results
 - Languages like PHP can access the database and create HTML that the web server returns

Web Servers

- Popular Web Server Programs
 - Apache → <http://www.apache.org/> Free!!!
 - Internet Information Services (IIS) - <https://www.iis.net/>
 - Nginx - <https://www.nginx.com/> (open source)
- You can install and run a web server in your computer
 - <http://www.apachefriends.org/en/xampp.html>
 - XAMPP - includes Apache, MariaDB, PHP, OpenSSL, etc.
- Web server local address: <http://localhost> or <http://127.0.0.1/>
- htdocs – location of documents in Apache
 - Windows → C:\xampp\htdocs
 - Mac → /Applications/XAMPP/htdocs
- **Example:** JavaToyWebServer.zip

htdocs

- You can define your own folders in htdocs
- Renaming index.php (e.g., indexOld.php) will list **htdocs** content
- **Example:** WebServersFormsCode/SamplePage.html
 - To run the example you need to start the web server, place the code distribution under the htdocs folder and open a browser window with the code. Here is a sample link:
<http://127.0.0.1/WebServersFormsCode/SamplePage.html>

htdocs

- **Example:** Let's create a user location (web page repository)
 - An **index.html** file in a folder represents the main index page for that folder
 - You will not see the contents of a folder if an index.html is present
 - » You need to type the file/folder name to see a particular file/folder in the folder
 - Which file represents the main index page can be defined in the Apache configuration file

Server Side Includes

- SSI - Server Side Includes is an interpreted server-side scripting language. **Common use is file inclusion.** This will prevent code duplication
- File Inclusion (following example includes **footer.html**)

```
<!--#include virtual="footer.html" -->
```
- For a web server to recognize an SSI-enabled HTML file the filename should end with a special extension, by default .shtml, .stm, .shtm (or by configuring the web server)
- **Example:** ServerSidesIncludesExample
 - After running the example, view the HTML code displayed (“View page source” option in the browser)
- **Example:** ServerSidesIncludesExample/modifiedExtension.html
 - Shows impact of file extension (notice it is not .shtml)

Server Side Includes

- [Apache](#), [LiteSpeed](#), [nginx](#), [lighttpd](#) and [IIS](#) are the five major web servers that support server side includes.
- **Reference:** Wikipedia
 - https://en.wikipedia.org/wiki/Server_Side_Includes
- Additional directives
 - **Timestamp for file modification**
`<!--#echo var="LAST_MODIFIED" -->`
 - **Timestamp for local date/time**
`<!--#echo var="DATE_LOCAL" -->`

Server Processing

- Server processing (back-end) can be supported by several languages
- Note: HTML/CSS/JavaScript → front-end
 - Although Node.js (that relies on JavaScript) is back-end
- One of them is PHP
 - Server-side, cross-platform, HTML Embedded scripting language
 - Text files with .php extension
- What does it allow us to do?
 - To dynamically generate HTML
 - To interact with other systems (e.g., DB Systems, File Systems)
- **Examples:**
 - Flight Information
 - Application System

HTML Forms

- Forms - means by which information passes from the user to the server. It is a functionality provided by HTML
- **<form></form>** - Includes elements that allows user to provide data
- **<input>** → Appears inside of the **<form>** tag and defines several input data alternatives. The general format is:

<input type="ALTERNATIVE" />

where **ALTERNATIVE** can be text, password, checkbox, radio, file, submit, image, button, reset, hidden

HTML Forms

- `<form>` → Defines the form. It has two attributes:
 - **action** → Indicates where the form contents will be sent when the form is submitted. It represents a script/program that will process the data
 - **method** → Defines how the contents will be sent
 - » **post** → Contents sent using the HTTP POST method.
Content is “hidden”
 - » **get** → Contents sent using the HTTP GET method.
Contents included in the URL. Parameters start after question mark (?) and are separated by &
- By using the `<input type="submit">` data in the form is sent to the resource specified by the action attribute
- **Example:** GetExample
- **Example:** PostExample

About Post and Get

- **HTTP get**

- All form information appears in the URL
- A get request can be bookmarked
- There is a limit of data (parameters) as URL has a size limit
- Not good for security as information available in the URL
- Intended for search operations that do not change the state of the server. **Examples of changing state:** adding/removing entry from a database

- **HTTP post**

- All form information sent in the body of the message
- A post request can NOT be bookmarked
- Better for security as information is not in the URL
- **Intended for operations that change the state of the server**
- Try reloading the script associated with the **PostExample**
 - » You get a warning about the impact of executing the script

Some PHP to understand scripts

- You are not responsible for knowing PHP, but some basic knowledge can help you write some useful scripts
- PHP code is enclosed in `<?php PHPCODE HERE ?>`
- PHP variable names start with \$ and can be followed by _ (underscore), letter, number
- We can use = operator to assign values to variables
- **Superglobal Variables** - variables that are always present and whose values are available to all your scripts. Each variable is actually an array of other variables
- **Some Superglobal Variables**
 - \$_GET - variables provided to a script through the GET method
 - \$_POST - variables provided to a script through the POST method
 - \$_COOKIE - variables provided to a script via a cookie
 - \$_FILES - variables provided to a script through file uploads
- **echo/print** - Used to generate HTML that is sent to the browser
 - E.g., echo “Hello”
- **isset** - whether a variable has been set

Form Examples

- **Example:** FormsSummary
- **Example:** SearchGoogleYouTube.html