

Manchester NodeJS Workshop

Getting Started with Express

darrylbrown.co.uk

   = ConsultDarryl

github.com/ConsultDarryl/manchester-node-workshop

<https://www.getpostman.com/>



   = ConsultDarryl

Recap

- ▶ Modules, request, response, npm, building a simple webserver
- ▶ Questions:
 - ▶ `url.parse()`
 - ▶ `const myURL = new URL('/foo', 'https://example.org/');`
`console.log (myURL); // https://example.org/foo`
 - ▶ Note the difference between url and URL
 - ▶ Examples have been updated:
 - ▶ `https://github.com/ConsultDarryl/manchester-node-workshop`
 - ▶ Is there a simple way to stop NodeJS returning a favicon.ico?



M.E.A.N. stack

- ▶ MongoDB (database)
- ▶ Express.js (webserver/html/apis)
- ▶ Angular.js (frontend/application)
- ▶ Node.js (platform)



Why use Express?

- ▶ Build dynamic and static websites
- ▶ Build web API's
- ▶ Simplify common activities e.g. routing, middleware, templating, error handling
- ▶ MVC pattern: Model, View, Controller
- ▶ Fast
- ▶ Many popular frameworks are based on Express



Hello world

- ▶ `npm install express`

```
const express = require('express')
const app = express()
app.get('/', (req, res) => res.send('Hello World!'))
app.listen(8080, () => console.log('http://localhost:8080'))
```



Routing

- ▶ <http://localhost:8080/customer>
- ▶ <http://localhost:8080/customer/25>

- ▶ GET, PUT, POST, DELETE etc.

- ▶ e.g.

```
app.get('/customer', function (request, response) { ... });
```



Postman

- ▶ <https://www.getpostman.com/>
- ▶ Makes it easy to call endpoints (API's)
- ▶ We are going to use it for testing



Middleware Functions

- ▶ Middleware functions are just JavaScript functions with three parameters:

```
function(req, res, next) {  
    // do something useful  
    next();  
}
```

- ▶ Middleware functions can:
 - ▶ Execute any code
 - ▶ Make changes to the request and the response objects
 - ▶ End the request-response cycle
 - ▶ Call the next middleware in the stack

Error Handling

- ▶ The Default Error Handler
 - ▶ Built-in
 - ▶ Automatically added at the end of the middleware function stack
- ▶ Create custom error-handling middleware functions
- ▶ error-handling functions have four arguments instead of three

```
app.use(function (err, req, res, next) {  
  console.error(err.stack)  
  res.status(500).send('Something broke!')  
})
```



Express application generator

- ▶ `npm install express-generator -g`
- ▶ `express --view=pug --git my-first-express-app`
- ▶ `cd my-first-express-app`
- ▶ `npm install`
- ▶ `set DEBUG=my-first-express-app:* & npm start`
- ▶ <http://localhost:3000/>




Darryl Brown

Modernising Websites with Angular



► Get in touch

- darrylbrown.co.uk/contact
-    = ConsultDarryl

► Suggested voluntary contribution towards organisation costs: £2

- The Manchester Node Workshop couldn't exist without the generous support of MadLab. Please make a donation because it enables MadLab to carry on providing spaces for meetups and enables MadLab to deliver free digital training and support to the people of Greater Manchester.

