JAVASCRIPT STORAGE

Cookies and Dom Storage

COS216
AVINASH SINGH
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF PRETORIA

LOCAL STORAGE – OVERVIEW

- Data can be stored on the server (sessions, files, database)
  - Requires the client to send a request to the server in order to get the data
- Sometimes you want to store data directly on the client's machine
  - Cannot be stored in files, since websites (HTML and JS) are not allowed to write to disk due to security reasons
- Data can be stored locally
  - Cookies (old limited method)
  - DOM Storage (new less limited method)

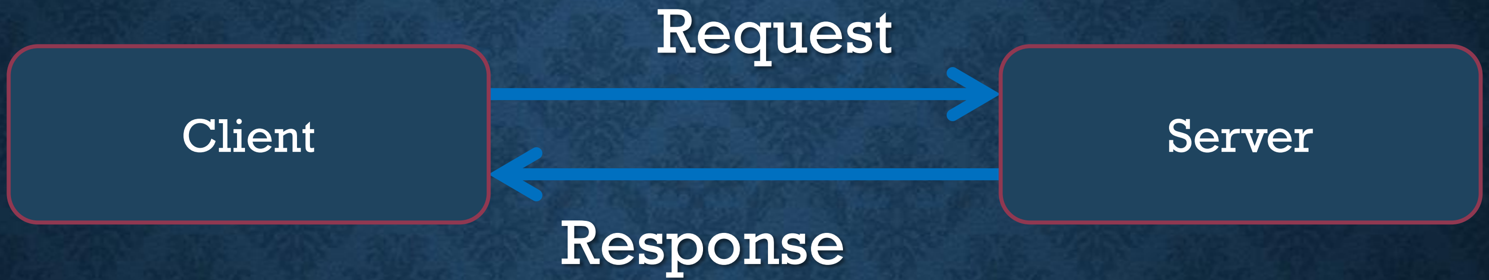
COOKIES – OVERVIEW

- A cookie is a variable that is stored by the client's browser for future use by a website
- The cookie can be used locally, or the value can be send to the server
- Name comes from "magic cookie" used by Unix programmers, describing a packet of data that is submitted unchanged
- A cookie consists of
  - **A name (required)**
  - **A value (required)**
  - **An expiration date (optional)**

COOKIES – OVERVIEW

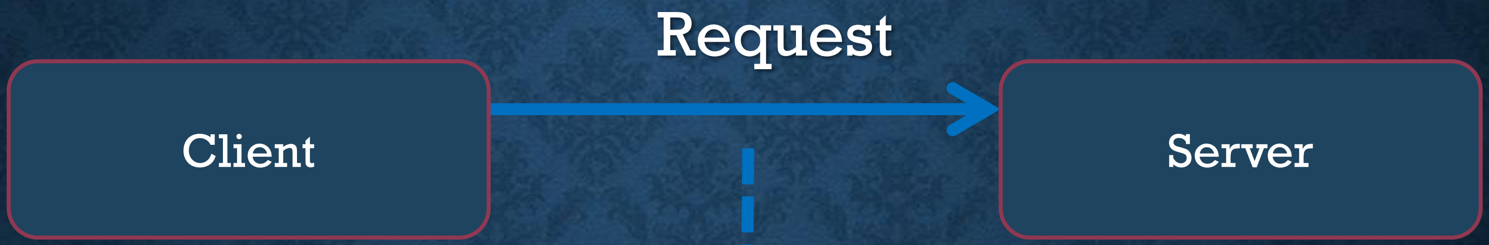
- Use cookies for
    - Session management
        - Example: Keep user logged in
        - PHP sessions store the PHP session ID as a cookie
- Personalisation
    - Example: Change theme for website
- User tracking
    - Example: Track user actions

COOKIES – SERVER COMMUNICATION



HTTP/1.1 200 OK
    Content-type: text/html
    Set-Cookie: name=value; Expires=Mon, 18-Mar-2013 09:20:00 GMT+2

COOKIES – SERVER COMMUNICATION



```
GET /login.html HTTP/1.1
    Host: www.server.com
    Cookie: name=value
```

COOKIES – CODE

- Write a custom generic function to set a cookie

```
function setCookie(name, value, lifetime)
{
        // Set name and value
        // Set the expiration date
        // Set the cookie
}
```

COOKIES – CODE

- Setting the name and value

```
function setCookie(name, value, lifetime)
{
        var result = name + "=" + escape(value);
}
```

COOKIES – CODE

- Certain characters are reserved by the syntax
- Replacements that conform to the syntax must be used instead
- Example:
  - File "happy tree friends.zip" on server "tree.com"
  - Escaped URL: http://tree.com/happy%20tree%20friends.zip

COOKIES – CODE

- Setting the expiration date (in days)

```
function setCookie(name, value, lifetime)
{
        var result = name + "=" + escape(value);
        if(lifetime != null)
        {
                var expiry = new Date();
                expiry.setDate(expiry.getDate() + lifetime);
                result += ";expires=";
                result += expiry.toUTCString();
        }
}
```

COOKIES – CODE

- Setting the actual cookie

```
function setCookie(name, value, lifetime)
{
        var result = name + "=" + escape(value);
        if(lifetime != null)
        {
                var expiry = new Date();
                expiry.setDate(expiry.getDate() + lifetime);
                result += ";expires=";
                result += expiry.toUTCString();
        }
        document.cookie = result;
}
```

COOKIES – CODE

- Set a cookie that holds the username for 7 days

```
setCookie("username", "lennon", 7);
```

COOKIES – CODE

- Retrieving a cookie that was previously saved

```
function getCookie(name)
{
        // Get all cookies
        // Look for cookie with the same name
        // Return the cookie value
}
```

COOKIES – CODE

- Retrieving all stored cookies

```
function getCookie(name)
{
        var cookies = document.cookie.split(";");
}
```

COOKIES – CODE

- Find the right cookie

```
function getCookie(name)
{
    var cookies = document.cookie.split(";");
    var cookie;
    for(var i = 0; i < cookies.length; ++i)
    {
        cookie = cookies[i].trim();
        if(cookie.indexOf(name) == 0)
        {
            var value = cookie.substring(name.length+1, cookie.length);
            return unescape(value);
        }
    }
    return "";
}
```

COOKIES – CODE

- Retrieve the username previously stored as a cookie

```
var user = getCookie("username");
// user = "lennon"
```

COOKIES – BROWSERS

- Note that most modern browsers do not store cookies for local files
- If you open a local file (file:///…) the browser will ignore cookies
- The website URL must start with http:// or https:// for the browser to store cookies

COOKIES – DRAWBACKS

- Very cumbersome to store and retrieve cookies, requires a lot of manual code
- Not very secure, due to cookie stealing
  - No authentication needed to access cookies
  - Some browsers now block access to cookies from a different domain
- Limited storage size of only 4KB
  - Not the best to store binary data, like profile pictures
- Can only store strings, no structured data
  - Structured data can be stringified and stored as a JSON string

DOM STORAGE – OVERVIEW

- Due to cookie limitations, DOM storage was introduced
- Originally part of the HTML5 standard
  - W3C now made it a separate specification
- Very similar to cookies, used to store data locally

DOM STORAGE – OVERVIEW

Session Storage

Separate storage for each page

Data is cleared when page/tab is closed

Local Storage

Same as session storage

However, data is stored permanently

Persistent even if tab or entire browser is closed

DOM STORAGE – BENEFITS

- Larger storage size
  - Between 5MB and 25MB, depending on the browser
- Temporary storage
  - Offers temporary and permanent data through the session and local storage
- Easy to use
  - Data can be stored and retrieved with a single line of code
- Structured data
  - Currently only strings can be stored
  - The DOM storage specifications plan to support the storage of structured data
  - Hence, objects and arrays can be stored without JSON stringification

DOM STORAGE – CODE

- Local storage (keeps data forever)

```
// Store value
localStorage.setItem('key', 'value');

// Retrieve value
var value = localStorage.getItem('key');
```

DOM STORAGE – CODE

- Session storage (keeps data until browser on window is closed)

```
// Store value
sessionStorage.setItem('key', 'value');

// Retrieve value
var value = sessionStorage.getItem('key');
```

DOM STORAGE – CODE

- Store structured data as JSON strings

```
// Store value
localStorage.setItem('key', JSON.stringify({"number": 1234}));

// Retrieve value
var value = JSON.parse(localStorage.getItem('key'));
```

WWW