

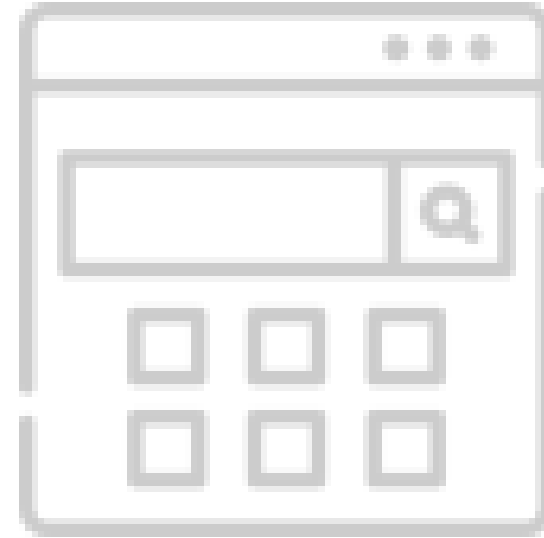
Node.js (쿠키, 세션)

김재호 정한이

목차



쿠키



세션

쿠키와 세션

	쿠키	세션
저장 위치	클라이언트(접속자의 PC)	웹 서버
저장 형식	텍스트	Object
만료 시점	쿠키 저장 시 설정 가능 브라우저 종료되어도 만료시점 지나지 않으면 자동 삭제되지 않음	브라우저 종료 시 삭제 기간 지정 가능
용량제한	하나의 쿠키 당 4KB 클라이언트 총 300개의 쿠키	용량제한 없음

쿠키



쿠키란?

- 클라이언트 로컬에 저장되는 키와 값이 들어있는 작은 데이터 파일
- 쿠키에는 이름, 값, 만료 기한, 경로 정보 등이 들어있다.
- 쿠키는 서버 측에서 만료 기한을 지정하여 정해진 시간동안 데이터를 유지할 수 있다. (로그인 상태 유지에 활용된다)

쿠키

사용 용도

- 1) 세션 관리
: 로그인 유지, 장바구니 유지, 게임 스코어 관리 용도
- 2) 개인화
: 사용자의 선호 언어, 테마 등을 유지하는 용도
- 3) 트래킹
: 사용자의 방문통계 등 행동을 기록하고 분석하기 위한 용도

쿠키



특징

- 1) 쿠키에 대한 정보를 매 헤더에 추가하여 보내기 때문에 상당한 트래픽을 발생시킨다.
- 2) 결제 정보 등을 쿠키에 저장하였을 때 쿠키가 유출되면 보안에 대한 문제점도 발생할 수 있다.
- 3) 클라이언트에 저장되어서 서버에 요청 시 빠르다.

쿠키 - 생성

```
var http = require('http');
http.createServer(function(request, response){
  response.writeHead(200, {
    'Set-Cookie': ['yummy_cookie=choco', 'tasty_cookie=strawberry']
  })
  response.end('Cookie!!');
}).listen(3000);
```

Chromium DevTools Cookies panel showing cookies for localhost.

Request Cookies

Name	Value	Domain	Path	Expires	Size	HttpOnly	Secure	SameSite	Priority
yummy_cookie	choco	localhost	/	Session	17				Medium
tasty_cookie	strawberry	localhost	/	Session	22				Medium

Response Cookies

Name	Value	Domain	Path	Expires	Size	HttpOnly	Secure	SameSite	Priority
yummy_cookie	choco	localhost		Session	19				Medium
tasty_cookie	strawberry	localhost		Session	23				Medium

쿠키 – cookie parser 사용

요청된 쿠키를 쉽게 추출할 수 있도록 도와주는 미들웨어

```
C:\Users\wjdgk\Desktop\Nodejs-master> npm install cookie-parser --save
```

```
var http = require('http');  
var express = require('express');  
var cookieParser = require('cookie-parser');  
var app = express();  
app.use(cookieParser());
```


쿠키 - count값 올라가는 프로그램

```
var express = require('express');
var app = express();
var cookieParser = require('cookie-parser');

app.use(cookieParser());

app.listen(3000, function() {
  console.log('Connected 3000 port');
});

app.get('/count', function(req, res) {
  if(req.cookies.count) { 요청한 쿠키값 읽어오기
    var count = parseInt(req.cookies.count);
  } else {
    var count = 0; count 변수 초기화
  }
  count = count + 1;
  res.cookie('count', count); cookie 값 설정
  res.send('count : ' + count);
});
```

← → ↺ 🏠 ⓘ localhost:3000/count

count : 3

← → ↺ 🏠 ⓘ localhost:3000/count

count : 16

쿠키 - 로그인

```
var http = require('http');
var url = require('url');
var querystring = require('querystring');

var server = http.createServer(function(request, response){
  response.writeHead(200, {'Content-Type': 'text/html'});
  response.end(text);
});

var text = `
  <html>
    <head>
    </head>
    <body>
      <form action='/login' method='post'>
        <input type="text" id="name" value="" placeholder="email">
        <input type="password" id="value" value="" placeholder="password">
        <input type="button" value="add" onclick="AddCookie()">
        <input type="button" value="del" onclick="DelCookie()">
      </form>
    </body>
  </html>
`;

server.listen(3000, function() {
  console.log('Server is running...');
});
```

← → ↻ 🏠 ⓘ localhost:3000/login

<input type="text" value="email"/>	<input type="password" value="password"/>	<input type="button" value="add"/>	<input type="button" value="del"/>
------------------------------------	---	------------------------------------	------------------------------------

Server is running...

쿠키 - 로그인

```
function SetCookie(strName, strValue, iSecond) {  
    var strCookie = strName + "=" + encodeURIComponent(strValue);  
    if(typeof iSecond === "number") {  
        strCookie += "; max-age=" + iSecond;  
    }  
    document.cookie = strCookie;  
}  
  
function AddCookie() {  
    var strName = document.getElementById("name").value;  
    var strValue = document.getElementById("value").value;  
    SetCookie(strName, strValue, null);  
}  
  
function DelCookie() {  
    var strName = document.getElementById("name").value;  
    var strValue = document.getElementById("value").value;  
    SetCookie(strName, strValue, 0);  
}
```

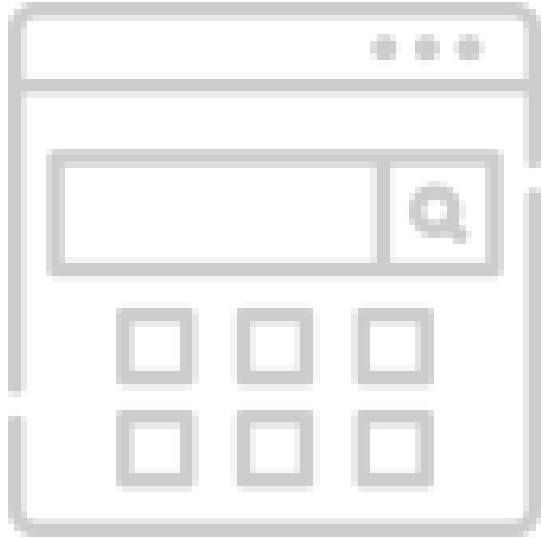
쿠키 - 로그인

← → ↻ 🏠 ⓘ localhost:3000/login

oyr@gmail.com

ole Sources Network Performance Memory Application Security Lighthouse										
🔄 Filter <input type="checkbox"/> Only show cookies with an issue										
Name	Value	Do...	Path	Expi...	Size	Http...	Sec...	Sa...	Sa...	Prio...
oyr@gmail.com	nodejs	loca...	/	Ses...	19					Me...

세션



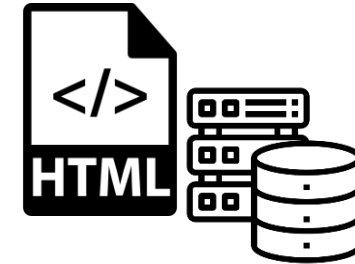
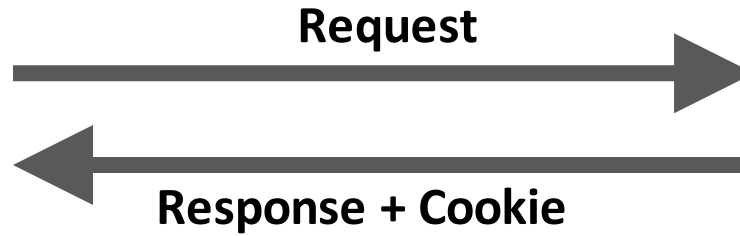
세션란?

- 서버는 클라이언트에게 사용자의 식별자(id값)만을 전송
- 해당 식별자의 여러 정보는 서버의 DB, 파일 형태로 저장
- 식별자만을 저장하는 것을 통해서 훨씬 안전하고 많은 정보를 유지 할 수 있도록 구현

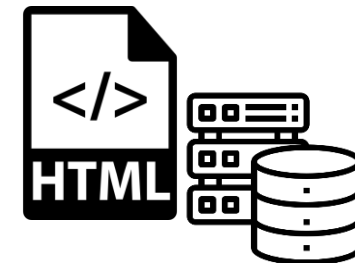
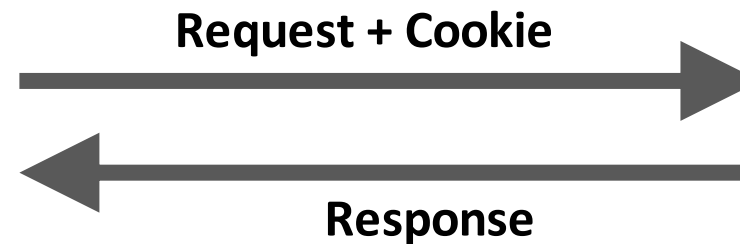
세션



Web Browser



Web Server



세션

Application										
Filter										
<input type="checkbox"/> Only show cookies with an issue										
Name	Value	Do...	Path	Expi...	Size	Http...	Sec...	Sa...	Sa...	Prio...
oyr@gmail.com	nodejs	loca...	/	Ses...	19					Me...

Application										
Filter										
<input type="checkbox"/> Only show cookies with an issue										
Name	Value	Domain	Path	Expires / ...	Size	HttpO...	Secure	Same...	Same...	Priority
oyr@gmail.com	s%253Aun6nq1_bigiUGPWvgIBUcyToWpYm9ik...	localhost	/	Session	19					Medi...

세션 - 구현

```
>npm install -s express-session_
```

express-session 모듈 설치

express-session

: express 프레임워크에서 session을 관리하기 위해 필요한 미들웨어

```
>npm install -s session-file-store_
```

Id, password 값을 파일에 저장하기 위한 모듈

세션 - 구현

```
var express = require('express');
var parseurl = require('parseurl');
var session = require('express-session');

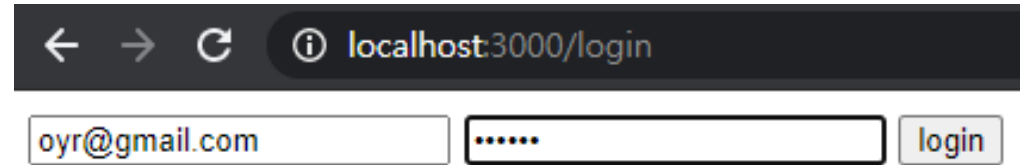
var app = express();

app.use(session({
  secret: 'random string',
  resave: false,
  saveUninitialized: true,
  store: new FileStore()
}));
// secret : 쿠키 값의 변조 방지하기 위한 값
```

```
app.post('/login', function(request, response){
  if(user.id == request.body.id && user.password == request.body.password){
    request.session.is_logged = true;
    response.send(`Success ${user.id}`);
  }
  else {
    response.send('try again');
  }
});
```

```
var user = {
  id: 'oyr@gmail.com',
  password: 'nodejs'
};
```

세션 - 구현



← → ↻ ⓘ localhost:3000/login

oyr@gmail.com login

Sources Network Performance Memory Application Security Lighthouse										
Filter		Only show cookies with an issue								
Name	Value	Domain	Path	Expires / ...	Size	HttpO...	Secure	Same...	Same...	Priority
oyr@gmail.com	s%253Aun6nq1_bigiUGPWvgIBUcyToWpYm9ik...	localhost	/	Session	19					Medi...

감 사 합 니 다
