

More REST APIs

Michael Chang
Spring 2023

Plan for today

Using REST APIs

Sending data: method, headers, body

Keeping data model up to date

Frontend wrap up

Keeping state in classes

Interaction between components and models

Sending data to server

`fetch(url[, options])`

`options` is an object with following keys

`method`: HTTP method

`headers`: Object of HTTP headers to include in request

`body`: request body (for non-GET) (as a string)

When sending data to server

Query string goes in the URL

When including request body, need to set Content-Type header

E.g. `headers: { "Content-Type": "application/json" }`

Sending data to server

```
const postData = async () => {  
  let data = { num: 42 };  
  let res = await fetch("/api/path?param=binky", {  
    method: "POST",  
    headers: { "Content-Type": "application/json" },  
    body: JSON.stringify(data)  
  });  
  ...  
};
```

Data models

Useful to encapsulate data in classes

E.g. Student, Course

Methods for reading from and updating API

Note: constructor cannot be async

Instead, use a static method

Data models

Useful function: `Object.assign(dest, src)`

Copy all the keys from src into dest (overwriting)

E.g. `Object.assign(this, data)`

`myClass.toJSON()`

Define this method to control how `JSON.stringify` converts object into JSON

E.g. include only public instance variables

class Usage

As DOM component

Encapsulate logic for creating and managing elements

Allows reuse of components

Common features:

- Event handlers (don't forget to bind!)

- Private instance variables for DOM elements

- Public methods take DOM element and/or other components as argument

- Methods take callbacks to notify other classes of events

class Usage

As data model

Encapsulate logic and state for a resource (a "noun" in our system)

Provides a public interface for retrieving/updating data from/to API

Common features:

- async methods that make API requests

- static method(s) to retrieve instances from the API

- Easiest to keep data returned from API "public"

- Private instance variables for client-side-only state

- toJSON method to control how resource is sent back to server

Summary

Today

Frontend wrap up

Before next time

assign2.2

assign3.1 out tomorrow

Next time

Intro to backends

How to build these APIs