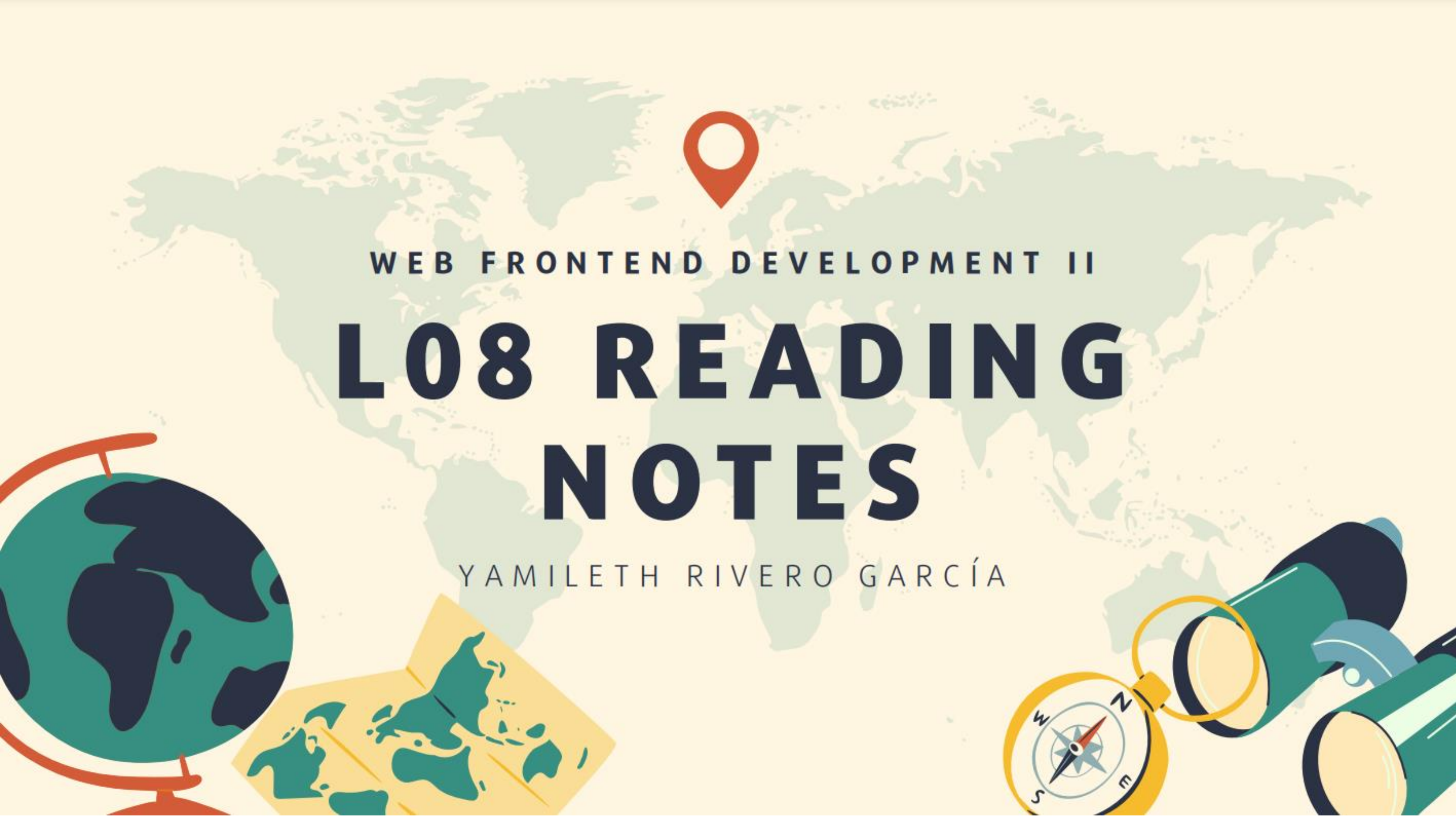


WEB FRONTEND DEVELOPMENT II

L08 READING NOTES

YAMILETH RIVERO GARCÍA

1.- Getting Started with JSON Web Tokens (JWT)

JSON Web Token (JWT) is an open standard that defines a compact and self-contained way for securely transmitting information between parties as a JSON object

- This information can be verified and trusted because it is digitally signed.
- JWTs can be signed using a secret or a public/private key

When should you use JSON Web Tokens?

These are some scenarios where JSON Web Tokens are useful, which are:

- Authentication: This is the typical scenario for using JWT, once the user is logged in, each subsequent request will include the JWT, allowing the user to access routes, services, and resources that are permitted with that token.

- Information Exchange: JWTs are a good way of securely transmitting information between parties, because as they can be signed.
    - **Example**: Using a public/private key pair, you can be sure that the sender is who they say they are.

Which is the JSON Web Token structure?

JWTs consist of three parts separated by dots (.), which are:

- Header
- Payload
- Signature

Therefore, a JWT typically looks like the following:

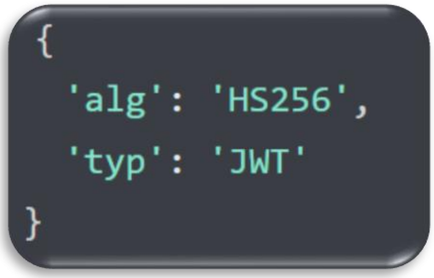
```
xxxxx.yyyyy.zzzzz
```

- Header -

The header typically consists of two parts:

-The type of the token (which is JWT)

-The hashing algorithm (such as HMAC SHA256 or RSA).



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

- Payload –

The second part of the token is the payload, which contains the **claims.**

- Claims are statements about an entity (typically, the user) and additional metadata. There are 3 types reserved, public, and private claims

Reserved Claims

- Set of predefined claims
- Are not mandatory but recommended
- Provide a set of useful, interoperable claims

EXAMPLES:

iss (issuer) exp (expiration time) sub (subject) aud (audience)

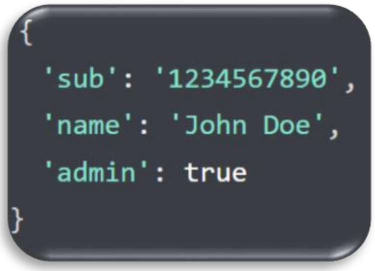
Public Claims

- Can be defined at will by those using JWTs
- To avoid collisions, they should be defined in the IANA JSON Web Token Registry
- Be defined as a URI that contains a collision resistant namespace.

Private Claims

- Custom claims created to share information between parties that agree on using them

Example of Payload:

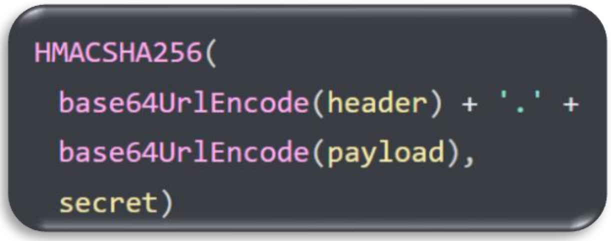


```
{
  'sub': '1234567890',
  'name': 'John Doe',
  'admin': true
}
```

- Signature –

To create the signature part you have to take the encoded header, the encoded payload, a secret, the algorithm specified in the header, and sign that.

EXAMPLE:



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

The signature is used to:

-Verify that the sender of the JWT is who it says it is

-Ensure that the message wasn’t changed in the way.

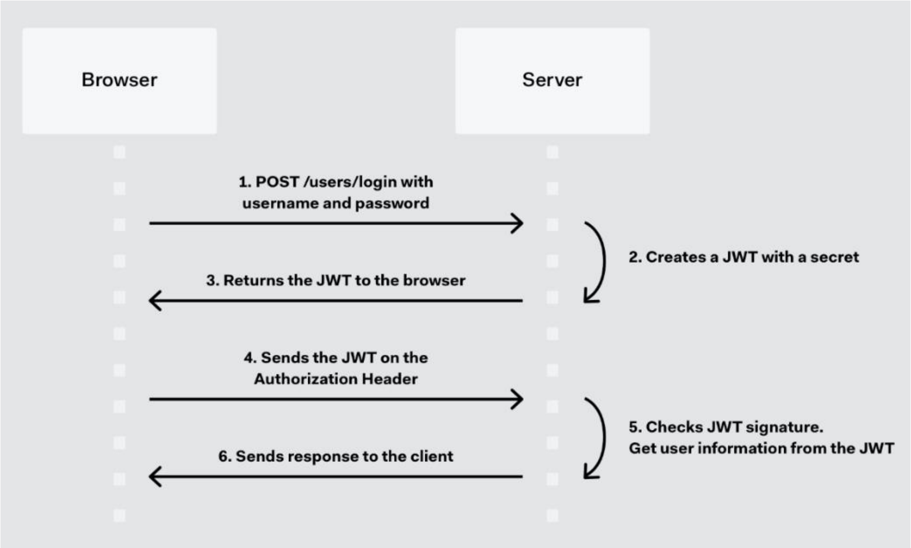
How JSON Web Tokens work?

In authentication, when the user successfully logs in using their credentials, a JSON Web Token will be returned.

Tokens are credentials, great care must be taken to **prevent security issues**.

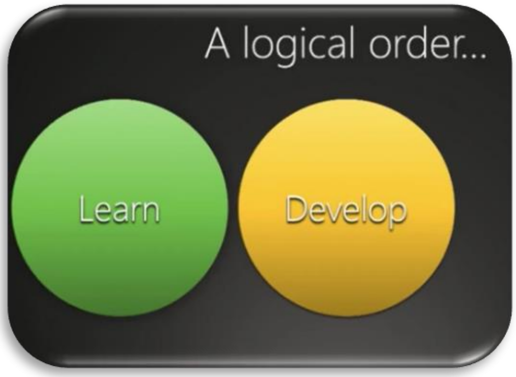
Note: You should not keep tokens longer than required!

Note 2: You also should not store sensitive session data in browser storage due to lack of security.

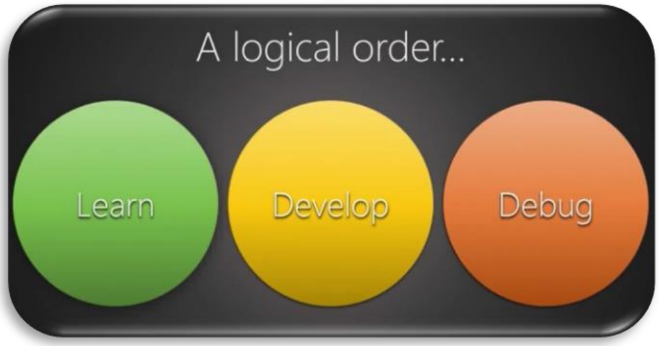

Browser
Server
1. POST /users/login with username and password
2. Creates a JWT with a secret
3. Returns the JWT to the browser
4. Sends the JWT on the Authorization Header
5. Checks JWT signature. Get user information from the JWT
6. Sends response to the client

2.- Develop, Debug, Learn?

- JavaScript allows us to create interactivity in the browser
- JavaScript give us the power we always wanted



- We first learn a language and then we develop, using a **logical order**



- But then we needed to learn how to debug too, because a lot of mistakes were happening, so the debugging process was added to the **logical order**

- Then we decided to add abstractions, libraries, and build processes.
- We now are using other people’s codes to build codes, but that sometimes is a problem, because debugging starts to be a problem! Because we don’t form that code and we are lost finding the issue

people.focus()

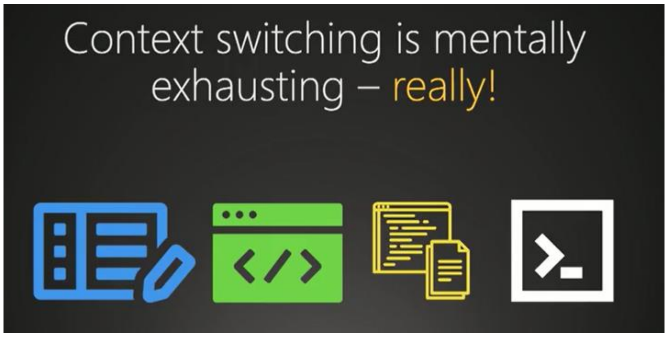
We need to focus more on the users, they pay our companies for the things that we create.

It’s not about us but about people who need our services!!

- There is a lot of code that nobody maintains because we are focused on new things, but that millions of users still use.

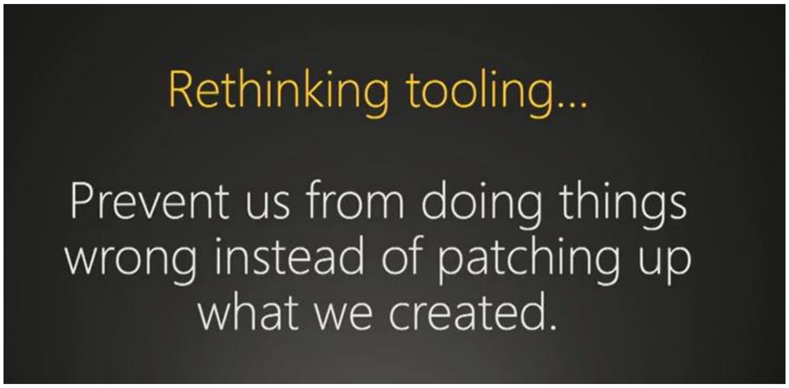
Sometimes because nobody knows how to make it we feel overwhelmed :c

- **We should be better than that!**
- We stopped thinking about JavaScript solutions!
- We stopped thinking about creating! :C
- WHERE IS THE HAPPINESS GONE?



Context switching is mentally exhausting – really!

- It’s time to re-think tooling



Rethinking tooling...

Prevent us from doing things wrong instead of patching up what we created.

- Learn things while doing it!

We can get as excited as we were when we started!

THE END!