Office_Hours02.27

Monday, February 27, 2023 7:48 PM

For JWT authentication how do we use generic principal/identity?

We are unable to convert principal to JWT due to claims

- In order to create principal, we need identity, which needs claims
- Principal is meant to show different contexts for each user
    - Same identity but different principals for a user, student vs teacher account for professor

Do we recreate the principal based on the JWT token in controller layer?

- Create principal in middleware layer
- Set current thread principal to the principal created in middleware
- HTTP context is typically only accessible in middleware
    - Technically invisible in controllers but thread is better alternative

For client-side, can we use secret key for creating the JWT in back-end and front-end?

We store it as an environment variable

- **FIRST WAY**
- Validate the token on front-end since JWT is meant to be transparent
    - If encrypted, then JWT becomes JSON Web Encrypted Token (JOSE?)
- If using HMAC hash a key needs to be provided
    - Or not if you don't mind it being less secure
- **SECOND WAY**
- Client side application launches a web worker which request a key value asynchronously, and the value stays within the web worker
    - Nothing has access to web worker, not even what initialized it
    - Safe from web dumps from client
- Web worker pushes back to application letting it know there are no issues
- AJAX request to server, get data, wrap in closure , take that value and store it in another closure
    - Closure is iffy block
    - Only access is exposing methods in closure
    - Unmodifiable
- **THIRD WAY**
- Store the secret key for validation as session storage, local storage, or index db
    - But accessible by anyone

Hard to see the AJAX request normally until its saved since it's stored in header (?)

(Not too sure about this part)

8 am start time for code review sign up