

HTTP Cookies & User Authentication

AGENDA

HTTP is Stateless

Cookies

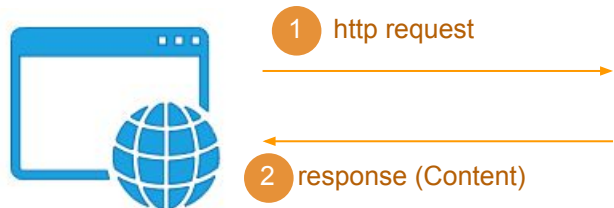
Code Demo Language Settings

Code Demo Login Flow

HTTP is Stateless

HTTP is Stateless

Web Browser (Client)



Web Server

- The server doesn't remember any info about who's making the request
- The server process every request like a new request

HTTP is Stateless

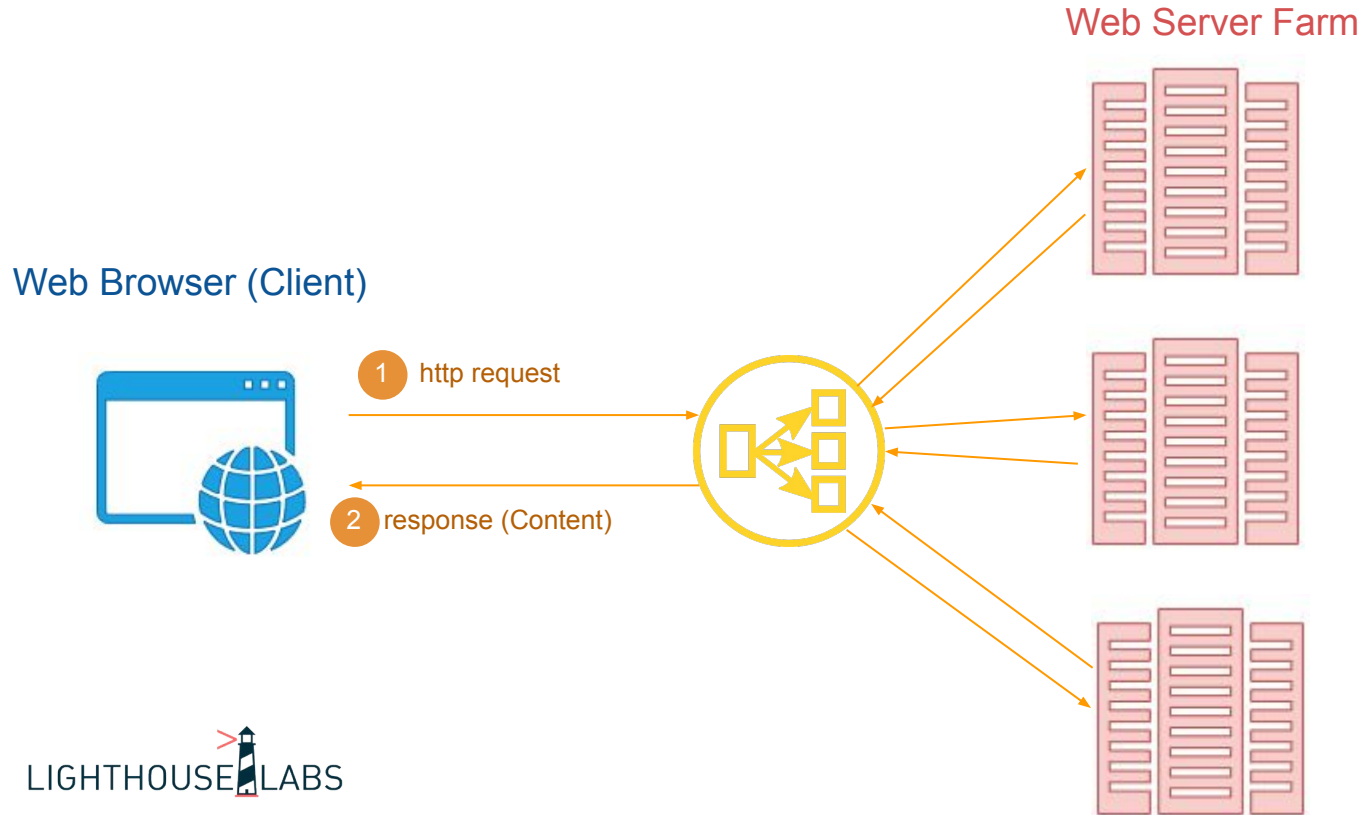
Pros

- Scalability - no session related dependency
- Each Server is Less complex
- Easier to cache
- The server cannot lose track of information

Cons

- Cannot easily keep track context
- Context has to be provided each time

Scalability



Browsers Store Cookies

Browsers Store Cookies

- When we say HTTP is stateless, we mean *the server doesn't store Cookie information*
- Browsers, however, are stateful - *the browser stores all of the Cookie information*
- The Server can tell the browser to remember a Cookie with the `Set-Cookie` header
- **The Browser** can set cookies at any time, because it **is the Cookie Master**



Anatomy of HTTP requests

method path protocol specification

POST /login HTTP/1.1
Host: lighthouselabs.ca
User-Agent: Mozilla/5.0
Accept: text/html

} headers

-VGhlIHF1aWNrIGJyb3duIGZveA==
username: idbentley
password: callyourmother
-VGhlIHF1aWNrIGJyb3duIGZveA==

} body

Anatomy of HTTP response

protocol specification

response status

```
HTTP/1.1 200 OK
content-length: 1500
content-type: text/html
Set-Cookie: UserID=123
```

headers

```
<html>
  <head>
  ...
  </body>
</html>
```

body

Cookie Header

- The **Set-Cookie** header is read by the browser
 - The browser will store this cookie information until expiration or deletion
 - All subsequent requests will contain a matching **Cookie** header

```
GET /profile HTTP/1.1
Host: lighthouselabs.ca
User-Agent: Mozilla/5.0
Accept: text/html
Cookie: UserId=123
```

Session Cookies

Session Cookies

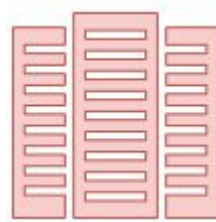
- Session Cookies are cookies that allow the server to identify the user
- Session Cookies contain an identifier that the server can validate
- The identifier can be used to look up the current user

User Login

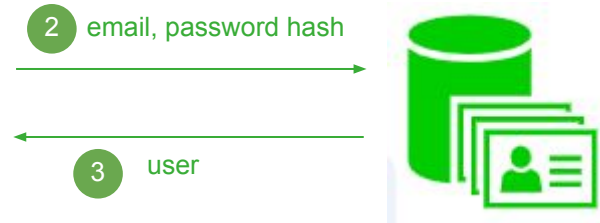
Web Browser (Client)



Web Server



Database



- Cookies are stored on the users' browsers
- Cookies are sent to the server with each request

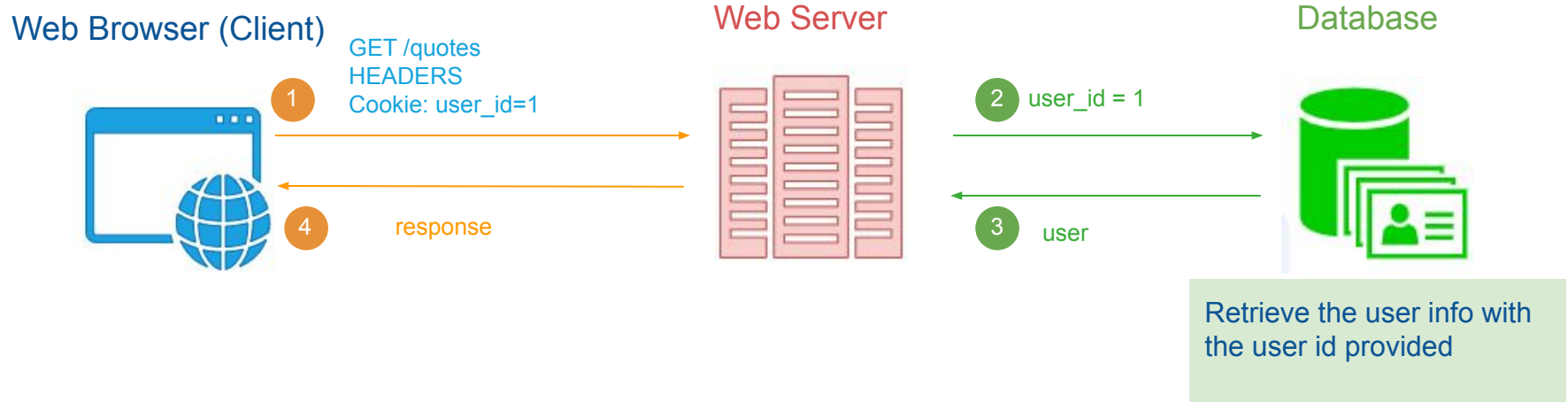
User Authentication



`setCookie(user_id, 1)`

Retrieve the user info for an account with a matching email and password

Logged In State



Cookies are sent with **every request** as request headers

CODE DEMOS

- Language Setting
- Login Flow

Questions?