

Express HTTP Server



Objectives

- Explain what Express is.
- Explain why Express is useful.
- Create an HTTP server with the `express` module.
- Handle some static and dynamic routes with Express HTTP server
- propose a solution to how Derek and Hansel could have handled this differently...



Big Picture

- ...So we will use our machines to create a server and serve up some files based on the requests that come in.
- Some of ya'll have already done this



TaT

1. what is Express?
2. why do we use it?
3. name 2 examples of sites you regularly use that have a database

Code Along



Key Steps (what are these doing?)

- `npm install -g nodemon`
- `npm init`
- `npm install --save express`
- `npm install --save dotenv`
- `brew install httpie`

Key Steps (what are these doing?)

- `npm install -g nodemon`
`// nodemon continuously monitors changes to our app`
- `npm init`
`// creates a package.json file`
- `npm install --save express`
`// installs and saves express as a dependency in our package.json file so`
`if you were to clone it from github, all you have to do is run npm install`
- `npm install --save dotenv`
`// gives us the ability to access private environmental variables that we`
`don't want to share on github ie) passwords, API keys`
- `brew install httpie`
`// kind of like curl but better`

There is always
music amongst the trees,
and in the garden,
but our hearts
must be still
to hear it.



A Simple 'Hello World'

```
'use strict';
```

```
const express = require('express');  
const app = express();  
const port = process.env.PORT || 8000;
```

```
app.use(function(req, res) {  
  res.send('Hello World');  
});
```

```
app.listen(port, function() {  
  console.log('Listening on port', port);  
});
```

Sending back some Data with the 'get' route

```
'use strict';

const express = require('express');
const app = express();
const port = process.env.PORT || 8000;

app.disable('x-powered-by');//not super important right now (ignore it)

app.use(function(req, res) {
  let guests = ['Mary', 'John'];
  res.send(guests);
});

app.listen(port, function() {
  console.log('Listening on port', port);
});
```

Let's put our guests in a file..

```
echo '["Mary", "Don"]' > guests.json
```

Serving a static file

```
'use strict';

const fs = require('fs');
const path = require('path');
const guestsPath = path.join(__dirname, 'guests.json');

const express = require('express');
const app = express();
const port = process.env.PORT || 8000;

app.disable('x-powered-by');

app.get('/guests', function(req, res) {
  fs.readFile(guestsPath, 'utf8', function(err, guestsJSON) {
    if (err) {
      console.error(err.stack);
      return res.sendStatus(500);
    }

    let guests = JSON.parse(guestsJSON);

    res.send(guests);
  });
});

app.use(function(req, res) {
  res.sendStatus(404);
  // we could reroute anywhere!
});

app.listen(port, function() {
  console.log('Listening on port', port);
});
```


Serving with Dyynamic Routes

```
'use strict';

const fs = require('fs');
const path = require('path');
const guestsPath = path.join(__dirname, 'guests.json');

const express = require('express');
const app = express();
const port = process.env.PORT || 8000;

app.disable('x-powered-by');

app.get('/guests', function(req, res) {
  fs.readFile(guestsPath, 'utf8', function(err, guestsJSON) {
    if (err) {
      console.error(err.stack);
      return res.sendStatus(500);
    }

    let guests = JSON.parse(guestsJSON);

    res.send(guests);
  });
});

app.get('/guests/:id', function(req, res) {
  fs.readFile(guestsPath, 'utf8', function(err, guestsJSON) {
    if (err) {
      console.error(err.stack);
      return res.sendStatus(500);
    }

    let id = Number.parseInt(req.params.id);
    // lets talk about req.params!
    let guests = JSON.parse(guestsJSON);

    if (id < 0 || id >= guests.length || Number.isNaN(id)) {
      return res.sendStatus(404);
    }

    res.set('Content-Type', 'text/plain');
    res.send(guests[id]);
  });
});

app.use(function(req, res) {
  res.sendStatus(404);
});

app.listen(port, function() {
  console.log('Listening on port', port);
});
```

a note on **req** and **res**

you can console.log() them in your routes...theres a lot of information there but req.params is a property on your request, lots of other good info there too!

ie) we'll use req.body when dealing with the data we send in posts posts

Review Objectives

- Explain what Express is.
- Explain why Express is useful.
- Create an HTTP server with the `express` module.
- Handle some static and dynamic routes with Express HTTP server
- propose a solution to how Derek and Hansel could have handled this differently...

Questions?