



Module 3 Day 18

Asynchronous Programming in JS

Definitions

Term	Definition
API	<p>Application Programming Interface. A set of functions and procedures allowing other applications to access the features or data of an operating system, application, or other service.</p> <p>An API may be for a web-based system, operating system, database system, computer hardware, or software library.</p>
Web API or Web Service	<p>A service (API) offered by an application or device to another application, which communicate with each other via the World Wide Web (typically HTTP).</p>
REST	<p>Representational State Transfer. One style of web service which utilizes HTTP features (statelessness, Request and Response, and GET, PUT, POST and DELETE methods) for accessing and updating data.</p>
Consumer	<p>A user of web services. This is software, not a person.</p>
Endpoint	<p>The "location" at which a service's features (methods / data) can be accessed. For a web service, this is usually a URL.</p>

Asynchronous Programming

- Start an operation, but don't wait for completion before moving on
- When we start the operation, specify "what to do" when the operation completes
 - The function returns us a "Promise"
 - The Promise is "pending"
- Move on to do more work while the operation is taking place
- When the operation finishes, the above "what to do" code is called
 - The Promise is "fulfilled"

Http with Axios

```
// Send an HTTP GET request to the nps parks api, returns a Promise of a Response
axios.get(npsURL)
  .then((response) => { // Get a Response object once the GET completes
    const parks = response.data; // The api returns a collection of parks
    if (parks.data.length > 0) {
```



Demo

JavaScript "Service" Objects

```
1  import axios from 'axios';
2
3  const http = axios.create({
4    |   baseURL: "http://localhost:3000"
5  });
6
7  export default {
8
9    |   list() {
10     |     return http.get('/docs');
11     |   },
12     |   get(id) {
13     |     |   return http.get(`/docs/${id}`)
14     |     |
15     |   }
```

Let's
Code