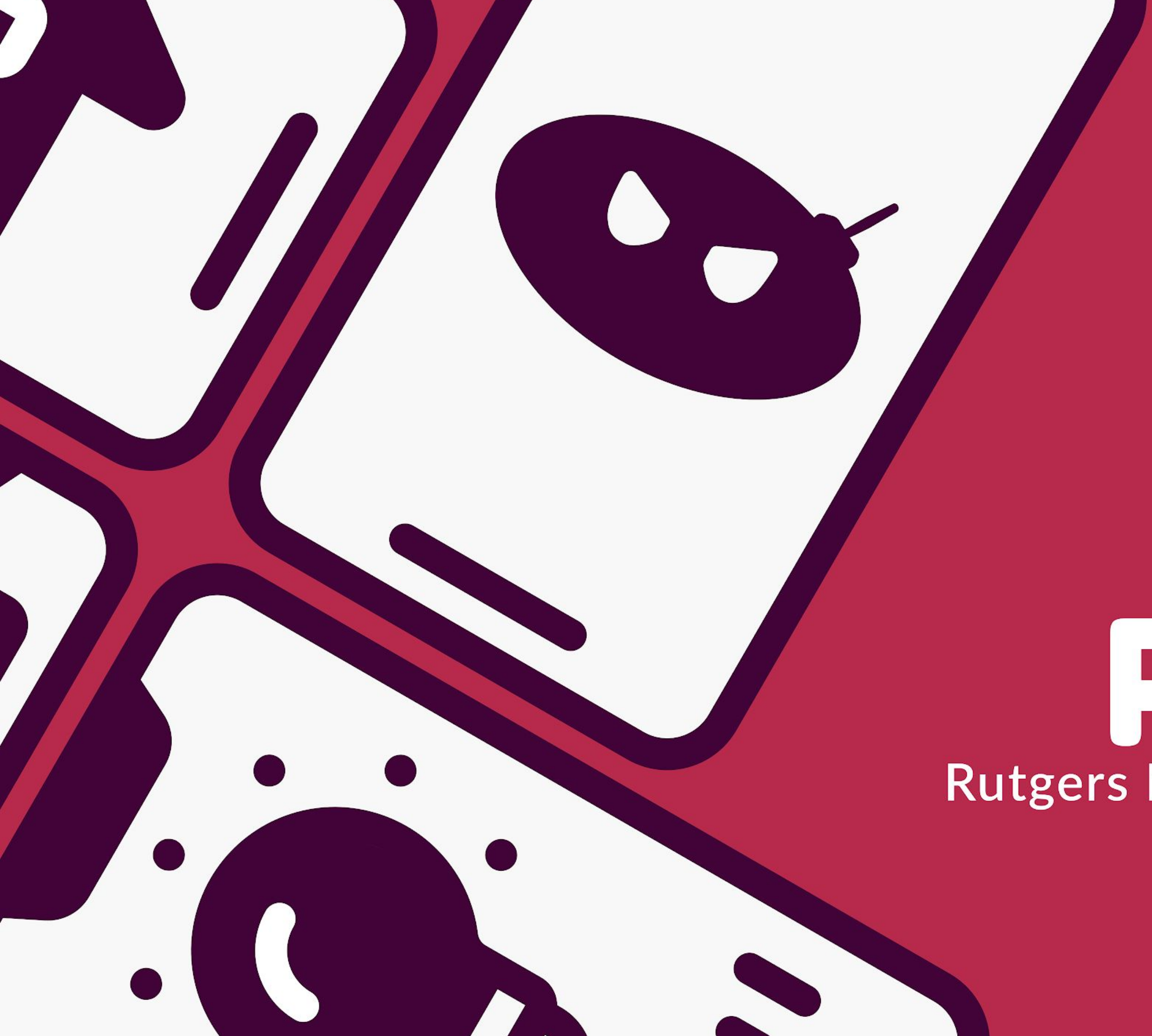


RUMAD

Rutgers Mobile App Development

Overview

For this section, we'll be covering:

- Finish Express Demo
- Review of Asynchronous Functions & Express
- Practice with Async Functions & Express
- If time (unlikely):
    - Introduction to Databases

Express Demo Cont.
(POST)

Async Functions (Review)

Asynchronous Functions

When requesting data from anywhere, data does not come immediately… (there is a delay)

- This delay is a big no-no when working with *synchronous code*. (Everything you’ve written so far)
- Code waits for nobody.

Asynchronous Functions - 2

- How do we handle waiting for data?
  - Declare a `async` function.
- A async function declares a function to rely on a "Promise-based" behavior.
  - Enables the code to continue running without needing to wait for something to return.

JavaScript “async” and “await”

What are the following?

- Promise
- async
- await

Promise

> ***“A Promise is a proxy for a value not necessarily known when the promise is created. It allows you to associate handlers with an asynchronous action's eventual success value or failure reason.”*** (mdn)

Imagine that you’re a top singer, and fans ask day and night for your upcoming song.

To get some relief, you promise to send it to them when it’s published. You give your fans a list. They can fill in their email addresses, so that when the song becomes available, all subscribed parties instantly receive it. And even if something goes very wrong, say, a fire in the studio, so that you can’t publish the song, they will still be notified. (javascript.info)

Promise

> ***“A Promise is a proxy for a value not necessarily known when the promise is created. It allows you to associate handlers with an asynchronous action's eventual success value or failure reason.”*** (mdn)

What is a promise?

- Signifies that something will be returned, whether it’s successful or not
- Has three states: pending, fulfilled, rejected
- When completed, return

Why is this important?

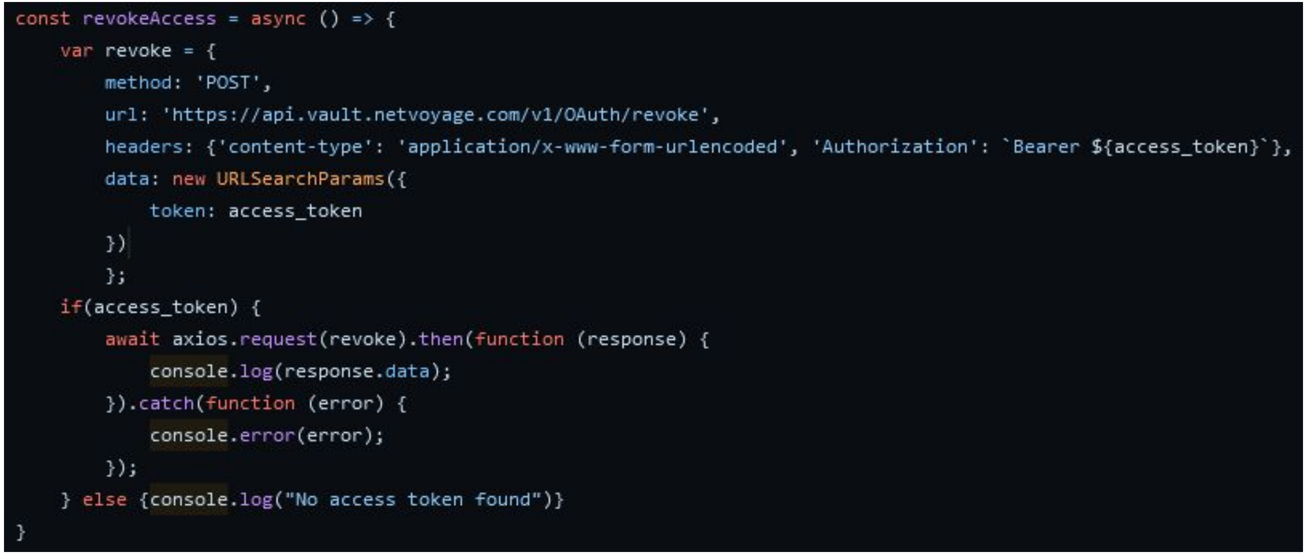
Many operations are never instant

- database operations… (we will go over this later)
- practically anything over the internet

Promises allow us to handle things when they are completed.

JavaScript “await” and “async”

await - ‘await’ a promise. yield until promise is fulfilled or rejected
async - this decorator indicates that a function returns a promise



```
const revokeAccess = async () => {
    var revoke = {
        method: 'POST',
        url: 'https://api.vault.netvoyage.com/v1/OAuth/revoke',
        headers: {'content-type': 'application/x-www-form-urlencoded', 'Authorization': `Bearer ${access_token}`},
        data: new URLSearchParams({
            token: access_token
        })
        };
    if(access_token) {
        await axios.request(revoke).then(function (response) {
            console.log(response.data);
        }).catch(function (error) {
            console.error(error);
        });
    } else {console.log("No access token found")}
}
```

Check-In Questions

Test Your Understanding 1

What is the main purpose of async functions?

a) Handling sequential processes in our code.
b) Accounting for the delay in data (Network requests, etc.)
c) Making our code more modular
d) Handling POST requests

Test Your Understanding 1

What is the main purpose of async functions?

a) Handling sequential processes in our code.

b) Accounting for the delay in data (Network requests, etc.)

c) Making our code more modular

d) Handling POST requests

Test Your Understanding 2

What are some methods of implementing async functions? (Select all that apply)

a) Async/Await keywords

b) If statements

c) Creating a Promise and writing callbacks for when it is resolved.

d) For loops

Test Your Understanding 2

What are some methods of implementing async functions? (Select all that apply)

a) **Async/Await keywords**

b) If statements

c) **Creating a Promise and writing callbacks for when it is fulfilled/rejected.**

d) For loops

Test Your Understanding 3

What is the bolded part of this URL http://localhost:3000/api/books/**1** an example of?

a. Query string
b. Path parameter
c. Both a and b
d. Neither

Test Your Understanding 3

What is the bolded part of this URL http://localhost:3000/api/books/**1** an example of?

a. Query string
b. **Path parameter**
c. Both a and b
d. Neither

Test Your Understanding 4

When handling a **POST** request in Express, where should we access the data? (Our request object is “req”)

a. req.body
b. req.query
c. req.params
d. None of the above

Test Your Understanding 4

When handling a **POST** request in Express, where should we access the data? (Our request object is “req”)

a. **req.body**
b. req.query
c. req.params
d. None of the above

Let’s Practice!

Get & Post Practice

Start on the homework!

a. You’ll be working on implementing a **task manager API** (Directions are on the problem) in breakout groups

b. Driver functions are provided.

c. We’ll be hopping into groups to answer any questions or issues

Databases

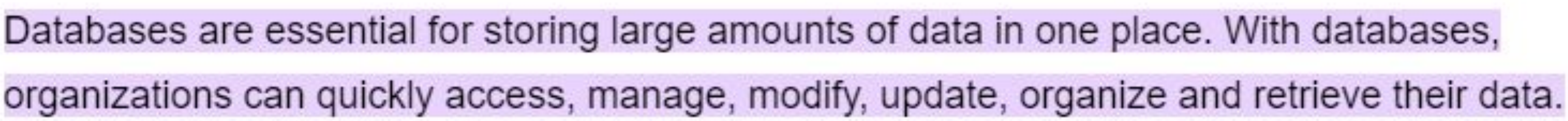
What is a database?

- Collection of structured information (**data**)
  - Relational Databases
    - Tabular format
    - Uses SQL for querying
  - Non-Relational (NoSQL) Databases
    - More flexibility in the way data is stored
    - No standardized querying language
- Allows users to store, retrieve, update, and delete data efficiently

Why do we need databases?

Recall the Express exercise we did:

- We had a list of dictionaries with **records** of books within the source code
- Now imagine if we wanted to display a catalog of **100** books, **1000** books, or maybe **10,000** books
  - *Would it be efficient to have a massive dictionary within our code?*



Databases are essential for storing large amounts of data in one place. With databases, organizations can quickly access, manage, modify, update, organize and retrieve their data.

What is a DBMS?

- Database Management System
  - Software for creating, managing, and interacting with databases

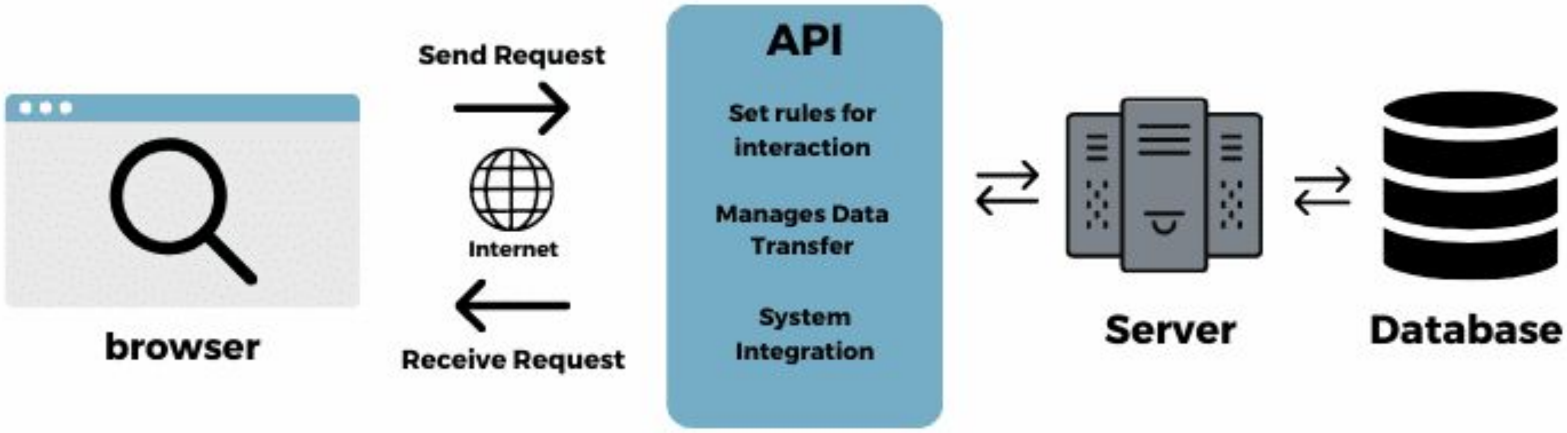
- Examples of DBMS:
  - mySQL
  - MongoDB
  - **Supabase**

How to interact with a DB?

C.R.U.D Operations

1. Create
2. Read
3. Update
4. Delete

We’ll go into more detail later!



Data Flow within an Application

What is Supabase?

- Supabase is an open-source cloud database
  - The database exists on a cloud server
  - You don’t need to host on your PC
  - BaaS – backend as a service

- Supabase is PostgreSQL-based
  - Real-time functionality
  - Storage & authentication

*PostgreSQL is another commonly used database

- Supabase is built upon Postgres (abstraction of Postgres)

Questions?

Please fill out the feedback form when you have a chance!

Feedback Form

Next week...

- Introduction to Supabase
  - Introduction to Databases
  - Setting up Supabase
  - Implementing CRUD

Did you complete Week 0?

Scan here for a guide to setting up your
development environment!