Node continued

Questions

- What is MVC?
- How should I start a new Node app?
- How do I debug Node.js with Chome Dev Tools?
- How do I make Node reload my code?
- How do I save shortcuts for long, annoying commands?
- How do I talk to Postgres from Node?
- How do I pass dynamic values to a SQL query?
- When do I use `db.one` vs `db.any` vs `db.result`
- When do I use `returning...`?

What is MVC?

- Model - talk to the database
- View - the UI that the user sees
- Controller - coordinates the View and the Model

It's a pattern for building web applications.

How should I start a new Node app?

```
git init
npm init -y
npm install --save-dev nodemon
touch index.js
code .
```

How do I debug Node.js with Chome Dev Tools?

- In Chrome, go to this address: `chrome://inspect`
- Click the link labeled "Open dedicated DevTools for Node"

Then run:

```
node --inspect-brk index.js
```

- It will pause on the first line of code
- You can add breakpoints in the DevTools
- Or add the `debugger` keyword to your code

How do I make Node reload my code?

Use nodemon!

```
npm install --save-dev nodemon
```

`--save-dev` saves it as a dev dependency

Then, add it as an "npm script" (see next slide).

How do I save shortcuts for long, annoying commands?

Add them to your package.json's `scripts`

```
"scripts": {
  "debug": "node --inspect-brk index.js",
  "dev": "nodemon index.js",
  "test": "echo \"Error: no test specified\" && exit 1"
},
```

How do I talk to Postgres from Node?

Use `pg-promise`

```
npm install pg-promise
```

Then...configure it!

```
const pgp = require('pg-promise')();

const db = pgp({
    host: 'localhost',
    port: 5432,
    database: 'node-todo-app-db'
});
```

How do I pass dynamic values to a SQL query?

Use $1, $2, $3, etc. in your SQL template string, and pass in an array of values to interpolate into the string.

```
db.result(`update todos
              set completed=true
           where id=$1`, [id])
```

```
function add(name, completed) {
    return db.one(`insert into todos (name, completed)
        values
            ($1, $2)
        returning id
    `, [name, completed])
}
```

When do I use `db.one` vs `db.any` vs `db.result`

- `db.any` - expect zero or more rows. (Zero rows means you get an empty array `[]`)
- `db.one` - expect one and only row from the database. (You're doing a select by id)
- `db.result` - you want a report on the transaction. (Useful for updates and deletions)
- `db.none` - you want no information or rows back from the query. (You're deleting, but you don't actually care if it works.)

When in doubt, use `db.any`

When do I use `returning...`?

Use when inserting into a table

In a web application, you almost always want to redirect the user to a page showing that newly created item.

For them to see that page, they need to be redirected to something like `/cats/5` or `/todos/7`.