

Build a REST API

NODE | EXPRESS | MONGO

What is HTTP...

HyperText Transfer Protocol

- Application Layer Protocol
- Built on top of TCP/IP protocol
- Rules for transferring resources
- Every HTTP Request is executed independently without the knowledge of requests that came before it

...What is HTTP

HyperText Transfer Protocol

- HTTP is stateless
- TCP/IP is not stateless
- Payload (body/data) can be anything as long as it is defined in header

...What is HTTP

HyperText Transfer Protocol

- HTTP is stateless
- TCP/IP is not stateless
- Payload (body/data) can be anything as long as it is defined in header

What is REST...

Representational State Transfer

- Architectural Pattern with design guidelines
- HTTP is usually the underlying protocol
- Use HTTP methods explicitly
- Every RESTful resource has a unique ID

...What is REST

Representational State Transfer

- Client state is not persisted between requests
- Caching Policy for responses
- Separation of concerns between clients and servers
- Layered system

What is an API...

Application Programming Interface

- Defines server-side functions that are supported
- Where requests should be made
- Format of the request and response

...What is an API

Application Programming
Interface

- No standard way of writing APIs
- REST provides guidelines
- CRUD Operations
- Create | Read | Update | Delete

IDE

Integrated Development Environment

- VS Code
- Atom
- Webstorm (Paid / Trial)

Node.js

Javascript Runtime

- Node.js 8.9.3+
- NPM (Node Package Manager)
- Using natively supported ES6/7
- <https://node.green>

MongoDB

NoSQL Database

- Local Installation
- mLab Cloud
- Docker

REST Client

API Testing

- Insomnia
- Postman

REST Client

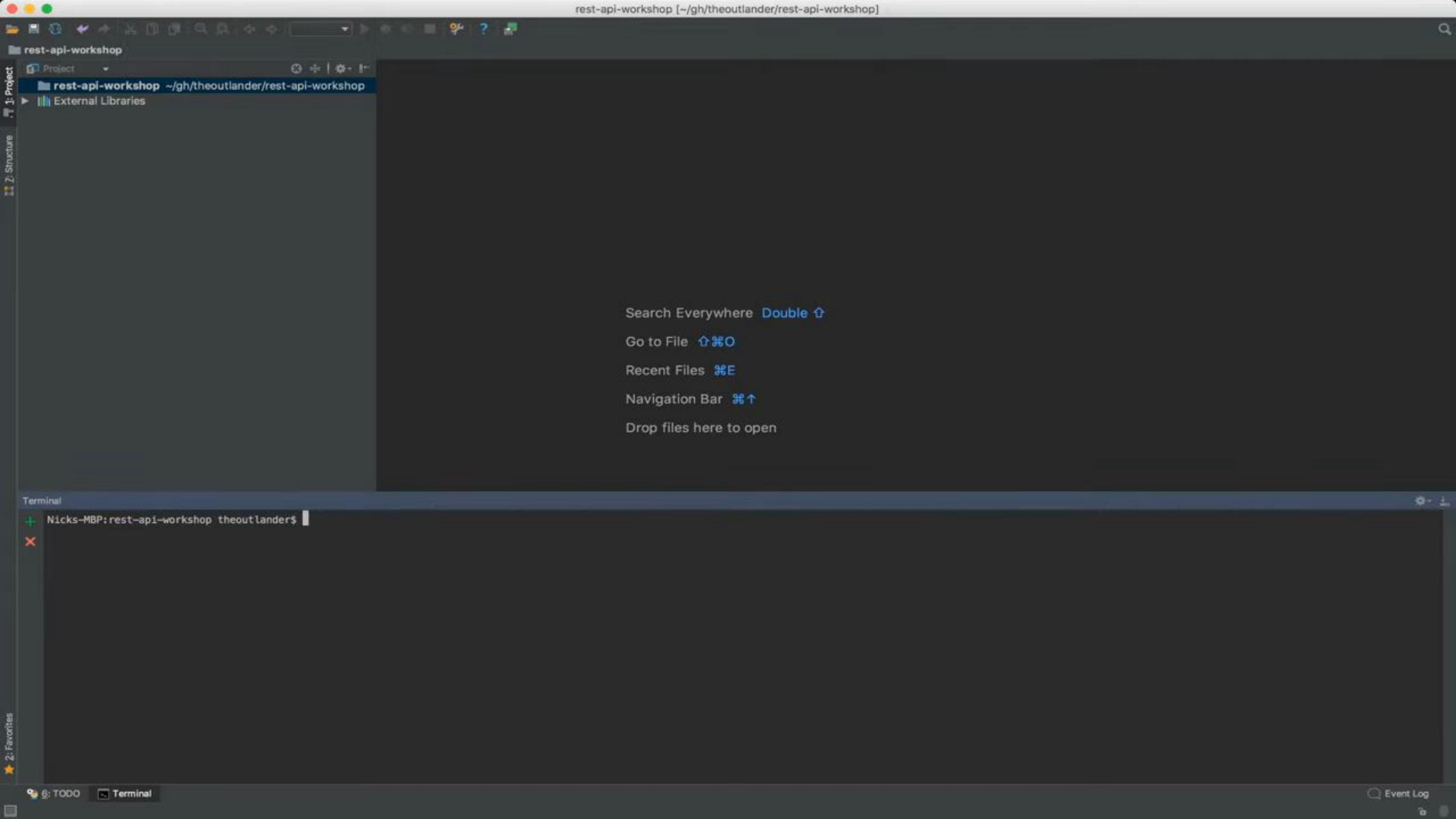
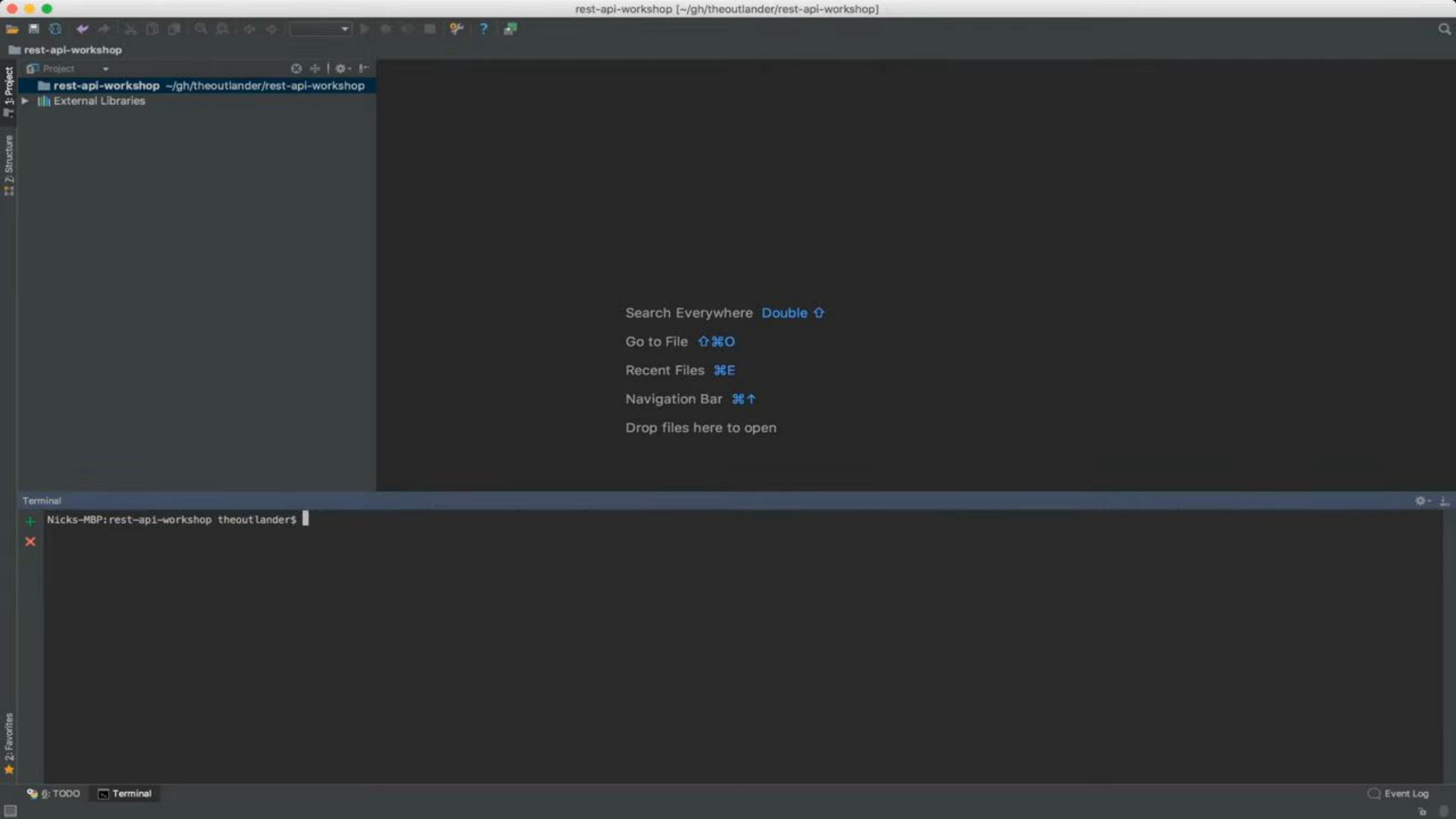
API Testing

- Insomnia
- Postman

Application Setup

Getting Started

- Project Folder
- Initialize Node.js Project
- Hello World
- Startup Scripts



The screenshot shows a dark-themed IDE interface with multiple windows. The main window displays the contents of a `package.json` file. The code is as follows:

```
1  {
2    "name": "rest-api-workshop",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \\\"Error: no test specified\\\" && exit 1"
8    },
9    "keywords": [],
10   "author": "",
11   "license": "ISC"
12 }
13 
```

Terminal

```
+ Nicks-MBP:rest-api-workshop theoutlander$ npm init -y
Wrote to /Users/theoutlander/gh/theoutlander/rest-api-workshop/package.json:
```

```
{
  "name": "rest-api-workshop",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

```
Nicks-MBP:rest-api-workshop theoutlander$
```

2: Favorites

6: TODO

Terminal

8:5 LF: UTF-8

Event Log

The screenshot shows a Java IDE interface with the following components:

- Project View:** Shows a tree structure for the "rest-api-workshop" project, which includes a "src" folder containing "package.json" and "index.js".
- Code Editor:** Displays the "index.js" file with the single line of code: `console.log('Hello World')`.
- Terminal:** Shows the command line output of running the application. It starts with the command `npm init -y`, followed by the creation of a package.json file at the specified path. The terminal then displays the contents of the package.json file, which is a JSON object with the following properties:

```
{  "name": "rest-api-workshop",  "version": "1.0.0",  "description": "",  "main": "index.js",  "scripts": {    "test": "echo \\\"Error: no test specified\\\" && exit 1"  },  "keywords": [],  "author": "",  "license": "ISC"}  
Nicks-MBP:rest-api-workshop theoutlander$ node src/index.js
```
- Status Bar:** Shows the status bar with "6: TODO", "Terminal", "Event Log", and system information like "1.27 n/a UTF-8".

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../package.json [rest-api-workshop]

rest-api-workshop package.json

Project rest-api-workshop ~/gh/theoutlander/rest-api-workshop src package.json External Libraries

1 {
2 "name": "rest-api-workshop",
3 "version": "1.0.0",
4 "description": "",
5 "main": "index.js",
6 "scripts": {
7 "start": "node src/index.js",
8 "test": "echo \\\"Error: no test specified\\\" && exit 1"
9 },
10 "keywords": [],
11 "author": "",
12 "license": "ISC"
13}
14

scripts > start

Terminal

Nicks-MBP:rest-api-workshop theoutlander\$ npm star

Event Log

Express

Web Server

- Install Dependency
- Reference the express library
- Create express application instance

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../package.json [rest-api-workshop]

Project rest-api-workshop package.json

rest-api-workshop ~/gh/theoutlander/rest-api-workshop

node_modules library root

src

package.json

package-lock.json

External Libraries

package.json

index.js

```
1  {
2      "name": "rest-api-workshop",
3      "version": "1.0.0",
4      "description": "",
5      "main": "index.js",
6      "scripts": {
7          "start": "node src/index.js",
8          "test": "echo \\\"Error: no test specified\\\" && exit 1"
9      },
10     "keywords": [],
11     "author": "",
12     "license": "ISC",
13     "dependencies": {
14         "express": "^4.16.3"
15     }
16 }
17
```

dependencies

Terminal

```
+ > rest-api-workshop@1.0.0 start /Users/theoutlander/gh/theoutlander/rest-api-workshop
> node src/index.js

Hello World
Nicks-MBP:rest-api-workshop theoutlander$ npm run start

> rest-api-workshop@1.0.0 start /Users/theoutlander/gh/theoutlander/rest-api-workshop
> node src/index.js

Hello World
Nicks-MBP:rest-api-workshop theoutlander$ npm install express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm warn rest-api-workshop@1.0.0 No description
npm warn rest-api-workshop@1.0.0 No repository field.

+ express@4.16.3
added 49 packages from 47 contributors in 2.784s
Nicks-MBP:rest-api-workshop theoutlander$
```

Favorites

Event Log

The screenshot shows a dark-themed IDE interface with the following components:

- Project View:** Shows the project structure under "rest-api-workshop".
- Code Editor:** Displays `index.js` with the following code:

```
1 let express = require('express')
2
3 let app = express()
```
- Terminal:** Shows the command-line output of running the application.

```
> rest-api-workshop@1.0.0 start /Users/theoutlander/gh/theoutlander/rest-api-workshop
> node src/index.js

Hello World
Nicks-MBP:rest-api-workshop theoutlander$ npm run start

> rest-api-workshop@1.0.0 start /Users/theoutlander/gh/theoutlander/rest-api-workshop
> node src/index.js

Hello World
Nicks-MBP:rest-api-workshop theoutlander$ npm install express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm warn rest-api-workshop@1.0.0 No description
npm warn rest-api-workshop@1.0.0 No repository field.

+ express@4.16.3
added 49 packages from 47 contributors in 2.784s
Nicks-MBP:rest-api-workshop theoutlander$
```
- Bottom Status Bar:** Shows "6: TODO" and "Event Log".

The screenshot shows a dark-themed IDE interface with the following components:

- Project View:** On the left, it shows the project structure under "rest-api-workshop". It includes a "node_modules" folder, a "public" folder, and a "src" folder which contains "index.js", "package.json", and "package-lock.json".
- Code Editor:** The main area displays the contents of "index.js". The code is as follows:

```
let express = require('express')
let app = express()
```

- Terminal:** At the bottom, the terminal window shows the command-line history and output:

```
> rest-api-workshop@1.0.0 start /Users/theoutlander/gh/theoutlander/rest-api-workshop
> node src/index.js

Hello World
Nicks-MBP:rest-api-workshop theoutlander$ npm run start

> rest-api-workshop@1.0.0 start /Users/theoutlander/gh/theoutlander/rest-api-workshop
> node src/index.js

Hello World
Nicks-MBP:rest-api-workshop theoutlander$ npm install express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm warn rest-api-workshop@1.0.0 No description
npm warn rest-api-workshop@1.0.0 No repository field.

+ express@4.16.3
added 49 packages from 47 contributors in 2.784s
Nicks-MBP:rest-api-workshop theoutlander$
```
- Bottom Bar:** The bottom bar includes tabs for "TODO" and "Terminal", and status indicators for "Event Log" and "5:1 LF: UTF-8".

The screenshot shows a dark-themed IDE interface with the following components:

- Project View:** Shows the project structure under "rest-api-workshop".
- Code Editor:** Displays the contents of `index.js`. The code uses the Express framework to serve static files from the `public` directory and listen on port 3000.
- Terminal:** Shows the command-line output of running the application. It includes the command `npm run start`, the application's response "Hello World", and the output of `npm install express`.
- Status Bar:** Includes the date and time (7:14), language (UTF-8), and file encoding (Event Log).

```
let express = require('express')
let app = express()
app.use(express.static('public'))
const PORT = |
app.listen(3000)
```

```
+ rest-api-workshop@1.0.0 start /Users/theoutlander/gh/theoutlander/rest-api-workshop
+ node src/index.js

Hello World
Nicks-MBP:rest-api-workshop theoutlander$ npm run start

> rest-api-workshop@1.0.0 start /Users/theoutlander/gh/theoutlander/rest-api-workshop
> node src/index.js

Hello World
Nicks-MBP:rest-api-workshop theoutlander$ npm install express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm warn rest-api-workshop@1.0.0 No description
npm warn rest-api-workshop@1.0.0 No repository field.

+ express@4.16.3
added 49 packages from 47 contributors in 2.784s
Nicks-MBP:rest-api-workshop theoutlander$
```

A screenshot of a dark-themed IDE interface, likely WebStorm or similar, showing a Node.js project structure and code editor.

Project Structure:

- rest-api-workshop
- node_modules
- public
- src
 - index.js
 - package.json
 - package-lock.json
- External Libraries

Code Editor (index.js):

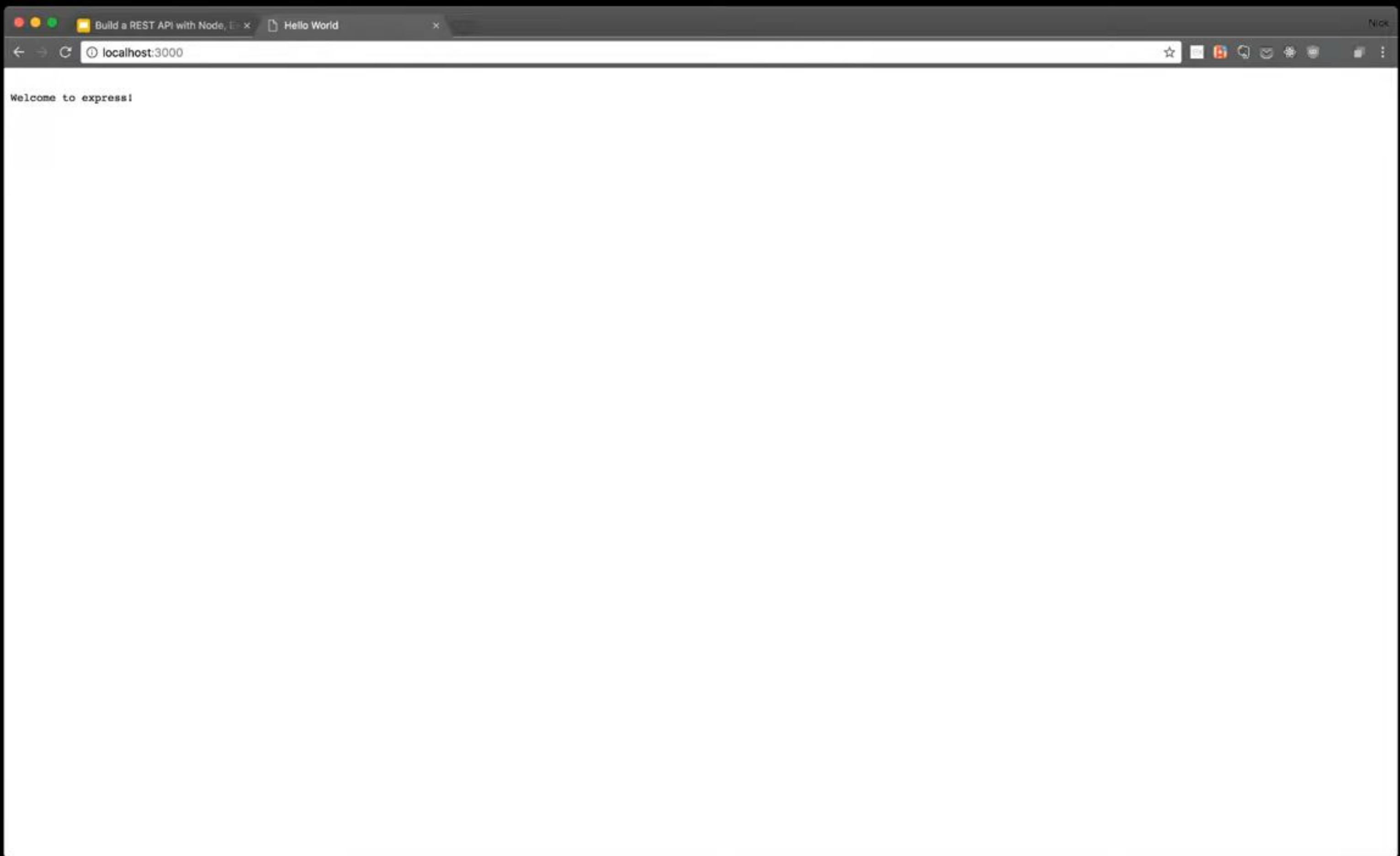
```
1 let express = require('express')
2
3 let app = express()
4
5 app.use(express.static('public'))
6
7 const PORT = process.env.PORT || 3000
8 app.listen(PORT, () => console.info(`Server has started on ${PORT}`))
```

Terminal:

```
Nicks-MBP:rest-api-workshop theoutlander$ node
```

Bottom Status Bar:

- TODO
- Terminal
- Event Log



Route

Mount to Express

- GET Request
- Request Parameters
- Query Parameters

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/person.js [rest-api-workshop]

rest-api-workshop src routes person.js

Project rest-api-workshop ~/gh/theoutlander/rest-api-workshop node_modules library root public src routes person.js Index.js package.json package-lock.json External Libraries

Structure

Terminal

```
Nicks-MBP:rest-api-workshop theoutlander$ npm start
> rest-api-workshop@1.0.0 start /Users/theoutlander/gh/theoutlander/rest-api-workshop
> node src/index.js

Server has started on 3000
```

TODO Terminal Event Log

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/person.js [rest-api-workshop]

Project

rest-api-workshop ~/gh/theoutlander/rest-api-workshop

- node_modules library root
- public
- src
 - routes
 - person.js
 - index.js
 - package.json
 - package-lock.json

External Libraries

Structure

1 package.json x 2 index.js x 3 person.js x 4 index.html x

```
1 let path: PathParams, ... handlers: RequestHandler
2 let path: PathParams, ... handlers: RequestHandlerParams
3
4 router.get('/person', )
5
6 module.exports = router
7
```

Terminal

```
+ Nicks-MBP:rest-api-workshop theoutlander$ npm start
> rest-api-workshop@1.0.0 start /Users/theoutlander/gh/theoutlander/rest-api-workshop
> node src/index.js

Server has started on 3000
```

Event Log

A screenshot of a code editor interface, likely IntelliJ IDEA, displaying a Node.js application named "rest-api-workshop".

The left sidebar shows the project structure:

- Project: rest-api-workshop (~/gh/theoutlander/rest-api-workshop)
- node_modules (library root)
- public
- src
 - routes
 - person.js
 - index.js
- package.json
- package-lock.json

The right pane shows the content of the file `src/index.js`:

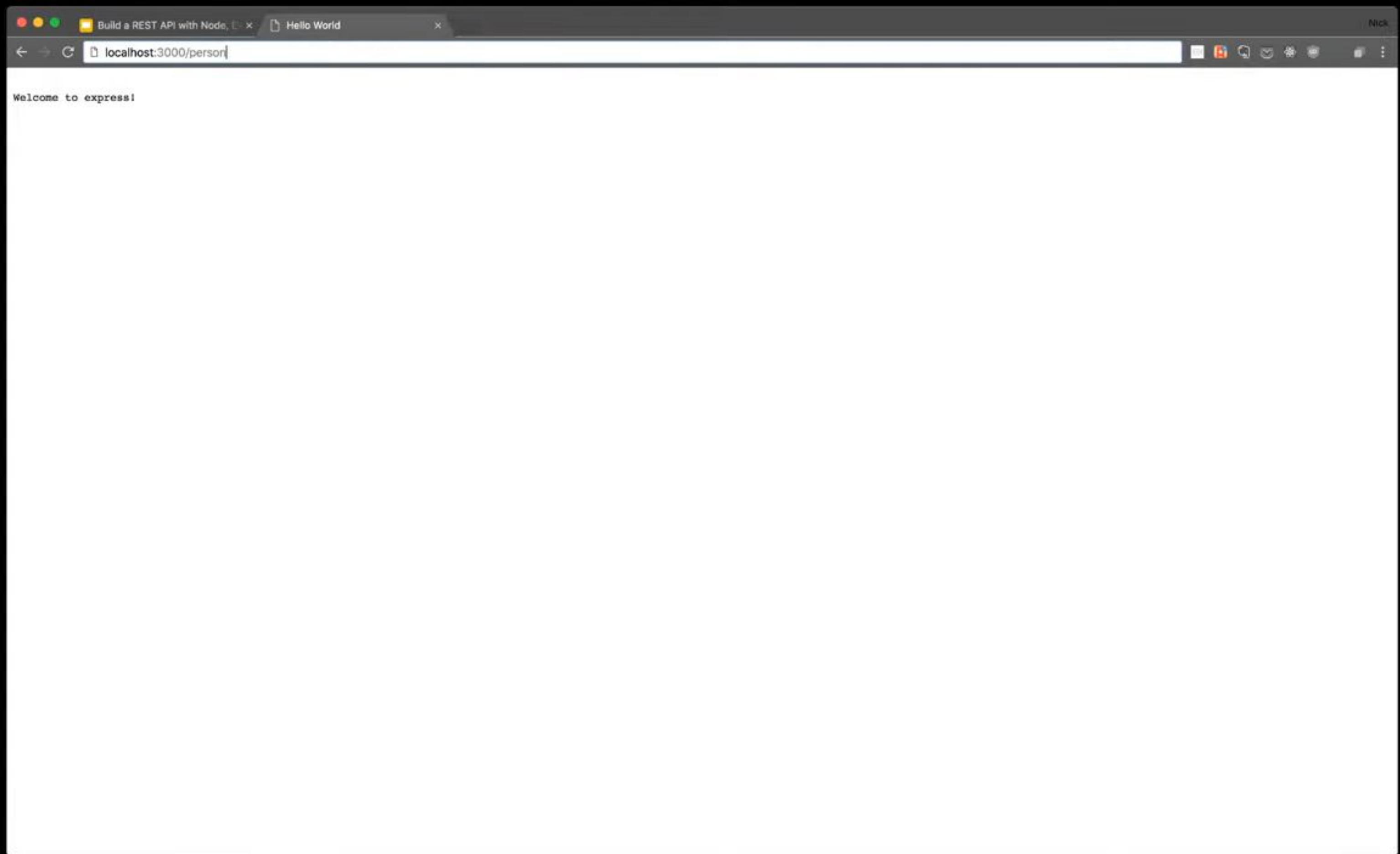
```
1 let express = require('express')
2
3 let app = express()
4
5 let personRoute = [
6   app.use(express.static('public'))
7
8 const PORT = process.env.PORT || 3000
9 app.listen(PORT, () => console.info(`Server has started on ${PORT}`))
```

The word `personRoute` is highlighted in blue.

The bottom-left terminal window shows the command line output:

```
Nicks-MBP:rest-api-workshop theoutlander$ npm start
> rest-api-workshop@1.0.0 start /Users/theoutlander/gh/theoutlander/rest-api-workshop
> node src/index.js
Server has started on 3000
```

The bottom status bar indicates the time as 5:19 and the encoding as UTF-8.





Insomnia

GET https://api.myproduct.com/v1/users:

Send

No Environment

Cookies

Filter

- GET
 - POST
 - PUT
 - PATCH
 - DELETE
 - OPTIONS
 - HEAD
-
- Custom Method

REST API Workshop

GET GET Person

Projects



Select a body type from above

Send Request

⌘Enter

Focus Url Bar

⌘L

Manage Cookies

⌘K

Edit Environments

⌘E



No Environment

Cookies

GET

Query

Header

Docs

Filter:



REST API Workshop

GET GET Person

Projects

POST

PUT

PATCH

DELETE

OPTIONS

HEAD

Custom Method



Select a body type from above

Send Request

⌘Enter

Focus Url Bar

⌘L

Manage Cookies

⌘K

Edit Environments

⌘E

Insomnia - GET Person

Send 200 OK TIME 5.44 ms SIZE 27 B

No Environment Cookies Body Auth Query Header Docs Preview Header Cookie Timeline

You have requested a person

REST API Workshop

GET Person

Projects



Select a body type from above

Insomnia - GET Person

GET http://localhost:3000/person

Send 200 OK 2019-07-14T00:21:19 5026-2119

No Environment Cookies Body

REST API Workshop

GET Person

Projects

Generate Client Code

Node.js Request

Copy to Clipboard

```
1 var request = require("request");
2
3 var jar = request.jar();
4 jar.setCookie(request.cookie("connect.sid=s%253A8Iq233D3GSmRRHoqqAX4t_7sV1ZNBAfe.6Ys4HkUv0h91dGSvXVoSzRSv7MzXcgFZC4zU04jbb8Y"), "http://localhost:3000/person");
5
6 var options = { method: 'GET',
7   url: 'http://localhost:3000/person',
8   jar: 'JAR' };
9
10 request(options, function (error, response, body) {
11   if (error) throw new Error(error);
12
13   console.log(body);
14 });
15
```

* Code snippets generated by httpsnippet

Done

The screenshot shows a software development environment with a dark theme. At the top, there's a tab bar with the title "rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../package.json [rest-api-workshop]" and several other tabs. Below the tabs is a project structure sidebar titled "Project" and "Structure". The "Project" section shows a tree view of the "rest-api-workshop" directory, which includes "node_modules", "public", "src" (containing "routes", "person.js", "index.js", "package.json", and "package-lock.json"), and "External Libraries". The "Structure" section shows the same directory structure. In the main workspace, there are four tabs: "package.json", "index.js", "person.js", and "index.html". The "package.json" tab is active, displaying the following JSON code:

```
1  {
2    "name": "rest-api-workshop",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "start": "node src/index.js",
8      "test": "echo \\\"Error: no test specified\\\" && exit 1"
9    },
10   "keywords": [],
11   "author": "",
12   "license": "ISC",
13   "dependencies": {
14     "express": "^4.16.3"
15   }
16 }
17
```

Below the workspace is a "Terminal" window showing command-line history:

```
Nicks-MBP:rest-api-workshop theoutlander$ npm start
> rest-api-workshop@1.0.0 start /Users/theoutlander/gh/theoutlander/rest-api-workshop
> node src/index.js

Server has started on 3000
^C

Nicks-MBP:rest-api-workshop theoutlander$ npm start
> rest-api-workshop@1.0.0 start /Users/theoutlander/gh/theoutlander/rest-api-workshop
> node src/index.js

Server has started on 3000
^C

Nicks-MBP:rest-api-workshop theoutlander$ npm i nodemon
```

At the bottom of the interface, there are several status indicators: "TODO" (with a count of 46), "Terminal", "Event Log", and "46 chars, 2 line breaks 15:4 LF: UTF-8".

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../package.json [rest-api-workshop]

rest-api-workshop package.json

Project rest-api-workshop ~/gh/theoutlander/rest-api-workshop node_modules library root public src routes person.js index.js package.json package-lock.json External Libraries

package.json index.js person.js index.html

```
1  {
2      "name": "rest-api-workshop",
3      "version": "1.0.0",
4      "description": "",
5      "main": "index.js",
6      "scripts": {
7          "start": "node src/index.js",
8          "test": "echo \\"Error: no test specified\\" && exit 1"
9      },
10     "keywords": [],
11     "author": "",
12     "license": "ISC",
13     "dependencies": {
14         "express": "^4.16.3"
15     }
16 }
17 }
```

Terminal

```
+ > fsevents@1.1.3 install /Users/theoutlander/gh/theoutlander/rest-api-workshop/node_modules/fsevents
> node install

[fsevents] Success: "/Users/theoutlander/gh/theoutlander/rest-api-workshop/node_modules/fsevents/lib/binding/Release/node-v57-darwin-x64/fse.node" already installed
Pass --update-binary to reinstall or --build-from-source to recompile

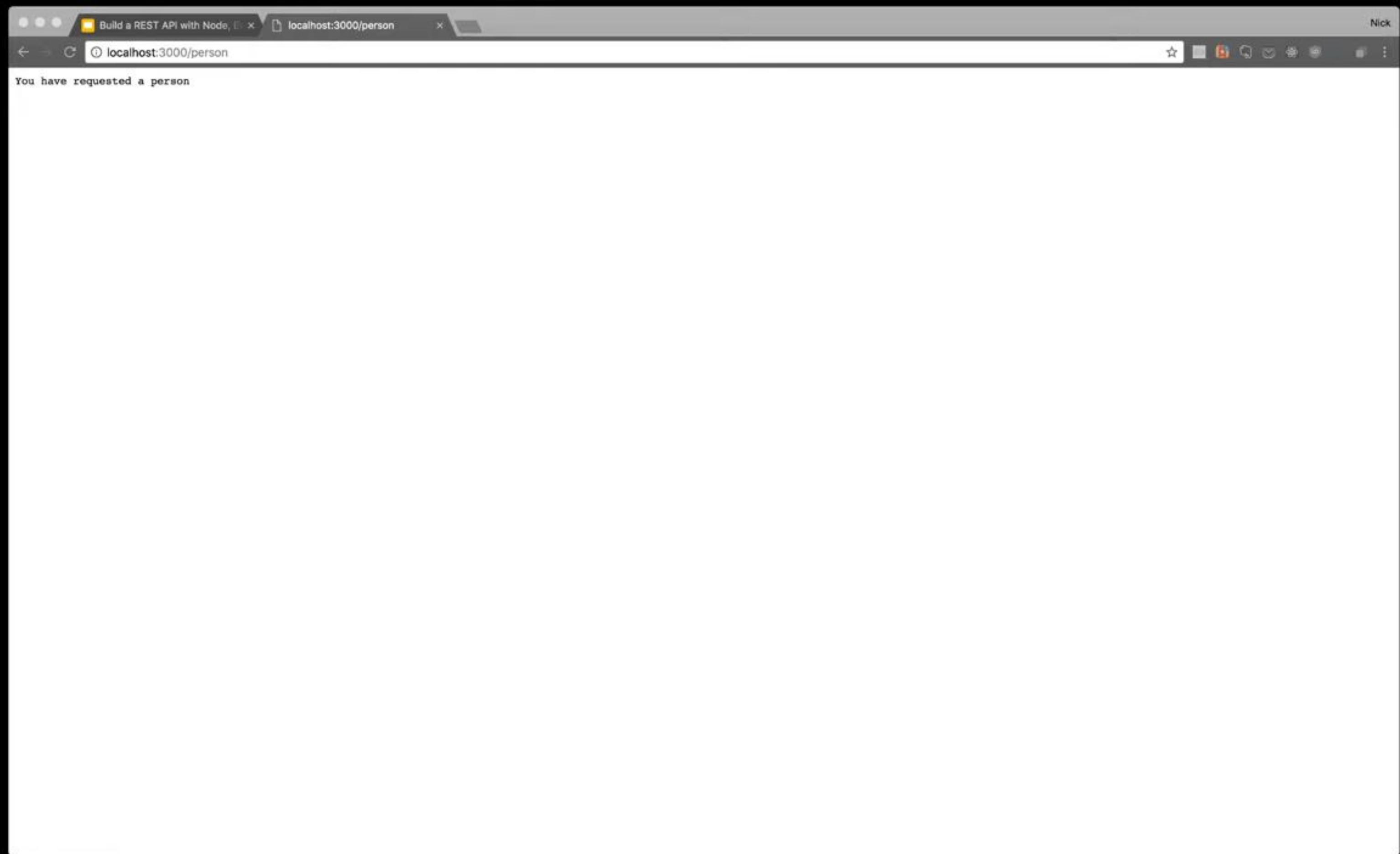
> nodemon@1.17.2 postinstall /Users/theoutlander/gh/theoutlander/rest-api-workshop/node_modules/nodemon
> node -e "console.log('"\u001b[32mLove nodemon? You can now support the project via the open collective:\u001b[22m\u001b[39m\n > \u001b[96m\u001b[1mhttps://opencollective.com/nodemon/donate\u001b[0m'\n')" || exit 0

Love nodemon? You can now support the project via the open collective:
> https://opencollective.com/nodemon/donate

npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ nodemon@1.17.2
added 355 packages from 201 contributors in 11.914s
Nicks-MBP:rest-api-workshop theoutlander$ npm i nodemon -D
```

6: TODO Terminal Event Log



rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/person.js [rest-api-workshop]

Project

rest-api-workshop -/gh/theoutlander/rest-api-workshop

- node_modules library root
- public
- src
 - routes
 - person.js
 - index.js
- package.json
- package-lock.json

External Libraries

File

Project

Structure

person.js

package.json index.js person.js index.html

```
1 let express = require('express')
2 let router = express.Router()
3
4 router.get('/person', (req, res) => {
5   res.send('You have requested a person')
6 })
7
8 router.get('/person/:name', (req, res) => {
9   res.send('You have requested a person')
10 })
11
12
13
14
15 module.exports = router
```

Terminal

```
Nicks-MBP:rest-api-workshop theoutlander$ npm run start-watch
> rest-api-workshop@1.0.0 start-watch /Users/theoutlander/gh/theoutlander/rest-api-workshop
> nodemon src/index.js

[nodemon] 1.17.2
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: ***!
[nodemon] starting 'node src/index.js'
Server has started on 3000
```

TODO Terminal Event Log

8:26 LF: UTF-8: a

The screenshot shows a code editor interface with several windows open. The left sidebar displays the project structure:

- rest-api-workshop
- src
- routes
- person.js

The main editor window contains the file `person.js` with the following code:

```
1  let express = require('express')
2  let router = express.Router()
3
4  router.get('/person', (req, res) => {
5      res.send('You have requested a person')
6  })
7
8  router.get('/person/:name', (req, res) => {
9      res.send(`You have requested a person ${req.params.name}`)
10 }
11
12
13
14
15 module.exports = router
```

A tooltip "callback for get()" is visible near the first `router.get` line.

The bottom terminal window shows the command output:

```
Nicks-MBP:rest-api-workshop theoutlander$ npm run start-watch
> rest-api-workshop@1.0.0 start-watch /Users/theoutlander/gh/theoutlander/rest-api-workshop
> nodemon src/index.js

[nodemon] 1.17.2
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: ***!
[nodemon] starting 'node src/index.js'
Server has started on 3000
[nodemon] restarting due to changes...
[nodemon] starting 'node src/index.js'
Server has started on 3000
```

The screenshot shows a code editor interface with a dark theme. On the left, the project structure is displayed:

- rest-api-workshop (library root)
- node_modules
- public
- src
 - routes
 - person.js
 - index.js
 - package.json
 - package-lock.json
- External Libraries

The current file being edited is `person.js`, which contains the following code:

```
1  let express = require('express')
2  let router = express.Router()
3
4  router.get('/person', (req, res) => {
5      res.send('You have requested a person')
6  })
7
8  router.get('/person/:name', (req, res) => {
9      res.send(`You have requested a person ${req.params.name}`)
10 }
11
12
13
14
15
16 module.exports = router
```

In the bottom-left corner, there is a terminal window showing the command-line output:

```
+ Nicks-MBP:rest-api-workshop theoutlander$ npm run start-watch
> rest-api-workshop@1.0.0 start-watch /Users/theoutlander/gh/theoutlander/rest-api-workshop
> nodemon src/index.js

[nodemon] 1.17.2
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: ***!
[nodemon] starting 'node src/index.js'
Server has started on 3000
[nodemon] restarting due to changes...
[nodemon] starting 'node src/index.js'
Server has started on 3000
```

The status bar at the bottom indicates "Expression statement is not assignment or call".

The screenshot shows a code editor interface with several windows open. The top window displays the command line output of an npm run command. Below it, the main editor area shows the content of a file named `person.js` under the `routes` directory of a project named `rest-api-workshop`. The code in `person.js` defines an Express Router for handling requests to '/person'. It includes logic to respond with the name if a query parameter 'name' is present, or a general message otherwise. It also handles a route '/person/:name' using a params object. The code ends with an export statement for the router.

```
rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/person.js [rest-api-workshop]
Project
rest-api-workshop -/gh/theoutlander/rest-api-workshop
  node_modules library root
  public
  src
    routes
      person.js
      index.js
    package.json
    package-lock.json
External Libraries

package.json x index.js x person.js x index.html x
1 let express = require('express')
2 let router = express.Router()
3
4 // QueryString => query property on the request object
5 router.get('/person', (req, res) => {
6   if(req.query.name) {
7     |
8   }
9   else
10  {
11    res.send('You have requested a person')
12  }
13})
14
15 // Params property on the request object
16 router.get('/person/:name', (req, res) => {
17   res.send(`You have requested a person ${req.params.name}`)
18 })
19
20
21
22
23 module.exports = router

callback for get()

Terminal
+ Nicks-MBP:rest-api-workshop theoutlander$ npm run start-watch
> rest-api-workshop@1.0.0 start-watch /Users/theoutlander/gh/theoutlander/rest-api-workshop
> nodemon src/index.js

[nodemon] 1.17.2
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: ***!
[nodemon] starting 'node src/index.js'
Server has started on 3000
[nodemon] restarting due to changes...
[nodemon] starting 'node src/index.js'
Server has started on 3000

Event Log
```

The screenshot shows a code editor interface with several windows open. The top window displays the command line output of a terminal session. Below it, the main workspace shows the project structure and the content of the `person.js` file.

Project Structure:

- rest-api-workshop
- src
- routes
- person.js
- index.js
- package.json
- package-lock.json

Terminal Output:

```
Nicks-MBP:rest-api-workshop theoutlander$ npm run start-watch
> rest-api-workshop@1.0.0 start-watch /Users/theoutlander/gh/theoutlander/rest-api-workshop
> nodemon src/index.js

[nodemon] 1.17.2
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: ***!
[nodemon] starting 'node src/index.js'
Server has started on 3000
[nodemon] restarting due to changes...
[nodemon] starting 'node src/index.js'
Server has started on 3000
```

Code Editor Content (person.js):

```
1  let express = require('express')
2  let router = express.Router()
3
4  // QueryString => query property on the request object
5  // localhost:3000/person?name=thomas&age=20
6  router.get('/person', (req, res) => {
7    if(req.query.name) {
8      res.send(`You have requested a person ${req.query.name}`)
9    }
10   else {
11     res.send('You have requested a person')
12   }
13 })
14
15 // Params property on the request object
16 router.get('/person/:name', (req, res) => {
17   res.send(`You have requested a person ${req.params.name}`)
18 })
19
20
21
22
23 module.exports = router
```



No Environment

Cookies

Body

Auth

Query

Header

Docs

Preview

Header

Cookie

Timeline

Filter

②

REST API Workshop

GET GET Person

GET GET Person by QueryString

GET GET Person by Name

Projects



Select a body type from above

Middleware

Functions that execute serially

- Access to request/response objects and the next middleware function in the pipeline
- Execute any code
- Make changes to the request and the response objects
- End the request-response cycle
- Call the next middleware function in the stack

A screenshot of a code editor interface showing a Node.js project structure and its source code.

The project structure on the left includes:

- rest-api-workshop
- node_modules
- public
- src
 - routes
 - person.js
 - index.js
- package.json
- package-lock.json

The current file being edited is `index.js`, which contains the following code:

```
let express = require('express')
let app = express()
let personRoute = require('./routes/person')
app.use(personRoute)
app.use(express.static('public'))
const PORT = process.env.PORT || 3000
app.listen(PORT, () => console.info(`Server has started on ${PORT}`))
```

In the bottom terminal window, the command `npm run start-watch` is being run, and the output shows the server starting on port 3000 and restarting due to changes:

```
Nicks-MBP:rest-api-workshop theoutlander$ npm run start-watch
> rest-api-workshop@1.0.0 start-watch /Users/theoutlander/gh/theoutlander/rest-api-workshop
> nodemon src/index.js

[nodemon] 1.17.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: ***!
[nodemon] starting 'node src/index.js'
Server has started on 3000
[nodemon] restarting due to changes...
[nodemon] starting 'node src/index.js'
Server has started on 3000
[nodemon] restarting due to changes...
[nodemon] starting 'node src/index.js'
Server has started on 3000
```

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/index.js [rest-api-workshop]

rest-api-workshop src Index.js

Project rest-api-workshop - /gh/theoutlander/rest-api-workshop node_modules library root public src routes person.js index.js package.json package-lock.json External Libraries

package.json index.js person.js index.html

```
1 let express = require('express')
2
3 let app = express()
4
5 let personRoute = require('./routes/person')
6
7 app.use((req, res, next) => {
8   console.log()
9 })
10
11 app.use(personRoute)
12 app.use(express.static('public'))
13
14 const PORT = process.env.PORT || 3000
15 app.listen(PORT, () => console.info(`Server has started on ${PORT}`))
```

callback for use()

Terminal

```
[nodemon] starting 'node src/index.js'
+ /Users/theoutlander/gh/theoutlander/rest-api-workshop/node_modules/express/lib/application.js:210
  throw new TypeError('app.use() requires a middleware function')
^

TypeError: app.use() requires a middleware function
  at Function.use (/Users/theoutlander/gh/theoutlander/rest-api-workshop/node_modules/express/lib/application.js:210:11)
  at Object.<anonymous> (/Users/theoutlander/gh/theoutlander/rest-api-workshop/src/index.js:7:5)
  at Module._compile (module.js:635:30)
  at Object.Module._extensions..js (module.js:646:10)
  at Module.load (module.js:554:32)
  at tryModuleLoad (module.js:497:12)
  at Function.Module._load (module.js:489:3)
  at Function.Module.runMain (module.js:676:10)
  at startup (bootstrap_node.js:187:16)
  at bootstrap_node.js:608:3
[nodemon] app crashed - waiting for file changes before starting...
```

TODO Terminal Event Log

The screenshot shows a code editor interface with a dark theme. On the left, the project structure is displayed under the 'Project' tab, showing a folder named 'rest-api-workshop' containing 'node_modules', 'public', and 'src' directories. Inside 'src', there are 'routes' and 'index.js' files, along with 'package.json' and 'package-lock.json'. The 'Structure' tab is also visible. The main workspace shows an 'index.js' file being edited:

```
1 let express = require('express')
2
3 let app = express()
4
5 let personRoute = require('./routes/person')
6
7 app.use((req, res, next) => {
8     console.log(`new Date().toString() => ${req.originalUrl}`)
9
10    next()
11})
12
13 app.use(personRoute)
14 app.use(express.static('public'))
15
16 const PORT = process.env.PORT || 3000
17 app.listen(PORT, () => console.info(`Server has started on ${PORT}`))
```

In the bottom-left corner, a terminal window displays the following error message:

```
TypeError: app.use() requires a middleware function
  at Function.use (/Users/theoutlander/gh/theoutlander/rest-api-workshop/node_modules/express/lib/application.js:210:11)
  at Object.<anonymous> (/Users/theoutlander/gh/theoutlander/rest-api-workshop/src/index.js:7:5)
  at Module._compile (module.js:635:30)
  at Object.Module._extensions..js (module.js:646:10)
  at Module.load (module.js:554:32)
  at tryModuleLoad (module.js:497:12)
  at Function.Module._load (module.js:489:3)
  at Function.Module.runMain (module.js:676:10)
  at startup (bootstrap_node.js:187:16)
  at bootstrap_node.js:608:3
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting 'node src/index.js'
Server has started on 3000
```

A screenshot of a code editor (IntelliJ IDEA) displaying a Node.js application structure and its source code.

Project Structure:

- rest-api-workshop
- node_modules
- public
- src
 - routes
 - person.js
 - index.js
 - package.json
 - package-lock.json
- External Libraries

Code Editor (Index.js):

```
1 let express = require('express')
2
3 let app = express()
4
5 let personRoute = require('./routes/person')
6
7 app.use((req, res, next) => {
8     console.log(`new Date().toString() => ${req.originalUrl}`)
9     next()
10})
11
12 app.use(personRoute)
13 app.use(express.static('public'))
14
15 const PORT = process.env.PORT || 3000
16 app.listen(PORT, () => console.info(`Server has started on ${PORT}`))
```

Terminal:

```
+ Thu Mar 22 2018 10:49:46 GMT-0700 (PDT) => /person/thomas
```

Bottom Status Bar:

- TODO
- Terminal
- Event Log

Error Pages

Custom Handlers

- 404 Not Found Handler
- 500 Internal Server Error

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/index.js [rest-api-workshop]

The screenshot shows a code editor interface with the following details:

- Project Structure:** The left sidebar displays the project structure under "rest-api-workshop". It includes a "node_modules" folder, a "public" folder, and a "src" folder containing "routes", "person.js", and "index.js".
- Code Editor:** The main area shows the content of "index.js". The code initializes an Express app, uses a person route, and handles 404 errors. A tooltip "callback for use()" is visible near the "next()" call.
- Terminal:** At the bottom, a terminal window shows the output of a command: "Thu Mar 22 2018 10:49:46 GMT-0700 (PDT) => /person/thomas".

```
let express = require('express')
let app = express()
let personRoute = require('./routes/person')

app.use((req, res, next) => {
  console.log(`new Date().toString() => ${req.originalUrl}`)
  next()
})

app.use(personRoute)
app.use(express.static('public'))

// Handler for 404 - Resource Not Found
app.use((req, res, next) => {
  res.status(404).|
})

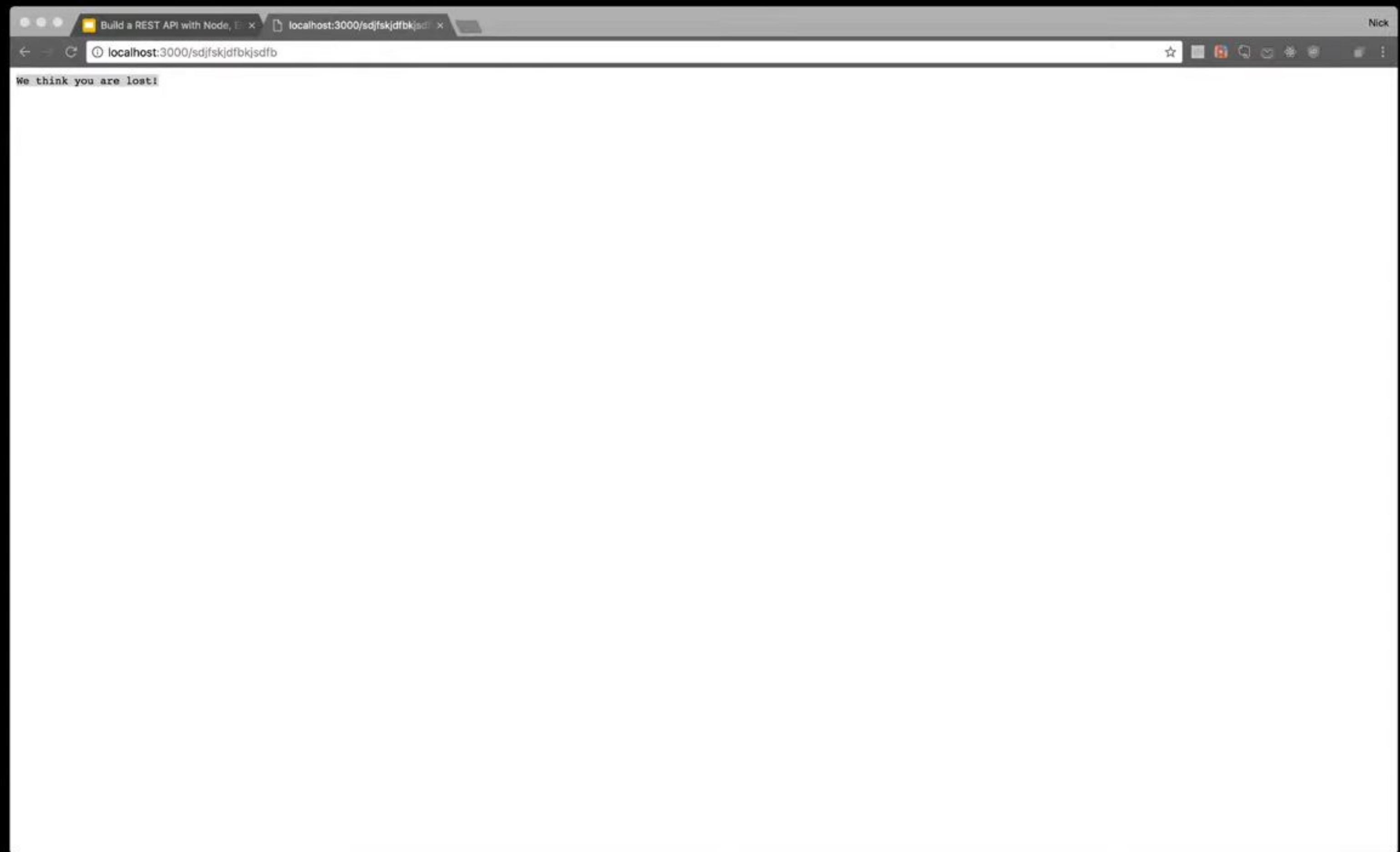
const PORT = process.env.PORT || 3000
app.listen(PORT, () => console.info(`Server has started on ${PORT}`))
```

callback for use()

Terminal

+ Thu Mar 22 2018 10:49:46 GMT-0700 (PDT) => /person/thomas

Event Log



Build a REST API with Node.js

localhost:3000/sdjfskjdfbkjsd

Nick

Secure https://docs.google.com/presentation/d/1VSlo9JEsoVjNNH4DqOJtSWInKlis5p39ImHvAcLifSw/edit#slide=id.g361772e256_1_57

Search the menus (Option+/)

18

B I U A

Format options...

13

Node.js

14

MongoDB

15

REST Client

16

Application Setup

17

Express

18

Serve Static Content

19

Route

20

Middlewares

Error Pages

Click to add speaker notes

Explore

Error Pages

Custom Handlers

- 404 Not Found Handler
- 500 Internal Server Error

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../public/500.html [rest-api-workshop]

rest-api-workshop public 500.html

Project rest-api-workshop ~/gh/theoutlander/rest-api-workshop node_modules library root public 500.html index.html src routes person.js index.js package.json package-lock.json External Libraries

1 package.json 2 index.js 3 500.html 4 person.js 6 index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Error 500</title>
</head>
<body>
    Some<br>
</body>
</html>
```

html body

Terminal

```
+ Thu Mar 22 2018 10:49:46 GMT-0700 (PDT) => /person/thomas
✖ [nodemon] restarting due to changes...
✖ [nodemon] starting 'node src/index.js'
Server has started on 3000
Thu Mar 22 2018 10:52:24 GMT-0700 (PDT) => /sdjfskjdfbkjsdfb
✖ [nodemon] restarting due to changes...
✖ [nodemon] starting 'node src/index.js'
Server has started on 3000
```

2: Favorites

Event Log

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/index.js [rest-api-workshop]

Project rest-api-workshop ~/gh/theoutlander/rest-api-workshop

Structure

Index.js

```
package.json index.js person.js index.html
10 app.use(req, res, next) => {
11   console.log(`${new Date().toString()} => ${req.originalUrl}`)
12 }
13
14 app.use(personRoute)
15 app.use(express.static('public'))
16
17 // Handler for 404 – Resource Not Found
18 app.use((req, res, next) => {
19   res.status(404).send('We think you are lost!')
20 })
21
22 // Handler for Error 500
23 app.use((err, req, res, next) => {
24   console.error(err.stack)
25   res.sendFile(path.join(__dirname, ''))
26 })
27
28
29 const PORT = process.env.PORT || 3000
30 app.listen(PORT, () => console.info(`Server has started on ${PORT}`))
```

callback for use()

Terminal

```
+ Thu Mar 22 2018 10:49:46 GMT-0700 (PDT) => /person/thomas
x [nodemon] restarting due to changes...
x [nodemon] starting `node src/index.js`
Server has started on 3000
Thu Mar 22 2018 10:52:24 GMT-0700 (PDT) => /sdjfskjdfbkjsdfb
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
Server has started on 3000
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
Server has started on 3000
```

6: TODO Terminal Event Log

The screenshot shows a code editor interface with a dark theme. The left sidebar displays the project structure:

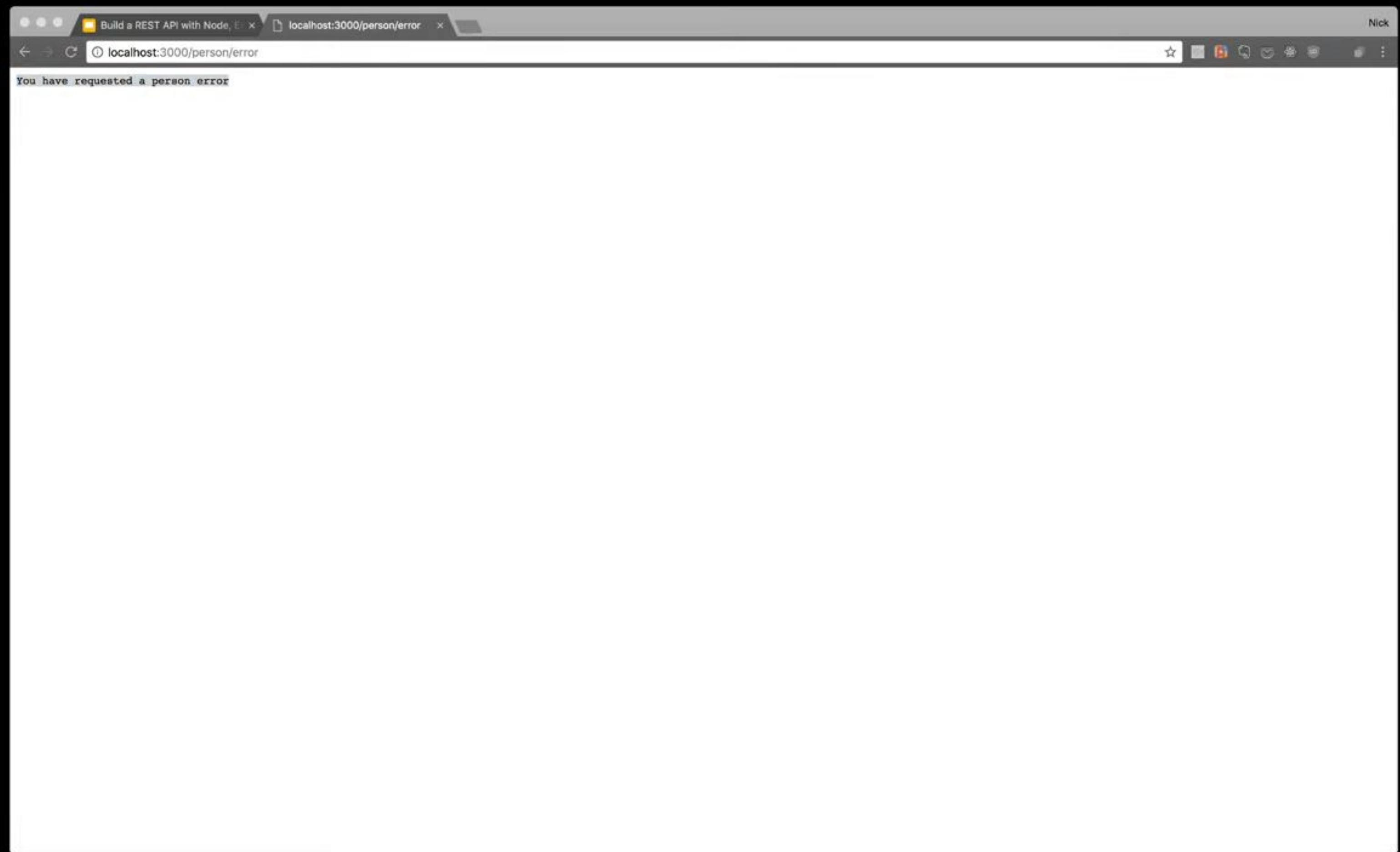
- Project: rest-api-workshop (~/gh/theoutlander/rest-api-workshop)
- node_modules (library root)
- public:
 - 500.html
 - index.html
- src:
 - routes:
 - person.js
 - index.js
 - package.json
 - package-lock.json
- External Libraries

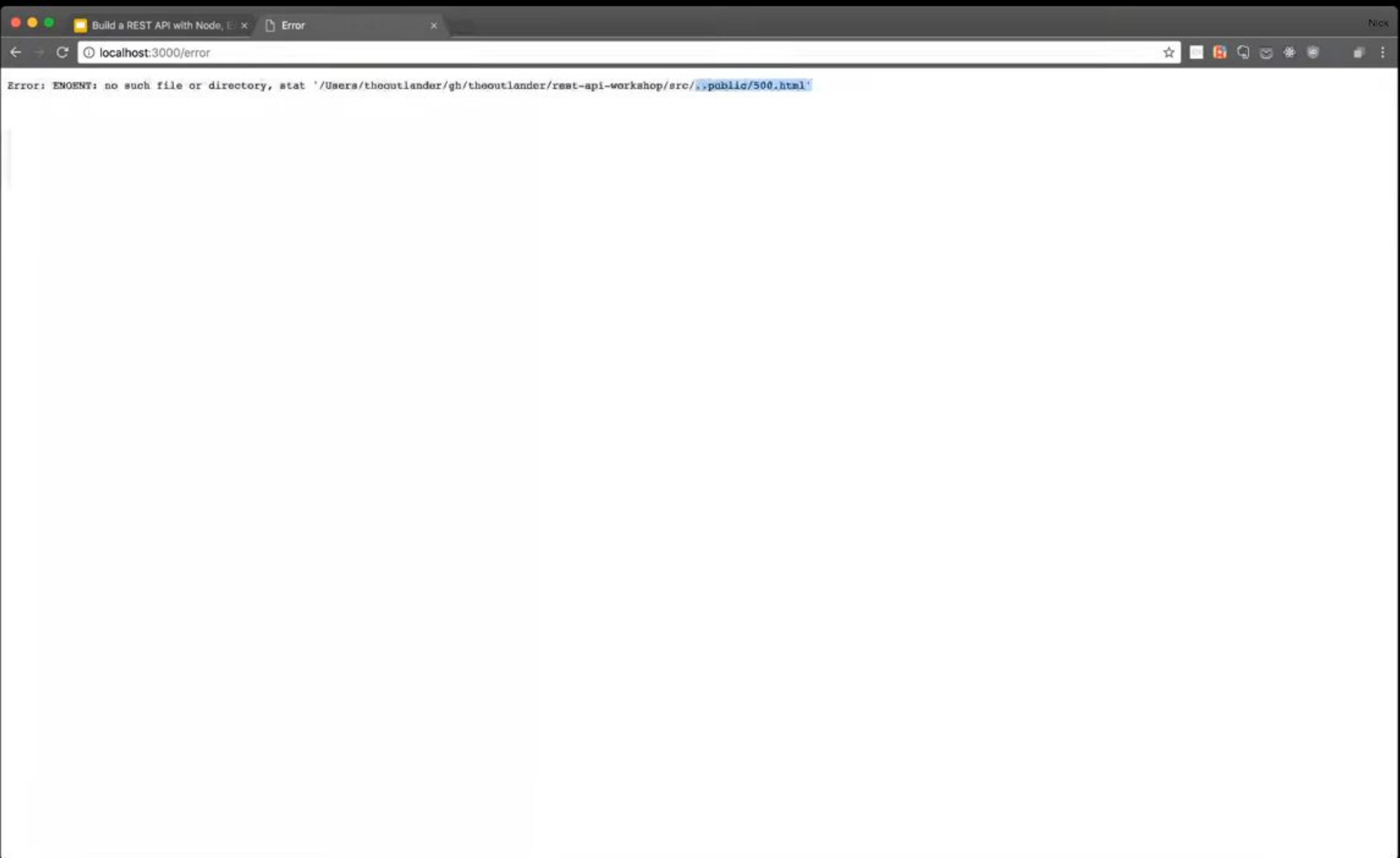
The main editor window shows the content of `person.js`:

```
1  let express = require('express')
2  let router = express.Router()
3
4  // QueryString => query property on the request object
5  // localhost:3000/person?name=thomas&age=20
6  router.get('/person', (req, res) => {
7    if(req.query.name) {
8      res.send(`You have requested a person ${req.query.name}`)
9    }
10   else {
11     res.send('You have requested a person')
12   }
13 })
14
15 // Params property on the request object
16 // localhost:3000/person/:name
17 router.get('/person/:name', (req, res) => {
18   res.send(`You have requested a person ${req.params.name}`)
19 })
20
21 router
22
23
24
25 module.exports = router
```

The bottom terminal window shows the output of the nodemon process:

```
+ Thu Mar 22 2018 10:49:46 GMT-0700 (PDT) => /person/thomas
x [nodemon] restarting due to changes...
x [nodemon] starting `node src/index.js`
Server has started on 3000
Thu Mar 22 2018 10:52:24 GMT-0700 (PDT) => /sdjfskjdfbkjsdfb
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
Server has started on 3000
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
Server has started on 3000
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
Server has started on 3000
```





Mongoose

For MongoDB

- Install dependency
- Reference Mongoose
- CRUD API

Mongoose

For MongoDB

- Install dependency
- Reference Mongoose
- CRUD API

The screenshot shows a code editor interface with a dark theme. The left sidebar displays the project structure:

- Project
- rest-api-workshop (~/gh/theoutlander/rest-api-workshop)
 - node_modules (library root)
 - public
 - 500.html
 - index.html
 - src
 - routes
 - person.js
 - index.js
 - package.json
 - package-lock.json
- External Libraries

The main editor area shows the content of `index.js`:

```
4 let personRoute = require('./routes/person')
5
6 let path = require('path')
7
8 app.use((req, res, next) => {
9   console.log(`${new Date().toString()} - ${req.originalUrl}`)
10  next()
11})
12
13 app.use(personRoute)
14 app.use(express.static('public'))
15
16 // Handler for 404 - Resource Not Found
17 app.use((req, res, next) => {
18   res.status(404).send('We think you are lost!')
19})
20
21 // Handler for Error 500
22 app.use((err, req, res, next) => {
23   console.error(err.stack)
24   res.sendFile(path.join(__dirname, '../public/500.html'))
25})
26
27
28 const PORT = process.env.PORT || 3000
path
```

The bottom terminal window shows the command `npm i mongoose` being run:

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$
```

The screenshot shows a Node.js project structure and its corresponding code files in a code editor.

Project Structure:

- rest-api-workshop
- src
 - models (selected)
 - routes
 - person.js
 - index.js
 - package.json
 - package-lock.json
- External Libraries

Code Editor:

File: customer.model.js

```
let mongoose = require('mongoose')
```

Terminal:

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$
```

Bottom Status Bar:

- 6: TODO
- Terminal
- Event Log

Invalid number of arguments, expected 1

1:29 n/a UTF-8

The screenshot shows a code editor interface with several windows open. The top window displays the file `customer.model.js` from a project named `rest-api-workshop`. The code imports `mongoose` and defines connection constants for a MongoDB database. The project structure on the left includes `node_modules`, `public` (with files `500.html` and `index.html`), and `src` (with `models` containing `customer.model.js`, `routes` containing `person.js` and `index.js`, and `package.json` and `package-lock.json`). The bottom window is a terminal showing the command `npm i mongoose` being run, which installs the `mongoose` package version 5.0.11. The terminal output also shows two WARN messages about the lack of a description and repository field in the package.json.

```
let mongoose = require('mongoose')

const server = 'ds221609.mlab.com:21609'
const database = 'rest-api-workshop'
const user = 'theoutlander'
const password = 'fZsMGZXOMxBFCTgkDwgFtEvwD7ML'

Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$
```

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/models/customer.model.js [rest-api-workshop]

rest-api-workshop src models customer.model.js

Project rest-api-workshop -/gh/theoutlander/rest-api-workshop node_modules library root public 500.html index.html src models customer.model.js routes person.js index.js package.json package-lock.json External Libraries

customer.model.js

```
1 let mongoose = require('mongoose')
2
3 const server = 'ds221609.mlab.com:21609'
4 const database = 'rest-api-workshop'
5 const user = 'theoutlander'
6 const password = 'fZsMGZXOMxBFCTgkBwgFtEvwD7ML'
7
8 mongoose.connect(`mongodb://${user}:${password}@${server}/${database}`)
9
```

Terminal

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$
```

Event Log

The screenshot shows a Java IDE interface with several windows and toolbars. The main window displays the code for `customer.model.js` under the `src/models` directory. The code imports `mongoose`, sets up a connection to a MongoDB database, and defines a schema for a `Customer` document.

```
let mongoose = require('mongoose')
const server = 'ds221609.mlab.com:21609'
const database = 'rest-api-workshop'
const user = 'theoutlander'
const password = 'fZsMGZXQMy8FCTgkBwgFtEvwD7ML'

mongoose.connect(`mongodb://${user}:${password}@${server}/${database}`)

let CustomerSchema = new mongoose.Schema({
  name: String,
})
```

The terminal window at the bottom shows the output of running `npm i mongoose`. It includes npm warnings about the lack of a description and repository for the `rest-api-workshop` package, and it lists the successful installation of `mongoose@5.0.11`.

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$
```

The screenshot shows a code editor interface with several tabs and panels.

Project Structure:

- rest-api-workshop
- src
 - models
 - customer.model.js
 - routes
 - person.js
 - index.js
 - package.json
 - package-lock.json
- External Libraries

Code Editor (customer.model.js):

```
1  let mongoose = require('mongoose')
2
3  const server = 'ds221609.mlab.com:21609'
4  const database = 'rest-api-workshop'
5  const user = 'theoutlander'
6  const password = 'fZsMGZXQMX8FCTgkBwgFtEvwD7ML'
7
8  mongoose.connect(`mongodb://${user}:${password}@${server}/${database}`)
9
10 let CustomerSchema = new mongoose.Schema({
11   name: String,
12   email: {
13     type: String,
14     require: true,
15     unique: true
16   }
17 })
18
19 module.exports = CustomerSchema
```

Terminal:

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$
```

Bottom Status Bar:

- 6: TODO
- Terminal
- Event Log

The screenshot shows a code editor interface with a dark theme. The left sidebar displays the project structure:

- rest-api-workshop
- src
 - models (selected)
 - routes
 - index.js
- public
 - 500.html
 - index.html
- node_modules (library root)
- package.json
- package-lock.json
- External Libraries

The main editor area shows the file `customer.model.js` with the following code:

```
1 let mongoose = require('mongoose')
2
3 const server = 'ds221609.mlab.com:21609'
4 const database = 'rest-api-workshop'
5 const user = 'theoutlander'
6 const password = 'fZsMGZXQMX8FCTgkBwgFtEvwD7ML'
7
8 mongoose.connect(`mongodb://${user}:${password}@${server}/${database}`)
9
10 let CustomerSchema = new mongoose.Schema({
11   name: String,
12   email: {
13     type: String,
14     required: true,
15     unique: true
16   }
17 })
18
19 module.exports = mongoose.model('Customer', CustomerSchema)
20
```

The code editor has syntax highlighting for JavaScript and MongoDB schema definitions. A tooltip "CustomerSchema" is visible over the variable name.

Below the editor is a terminal window showing the output of an npm command:

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$
```

The bottom status bar indicates the time as 14:20 and the encoding as UTF-8.

rest-api-workshop [-/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

rest-api-workshop src routes customer.js

Project rest-api-workshop -/gh/theoutlander/rest-api-workshop node_modules library root public 500.html index.html src models customer.model.js routes customer.js person.js index.js package.json package-lock.json External Libraries

CustomerModel

```
let CustomerModel = require();
```

Terminal

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$
```

6: TODO Terminal Event Log

Invalid number of arguments, expected 1 1:29 n/a UTF-8

The screenshot shows a dark-themed IDE interface with several windows and panels.

Project View: On the left, the project structure is displayed under "rest-api-workshop". It includes a "node_modules" folder, a "public" folder containing "500.html" and "index.html", and an "src" folder containing "models" (with "customer.model.js") and "routes" (with "customer.js", "person.js", "index.js", "package.json", and "package-lock.json").

Code Editor: The main editor window displays "customer.js" code:

```
1 let CustomerModel = require('../models/customer.model')
2 let express = require('express')
3 let router = express.Router()
4
5 // Create a new customer
6 router.post()
```

Terminal: Below the editor is a terminal window showing the command-line output of an npm install:

```
+ Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.8.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$
```

Bottom Status Bar: The status bar at the bottom shows "4 chars" and "Event Log".

The screenshot shows a code editor interface with a dark theme. On the left, the Project Explorer displays the file structure of a 'rest-api-workshop' project. The 'src' directory contains 'models' (with 'customer.model.js'), 'routes' (with 'customer.js', 'person.js', and 'index.js'), and other files like 'package.json' and 'package-lock.json'. The 'public' directory contains '500.html' and 'index.html'. The 'node_modules' folder is listed as a library root. The 'index.js' file in the 'src' directory is open in the main editor area, showing code for setting up an Express app, loading routes, and handling errors. Below the editor is a Terminal window showing the command 'npm i mongoose' being run, with npm warnings about package descriptions and repository fields. The bottom status bar indicates the file is 6: TODO, the terminal tab is active, and the event log tab is available.

```
let express = require('express')
let app = express()
let personRoute = require('./routes/person')
let path = require('path')

app.use((req, res, next) => {
  console.log(`new Date().toString() => ${req.originalUrl}`)
  next()
})

app.use(personRoute)
app.use(express.static('public'))

// Handler for 404 - Resource Not Found
app.use((req, res, next) => {
  res.status(404).send('We think you are lost!')
})

// Handler for Error 500
app.use((err, req, res, next) => {
  console.error(err.stack)
  res.sendFile(path.join(__dirname, '../public/500.html'))
})
```

Nicks-MBP:rest-api-workshop theoutlander\$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander\$

The screenshot shows a code editor interface with several windows open. The main window displays the `index.js` file from a `rest-api-workshop` project. The code implements an Express.js application with routes for customers and persons, static file serving, and error handling. The project structure on the left includes `node_modules`, `public`, and `src` directories containing `models` and `routes`. Below the code editor is a terminal window showing npm installation logs for `mongoose` and `body-parser`.

```
let express = require('express')
let app = express()
let personRoute = require('./routes/person')
let path = require('path')
let bodyParser = require('body-parser')

app.use((req, res, next) => {
  console.log(` ${new Date().toString()} - ${req.originalUrl}`)
  next()
})

app.use(personRoute)
app.use(express.static('public'))

// Handler for 404 - Resource Not Found
app.use((req, res, next) => {
  res.status(404).send('We think you are lost!')
})

// Handler for Error 500
app.use((err, req, res, next) => {
```

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$ npm i body-parser
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ body-parser@1.18.2
updated 1 package in 2.911s
Nicks-MBP:rest-api-workshop theoutlander$
```

The screenshot shows a code editor interface with a dark theme. On the left, the Project Explorer displays the file structure of a Node.js application named "rest-api-workshop". The "src" directory contains "models" (with "customer.model.js") and "routes" (with "customer.js", "person.js", "index.js", "package.json", and "package-lock.json"). The "public" directory includes "500.html" and "index.html". The "node_modules" folder is listed as a library root. The "Terminal" tab at the bottom shows the command-line history:

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$ npm i body-parser
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ body-parser@1.18.2
updated 1 package in 2.911s
Nicks-MBP:rest-api-workshop theoutlander$
```

The main code editor window shows "customer.js" with the following content:

```
1 let CustomerModel = require('../models/customer.model')
2 let express = require('express')
3 let router = express.Router()
4
5 // Create a new customer
6 // POST localhost:3000/customer
7 router.post('/customer', (req, res) => {
8     // req.body
9 })
10
```

A tooltip "callback for post()" is visible near the closing brace of the router.post() call.

The screenshot shows a code editor interface with a dark theme. On the left, the project structure is displayed:

- Project: rest-api-workshop
- rest-api-workshop (library root)
- public: 500.html, index.html
- src: models (customer.model.js), routes (customer.js, person.js, index.js), package.json, package-lock.json
- External Libraries

The main editor area shows the content of `customer.js`:

```
1 let CustomerModel = require('../models/customer.model')
2 let express = require('express')
3 let router = express.Router()
4
5 // Create a new customer
6 // POST localhost:3000/customer
7 router.post('/customer', (req, res) => {
8     if(!req.body) {
9         return res.status(400).se
10    }
11})
```

A tooltip "callback for post()" is visible near the bottom of the code editor.

In the bottom-left corner, the Terminal tab is active, showing the following npm command history:

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$ npm i body-parser
+ body-parser@1.18.2
updated 1 package in 2.911s
Nicks-MBP:rest-api-workshop theoutlander$
```

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar titled 'Project' showing the directory structure of a Node.js application named 'rest-api-workshop'. The 'routes' folder contains 'customer.js', 'index.js', 'person.js', and 'customer.model.js'. The 'models' folder contains 'customer.model.js'. The 'public' folder contains '500.html' and 'index.html'. The main editor window displays 'customer.js' with the following code:

```
let CustomerModel = require('../models/customer.model')
let express = require('express')
let router = express.Router()

// Create a new customer
// POST localhost:3000/customer
router.post('/customer', (req, res) => {
  if(!req.body) {
    return res.status(400).send('Request body is missing')
  }

  let model = new CustomerModel(req.body)
})
```

A tooltip 'callback for post()' is visible above the closing brace of the router's post method. Below the editor is a terminal window showing the output of npm install commands for 'mongoose' and 'body-parser'.

Terminal

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$ npm i body-parser
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ body-parser@1.18.2
updated 1 package in 2.911s
Nicks-MBP:rest-api-workshop theoutlander$
```

Bottom status bar: 6: TODO, Terminal, Event Log, 12:39 LF: UTF-8

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

Project rest-api-workshop src routes customer.js

1: Project 2: Structure 3: TODO 4: Terminal 5: Favorites 6: Event Log

1 rest-api-workshop ~/gh/theoutlander/rest-api-workshop
2 node_modules library root
3 public 500.html index.html
4 src models customer.model.js
5 routes customer.js person.js index.js package.json package-lock.json
6 External Libraries

customer.js

```
1 let CustomerModel = require('../models/customer.model')
2 let express = require('express')
3 let router = express.Router()
4
5 // Create a new customer
6 // POST localhost:3000/customer
7 router.post('/customer', (req, res) => {
8     if(!req.body) {
9         return res.status(400).send('Request body is missing')
10    }
11
12    // let user = {
13    //     name: 'firstname lastname',
14    //     email: 'email@gmail.com'
15    // }
16
17    let model = new CustomerModel(req.body)
18    model.save()
19
20})

```

callback for post() > model

Terminal

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.0.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$ npm i body-parser
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ body-parser@1.18.2
updated 1 package in 2.911s
Nicks-MBP:rest-api-workshop theoutlander$
```

The screenshot shows a code editor interface with a dark theme. The top bar displays the title "rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]" and various toolbars.

The left sidebar shows the project structure:

- Project
- rest-api-workshop (~/gh/theoutlander/rest-api-workshop)
 - node_modules library root
 - public
 - 500.html
 - index.html
 - src
 - models
 - customer.model.js
 - routes
 - customer.js
 - person.js
 - index.js
 - package.json
 - package-lock.json
- External Libraries

The main editor area is focused on the file `customer.js` under the `routes` directory. The code implements a POST endpoint for creating a new customer:

```
let CustomerModel = require('../models/customer.model')
let express = require('express')
let router = express.Router()

// Create a new customer
// POST localhost:3000/customer
router.post('/customer', (req, res) => {
  if(!req.body) {
    return res.status(400).send('Request body is missing')
  }

  // let user = {
  //   name: 'firstname lastname',
  //   email: 'email@gmail.com'
  // }

  let model = new CustomerModel(req.body)
  model.save()
    .then(doc => {
      if(!doc || doc.length === 0) {
        return res.status(500).send(doc)
      }

      res.status(201)
    })
})
```

The terminal below shows the command-line output of running npm install:

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.8.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$ npm i body-parser
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ body-parser@1.18.2
updated 1 package in 2.911s
Nicks-MBP:rest-api-workshop theoutlander$
```

The bottom navigation bar includes tabs for "TODO", "Terminal", and "Event Log".

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

rest-api-workshop src routes customer.js

Project 1: rest-api-workshop ~/gh/theoutlander/rest-api-workshop

node_modules library root

public 500.html index.html

src models customer.model.js

routes customer.js person.js index.js package.json package-lock.json

External Libraries

customer.js

```
7   router.post('/customer', (req, res) => {
8     if(!req.body) {
9       return res.status(400).send('Request body is missing')
10    }
11
12    // let user = {
13    //   name: 'firstname lastname',
14    //   email: 'email@gmail.com'
15    // }
16
17    let model = new CustomerModel(req.body)
18    model.save()
19      .then(doc => {
20        if(!doc || doc.length === 0) {
21          return res.status(500).send(doc)
22        }
23
24        res.status(201).send(doc)
25      })
26      .catch(err => {
27        res.status(500).json(err)
28      })
29  })
```

callback for post() > callback for catch()

Terminal

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i mongoose
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ mongoose@5.8.11
added 17 packages from 12 contributors in 5.424s
Nicks-MBP:rest-api-workshop theoutlander$ npm i body-parser
npm WARN rest-api-workshop@1.0.0 No description
npm WARN rest-api-workshop@1.0.0 No repository field.

+ body-parser@1.18.2
updated 1 package in 2.911s
Nicks-MBP:rest-api-workshop theoutlander$
```

6: TODO Terminal Event Log

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

Project Structure

rest-api-workshop

- node_modules
- public
 - 500.html
 - index.html
- src
 - models
 - customer.model.js
 - routes
 - customer.js
 - person.js
 - index.js
 - package.json
 - package-lock.json

External Libraries

callback for post()

```
7 router.post('/customer', (req, res) => {
8   if(!req.body) {
9     return res.status(400).send('Request body is missing')
10  }
11
12  // let user = {
13  //   name: 'firstname lastname',
14  //   email: 'email@gmail.com'
15  // }
16
17  let model = new CustomerModel(req.body)
18  model.save()
19  .then(doc => {
20    if(!doc || doc.length === 0) {
21      return res.status(500).send(doc)
22    }
23
24    res.status(201).send(doc)
25  })
26  .catch(err => {
27    res.status(500).json(err)
28  })
29})
```

Terminal

```
Nicks-MBP:rest-api-workshop theoutlander$ npm i body-parser
+ body-parser@1.18.2
updated 1 package in 2.911s
Nicks-MBP:rest-api-workshop theoutlander$ npm run start-watch

> rest-api-workshop@1.0.0 start-watch /Users/theoutlander/gh/theoutlander/rest-api-workshop
> nodemon src/index.js

[nodemon] 1.17.2
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: ***!
[nodemon] starting 'node src/index.js'
Server has started on 3000
Thu Mar 22 2018 11:18:48 GMT-0700 (PDT) => /customer {}
```

Event Log

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/index.js [rest-api-workshop]

Project rest-api-workshop src index.js package.json index.js customer.model.js customer.js person.js index.html

```
let express = require('express')
let app = express()
let personRoute = require('./routes/person')
let customerRoute = require('./routes/customer')
let path = require('path')
let bodyParser = require('body-parser')

app.use(bodyParser.json())

app.use((req, res, next) => {
  console.log(` ${new Date().toString()} => ${req.originalUrl} `, req.body)
  next()
})

app.use(personRoute)
app.use(customerRoute)
app.use(express.static('public'))

// Handler for 404 – Resource Not Found
app.use((req, res, next) => {
  res.status(404).send('We think you are lost!')
})

// Handler for Error 500
app.use((err, req, res, next) => {
  callbackForUse()
})
```

Terminal

```
[nodemon] starting `node src/index.js`
[nodemon] info App will attempt recovery if files change
[nodemon] info Starting `node src/index.js` in /Users/theoutlander/gh/theoutlander/rest-api-workshop
[nodemon] info File changes detected. Waiting for file changes before starting...
[nodemon] info App crashed - waiting for file changes before starting...
```

Insomnia - POST Customer

POST localhost:3000/customer

Send 404 Not Found TIME 21 ms SIZE 22 B

No Environment Cookies Body Auth Query Header Docs Preview Header Cookie Timeline

We think you are lost!

Filter

REST API Workshop

POST POST Customer

GET GET Person

GET GET Person by QueryString

GET GET Person by Params

Projects



Select a body type from above

Insomnia - POST Customer

POST localhost:3000/customer

Send 201 Created TIME 116 ms SIZE 94 B

No Environment Cookies

JSON Auth Query Header Docs

Filter

REST API Workshop

POST POST Customer

GET GET Person

GET GET Person by QueryString

GET GET Person by Params

Projects

Preview Header Cookie Timeline

```
1 = {  
2   "name": "Thomas Anderson",  
3   "email": "thomas@gmail.com"  
4 }  
  
1 = {  
2   "_id": "5ab3f442f3d76b4ae4cb433b",  
3   "name": "Thomas Anderson",  
4   "email": "thomas@gmail.com",  
5   "__v": 0  
6 }
```



mLab (1)

rest-api-workshop

- Collections (2)
 - System
 - customers
- Functions
- Users

db.getCollection('customers').find({})

mLab ds221609.mlab.com:21609 rest-api-workshop

1 db.getCollection('customers').find({})

customers 0.089 sec.

Key	Value	Type
(1) ObjectId("5ab3f442f3d76b4ae4cb433b")	{ 4 fields }	Object
__id	ObjectId("5ab3f442f3d76b4ae4cb433b")	ObjectId
name	Thomas Anderson	String
email	thomas@gmail.com	String
__v	0	Int32

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

The screenshot shows a code editor with a dark theme. On the left is a project tree for 'rest-api-workshop'. It includes a 'node_modules' folder, a 'public' folder with '500.html' and 'index.html', a 'src' folder containing 'models' (with 'customer.model.js') and 'routes' (with 'customer.js', 'person.js', 'index.js', 'package.json', and 'package-lock.json'). A 'External Libraries' section is also present. The main panel displays the 'customer.js' file. The code handles a POST request to '/customer'. It checks if 'req.body.email' is provided. If not, it returns a 400 Bad Request response. Otherwise, it creates a 'CustomerModel' instance from the request body, saves it to the database, and returns a 201 Created response with the saved document. If there's an error during saving or if no document is found, it returns a 500 Internal Server Error with the error object as JSON. The file concludes with an export statement for the router.

```
if(!req.body.email) {
    // ...
}

// let user = {
//   name: 'firstname lastname',
//   email: 'email@gmail.com'
// }

let model = new CustomerModel(req.body)
model.save()
.then(doc => {
    if(!doc || doc.length === 0) {
        return res.status(500).send(doc)
    }

    res.status(201).send(doc)
})
.catch(err => {
    res.status(500).json(err)
})
}

router

module.exports = router
```

Terminal

```
+ TypeError: app.use() requires a middleware function
  at Function.use (/Users/theoutlander/gh/theoutlander/rest-api-workshop/node_modules/express/lib/application.js:210:11)
  at Object.<anonymous> (/Users/theoutlander/gh/theoutlander/rest-api-workshop/src/index.js:16:5)
  at Module._compile (module.js:635:30)
  at Object.Module._extensions..js (module.js:646:10)
  at Module.load (module.js:554:32)
  at tryModuleLoad (module.js:497:12)
  at Function.Module._load (module.js:489:3)
  at Function.Module.runMain (module.js:676:10)
  at startup (bootstrap_node.js:187:16)
  at bootstrap_node.js:608:3
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
Server has started on 3000
Thu Mar 22 2018 11:21:00 GMT-0700 (PDT) => /customer {}
Thu Mar 22 2018 11:21:54 GMT-0700 (PDT) => /customer { name: 'Thomas Anderson', email: 'thomas@gmail.com' }
```

rest-api-workshop [-/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

rest-api-workshop src routes customer.js

Project rest-api-workshop node_modules library root public 500.html index.html src models customer.model.js routes customer.js person.js index.js package.json package-lock.json External Libraries

```
1 let CustomerModel = require('../models/customer.model')
2 let express = require('express')
3 let router = express.Router()
4
5 // Create a new customer
6 // POST localhost:3000/customer
7 router.post('/customer', (req, res) => {
8   if(!req.body) {
9     return res.status(400).send('Request body is missing')
10  }
11
12  if(!req.body.email) {
13    // ...
14  }
15
16  // let user = {
17  //   name: 'firstname lastname',
18  //   email: 'email@gmail.com'
19  // }
20
21  let model = new CustomerModel(req.body)
22  model.save()
23  .then(doc => {
24    if(!doc || doc.length === 0) {
25      return res.status(500).send(doc)
26    }
27
28    res.status(201).send(doc)
29  })
30  .catch(err => {
31    res.status(500).json(err)
32  })
33}
34
35 router.get('/customer', (req, res) => {
36   CustomerModel.findOne({
37   })
38 })
39
40 module.exports = router
```

callback for get()

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

```
Project rest-api-workshop src routes customer.js package.json index.js customer.model.js customer.js person.js index.html 1 router.post('/customer', (req, res) => { 2   if(!req.body) { 3     return res.status(400).send('Request body is missing') 4   } 5   if(!req.body.email) { 6     // ... 7   } 8   // let user = { 9   //   name: 'firstname lastname', 10  //   email: 'email@gmail.com' 11  // } 12 13  let model = new CustomerModel(req.body) 14  model.save() 15  .then(doc => { 16    if(!doc || doc.length === 0) { 17      return res.status(500).send(doc) 18    } 19 20    res.status(201).send(doc) 21 22  }) 23  .catch(err => { 24    res.status(500).json(err) 25 26  }) 27 28 29 30 31 32 33 34 35  router.get('/customer', (req, res) => { 36    if(!req.query.email) { 37      return res.status(400).send('Missing URL parameter: email') 38    } 39 40    CustomerModel.findOne({ 41      email: req.query.email 42    }) 43    .then(doc => { 44      | 45    }) 46 47 48  module.exports = router
```

callback for get() - callback for then()

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

Project Structure

rest-api-workshop

- node_modules
- public
 - 500.html
 - index.html
- src
 - models
 - customer.model.js
 - routes
 - customer.js
 - person.js
 - index.js
 - package.json
 - package-lock.json

External Libraries

customer.js

```
14      }
15
16      // let user = {
17      //   name: 'firstname lastname',
18      //   email: 'email@gmail.com'
19      // }
20
21      let model = new CustomerModel(req.body)
22      model.save()
23        .then(doc => {
24          if(!doc || doc.length === 0) {
25            return res.status(500).send(doc)
26          }
27
28          res.status(201).send(doc)
29        })
30        .catch(err => {
31          res.status(500).json(err)
32        })
33    })
34
35    router.get('/customer', (req, res) => {
36      if(!req.query.email) {
37        return res.status(400).send('Missing URL parameter: email')
38      }
39
40      CustomerModel.findOne({
41        email: req.query.email
42      })
43        .then(doc => {
44          res.json(doc)
45        })
46        .catch(err => {
47          res.stats(500).json(err)
48        })
49    })
50
51    router.put('/cusomter', (req, ))
52
53    module.exports = router
```

TODO Terminal Event Log

expression expected 51:31 LF: UTF-8: 0

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

Project Structure

rest-api-workshop

- node_modules
- public
 - 500.html
 - index.html
- src
 - models
 - customer.model.js
 - routes
 - customer.js
 - person.js
 - index.js
 - package.json
 - package-lock.json

External Libraries

customer.js

```
30     .catch(err => {
31         res.status(500).json(err)
32     })
33 }
34
35 router.get('/customer', (req, res) => {
36     if(!req.query.email) {
37         return res.status(400).send('Missing URL parameter: email')
38     }
39
40     CustomerModel.findOne({
41         email: req.query.email
42     })
43     .then(doc => {
44         res.json(doc)
45     })
46     .catch(err => {
47         res.stats(500).json(err)
48     })
49 })
50
51 router.put('/cusomter', (req, res) => {
52     if(!req.query.email) {
53         return res.status(400).send('Missing URL parameter: email')
54     }
55
56     CustomerModel.findOneAndUpdate({
57         email: req.query.email
58     }, req.body, {
59         new: true
60     })
61     .then(doc => {
62         res.json(doc)
63     })
64     .catch(err => {
65         res.stats(500).json(err)
66     })
67 })
68
69 module.exports = router
```

callback for put() - callback for catch()

6: TODO Terminal Event Log

Unresolved function or method stats()

65:15 LF: UTF-8: ☰

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

The screenshot shows a code editor interface with a dark theme. On the left, there's a sidebar with 'Project' and 'Structure' tabs, displaying the file tree of a 'rest-api-workshop' project. The 'Structure' tab is active, showing the following directory structure:

- rest-api-workshop (~/gh/theoutlander/rest-api-workshop)
 - node_modules
 - public
 - 500.html
 - index.html
 - src
 - models
 - customer.model.js
 - routes
 - customer.js
 - person.js
 - index.js
 - package.json
 - package-lock.json
- External Libraries

The main editor area has tabs for package.json, index.js, customer.model.js, customer.js (which is the active tab), person.js, and index.html. The code in customer.js is as follows:

```
44     .then(doc => {
45       res.json(doc)
46     })
47     .catch(err => {
48       res.status(500).json(err)
49     })
50   })
51
52   // UPDATE
53   router.put('/customer', (req, res) => {
54     if(!req.query.email) {
55       return res.status(400).send('Missing URL parameter: email')
56     }
57
58     CustomerModel.findOneAndUpdate({
59       email: req.query.email
60     }, req.body, {
61       new: true
62     })
63     .then(doc => {
64       res.json(doc)
65     })
66     .catch(err => {
67       res.status(500).json(err)
68     })
69   })
70
71   router.delete('/customer', (req, res) => {
72     if(!req.query.email) {
73       return res.status(400).send('Missing URL parameter: email')
74     }
75
76     CustomerModel.findOneAndUpdate({
77       email: req.query.email
78     }, req.body, {
79       new: true
80     })
81     .then(doc => {
82       res.json(doc)
83     })
84     .catch(err => {
85       res.status(500).json(err)
86     })
87   })
88 }

callback for delete()
```

At the bottom of the editor, there are status bars for 'TODO' (6 chars), 'Terminal', 'Event Log', and file encoding ('UTF-8'). A tooltip at the bottom left says 'Unresolved function or method findOneAndUpdate()'.

Insomnia - POST Customer

PUT localhost:3000/customer?email=thomas@gmail.com Send 201 Created TIME 116 ms SIZE 94 B

No Environment Cookies JSON Auth Query Header Docs Preview Header Cookie Timeline

Filter REST API Workshop PUT POST Customer POST PUT Customer GET GET Person GET GET Person by QueryString GET GET Person by Params Projects

1 = {
2 "name": "Thomas Updated Anderson",
3 "email": "thomas@gmail.com"
4 }

1 = {
2 "_id": "5ab3f442f3d76b4ae4cb433b",
3 "name": "Thomas Anderson",
4 "email": "thomas@gmail.com",
5 "<_v": 8
6 }

Insomnia - GET Customer

Send 200 OK TIME 97.3 ms SIZE 102 B

No Environment Cookies JSON Auth Query Header 1 Docs Preview Header 8 Cookie Timeline

Filter: ...

REST API Workshop

PUT POST Customer

POST PUT Customer

GET GET Customer

GET GET Person

GET GET Person by QueryString

GET GET Person by Params

Projects

1 = {
2 " _id": "5ab3f442f3d76b4ae4cb433b",
3 " name": "Thomas Updated Anderson",
4 " email": "thomas@gmail.com",
5 " __v": 0
6 }

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

Project Structure

rest-api-workshop

- node_modules
- public
 - 500.html
 - index.html
- src
 - models
 - customer.model.js
 - routes
 - customer.js
 - person.js
 - index.js
 - package.json
 - package-lock.json

External Libraries

customer.js

```
1 let CustomerModel = require('../models/customer.model')
2 let express = require('express')
3 let router = express.Router()
4
5 // Create a new customer
6 // POST localhost:3000/customer
7 router.post('/customer', (req, res) => {
8     if(!req.body) {
9         return res.status(400).send('Request body is missing')
10    }
11
12    if(!req.body.email) {
13        // ...
14    }
15
16    // let user = {
17    //     name: 'firstname lastname',
18    //     email: 'email@gmail.com'
19    // }
20
21    let model = new CustomerModel(req.body)
22    model.save()
23        .then(doc => {
24            if(!doc || doc.length === 0) {
25                return res.status(500).send(doc)
26            }
27
28            res.status(201).send(doc)
29        })
29        .catch(err => {
30            res.status(500).json(err)
31        })
32    })
33
34
35 // GET
// callback for put()
```

Terminal

```
+ Server has started on 3000
x [nodemon] restarting due to changes...
x [nodemon] starting 'node src/index.js'
Server has started on 3000
Thu Mar 22 2018 11:29:58 GMT-0700 (PDT) => /customer?email=thomas%40gmail.com { name: 'Thomas Updated Anderson', email: 'thomas@gmail.com' }
Thu Mar 22 2018 11:30:51 GMT-0700 (PDT) => /customer?email=thomas%40gmail.com {}
Thu Mar 22 2018 11:31:13 GMT-0700 (PDT) => /customer?email=thomas%40gmail.com {}
Thu Mar 22 2018 11:31:20 GMT-0700 (PDT) => /customer?email=thomas%40gmail.com {}
```

TODO Terminal Event Log

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

Project 1: rest-api-workshop

Structure 2: routes

customer.js

File package.json index.js customer.model.js customer.js person.js index.html

```
10      }
11      if(!req.body.email) {
12          // ...
13      }
14      // let user = {
15      //     name: 'firstname lastname',
16      //     email: 'email@gmail.com'
17      // }
18
19      let model = new CustomerModel(req.body)
20      model.save()
21          .then(doc => {
22              if(!doc || doc.length === 0) {
23                  return res.status(500).send(doc)
24              }
25
26                  res.status(201).send(doc)
27              })
28          .catch(err => {
29              res.status(500).json(err)
30          })
31
32      })
33
34      // GET
35      router.get('/customer', (req, res) => {
36          if(!req.query.email) {
37              return res.status(400).send('Missing URL parameter: email')
38          }
39
40          CustomerModel.findOne({
41              email: req.query.email
42          })
43          .then(doc => {
44              res.json(doc)
callback for post()
        })
```

Terminal

- + Server has started on 3000
- [nodemon] restarting due to changes...
- [nodemon] starting 'node src/index.js'

Server has started on 3000

Thu Mar 22 2018 11:29:58 GMT-0700 (PDT) => /customer?email=thomas%40gmail.com { name: 'Thomas Updated Anderson', email: 'thomas@gmail.com' }

Thu Mar 22 2018 11:30:51 GMT-0700 (PDT) => /customer?email=thomas%40gmail.com {}

Thu Mar 22 2018 11:31:13 GMT-0700 (PDT) => /customer?email=thomas%40gmail.com {}

Thu Mar 22 2018 11:31:20 GMT-0700 (PDT) => /customer?email=thomas%40gmail.com {}

39 chars 21:3 LF: UTF-8

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]

```
33     })
34
35     // GET
36     router.get('/customer', (req, res) => {
37       if(!req.query.email) {
38         return res.status(400).send('Missing URL parameter: email')
39       }
40
41       CustomerModel.findOne({
42         email: req.query.email
43       })
44         .then(doc => {
45           res.json(doc)
46         })
47         .catch(err => {
48           res.status(500).json(err)
49         })
50     })
51
52     // UPDATE
53     router.put('/customer', (req, res) => {
54       if(!req.query.email) {
55         return res.status(400).send('Missing URL parameter: email')
56       }
57
58       CustomerModel.findOneAndUpdate({
59         email: req.query.email
60       }, req.body, {
61         new: true
62       })
63         .then(doc => {
64           res.json(doc)
65         })
66         .catch(err => {
67           res.status(500).json(err)
68         })
69   })
70
71   callback for get()
```

Server has started on 3000
[nodemon] restarting due to changes...
[nodemon] starting 'node src/index.js'
Server has started on 3000
Thu Mar 22 2018 11:29:58 GMT-0700 (PDT) => /customer?email=thomas%40gmail.com { name: 'Thomas Updated Anderson', email: 'thomas@gmail.com' }
Thu Mar 22 2018 11:30:51 GMT-0700 (PDT) => /customer?email=thomas%40gmail.com {}
Thu Mar 22 2018 11:31:13 GMT-0700 (PDT) => /customer?email=thomas%40gmail.com {}
Thu Mar 22 2018 11:31:20 GMT-0700 (PDT) => /customer?email=thomas%40gmail.com {}

The screenshot shows a code editor interface with a dark theme. The top navigation bar displays the title "rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../package.json [rest-api-workshop]" and a search bar. The left sidebar shows the project structure under "Project" and "Structure". The main editor area displays the contents of the `package.json` file, which defines a Node.js application with various configurations like scripts, dependencies, and devDependencies. Below the editor is a terminal window showing the command `ls` being run in the project directory, listing files like `node_modules`, `package-lock.json`, `package.json`, `public`, and `src`. The bottom status bar includes tabs for "TODO", "Terminal", and "Event Log", along with system information like "22:2 LF: UTF-8".

```
1 {  
2   "name": "rest-api-workshop",  
3   "version": "1.0.0",  
4   "description": "",  
5   "main": "index.js",  
6   "scripts": {  
7     "start": "node src/index.js",  
8     "start-watch": "nodemon src/index.js",  
9     "test": "echo \\\"Error: no test specified\\\" && exit 1"  
10    },  
11    "keywords": [],  
12    "author": "",  
13    "license": "ISC",  
14    "dependencies": {  
15      "body-parser": "^1.18.2",  
16      "express": "^4.16.3",  
17      "mongoose": "^5.0.11"  
18    },  
19    "devDependencies": {  
20      "nodemon": "^1.17.2"  
21    }  
22  }  
23
```

```
Nicks-MBP:rest-api-workshop theoutlander$ ls  
+ node_modules          package-lock.json      package.json        public           src  
x Nicks-MBP:rest-api-workshop theoutlander$
```

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../package.json [rest-api-workshop]

rest-api-workshop package.json

Project rest-api-workshop ~/gh/theoutlander/rest-api-workshop

1: package.json Index.js customer.model.js customer.js person.js index.html

1 { "name": "rest-api-workshop", "version": "1.0.0", "description": "", "main": "index.js", "scripts": { "start": "node src/index.js", "start-watch": "nodemon src/index.js", "test": "echo \\\"Error: no test specified\\\" && exit 1" }, "keywords": [], "author": "", "license": "ISC", "dependencies": { "body-parser": "^1.18.2", "express": "^4.16.3", "mongoose": "^5.0.11" }, "devDependencies": { "nodemon": "^1.17.2" } }

2: Structure

3: Project

4: Public

5: src

6: models

7: routes

8: package.json

9: package-lock.json

10: External Libraries

Terminal

Nicks-MBP:rest-api-workshop theoutlander\$ now

> Deploying ~/gh/theoutlander/rest-api-workshop under theoutlander

> Your deployment's code and logs will be publicly accessible because you are subscribed to the OSS plan.

> > NOTE: You can use `now --public` or upgrade your plan (<https://zeit.co/account/plan>) to skip this prompt

> Using Node.js 8.10.0 (default)

> Ready! <https://rest-api-workshop-vwlwvsxvsk.now.sh> (copied to clipboard) [4s]

> Synced 10 files (25.46KB) [4s]

> Initializing...

> Building

> ▲ npm install

> ✓ Using "package-lock.json"

> ✓ Installed 421 modules [3s]

6: TODO Terminal Event Log

22:2 LF: UTF-8: 8

The screenshot shows a code editor interface with a project named "rest-api-workshop".

Project Structure:

- rest-api-workshop (~/gh/theoutlander/rest-api-workshop)
 - node_modules (library root)
 - public
 - 500.html
 - index.html
 - src
 - models
 - customer.model.js
 - routes
 - customer.js
 - person.js
 - index.js
 - package.json
 - package-lock.json
- External Libraries

package.json Content:

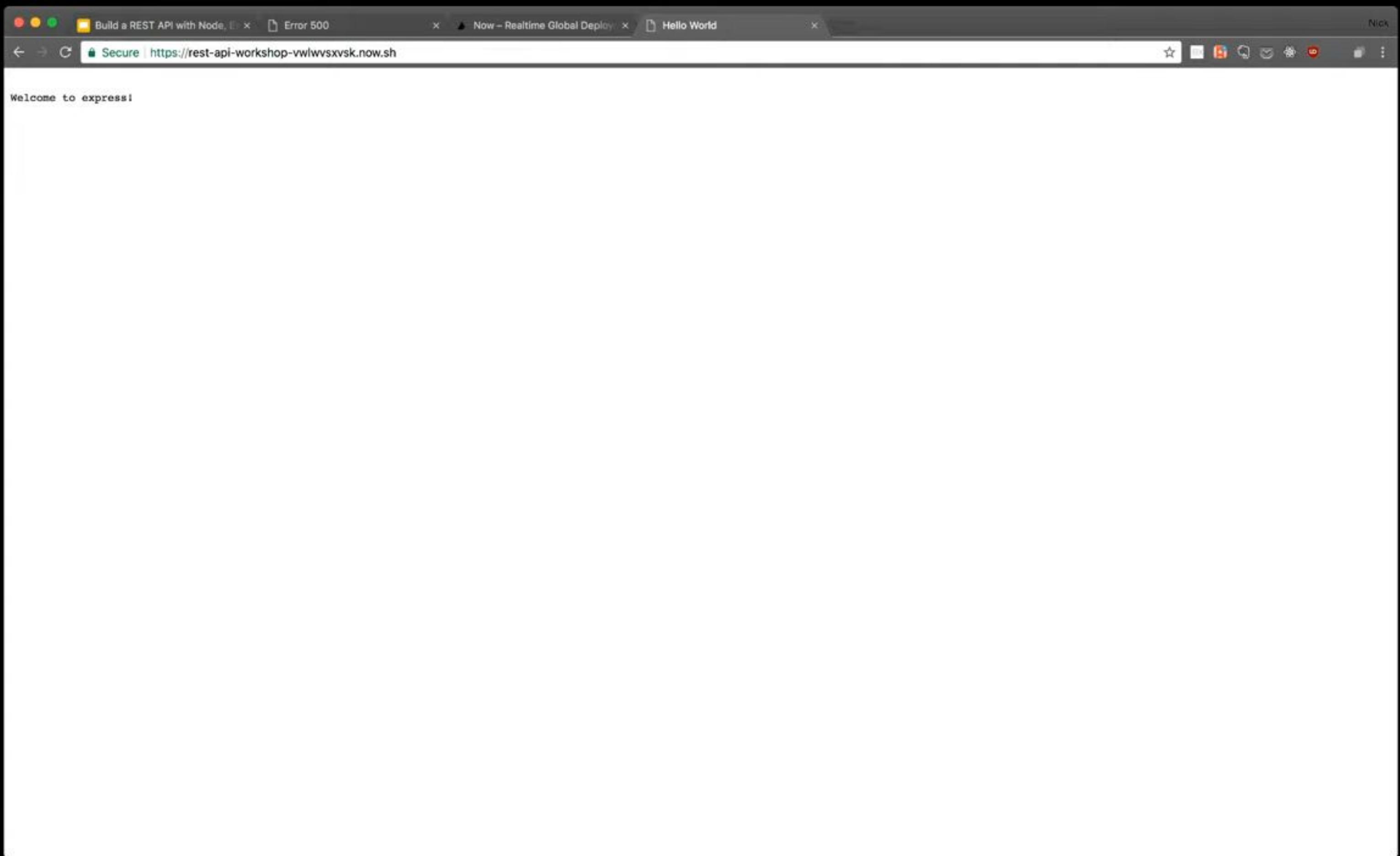
```
1  {
2    "name": "rest-api-workshop",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "start": "node src/index.js",
8      "start-watch": "nodemon src/index.js",
9      "test": "echo \"Error: no test specified\" && exit 1"
10    },
11    "keywords": [],
12    "author": "",
13    "license": "ISC",
14    "dependencies": {
15      "body-parser": "^1.18.2",
16      "express": "^4.16.3",
17      "mongoose": "^5.0.11"
18    },
19    "devDependencies": {
20      "nodemon": "^1.17.2"
21    }
22  }
```

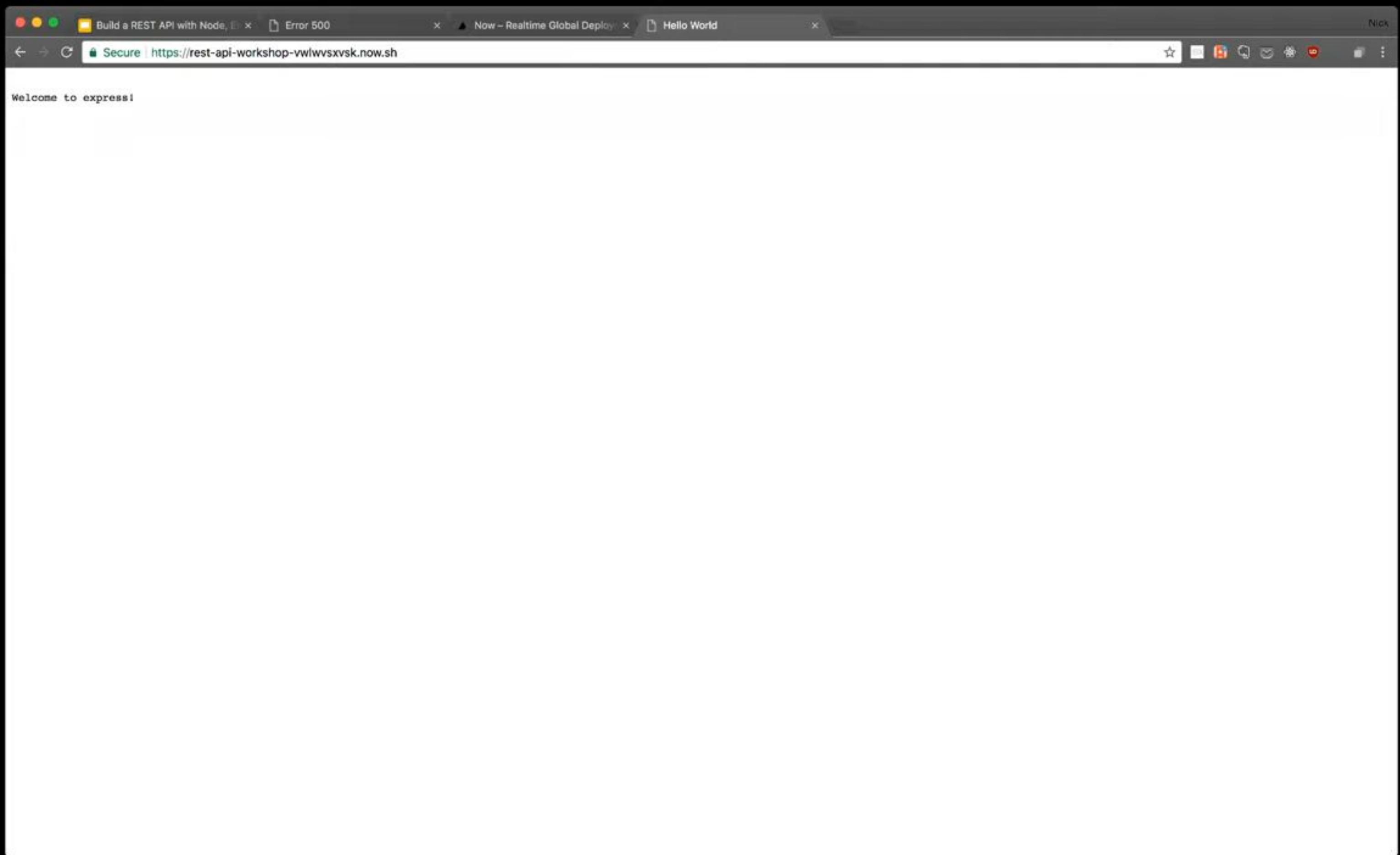
Terminal Log:

```
+ Nicks-MBP:rest-api-workshop theoutlander$ now
> Deploying ~/gh/theoutlander/rest-api-workshop under theoutlander
> Your deployment's code and logs will be publicly accessible because you are subscribed to the OSS plan.
> > NOTE: You can use `now --public` or upgrade your plan (https://zeit.co/account/plan) to skip this prompt
> Using Node.js 8.10.0 (default)
> Ready! https://rest-api-workshop-vwlwvsvsk.now.sh (copied to clipboard) [4s]
> Synced 10 files (25.46KB) [4s]
> Initializing...
> Building
> ▲ npm install
> ✓ Using "package-lock.json"
> ✓ Installed 421 modules [3s]
> ▲ npm start
> Deployment complete!
> Running in sf01
Nicks-MBP:rest-api-workshop theoutlander$
```

Bottom Status Bar:

- TODO
- Terminal
- Event Log





Build a REST API with Node.js x Error 500 Now – Realtime Global Deploy x Hello World Introduction to Mongoose for x How to build your own React Native App x Nick

Secure | <https://medium.freecodecamp.org/how-to-build-your-own-react-boilerplate-2f8cbbbe9b3f>

freeCodeCamp(🔥)

Writer  

Edit   

HOME DEV DESIGN DATA | LEARN TO CODE FOR FREE

and over again.

2.1K     

Let's assume your development stack consists of several libraries, such as React, Babel, Express, Jest, Webpack, etc. When you start a new project, you initialize all these libraries and configure them to work with each other.

With every new project that you start, you will be repeating yourself. You could also introduce inconsistencies in how these libraries are set up in each project. This can cause confusion when you switch between projects.

This is where boilerplates come in. A boilerplate is a template that you can clone and reuse for every project.

The modular Javascript ecosystem simplifies application development through various libraries, frameworks and tools. Boilerplates can be daunting if you don't understand the fundamentals of their underlying components. Let's learn about these basic building blocks while creating our own.

[Click here for source on GitHub](#)

I am using Webstorm, Git, NodeJS 8.9, NPM 5.6, and React 16. Fire up your favorite IDE, create a blank project, and let's get started!

Git Repository: Setup

 2.1K      Next story Insights from Stack Overflow'...