10. HTTP Cookies & User Authentication

HTTP Cookies & User Authentication

Github repo

HTTP

- HTTP is stateless - the connection between client and server stops once the response has been sent.
- Next time you ask a server for something, the server doesn't actually know who you are or remember that you asked it for something seconds before.
- request/response cycle
- request contains a verb and a path (GET /urls)
- response must contain a status code; may or may not contain a body

Cookies

- Allow us to store information about a user between HTTP requests
- Stored as key/value pairs in the client's browser
- Are passed to the server with every HTTP request
- Usually used to store unique value that identifies a particular user
- Domain specific ip address/port

Reading Cookies

- Cookies come in with the request
- We could parse the request header ourselves, but it's easier to use a library like `cookie-parser`
- `cookie-parser` will parse the cookies and add them to the `request` object

```
app.get("/protected", (req, res) => {
  const userId = req.cookies.userId;
  // do something with the userId
});
```

Setting Cookies

- Cookies are set on the `response` object

- The browser will receive the reponse and store the cookie as directed

```
app.post("/login", (req, res) => {
  // other authenticatey stuff
  res.cookie("userId", user.id); // set the cookie's key and value
  res.redirect("/");
});
```

Clearing Cookies

```
app.post("/logout", (req, res)=>{
    // Clearing the user cookie out
    res.clearCookie("userId")
    // Redirecting the user back to the login page
    res.redirect("/login")
})
```

Useful Links

- Restrictions on Cookies
- cookie-parser

View on Compass | Leave Feedback