Week 08 notes

Get started with JSON web tokens

* JSON web token (JWT) is an open standard
* You can use this as a compact and self-contained way for transmitting information between parties as a JSON object securely.
* This information can be verified and trusted because it is digitally signed
* JSON web tokens can be used to give users access to routes and services that they are permitted to with the token that they have.
* JWT’s consist of three parts separated by dots.
  + Header
  + Payload
  + Signature
* Typically looks like the following
  + Xxxxx.yyyyy.zzzzz
* Header
  + Typically consists of two parts:
    - The type of the token, which is JWT
    - The hashing algorithm such as HMAC SHA256 or RSA
* Payload
  + This contains the claims
  + These are statements about an entity (typically, the user) and additional metadata
  + There are three types of claims
    - Reserved claims
      * These are a set of predefined claims, which are not mandatory but recommended.
    - Public claims
      * These can be defined at will by those using JWT’s
      * To avoid collisions, they should be defined in the IANA JSON web token registry or be defined as a URI that contains a collision resistant namespace
    - Private claims
      * These are the custom claims created to share information between parties that agree on using them
* 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 it.
* Putting it together
  + The output is three Base64 strings separated by dots that can be easily passed in HTML and HTTP environments.
* In authentication, when the user successfully logs in using their credentials, a JSON web toden will be returned.
* Since tokens are credentials, great care must be taken to prevent security issues
* You also should not store sensitive session data in browser storage due to lack of security
* Whenever the user wants to access a protected route, it should send the JWT, typically in the Authorization header using the Bearer schema
* This is a stateless authentication mechanism as the user state is never saved in the server memory

Why should you use JSON web tokens

* They are smaller than SAML making them a good choice to be passed in HTML and HTTP environments
* It has tokens that can also use a public/private key pair in the form of a X.509 certificate to sign them
* They are common in most programming languages, because they map directly to objects
* They are used at an internet scale; this highlights the ease of client-side processing

Develop, debug, learn

* Be creative in your coding
* The debugging tool was an alert
* Logical order
  + Learn
  + Develop
  + Debug
* The end user shouldn’t suffer from our creativity
* Debugging doesn’t work as well as we want it too sometimes
* Caught in the rush of what is better
* We are focusing on the wrong thing
  + We need to be focusing on the end user
* We are a delivery service
* It is not us, it is about our tech legacy
* Developers are people too
* There is a demand overload for developers
* We are missing a lot of opportunities
* We have amazing technology and yet no one is happy
* We have to much choice
* We have breakpoints that we can use to debug and people are using console.log
* We get mentally exostied by jumping back and forth while coding
* The learning needs to be a part of the developing and debugging
* Most of what we do now days is not offline
* A lot of browsers have API’s now and this is a good thing
* You can learn while you are doing things
* You can use autocomplete for certain languages like python
* This is your world to build