



fetch(. . .) and React

CUNY Tech Prep (2019)
Edgardo Molina, PhD | Head Instructor

`fetch()` is built in to modern web browsers and allows us to request network resources asynchronously

(You may have heard of AJAX calls before)



How fetch() works

```
fetch('http://example.com/movies.json')  
  .then(function(response) {  
    return response.json();  
  })  
  .then(function(myJson) {  
    console.log(JSON.stringify(myJson));  
  });
```

MDN Using Fetch Documentation:

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

fetch()

- returns a *promise*
- runs asynchronously (later, not now)
- can perform any HTTP Request method (GET, POST, PUT, DELETE, etc.)



Promises

Functions that return *promises*, return right away (*without doing any work*), and will run their task at a later time

When the promise is fulfilled, we will have access to the result, we can provide a function to receive that result, and we pass it to the `.then()` function


`.then().then().then()`: this function can be chained so that multiple functions can run on the output in order

`.then()` can be called on any promise and any `.then()` call



What if there's an error?

```
fetch('https://example.com/posts', {  
  method: 'POST',  
  body: formData  
})  
  .then(response => response.json())  
  .then(response => console.log('Success:', JSON.stringify(response)))  
  .catch(error => console.error('Error:', error));
```



`.catch()` errors

`.catch()` can be used with any promise or `.then()` call

If any exception occurs during `fetch()` or during any of the `.then()` calls, the error will immediately go to the first `.catch()` call below the location of the error.

`.catch()` calls prior to the error happening are out of scope

How do we use fetch in React?

We use `fetch` to access *data* from an API, that our React components will *use* and *render*

Two common scenarios

- A react component needs data when it has loaded
- A react component needs data after a user interaction

There are more complex scenarios that come up with games, chats, and real time apps.



Options for calling fetch within components

NOT HERE

- `constructor()`
- `render()`
- `componentWillMount()`

Where?

- `componentDidMount()`
- Event handlers (`onClick`, `onChange`, etc)
 - (but **BEWARE**, this could cause a lot of `fetch()` calls, which will impact performance and costs!!!)

**How does a successful fetch
get React to use the data?**

Use `this.setState(...)`



Fetch should update state,
React takes care of the rest

```
fetch("https://ctp-zip-api.herokuapp.com/zip/11432")  
  .then(res => res.json())  
  .then(body => {  
    console.log(body);  
    this.setState({  
      cityName: body.City,  
      population: body.EstimatedPopulation,  
    });  
  })  
})
```



Additional Resources

That's so Fetch: <https://jakearchibald.com/2015/thats-so-fetch/>
(great detailed overview of ajax and fetch)

Fetch Polyfill: <https://github.com/github/fetch>
(in case you have to support an older browser that does not natively have fetch built in)

Get Started with Fetch API:
<https://www.freecodecamp.org/news/a-practical-es6-guide-on-how-to-perform-http-requests-using-the-fetch-api-594c3d91a547/>