

Day 1: Intro to Node and Express

Fast, unopinionated, minimalist web framework.

Agenda

1. 🏆 [05m] Learning Objectives
2. 📖 [15m] Node and Express
3. 🖥️ [50m] Activity: GIF Search
4. 🌴 [10m] BREAK
5. 📖 [30m] Recap: Node & Express
 1. A Basic Server
 2. Writing a Route
 3. The `req` & `res` Variables
 4. Writing a Template
6. 🌙 [05m] Wrap Up
7. 📚 Resources

🏆 [05m] Learning Objectives

By the end of this class, you should be able to...

1. Set up a fully functioning Node environment.
2. Write a route using Node and Express.js.
3. Use the Handlebars templating language to pass data to a template.
4. Compare and Contrast Express routes with Flask.

📖 [15m] Node and Express

Let's go over some definitions to get us started...

- **Node.js** is an open-source JavaScript runtime environment that executes JavaScript code outside of a browser.
- **NPM** is short for "Node Package Manager", and is used to download third-party packages for use with Node.
- **Express** is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.
- **Handlebars** is a lightweight templating language that works with Node and Express.

🌴 [10m] BREAK

🖥️ [50m] Activity: GIF Search

Choose pairs randomly. You will work with your pair for the rest of class.

Open the [GIF Search tutorial](#) and finish parts 1-4 (part 5 optional). As you work, discuss each step with your pair and make sure you understand each step.

You will submit the tutorial in Class 2. You may finish on your own (just make sure to `git push` before the end of class).

[30m] Recap: Node & Express

A Basic Server

This code will likely be the same for most projects you create. *You do not need to memorize this, but instead should reference*

```
const express = require('express');
const app = express();

const exphbs = require('express-handlebars');

app.engine('handlebars', exphbs({defaultLayout: 'main'}));
app.set('view engine', 'handlebars');
```

Writing a Route

We write routes in Express using `app.get`, `app.post`, etc.

```
app.get('/greetings/:name', (req, res) => {
  // grab the name from the path provided
  const name = req.params.name;
  // render the greetings view, passing along the name
  res.render('greetings', { name });
})
```

The `req` & `res` Variables

`req` = “Request”.

- `req.query` - Access the **query parameters** passed in the URL's query string.
- `req.params` - Access the **route variables** passed in the URL. E.g. `/greeting/Meredith`

`res` = “Response”.

Writing a Template

Take a look at the template syntax for Handlebars for showing a list of songs. How is it different than Jinja (or Django) syntax?

```
<ul>
  {{ #each songs }}
    <li>
      <a href="/song/{{ this._id }}">
        {{ this.name }}
      </a>
    </li>
  {{ /each }}
</ul>
```

[05m] Wrap Up

Finish the GIF Search tutorial before the start of next class. You may finish individually - just make sure to clone the repository

Resources

[Express/Node Introduction](#)