Office_Hours02.15

Wednesday, February 15, 2023 8:09 PM

AJAX DEMO

Startup backend

Startup frontend

Change url to backend

Will run into issues with CORS due to ports

Unless you use the same port as in the demo

How to pass principal to backend?

Must send information

Easiest way: send as authorization header, since it's a string value it must be formatted

Industry standard is JWT

Take principal -> JWT string -> put in header

Backend just says "request.header" to get header back on future requests

Will need to store it on client side to go across pages without constant checks to server

JWT is safe value to verify permissions are correct

Caveat: cookie is not stale

1. Pass along in header each time
   a. NOT RECOMMENDED
2. Create cookie and store inside
   a. Automatic creation by http response with create cookie
3. Store in session storage
   a. Newish technology in browser
      i. Index storage
         1) Not here
      ii. Session storage
         1) Yes here pls
      iii. Something else storage
   b. Session storage will work across different pages with the same origin

In production there're two different tokens:

1. ID token and represents user (only contains user name/id/whatever)
   a. Made on log-in
   b. Presence means user is already logged in

2. Access token: permissions within applications (principal)
    a. On page request, "Do you have right perms?"

Etag header

Execution

Max age header

Maximum time to live

How to check if data has been changed or is new to refresh automatically?

1. Every request response has specification of whether or not to alter
    a. Vary header
2. Cache
    a. If the data comes from backend to frontend and it is already present in cache, do not execute

An AJAX request will be required to determine if backend has changed if you're using HTTP

Push request has hub which checks if there has been a change

Libraries exist but more stuff you will have to learn (not recommended)

Socket.io

Signal R

RX.js or RX.net

CacheBuster to make sure the server will treat each request as unique

If a similar request is made in small interval, the same response will be made