

# Cookies and Session with NodeJS

# Cookie

- A cookie is a name/value pair that is passed between a browser and a server in the HTTP header
- They are small pieces of data that are stored on the client's browser.
- Cookies are implicitly deleted when their lifetimes are over

# What we can do with cookies

- Know who visits which part of the site
- Know when is the visit
- Record session id (session uses cookie too!)
  - See below
- What the server can do
  - generate customized interface
  - target advisement according to past interests

# You can have:

- Cookies for identifying users
- Cookies for recording clicks
- Cookies for shopping cart
- Cookies for recommendation

# Traffic

```
GET /somedir/page.php HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr
```

(extra carriage return, line feed)

Client → Server

```
HTTP/1.1 200 OK
Connection close
Date: Tu, 19 Nov 2024 11:27:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 20 Nov 2023 .....
Content-Length: 6821
Content-Type: text/html
Set-Cookie: name=value;expires= ...;path=/
data data data data data ...
```

Server → Client

## Traffic (3)

```
GET /somedir/page2.html HTTP/1.1
Host: www.someschool.edu
Cookie: name=value
User-agent: Mozilla/4.0
Connection: close
Accept-language:en
```

Client → Server

# Cookies in Your Browser

- View cookies
- Delete cookies
- Block cookies

## Cookies and Security

- Cookies are only returned to the server that created them
- Cookies can be used to determine usage patterns that might not otherwise be ascertained by a server
- Browsers generally allow users to limit how cookies are used
  - Browsers usually allow users to remove all cookies currently stored by the browser
- Systems that depend on cookies will fail if the browser refuses to store them



# cookie-parser middleware

- cookieParser is a middleware in the express framework that simplifies cookie handling in Node.js applications.
- To use cookieParser:
- Install the package using npm: `npm install cookie-parser`
- Require and initialize it in your Express app:

# Create cookie

```
const cookieParser = require("cookie-parser"); // Make sure you install it
app.use(cookieParser());
```

```
app.get("/setCookie", (request, response) => {
  response.cookie("course", "SOEN287", {
    maxAge: 5000, //in seconds; define maxAge or expires, if both are defined, maxAge has precedence
    expires: new Date("1 December 2024"),
    httpOnly: true, // The cookie will not be available to Console on browser through document.cookies
    secure: true, // only use with https, not with http
  });
  response.send("Cookie set");
});
```

# Read cookie

```
const cookieParser = require("cookie-parser");

const app = express();
app.use(cookieParser());

app.get("/readCookie", (req, res) => {
  const course = req.cookies.course;
  res.send(course);
});
```

# Session

- Similar to cookies
  - Hold information about one single user
    - username
    - shopping items
  - Pass information from one page to another in your application
- Different from cookies
  - Store the information at the server side instead of the client side
  - Session information is deleted after the user has left the website (permanent storage is possible)

## Set session

```
const session = require("express-session");  
app.use(session({ secret: "a-secret-key-to-encrypt-  
session-data" }));  
  
app.get("/setSession", (request, response) => {  
  request.session.user = "John";  
  response.send("Session set");  
});
```

## Get session

```
const session = require("express-session");

app.use(session({ secret: "a-secret-key-to-encrypt-session-data" }));

app.get("/getSession", (request, response) => {
  const user = request.session.user;
  response.send(user);
});
```