Lecture-22

JWT, Cookies and Express Sessions

Coding Blocks - Kartik Mathur

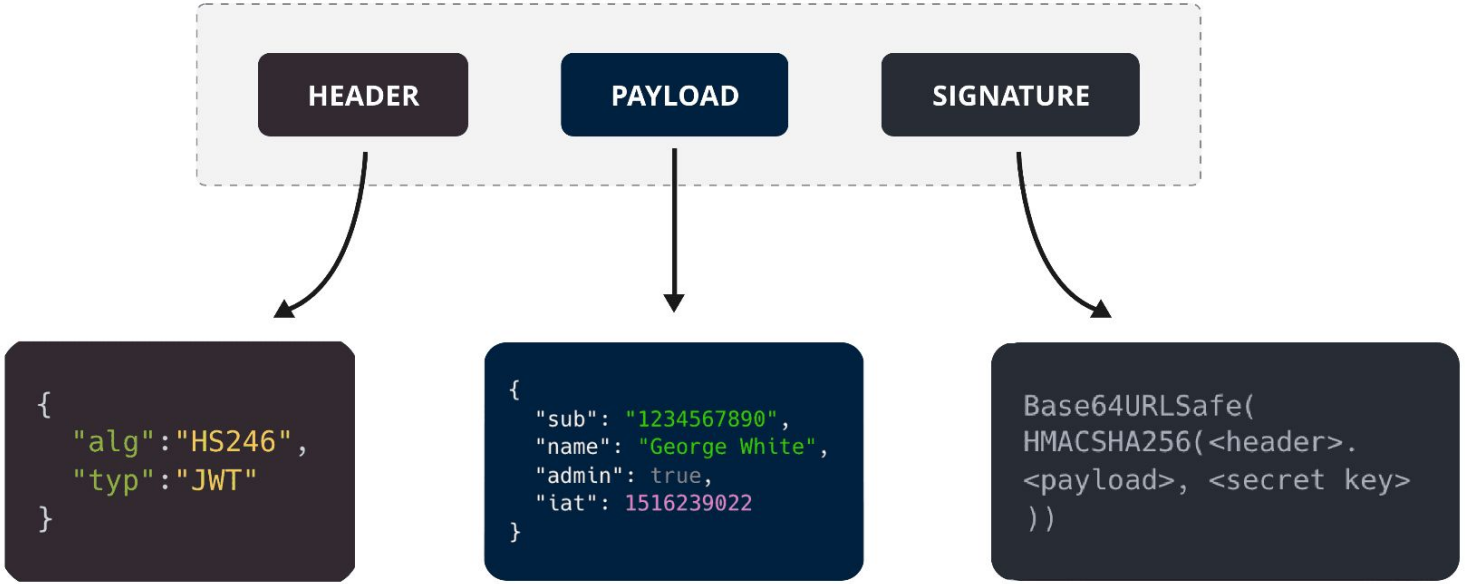
Class Agenda

01	JWT		
02	Cookies		
03	Express Sessions	04	CSR vs SSR
05	-		
06	-	07	-

JWT

1

Structure of a JSON Web Token (JWT)



HEADER.PAYLOAD.SIGNATURE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

`"alg"` is the algorithm used to seal the cookie (usually HMAC SHA-256)
`"typ"` just says: "Hey, I'm a JWT!"

```
{
  "_id": "12345",
  "name": "CB",
  "exp": 112378
}
```

This is where your **user info** goes.

Anyone can open this part and **read** it, but **can't change** it (unless they know the secret).

```
HMACSHA256(
  base64UrlEncode(header) + "." +
base64UrlEncode(payload),
  secret
)
```

If anybody changes header or payload, The signature won't be same

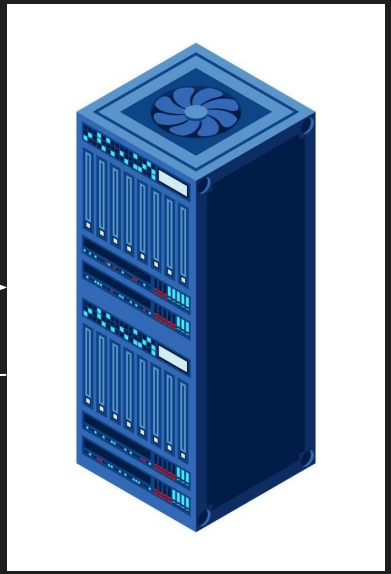
Cookies

2

Cookie

req +

res



Cookie

```
res.cookie('user', '123', {
        httpOnly: false,    // Allows JS access
        secure: false,      // Allows over HTTP
        sameSite: 'Strict',
        maxAge: 86400000
});
```

```
const user = {
  id: 123,
  name: 'Kartik',
  role: 'admin'
};

res.cookie('userData', JSON.stringify(user), {
        httpOnly: false,
        secure: process.env.NODE_ENV === 'production',
        maxAge: 86400000
});
```

To access cookie back on express lets use cookie-parser

Express Sessions

3

Express Session

This express session: To store the data on the server and not on the client

Let us see it in action.

CSR vs SSR

4

CSR vs SSR

Todo list using SSR: HBS