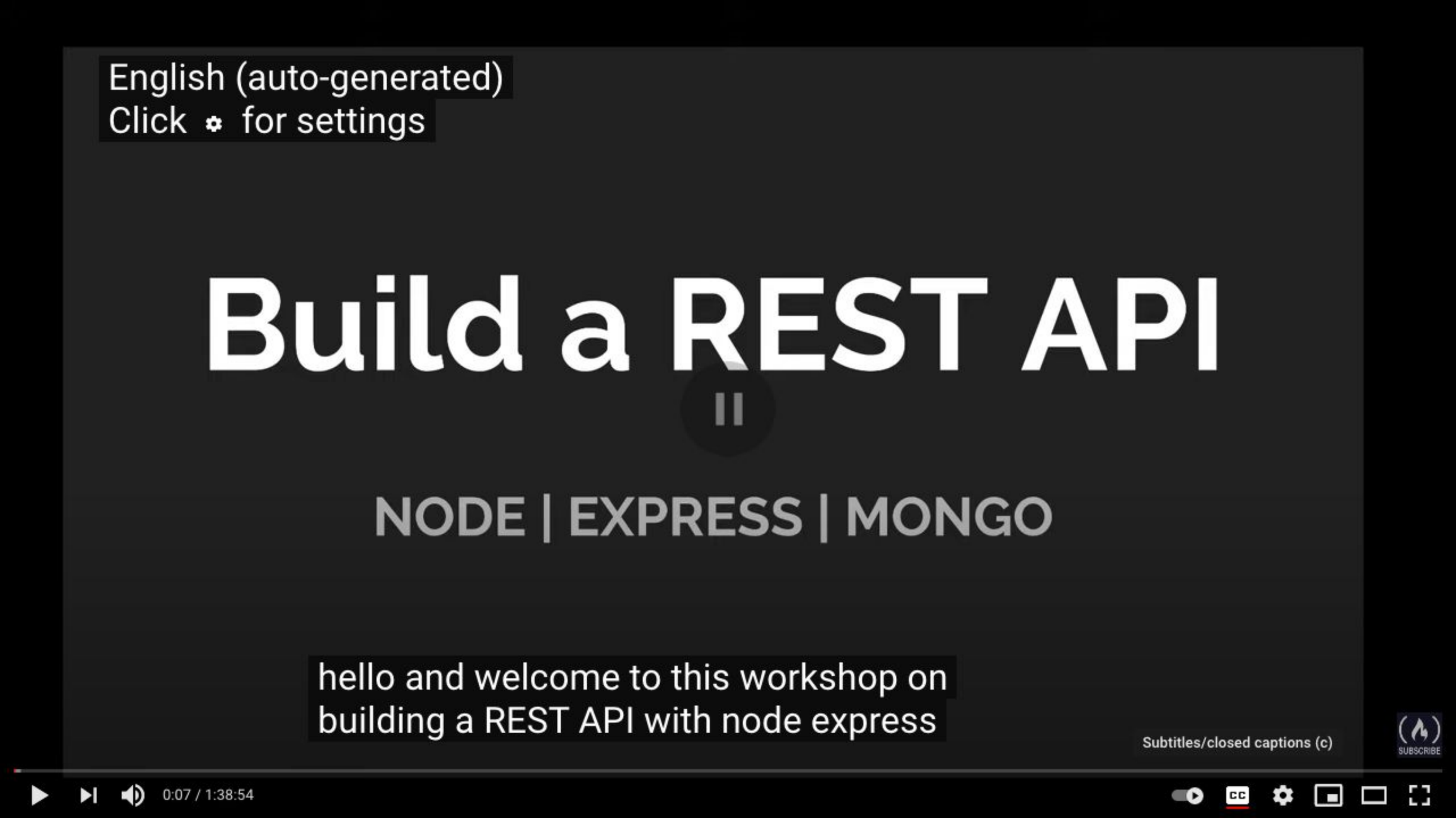

English (auto-generated)
Click for settings
Build a REST API
NODE | EXPRESS | MONGO
hello and welcome to this workshop on
building a REST API with node express
Subtitles/closed captions (c)
SUBSCRIBE
0:07 / 1:38:54

English (auto-generated)
Click ⚙ for settings

# 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

so what does HTTP it stands

Subtitles/closed captions (c)

SUBSCRIBE

1:07 / 1:38:54

English (auto-generated)
Click ⚙ for settings

# ...What is HTTP

HyperText Transfer Protocol

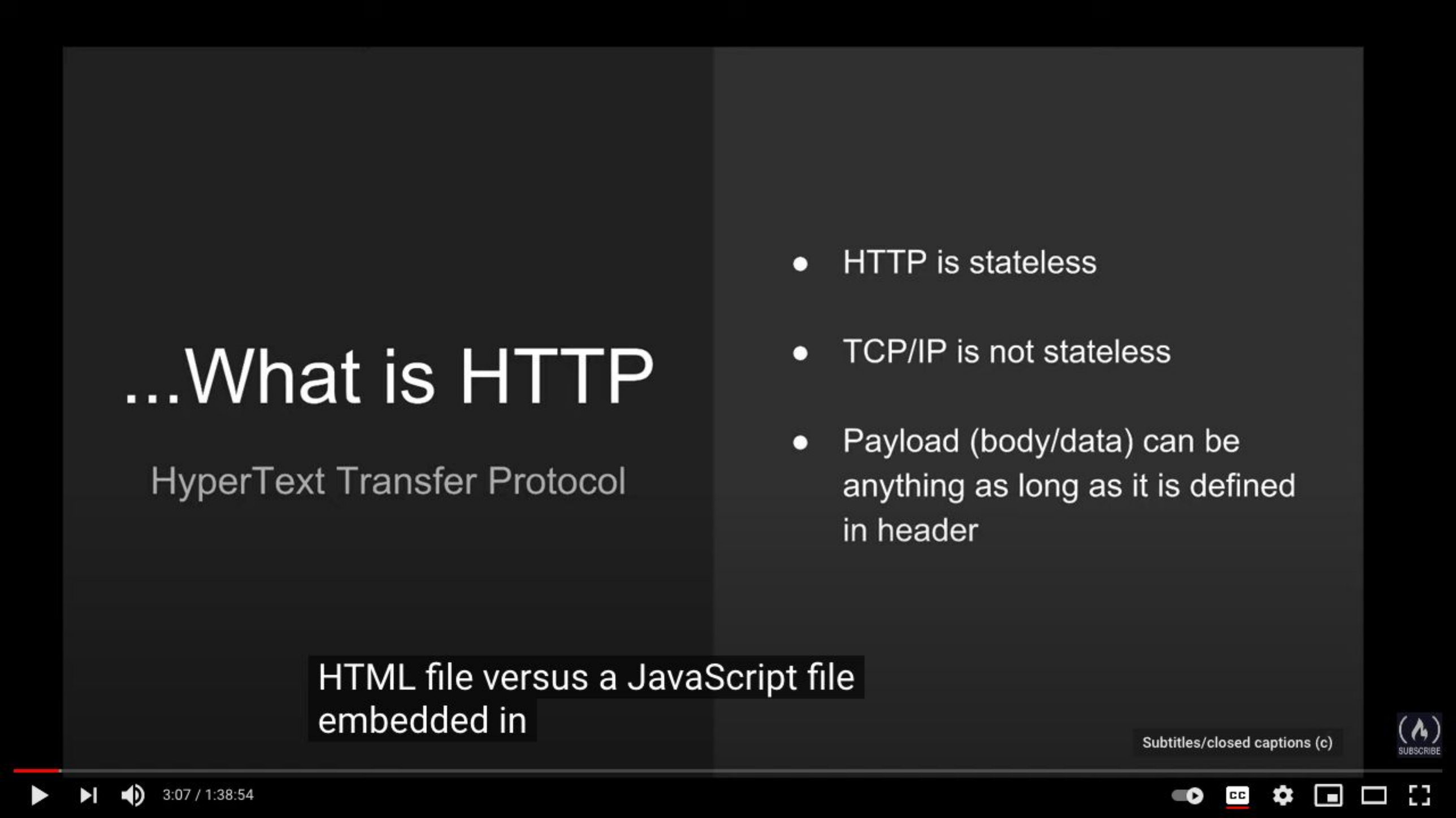
- HTTP is stateless
- TCP/IP is not stateless
- Payload (body/data) can be anything as long as it is defined in header

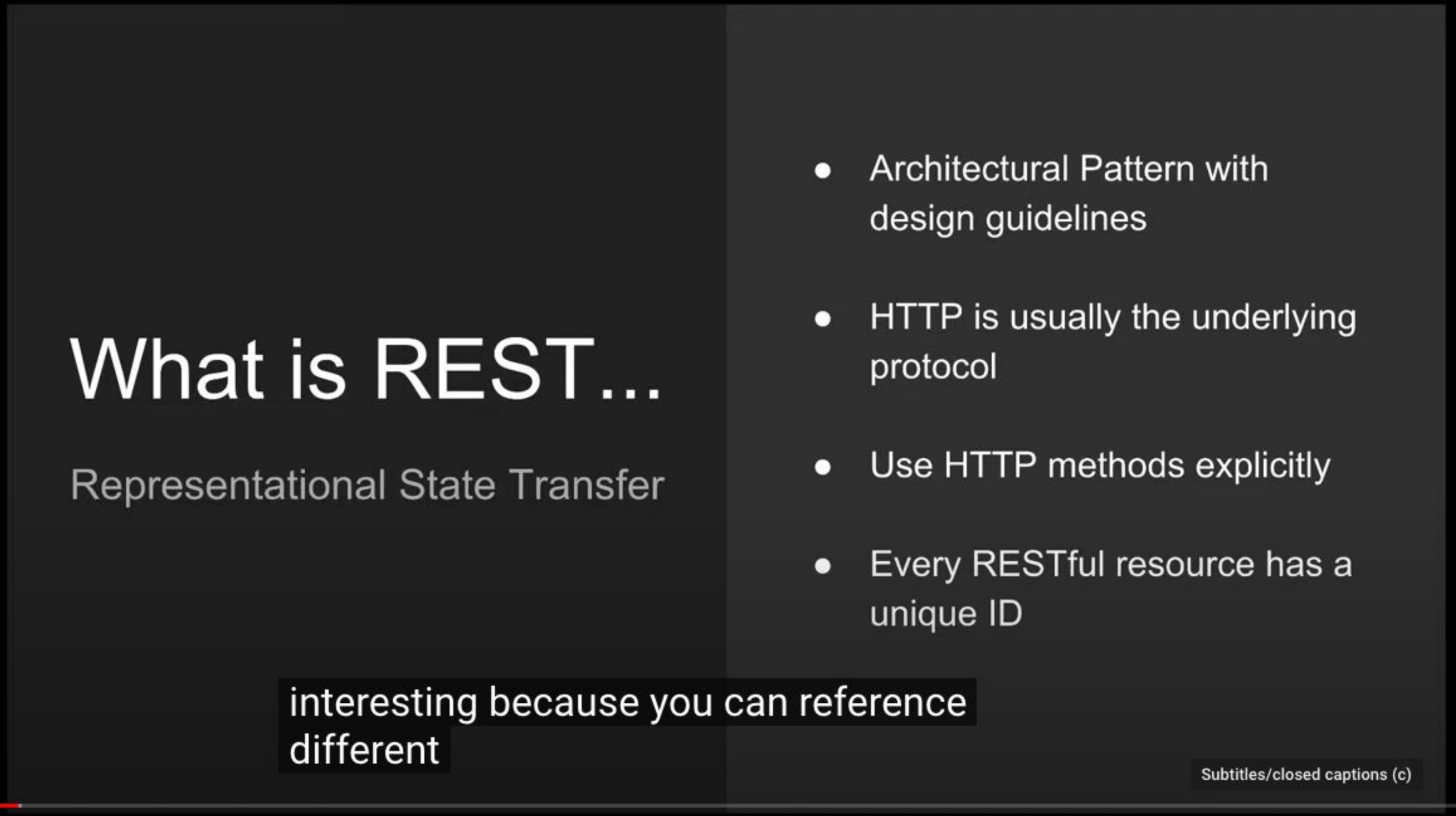
of the requests that came before it and
that's what makes HTTP stateless

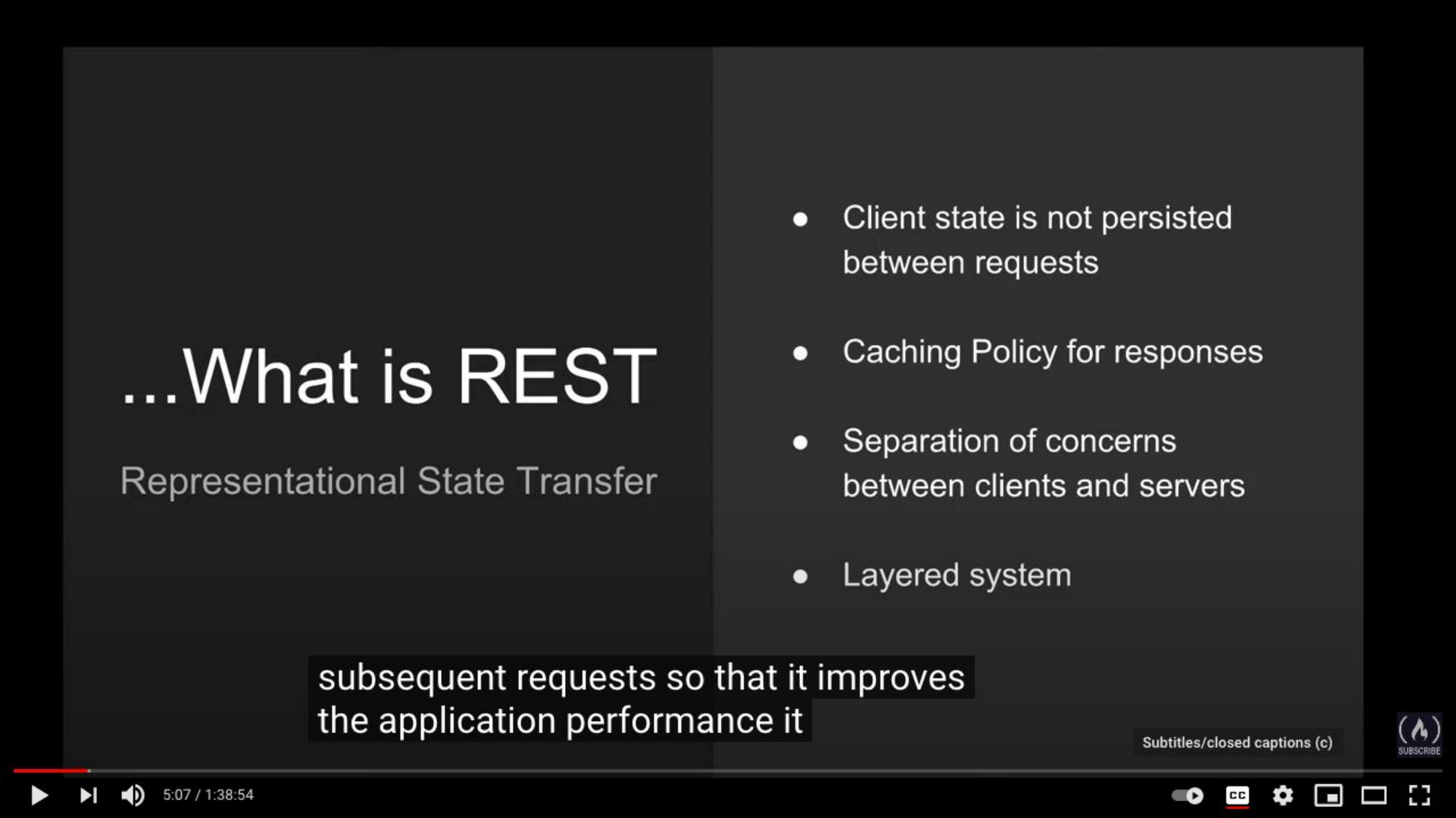
Subtitles/closed captions (c)

SUBSCRIBE

2:07 / 1:38:54


...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
HTML file versus a JavaScript file embedded in
Subtitles/closed captions (c)
SUBSCRIBE
3:07 / 1:38:54


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
interesting because you can reference different
Subtitles/closed captions (c)
SUBSCRIBE
4:07 / 1:38:54


...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
subsequent requests so that it improves
the application performance it
Subtitles/closed captions (c)
SUBSCRIBE
5:07 / 1:38:54

# What is an API...

Application Programming Interface

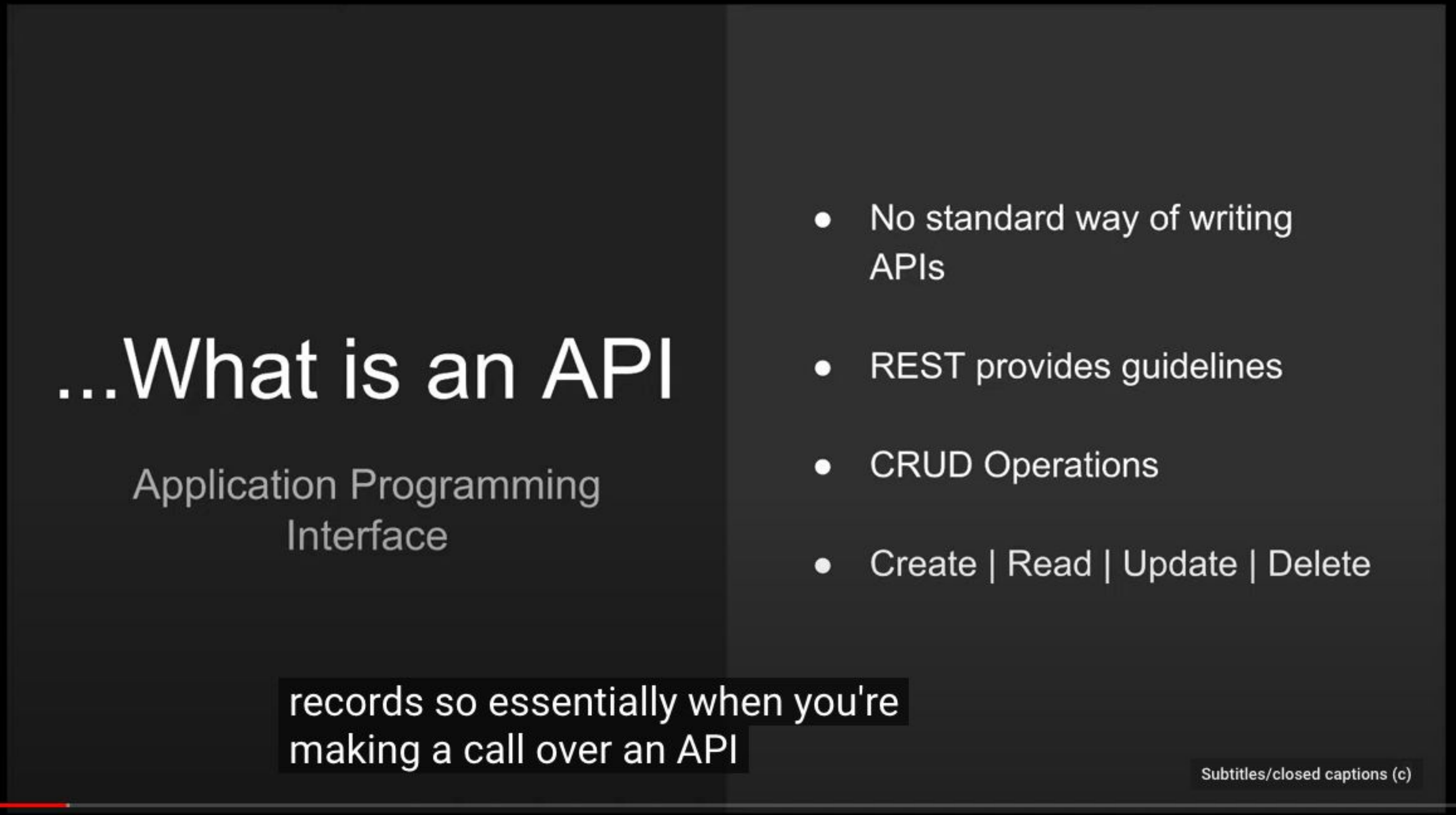
- Defines server-side functions that are supported
- Where requests should be made
- Format of the request and response

be made so essentially it's going to
tell you what the the pattern

Subtitles/closed captions (c)

SUBSCRIBE

6:07 / 1:38:54


...What is an API
Application Programming Interface
No standard way of writing APIs
REST provides guidelines
CRUD Operations
Create | Read | Update | Delete
records so essentially when you're making a call over an API
Subtitles/closed captions (c)
SUBSCRIBE
7:07 / 1:38:54

# IDE

Integrated Development Environment

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

which allows us to look at the data in
so as

Subtitles/closed captions (c)

SUBSCRIBE

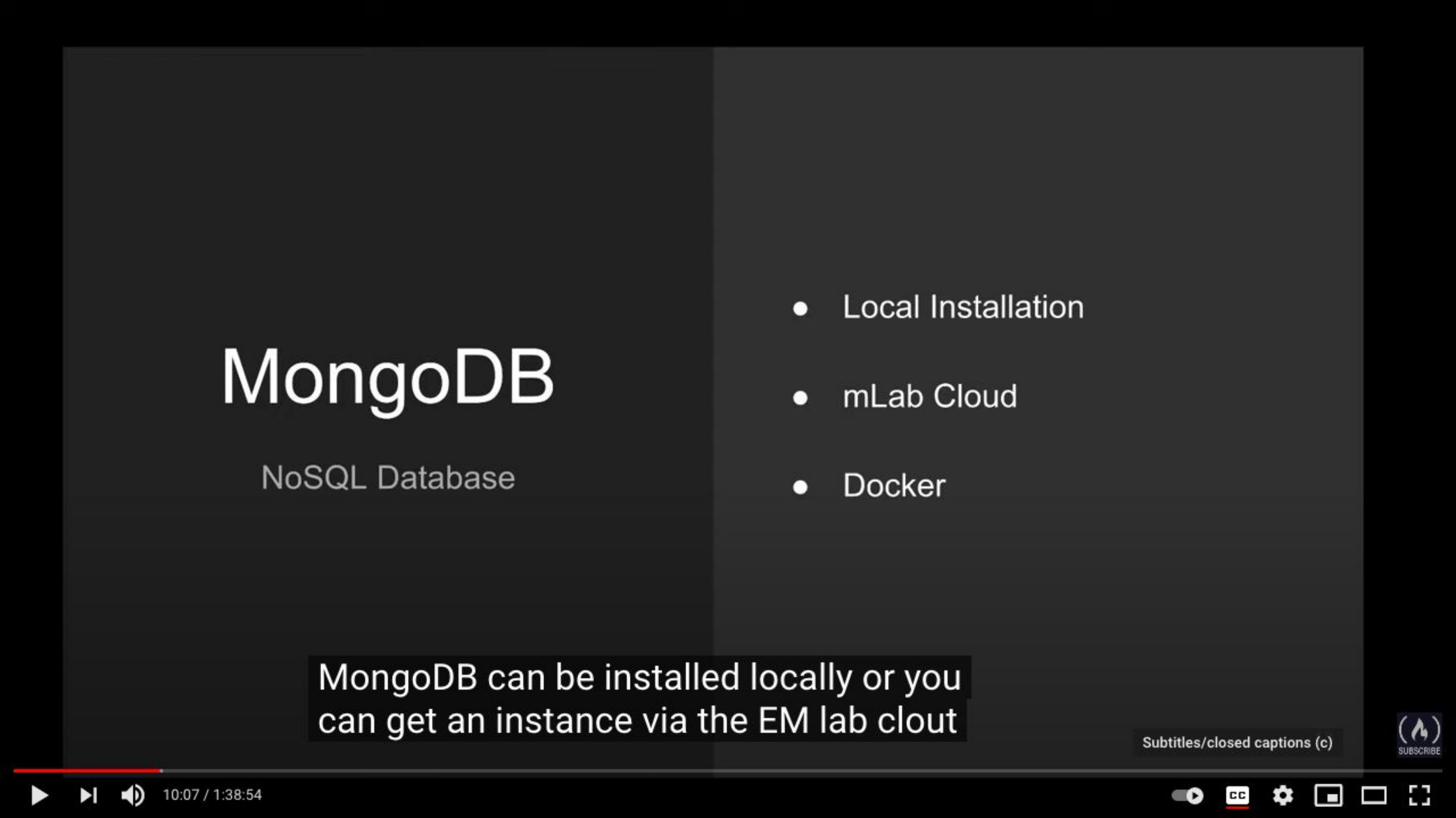
8:07 / 1:38:54

# Node.js

Javascript Runtime

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

babel or any other transpilers if you can


MongoDB
NoSQL Database
Local Installation
mLab Cloud
Docker
MongoDB can be installed locally or you
can get an instance via the EM lab clout
Subtitles/closed captions (c)
SUBSCRIBE
10:07 / 1:38:54

# REST Client

API Testing

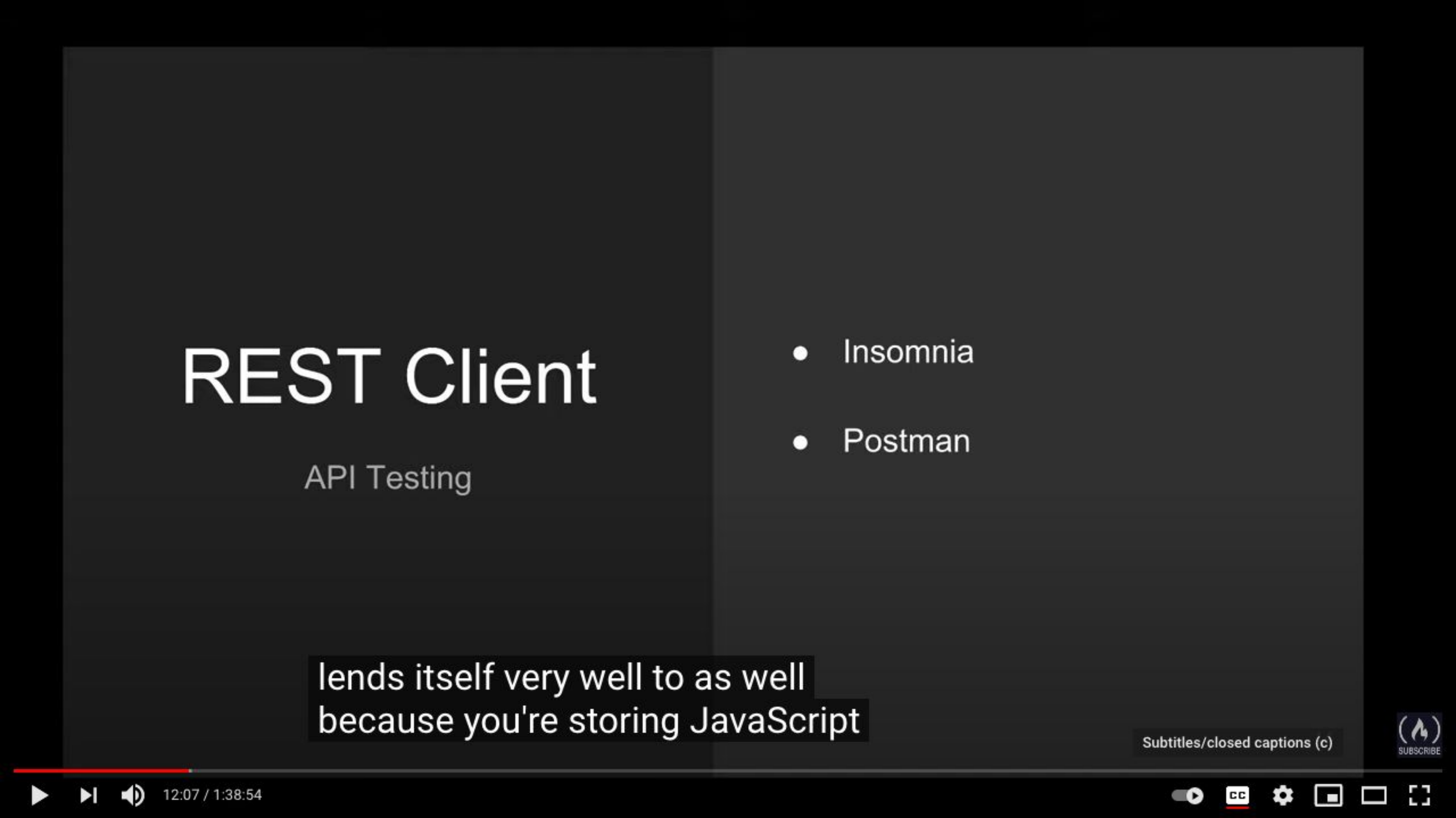
- Insomnia
- Postman

most of the other IDs require you to configure

Subtitles/closed captions (c)

SUBSCRIBE

11:07 / 1:38:54


REST Client
API Testing
Insomnia
Postman
lends itself very well to as well
because you're storing JavaScript
Subtitles/closed captions (c)
SUBSCRIBE
12:07 / 1:38:54

# Application Setup

Getting Started

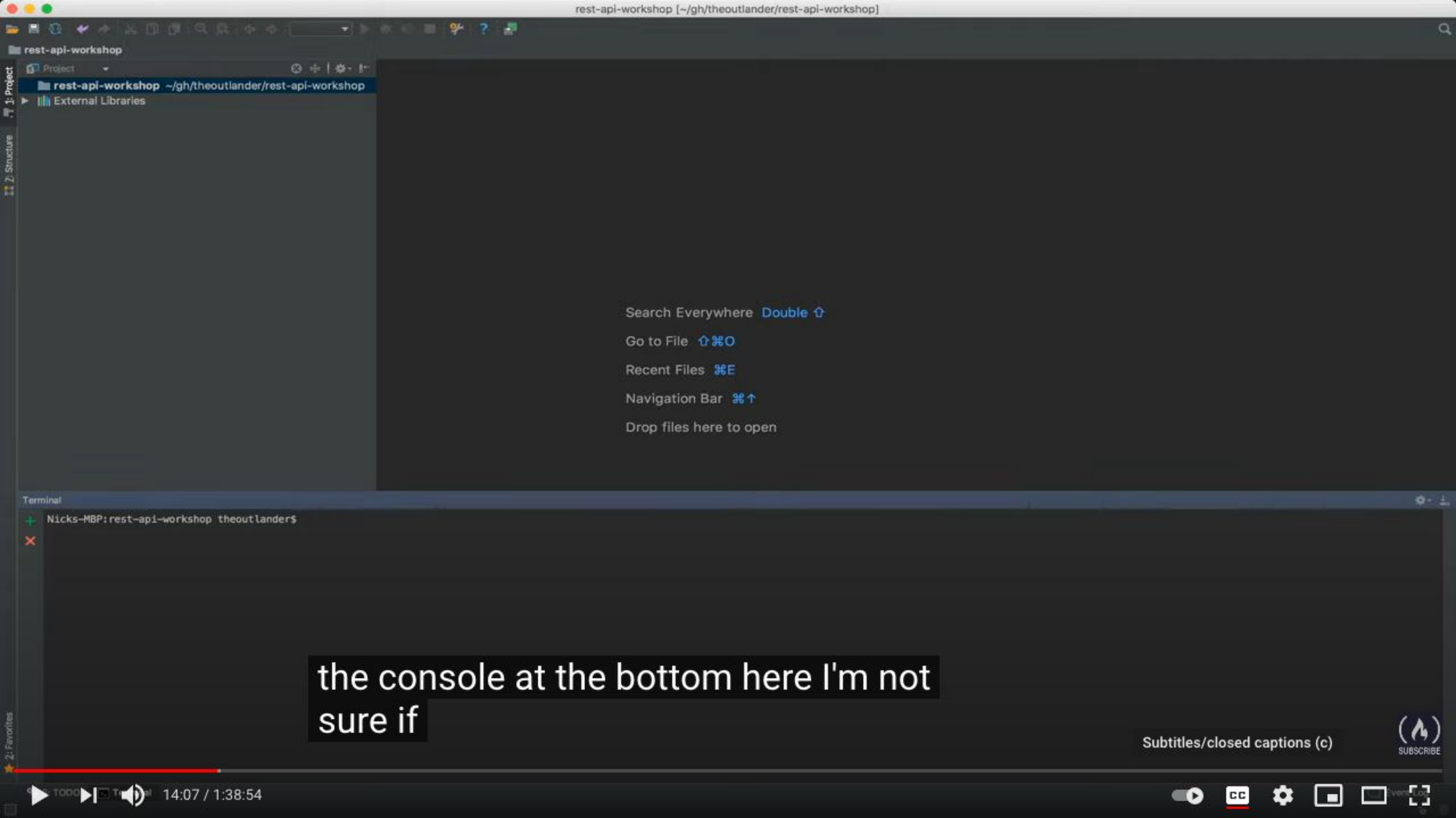
- Project Folder
- Initialize Node.js Project
- Hello World
- Startup Scripts

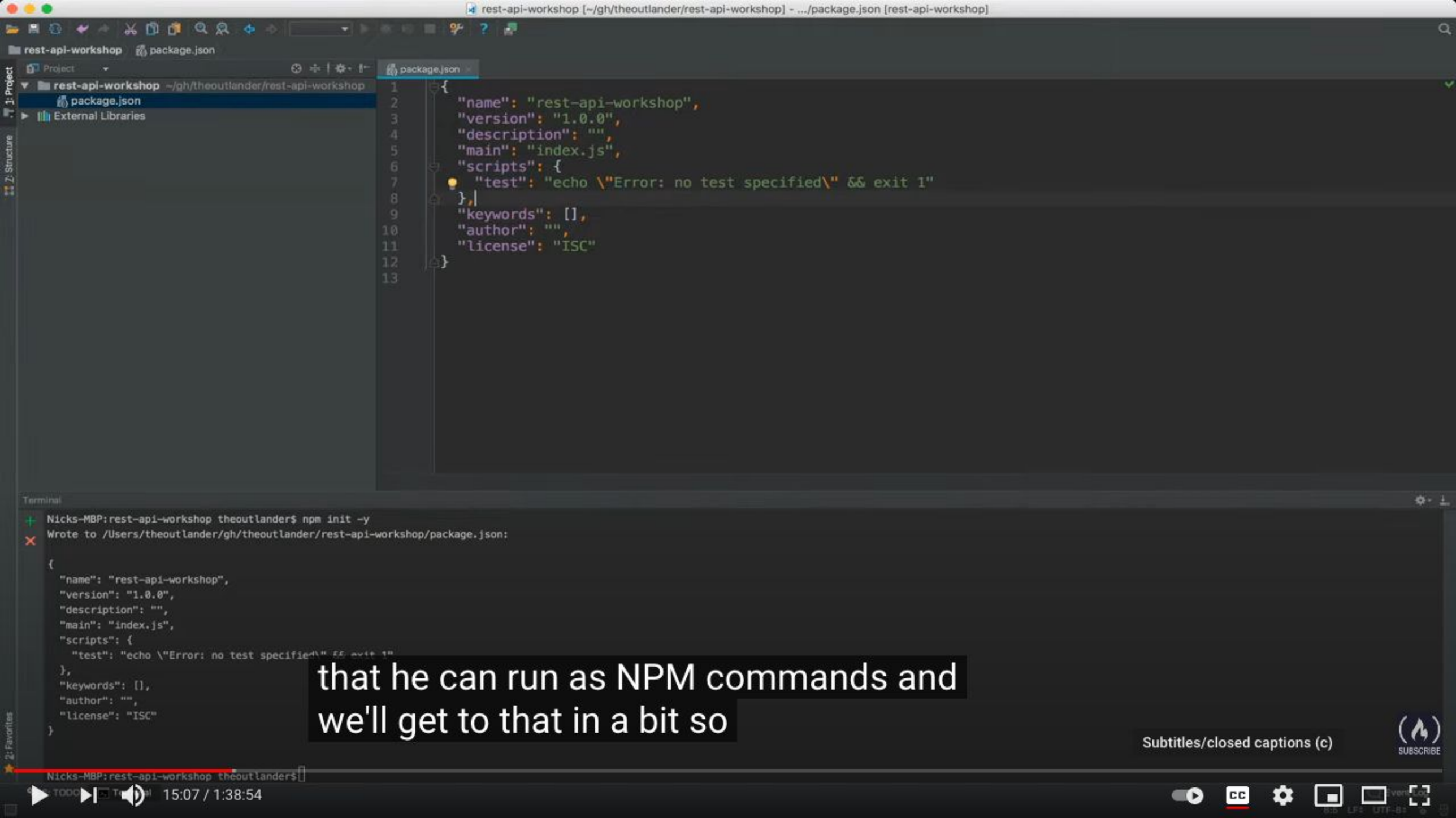
project folder and then you will open up
your IDE either

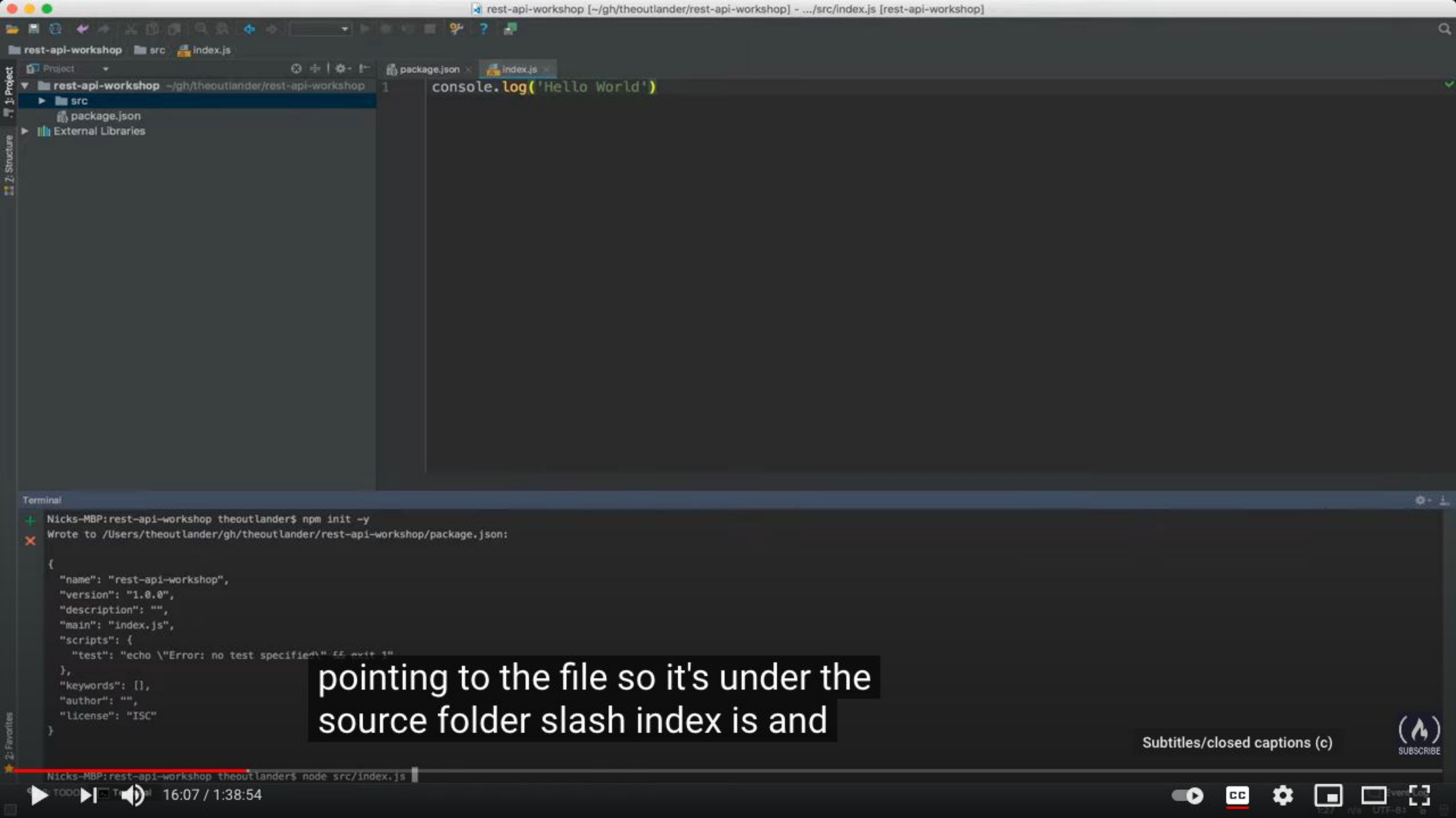
Subtitles/closed captions (c)

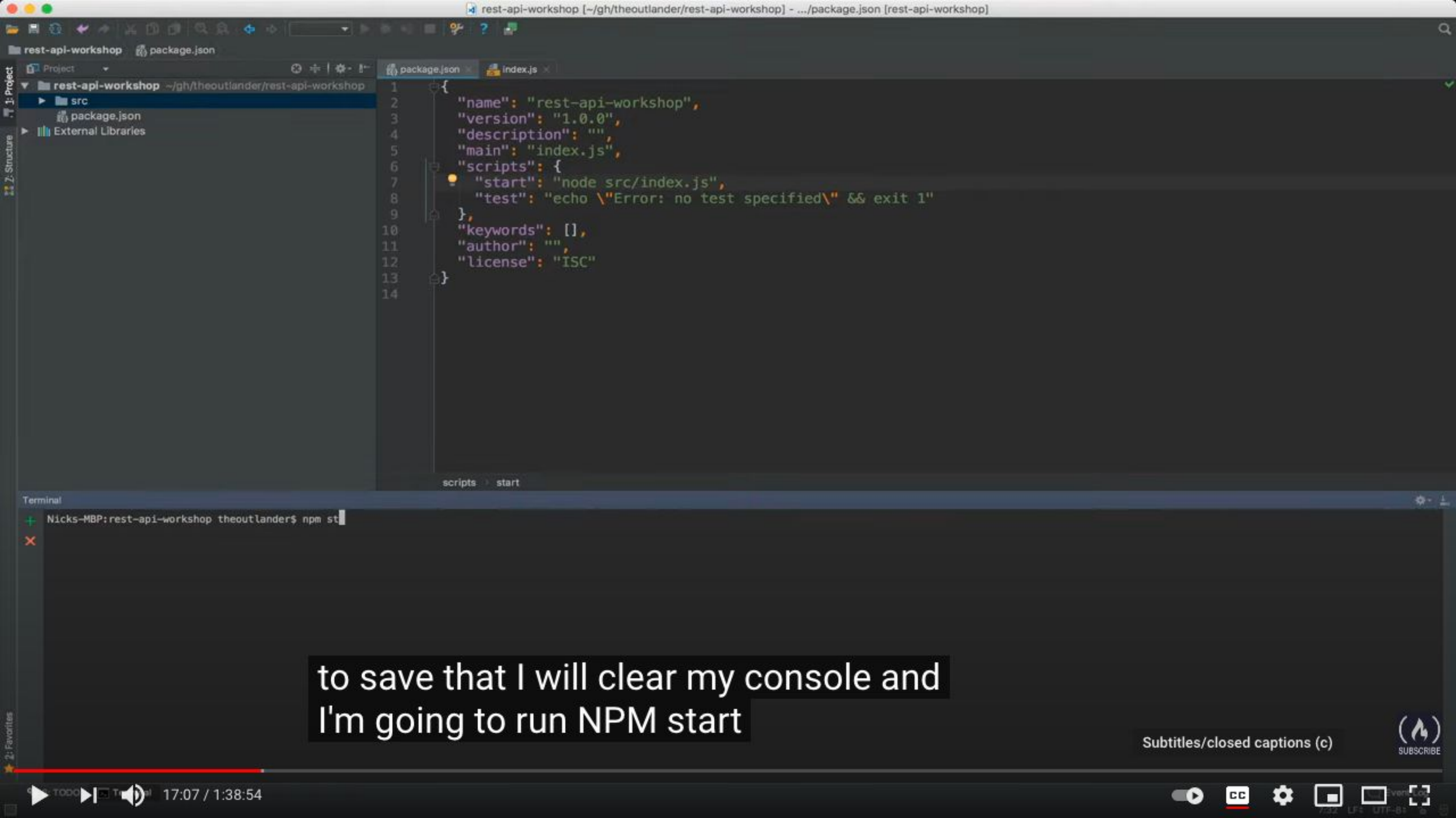
SUBSCRIBE

13:07 / 1:38:54


rest-api-workshop [~/gh/theoutlander/rest-api-workshop]
rest-api-workshop
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
External Libraries
Search Everywhere Double ⇧
Go to File ⇧⌘O
Recent Files ⌘E
Navigation Bar ⌘↑
Drop files here to open
Terminal
Nicks-MBP:rest-api-workshop theoutlander$
the console at the bottom here I'm not sure if
Subtitles/closed captions (c)
SUBSCRIBE
14:07 / 1:38:54


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
package.json
External Libraries
{
"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"
}
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$
that he can run as NPM commands and
we'll get to that in a bit so
Subtitles/closed captions (c)
SUBSCRIBE
15:07 / 1:38:54


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
src
package.json
External Libraries
package.json
index.js
1
console.log('Hello World')
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"
}
pointing to the file so it's under the
source folder slash index is and
Subtitles/closed captions (c)
SUBSCRIBE
Nicks-MBP:rest-api-workshop theoutlander$ node src/index.js
16:07 / 1:38:54


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
package.json
index.js
{
"name": "rest-api-workshop",
"version": "1.0.0",
"description": "",
"main": "index.js",
"scripts": {
"start": "node src/index.js",
"test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC"
}
scripts
start
Terminal
Nicks-MBP:rest-api-workshop theoutlander$ npm st
to save that I will clear my console and
I'm going to run NPM start
Subtitles/closed captions (c)
SUBSCRIBE
17:07 / 1:38:54

# Express

Web Server

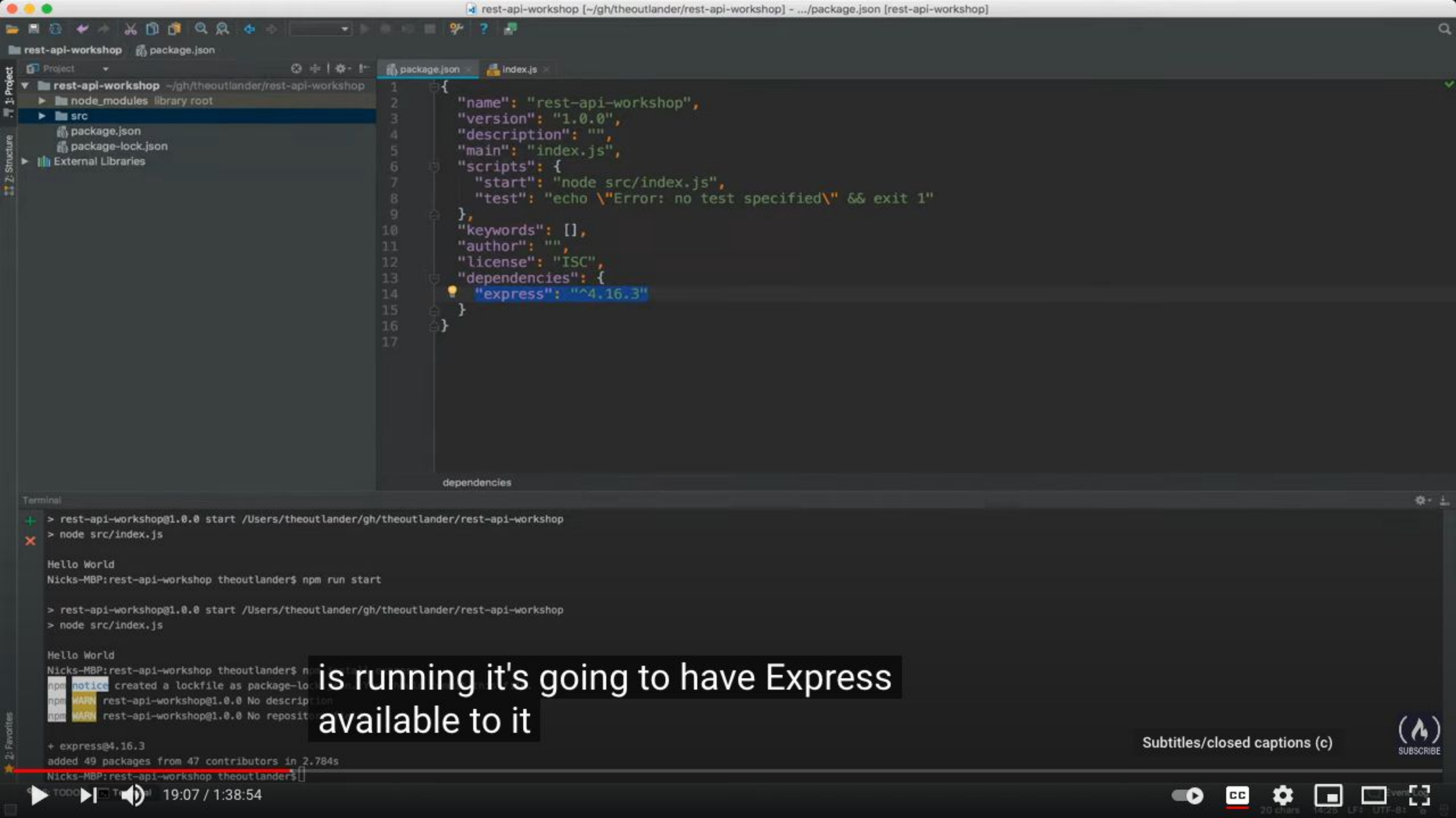
- Install Dependency
- Reference the express library
- Create express application instance

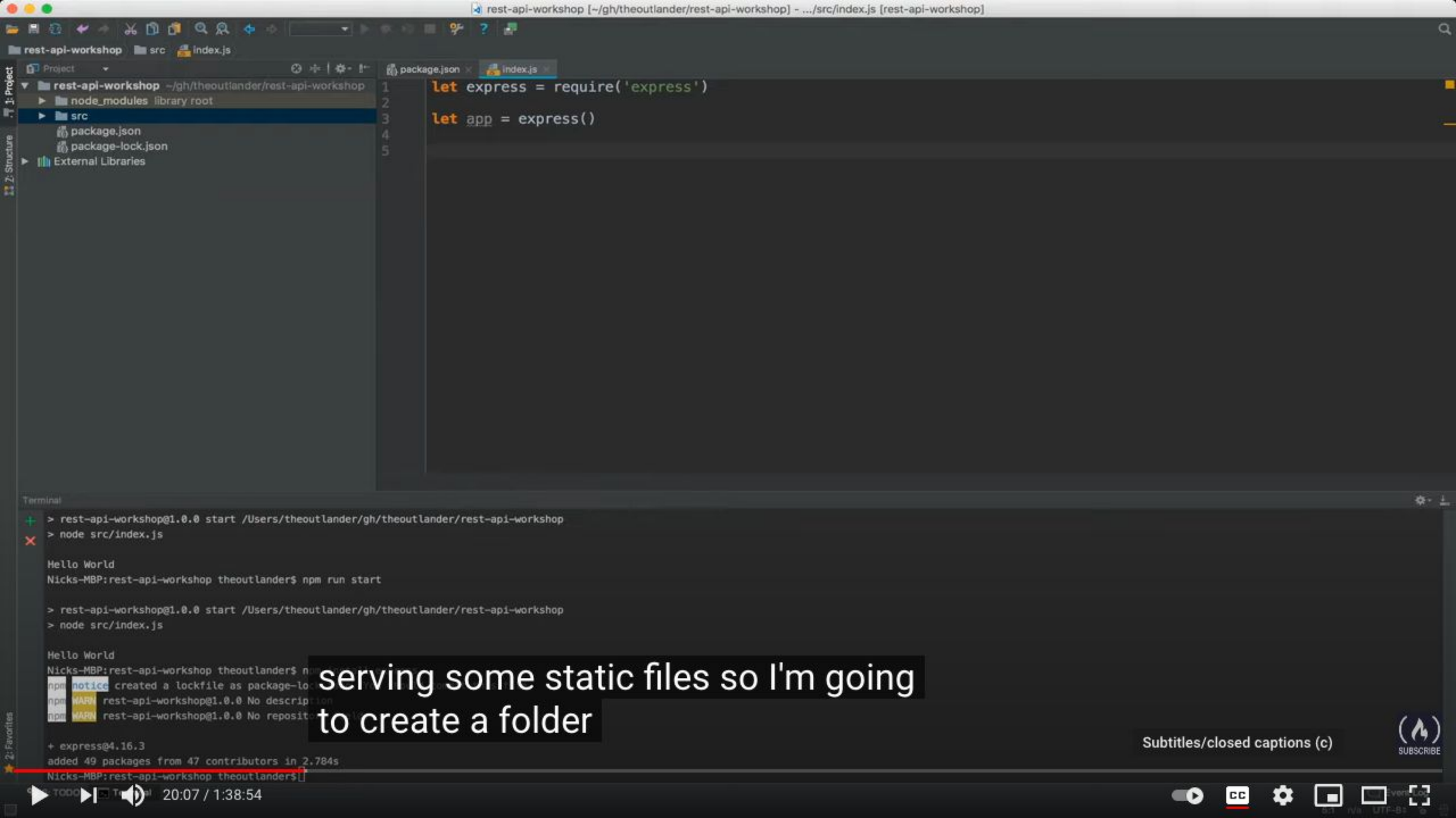
is basically a lightweight web server
what

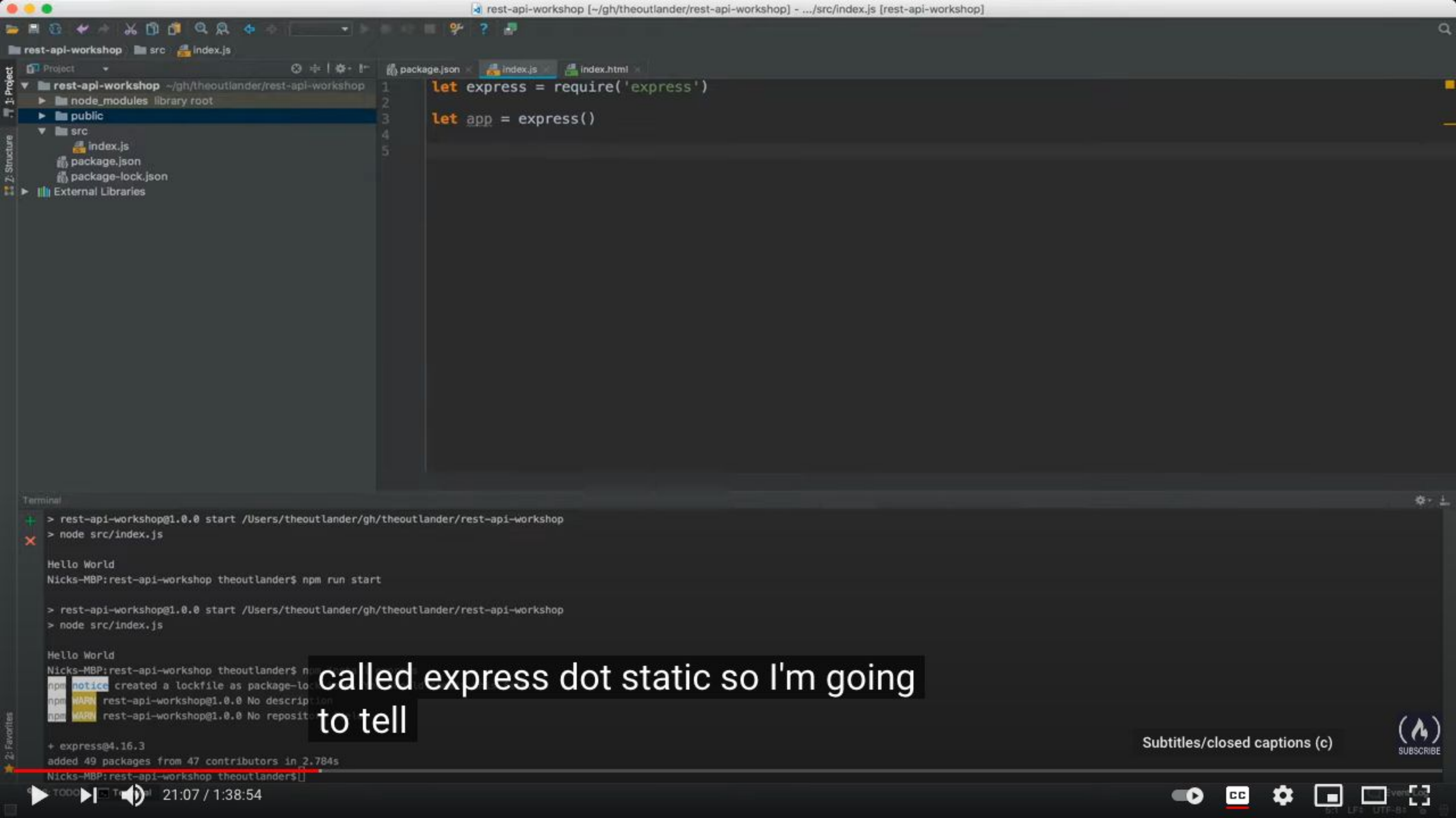
Subtitles/closed captions (c)

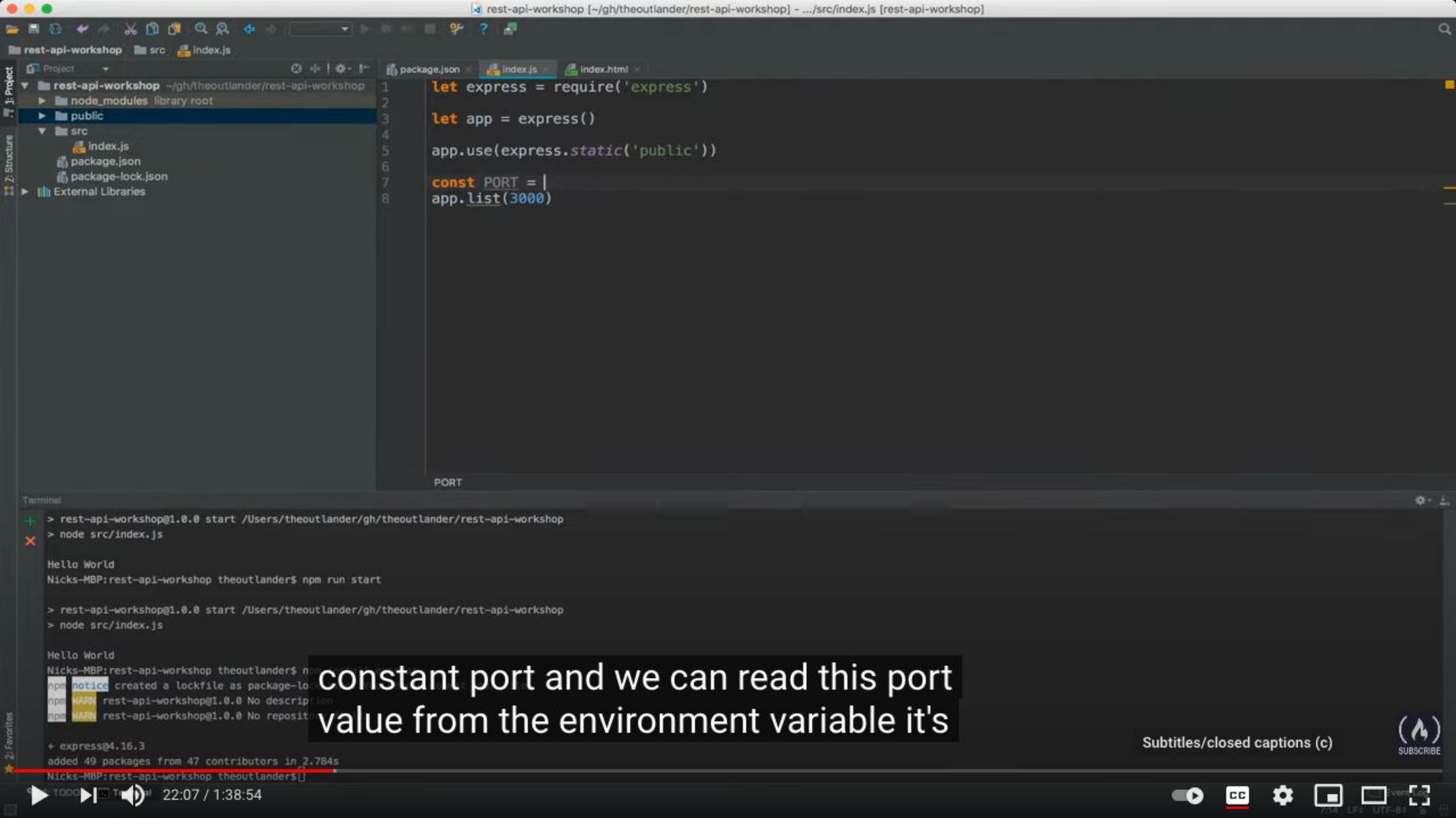
SUBSCRIBE

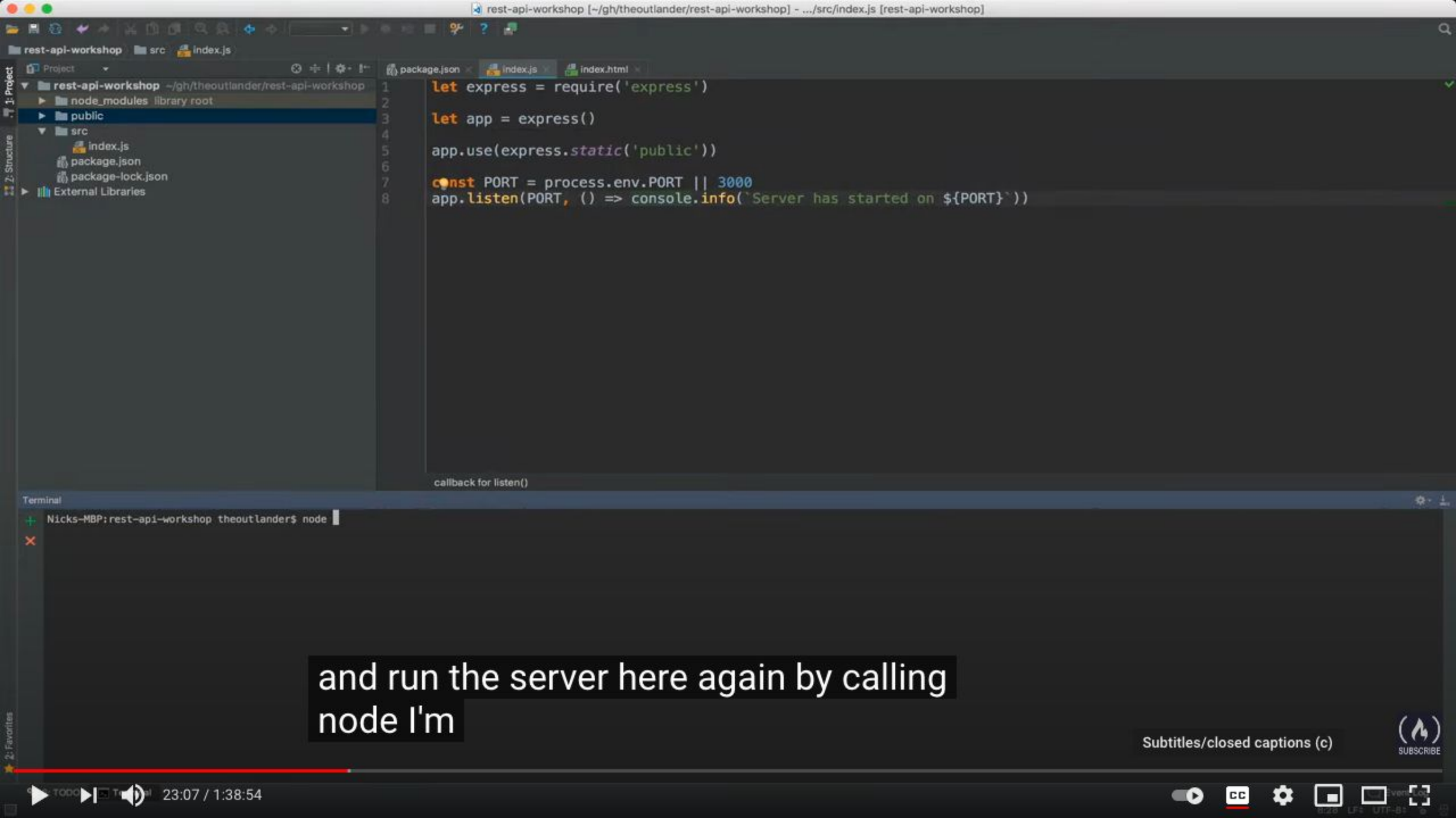
18:07 / 1:38:54

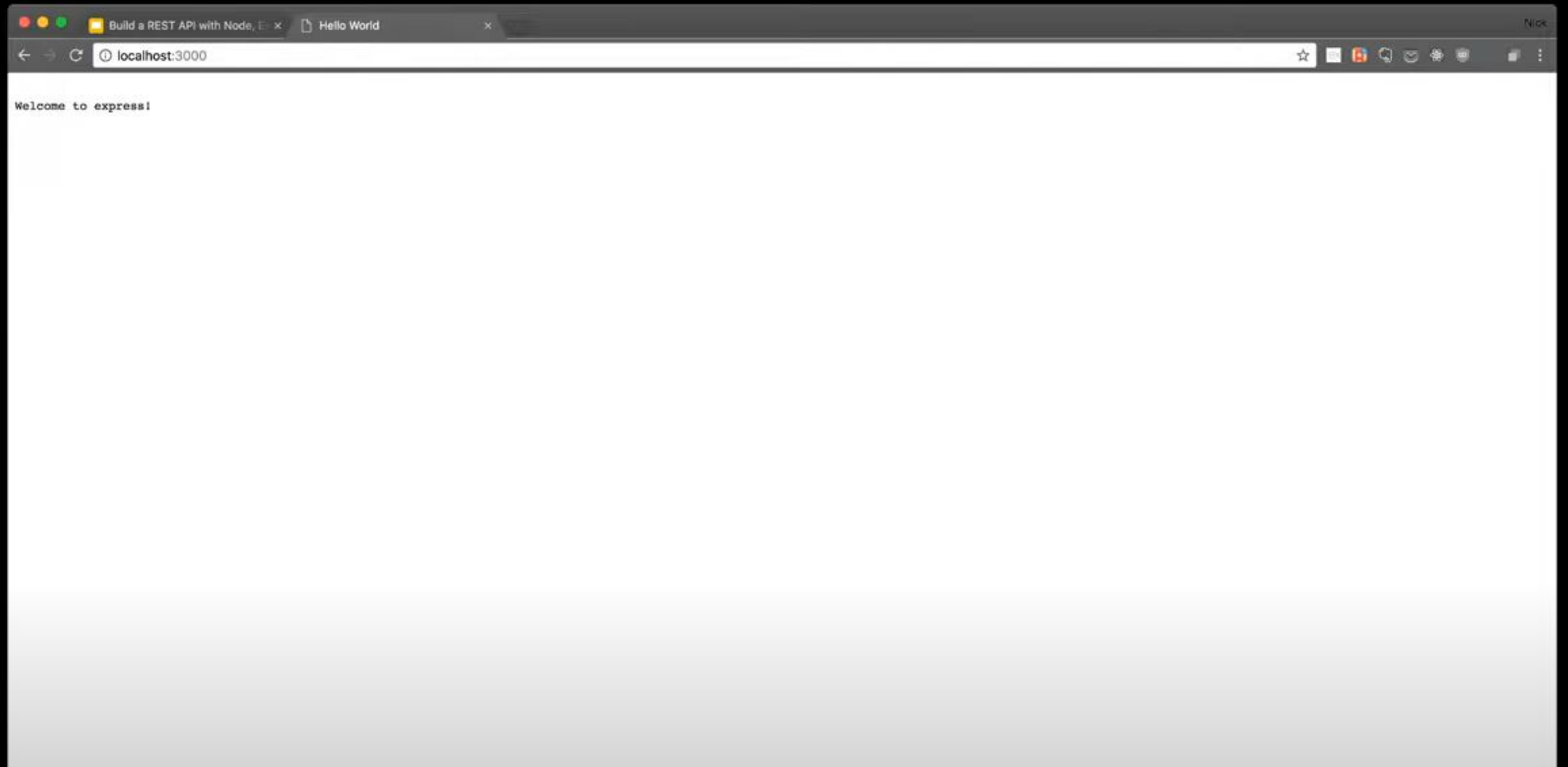

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../package.json [rest-api-workshop]
rest-api-workshop
package.json
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
src
package.json
package-lock.json
External Libraries
package.json
index.js
{
"name": "rest-api-workshop",
"version": "1.0.0",
"description": "",
"main": "index.js",
"scripts": {
"start": "node src/index.js",
"test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC",
"dependencies": {
"express": "^4.16.3"
}
}
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$ n
npm notice created a lockfile as package-lo
npm WARN rest-api-workshop@1.0.0 No descrip
npm WARN rest-api-workshop@1.0.0 No reposit
+ express@4.16.3
added 49 packages from 47 contributors in 2.784s
Nicks-MBP:rest-api-workshop theoutlander$
is running it's going to have Express available to it
Subtitles/closed captions (c)
SUBSCRIBE
19:07 / 1:38:54


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
src
package.json
package-lock.json
External Libraries
package.json
index.js
let express = require('express')
let app = express()
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 notice created a lockfile as package-lo
npm WARN rest-api-workshop@1.0.0 No descrip
npm WARN rest-api-workshop@1.0.0 No reposit
+ express@4.16.3
added 49 packages from 47 contributors in 2.784s
Nicks-MBP:rest-api-workshop theoutlander$
serving some static files so I'm going
to create a folder
Subtitles/closed captions (c)
SUBSCRIBE
20:07 / 1:38:54


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
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
index.html
1 let express = require('express')
2
3 let app = express()
4
5
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 notice created a lockfile as package-lo
npm WARN rest-api-workshop@1.0.0 No descrip
npm WARN rest-api-workshop@1.0.0 No reposit
+ express@4.16.3
added 49 packages from 47 contributors in 2.784s
Nicks-MBP:rest-api-workshop theoutlander$
called express dot static so I'm going
to tell
Subtitles/closed captions (c)
SUBSCRIBE
21:07 / 1:38:54


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
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
index.html
1 let express = require('express')
2
3 let app = express()
4
5 app.use(express.static('public'))
6
7 const PORT =
8 app.list(3000)
PORT
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 notice created a lockfile as package-lo
npm WARN rest-api-workshop@1.0.0 No descrip
npm WARN rest-api-workshop@1.0.0 No reposit
+ express@4.16.3
added 49 packages from 47 contributors in 2.784s
Nicks-MBP:rest-api-workshop theoutlander$
constant port and we can read this port
value from the environment variable it's
Subtitles/closed captions (c)
SUBSCRIBE
22:07 / 1:38:54


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
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
index.html
let express = require('express')
let app = express()
app.use(express.static('public'))
const PORT = process.env.PORT || 3000
app.listen(PORT, () => console.info(`Server has started on ${PORT}`))
callback for listen()
Terminal
Nicks-MBP:rest-api-workshop theoutlander$ node
and run the server here again by calling
node I'm
Subtitles/closed captions (c)
SUBSCRIBE
23:07 / 1:38:54




file into public folder and you can
serve it through through

Subtitles/closed captions (c)

SUBSCRIBE

24:07 / 1:38:54

# Route

Mount to Express

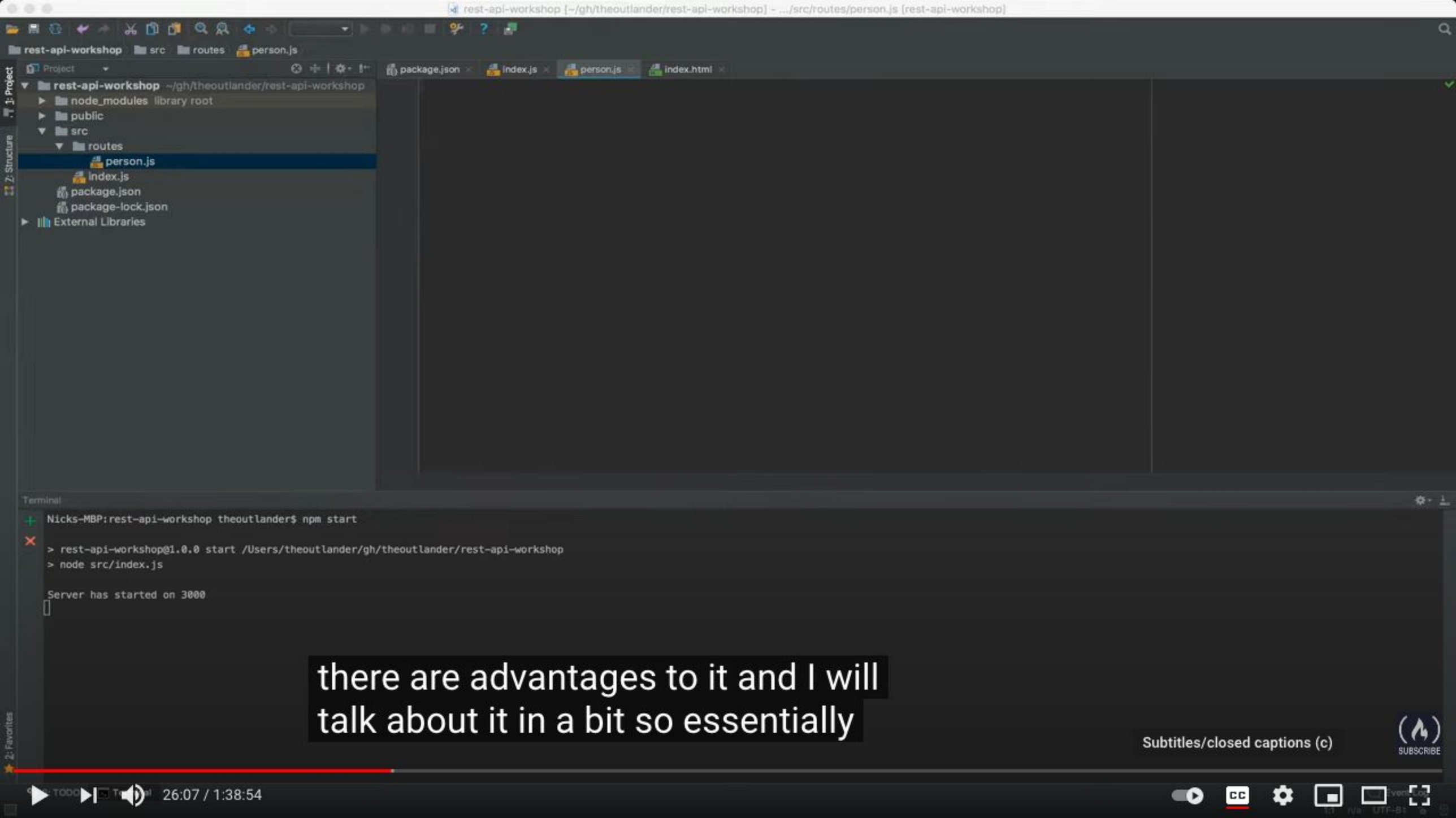
- GET Request
- Request Parameters
- Query Parameters

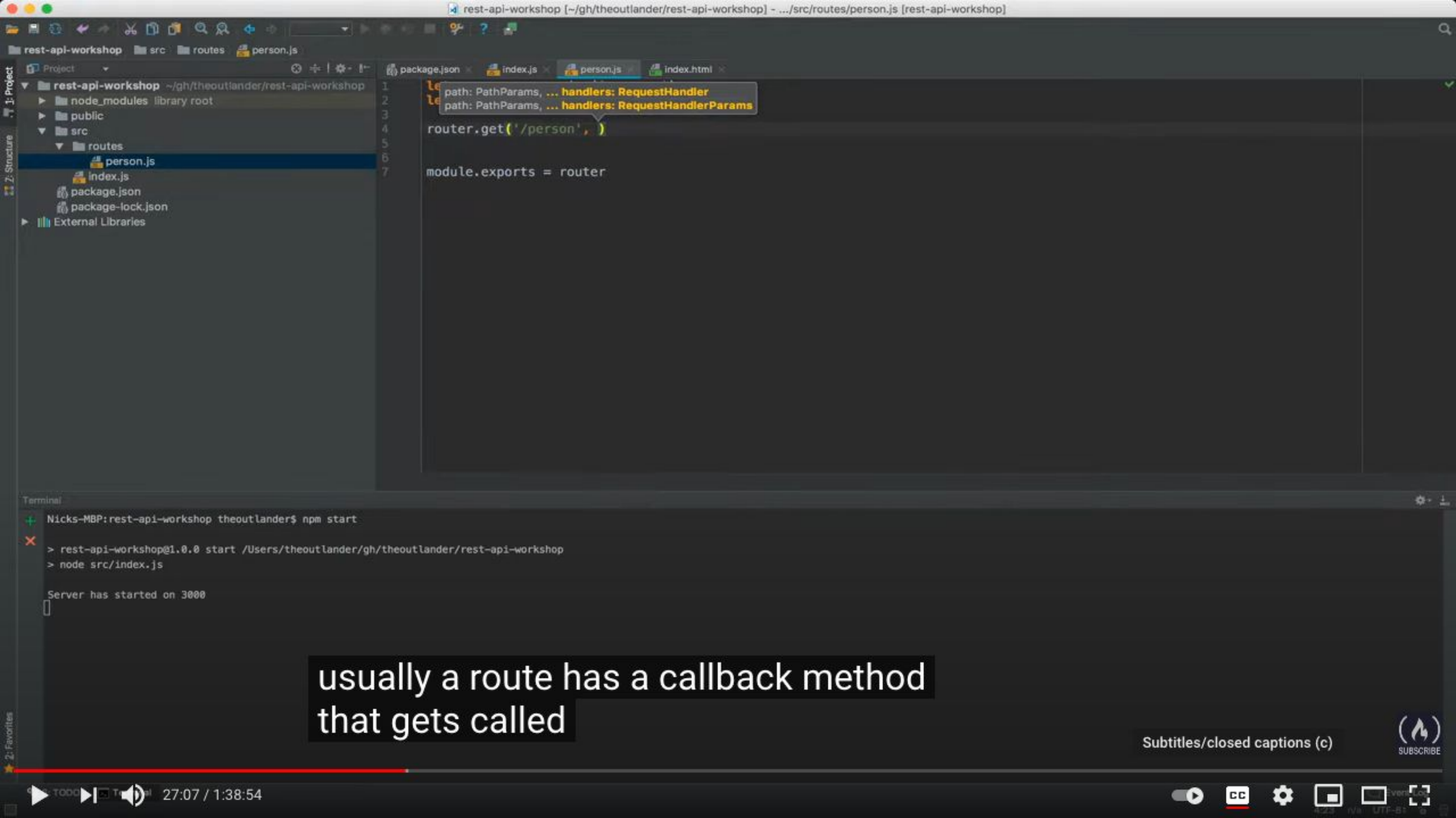
you may have heard about something called

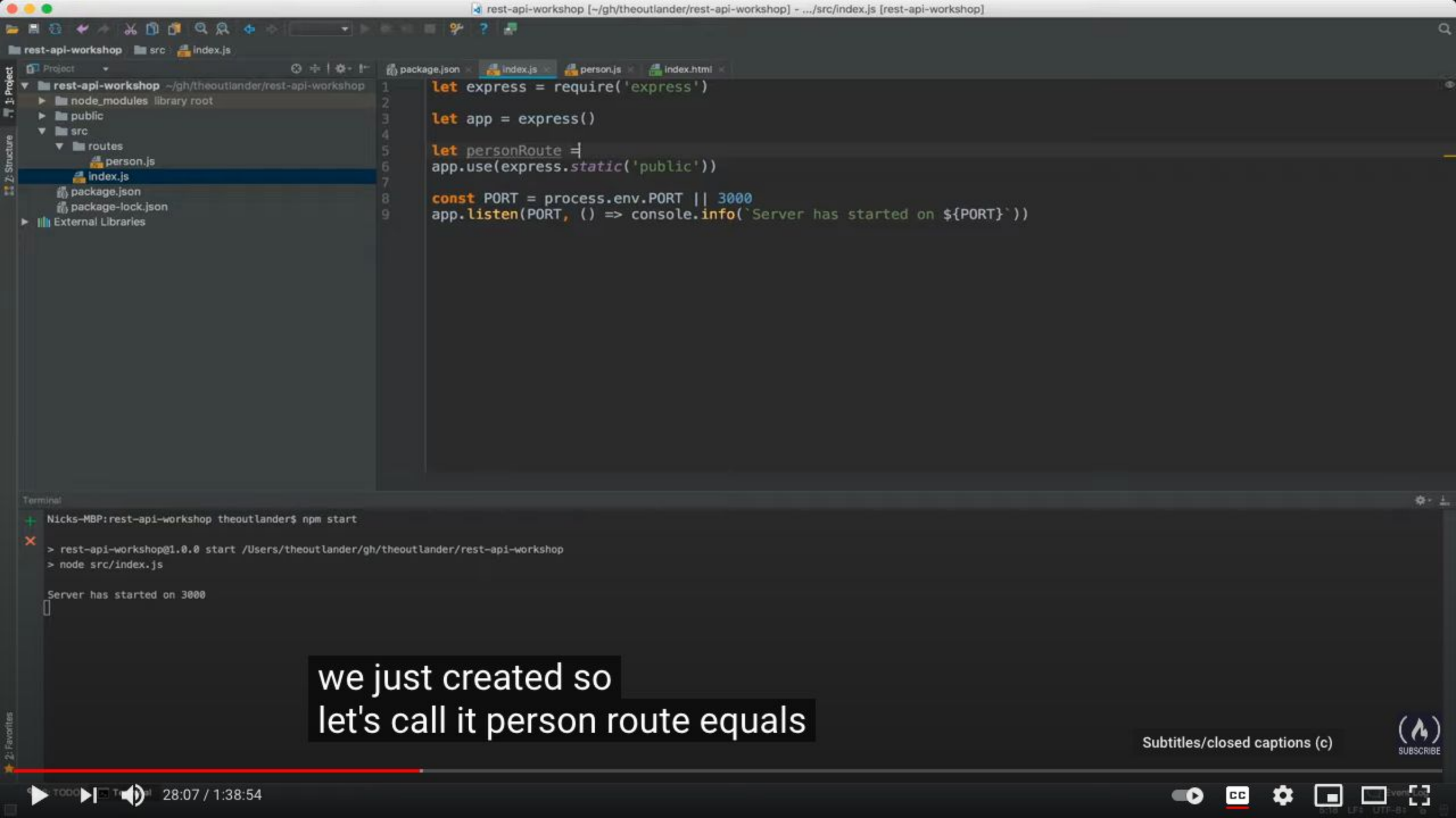
Subtitles/closed captions (c)

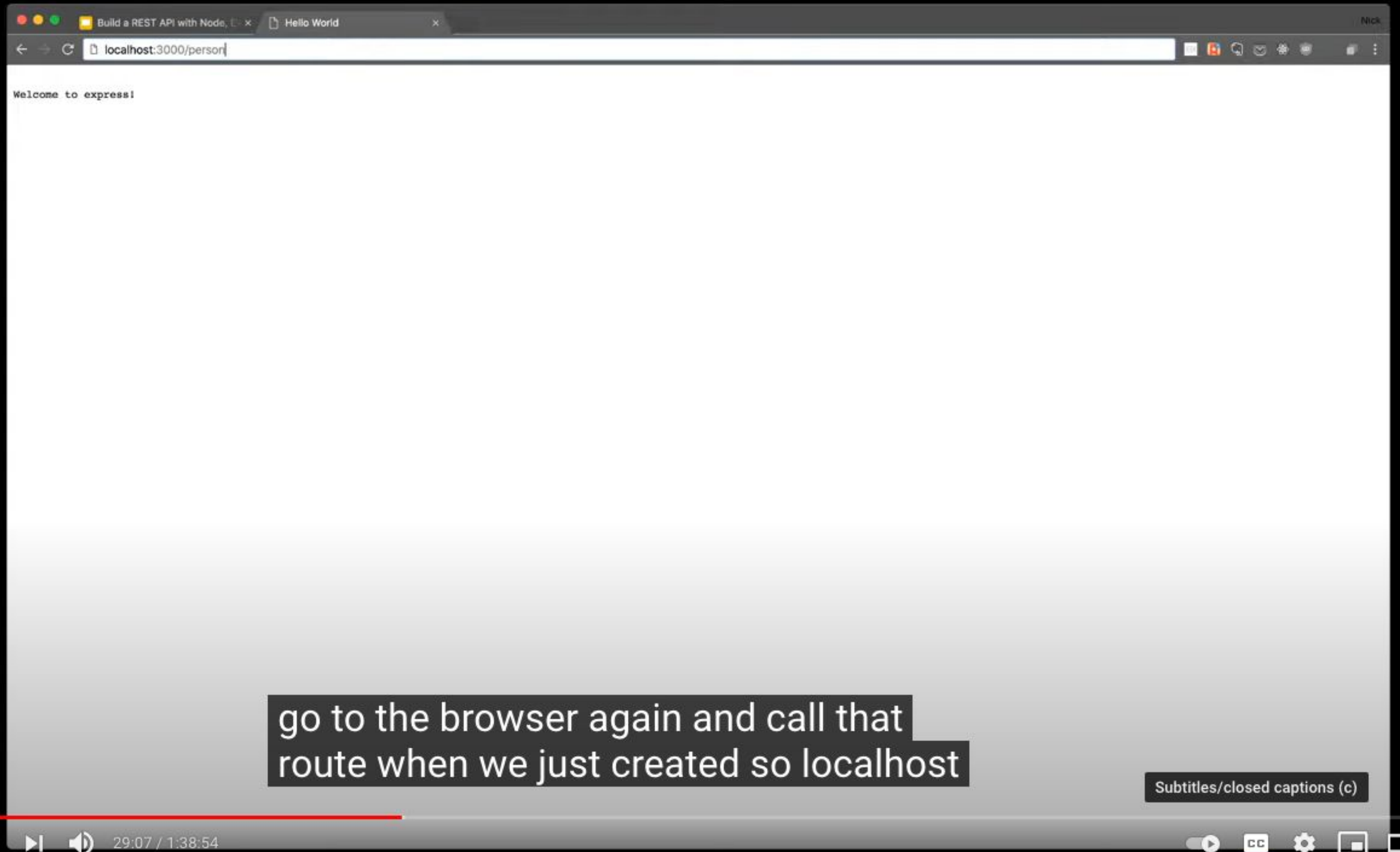
SUBSCRIBE

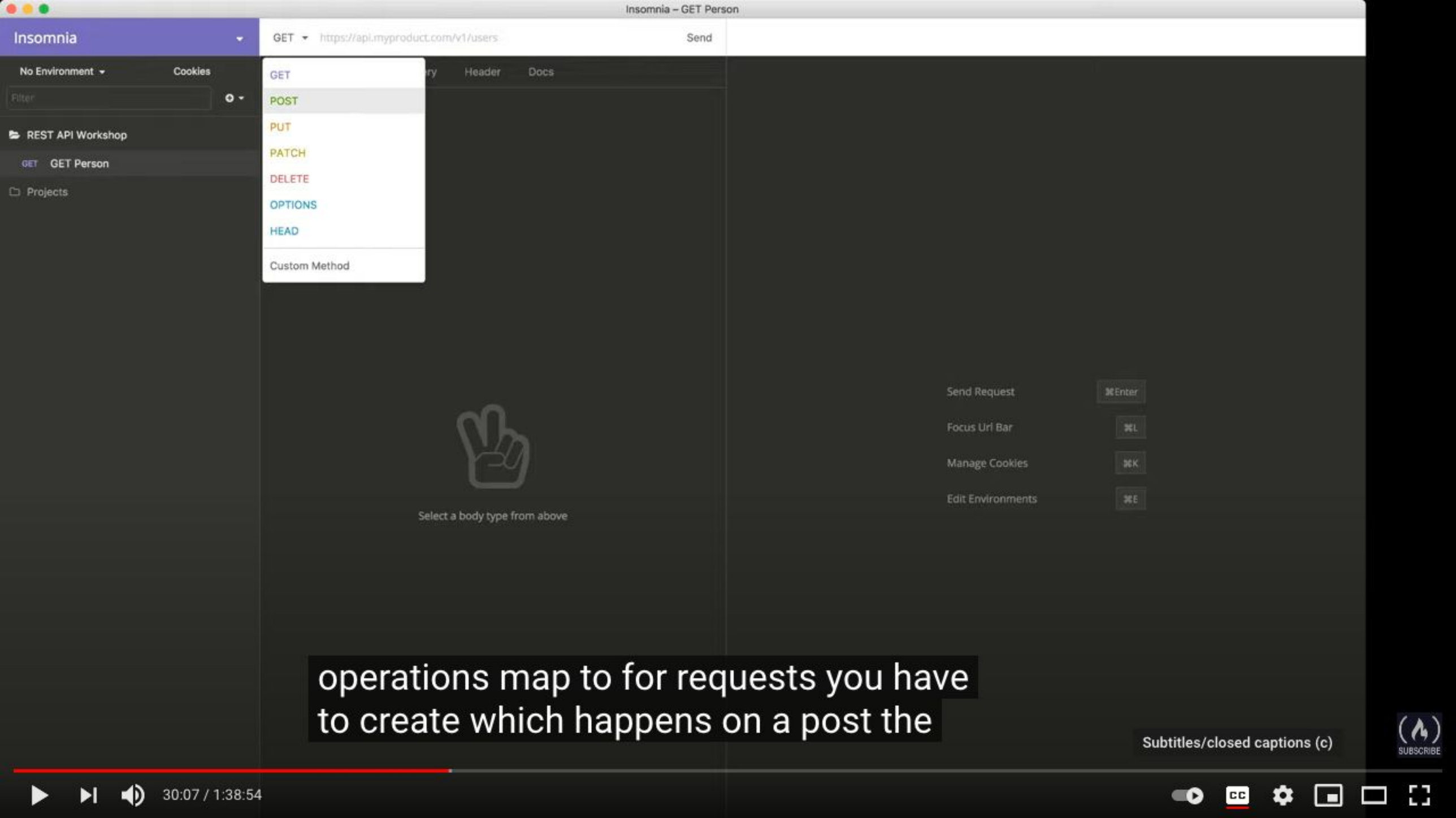
25:07 / 1:38:54

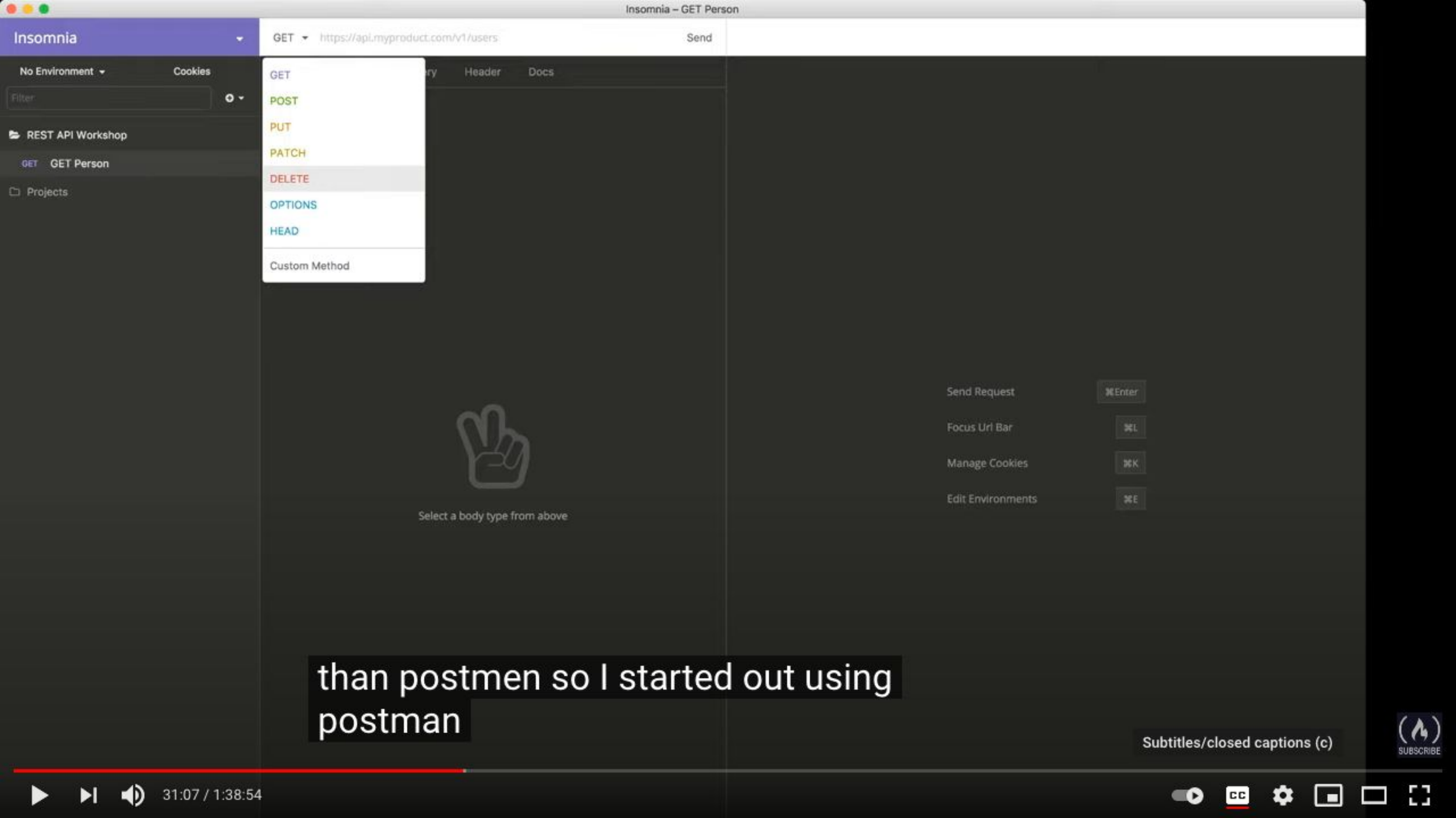

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/person.js [rest-api-workshop]
rest-api-workshop
src
routes
person.js
Project
package.json
index.js
person.js
index.html
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
src
routes
person.js
index.js
package.json
package-lock.json
External Libraries
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
there are advantages to it and I will
talk about it in a bit so essentially
Subtitles/closed captions (c)
SUBSCRIBE
26:07 / 1:38:54

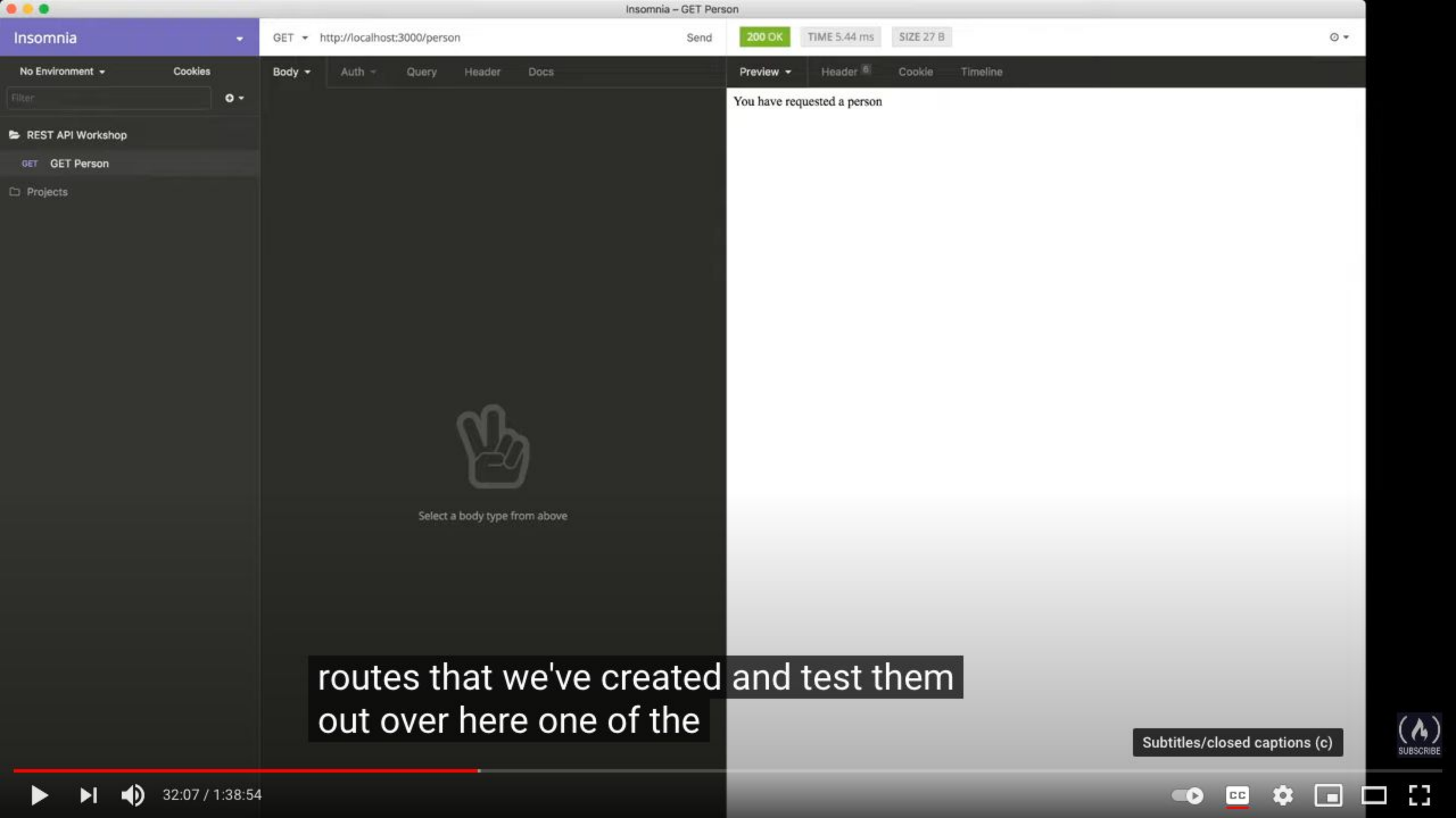

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/person.js [rest-api-workshop]
rest-api-workshop
src
routes
person.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
src
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
person.js
index.html
path: PathParams, ... handlers: RequestHandler
path: PathParams, ... handlers: RequestHandlerParams
router.get('/person', )
module.exports = router
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
usually a route has a callback method
that gets called
Subtitles/closed captions (c)
SUBSCRIBE
27:07 / 1:38:54

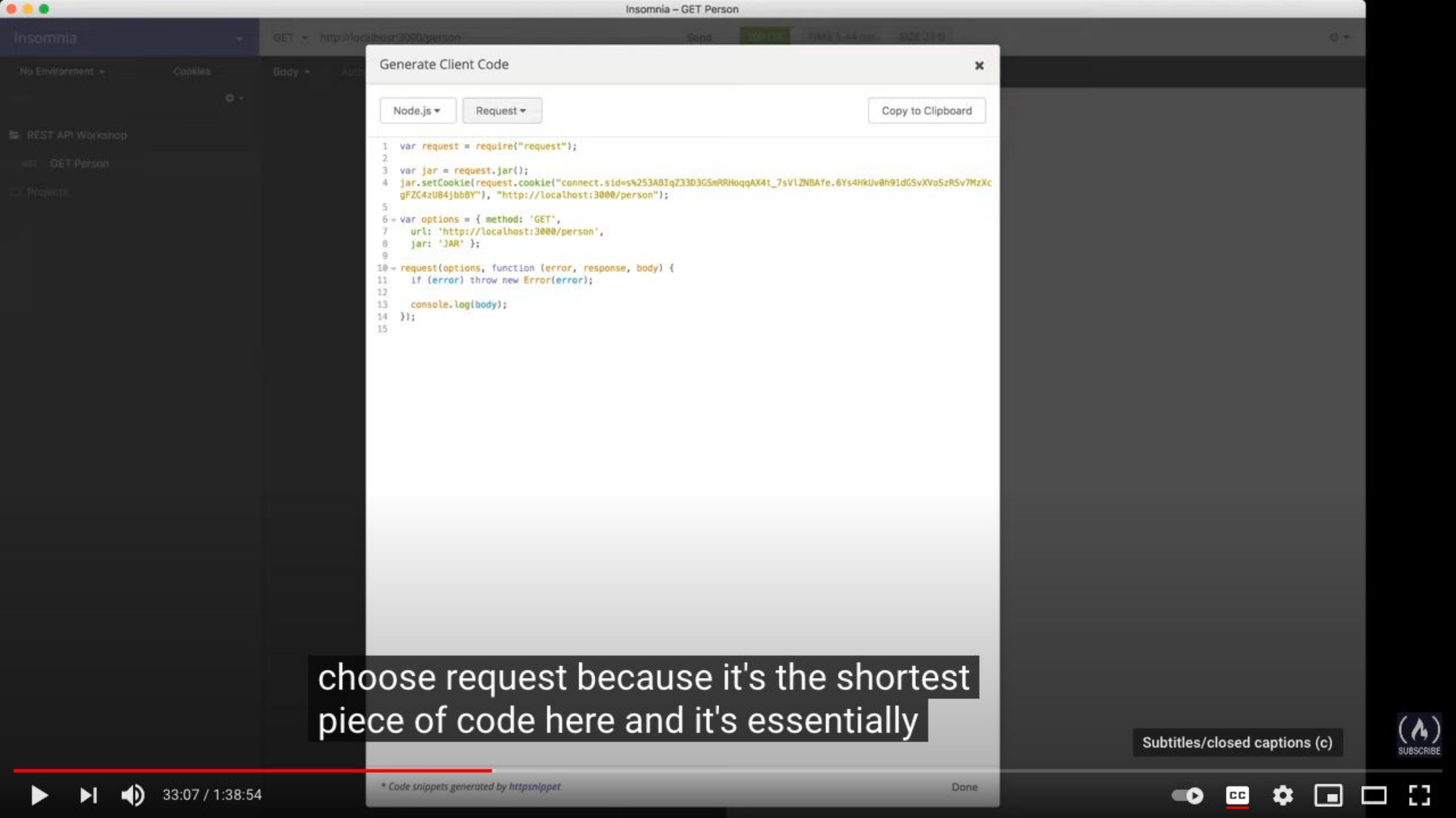

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/index.js [rest-api-workshop]
rest-api-workshop
src
index.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
src
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
person.js
index.html
let express = require('express')
let app = express()
let personRoute =
app.use(express.static('public'))
const PORT = process.env.PORT || 3000
app.listen(PORT, () => console.info(`Server has started on ${PORT}`))
Terminal
Nicks-MBP:rest-api-workshop theoutlander$ 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
we just created so
let's call it person route equals
Subtitles/closed captions (c)
SUBSCRIBE
28:07 / 1:38:54

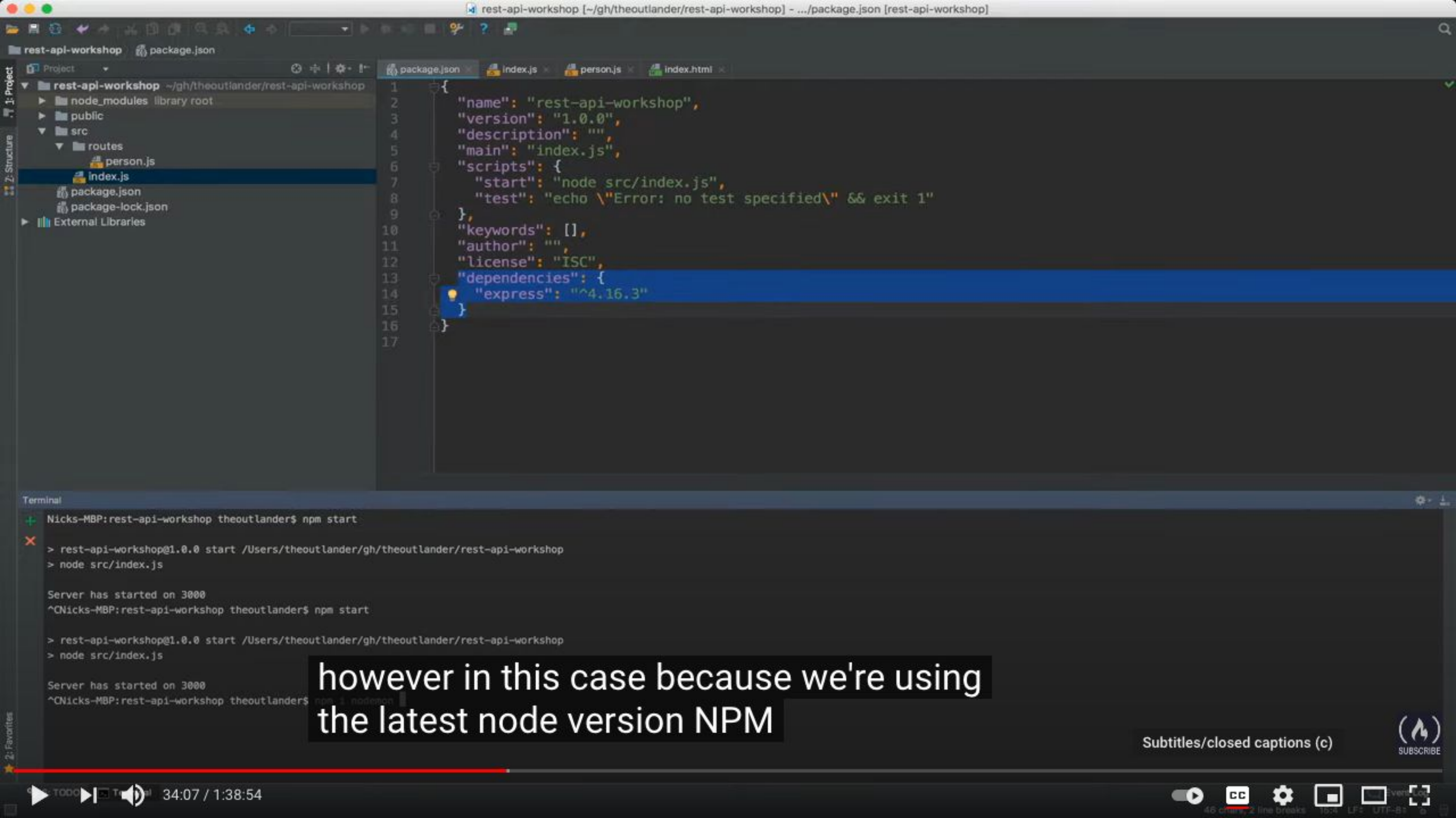

Build a REST API with Node,
Hello World
Nick
localhost:3000/person
Welcome to express!
go to the browser again and call that
route when we just created so localhost
Subtitles/closed captions (c)
SUBSCRIBE
29:07 / 1:38:54

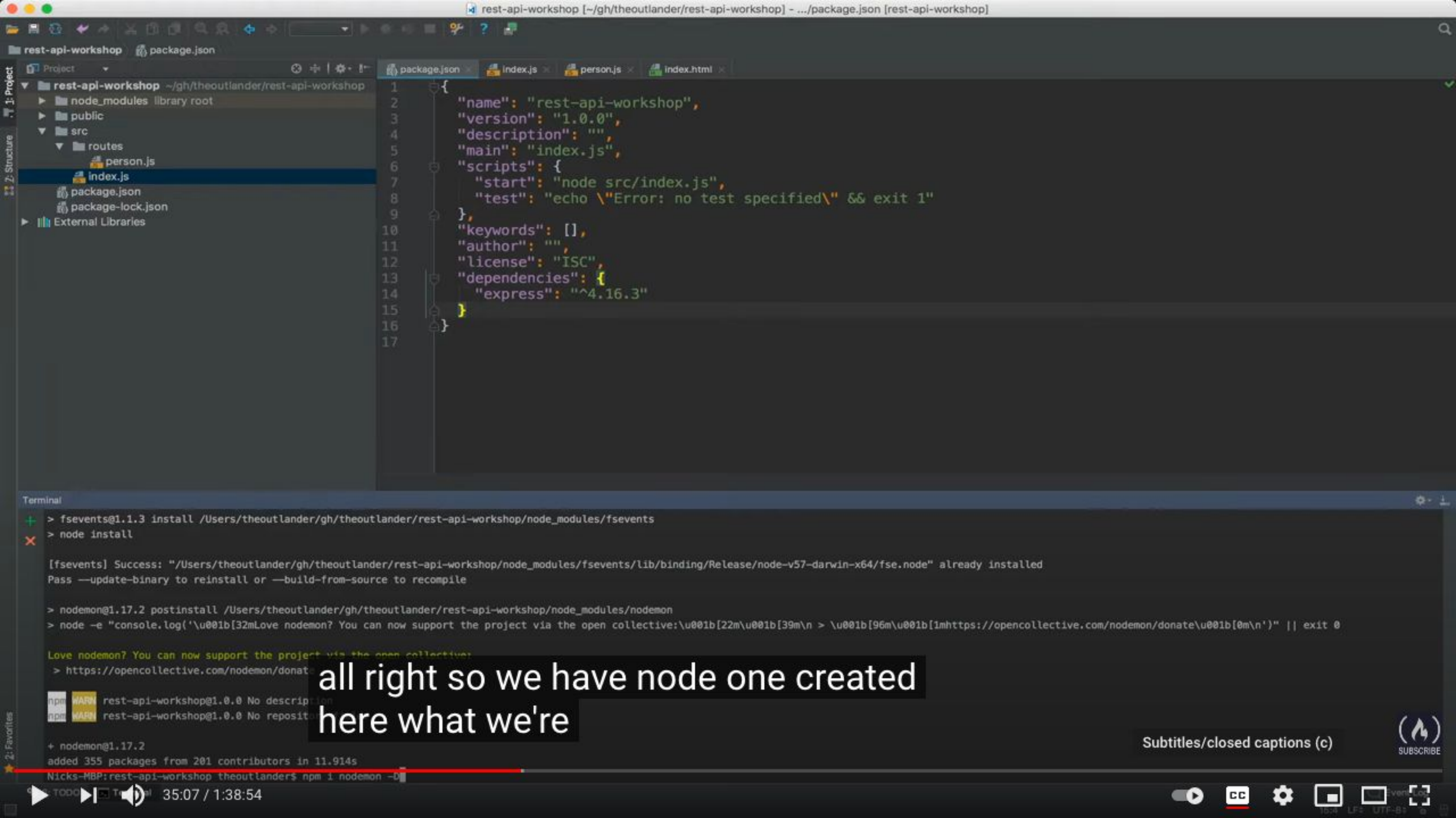

Insomnia – GET Person
Insomnia
GET
https://api.myproduct.com/v1/users
Send
No Environment
Cookies
Filter
REST API Workshop
GET Person
Projects
GET
POST
PUT
PATCH
DELETE
OPTIONS
HEAD
Custom Method
Header
Docs
Select a body type from above
Send Request ⌘Enter
Focus Url Bar ⌘L
Manage Cookies ⌘K
Edit Environments ⌘E
operations map to for requests you have
to create which happens on a post the
Subtitles/closed captions (c)
SUBSCRIBE
30:07 / 1:38:54

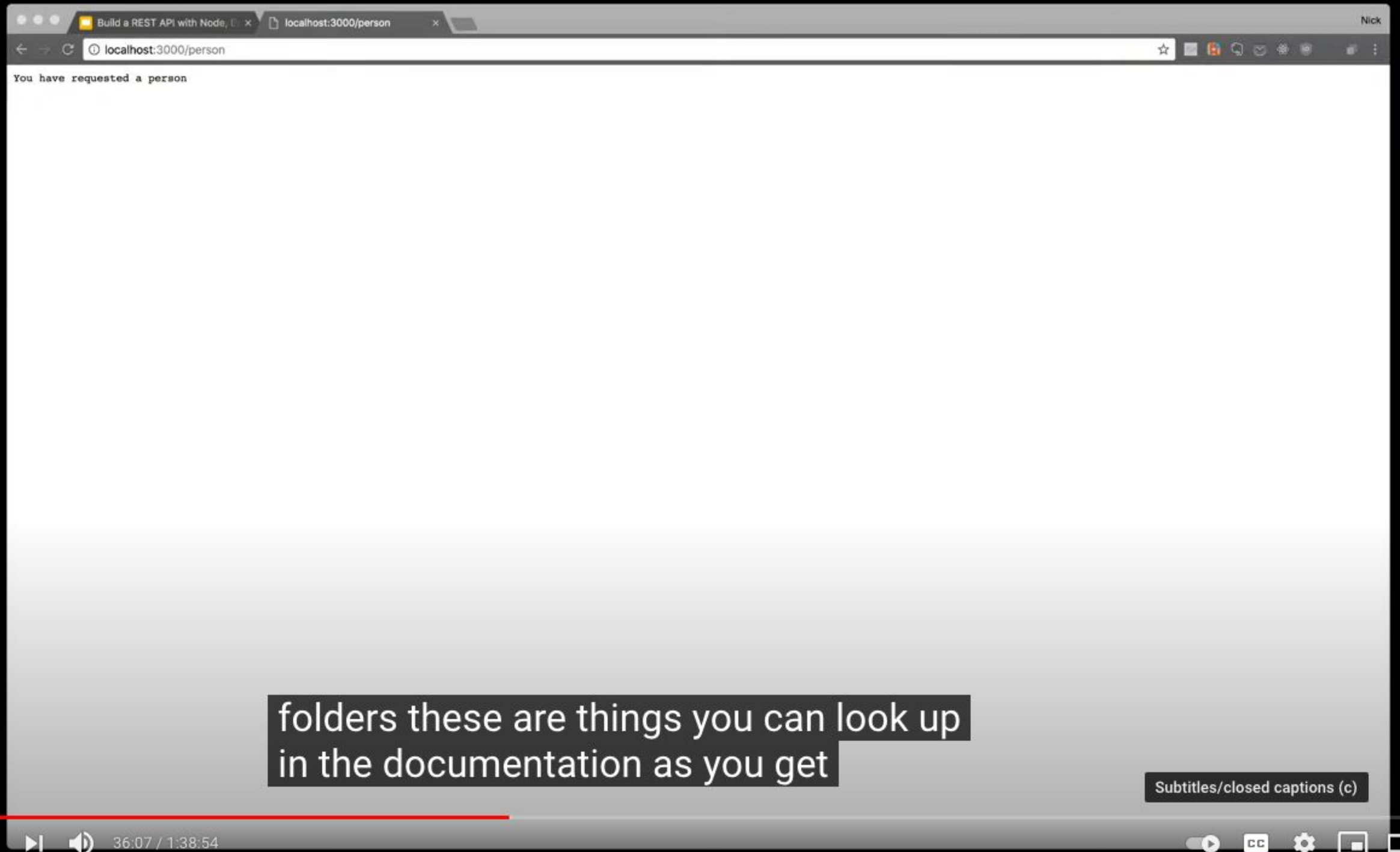

Insomnia – GET Person
Insomnia
GET
https://api.myproduct.com/v1/users
Send
No Environment
Cookies
Filter
REST API Workshop
GET Person
Projects
GET
POST
PUT
PATCH
DELETE
OPTIONS
HEAD
Custom Method
Header
Docs
Select a body type from above
Send Request ⌘Enter
Focus Url Bar ⌘L
Manage Cookies ⌘K
Edit Environments ⌘E
than postmen so I started out using postman
Subtitles/closed captions (c)
SUBSCRIBE
31:07 / 1:38:54

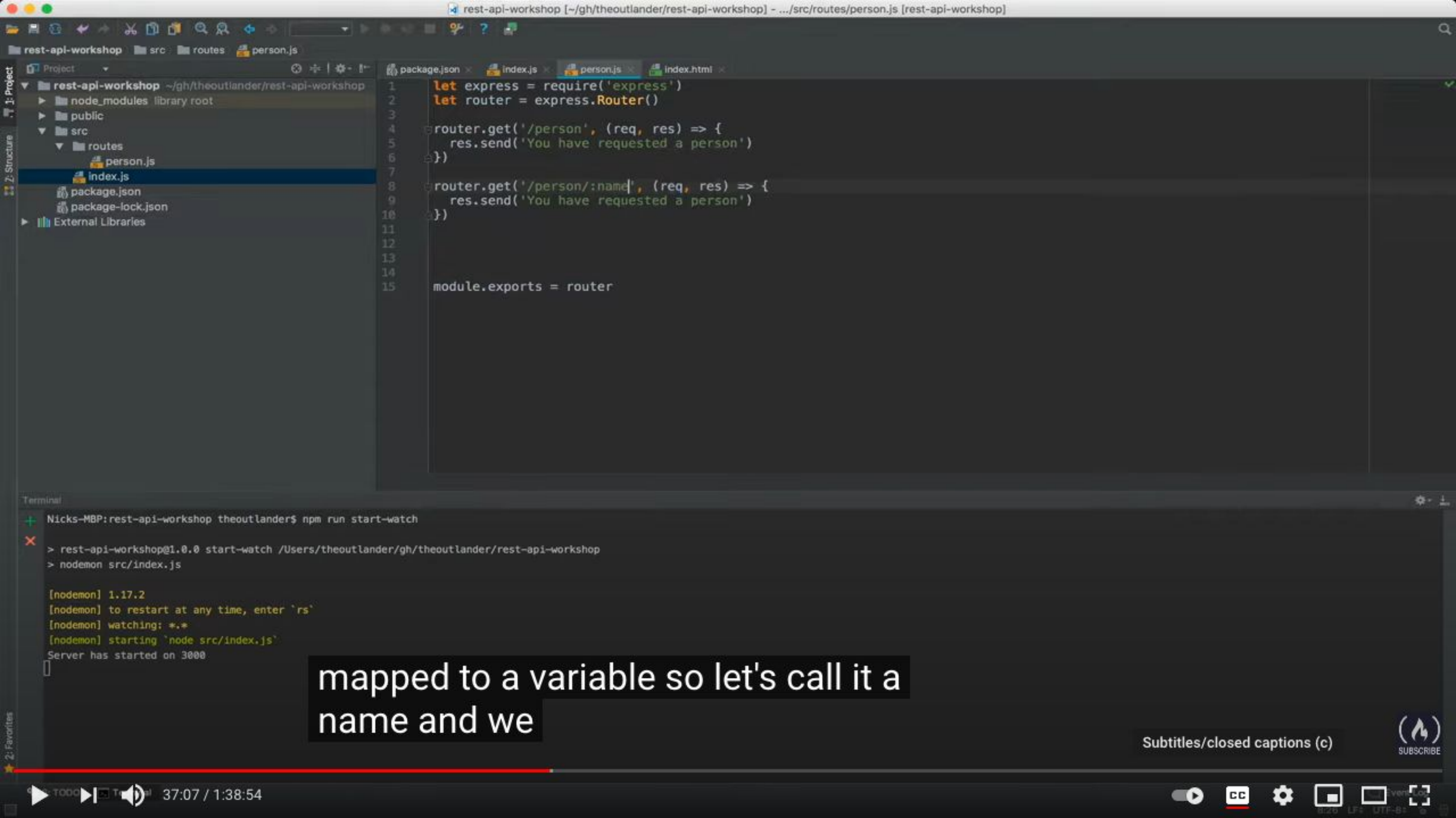

Insomnia – GET Person
Insomnia
GET http://localhost:3000/person
Send
200 OK
TIME 5.44 ms
SIZE 27 B
No Environment
Cookies
Filter
REST API Workshop
GET Person
Projects
Body
Auth
Query
Header
Docs
Preview
Header 6
Cookie
Timeline
You have requested a person
Select a body type from above
routes that we've created and test them
out over here one of the
Subtitles/closed captions (c)
SUBSCRIBE
32:07 / 1:38:54

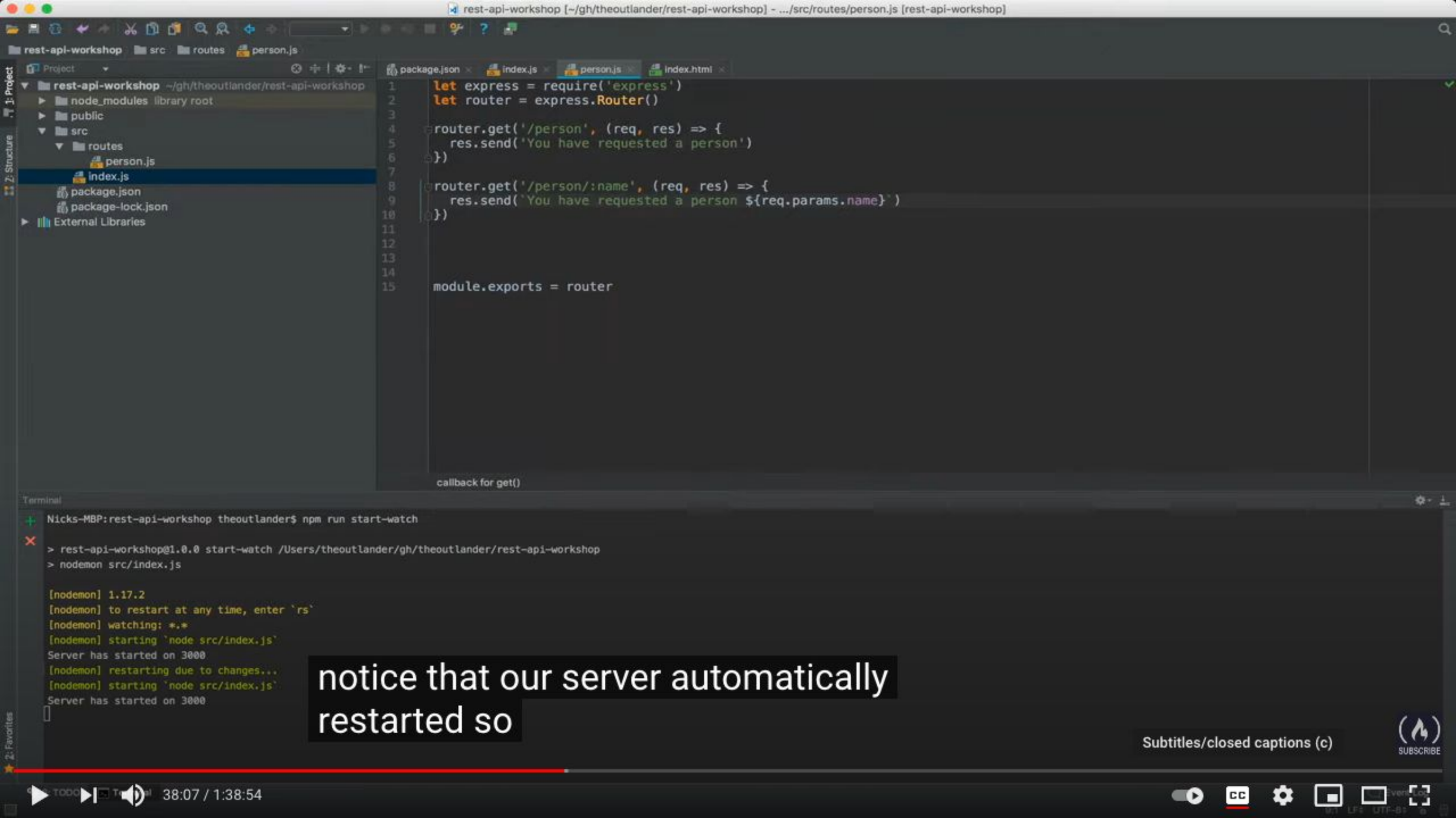

Insomnia – GET Person
Generate Client Code
Node.js
Request
Copy to Clipboard
var request = require("request");
var jar = request.jar();
jar.setCookie(request.cookie("connect.sid=s%253A8IqZ33D3GSmRRHoqqAX4t_7sVlZNBAfe.6Ys4HkUv0h91dGSvXVoSzRSv7MzXcgFZC4zU84jbbBY"), "http://localhost:3000/person");
var options = { method: 'GET',
url: 'http://localhost:3000/person',
jar: 'JAR' };
request(options, function (error, response, body) {
if (error) throw new Error(error);
console.log(body);
});
* Code snippets generated by httpsnippet
Done
choose request because it's the shortest
piece of code here and it's essentially
Subtitles/closed captions (c)
SUBSCRIBE
33:07 / 1:38:54

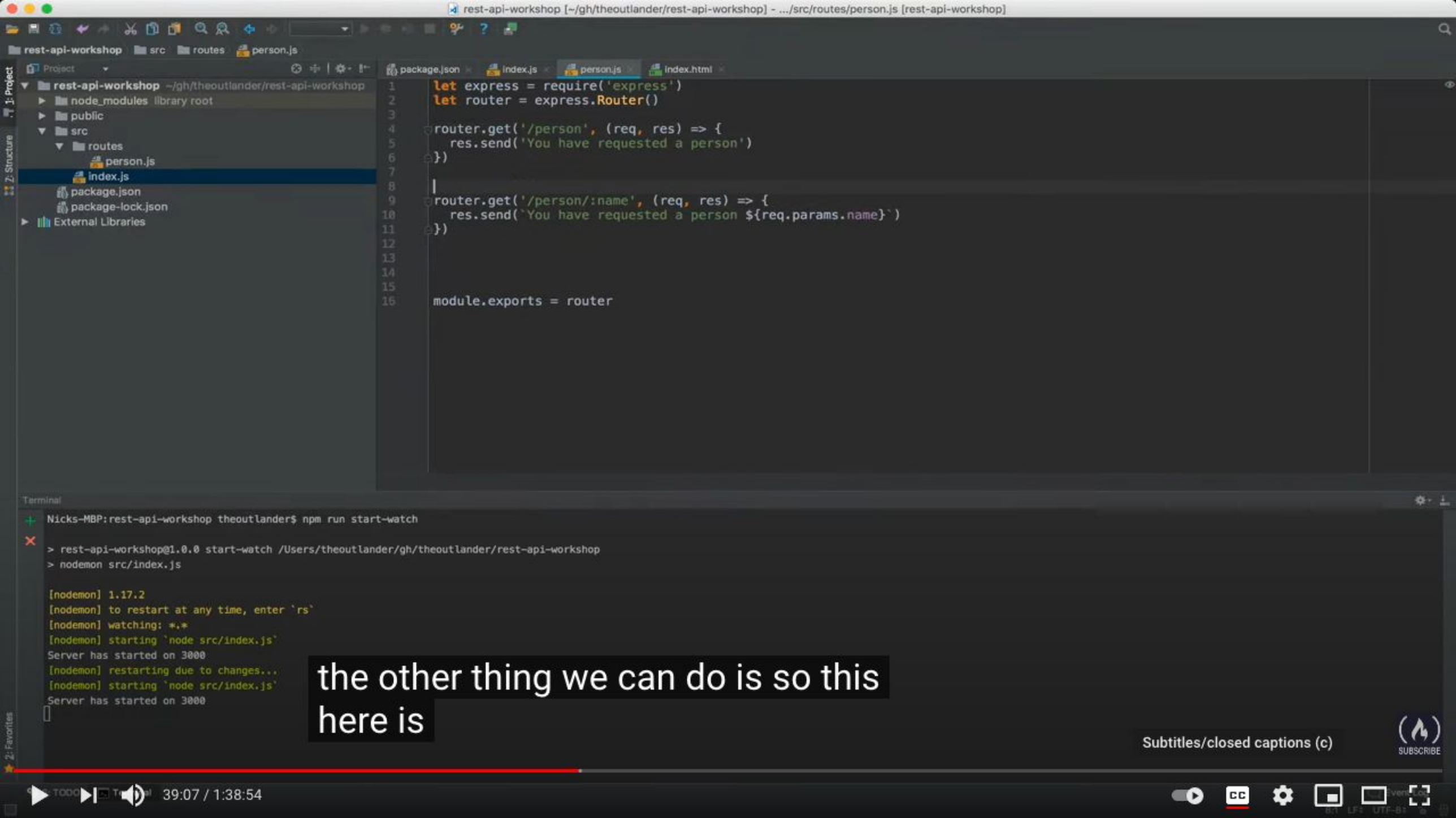

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../package.json [rest-api-workshop]
rest-api-workshop
package.json
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
src
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
person.js
index.html
{
"name": "rest-api-workshop",
"version": "1.0.0",
"description": "",
"main": "index.js",
"scripts": {
"start": "node src/index.js",
"test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC",
"dependencies": {
"express": "^4.16.3"
}
}
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
^CNicks-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
^CNicks-MBP:rest-api-workshop theoutlander$ npm i nodemon
however in this case because we're using
the latest node version NPM
Subtitles/closed captions (c)
SUBSCRIBE
34:07 / 1:38:54

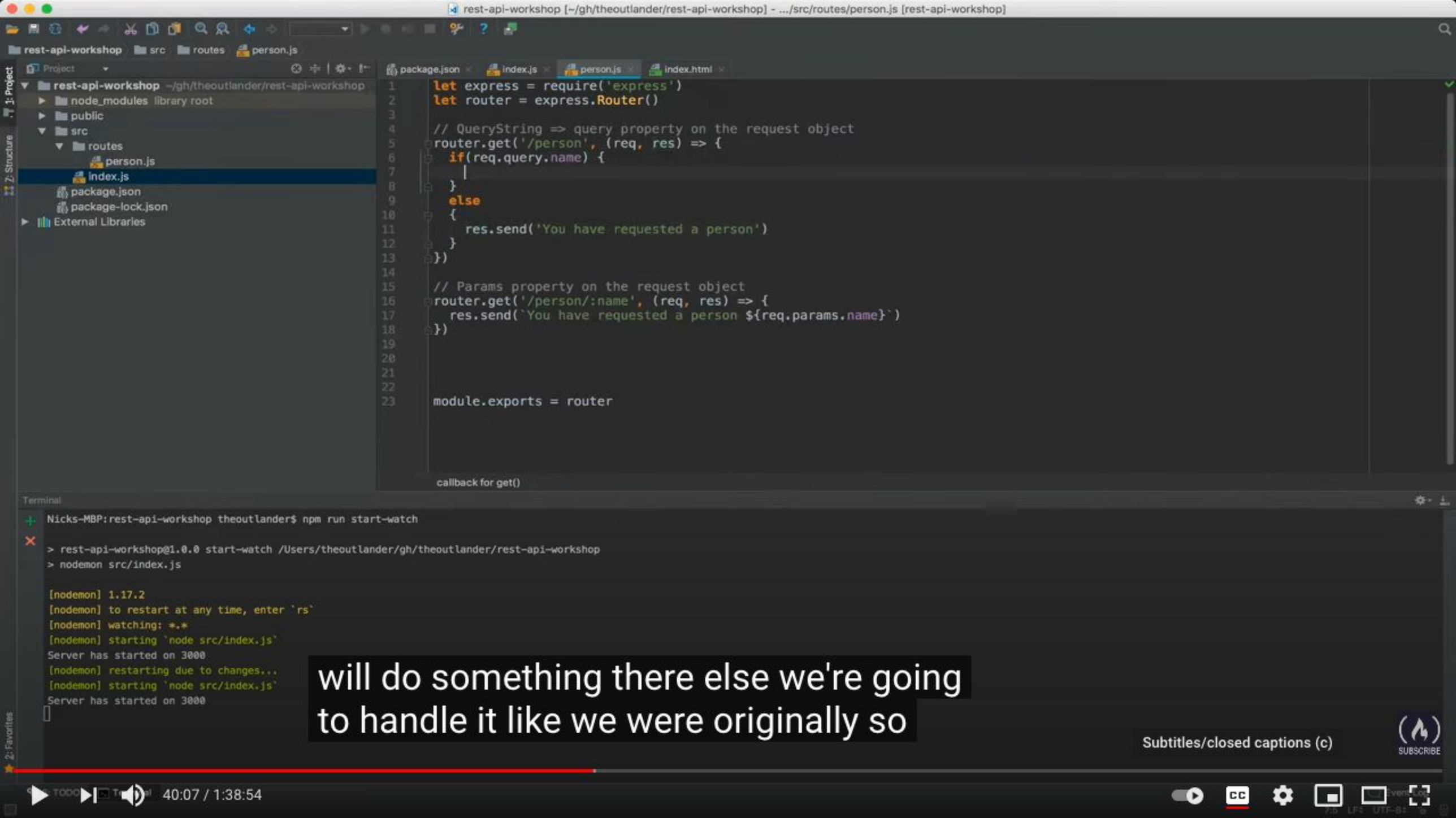

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../package.json [rest-api-workshop]
rest-api-workshop
package.json
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
src
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
person.js
index.html
{
"name": "rest-api-workshop",
"version": "1.0.0",
"description": "",
"main": "index.js",
"scripts": {
"start": "node src/index.js",
"test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC",
"dependencies": {
"express": "^4.16.3"
}
}
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/donat
npm WARN rest-api-workshop@1.0.0 No descrip
npm WARN rest-api-workshop@1.0.0 No reposit
+ nodemon@1.17.2
added 355 packages from 201 contributors in 11.914s
Nicks-MBP:rest-api-workshop theoutlander$ npm i nodemon -D
all right so we have node one created
here what we're
Subtitles/closed captions (c)
SUBSCRIBE
35:07 / 1:38:54

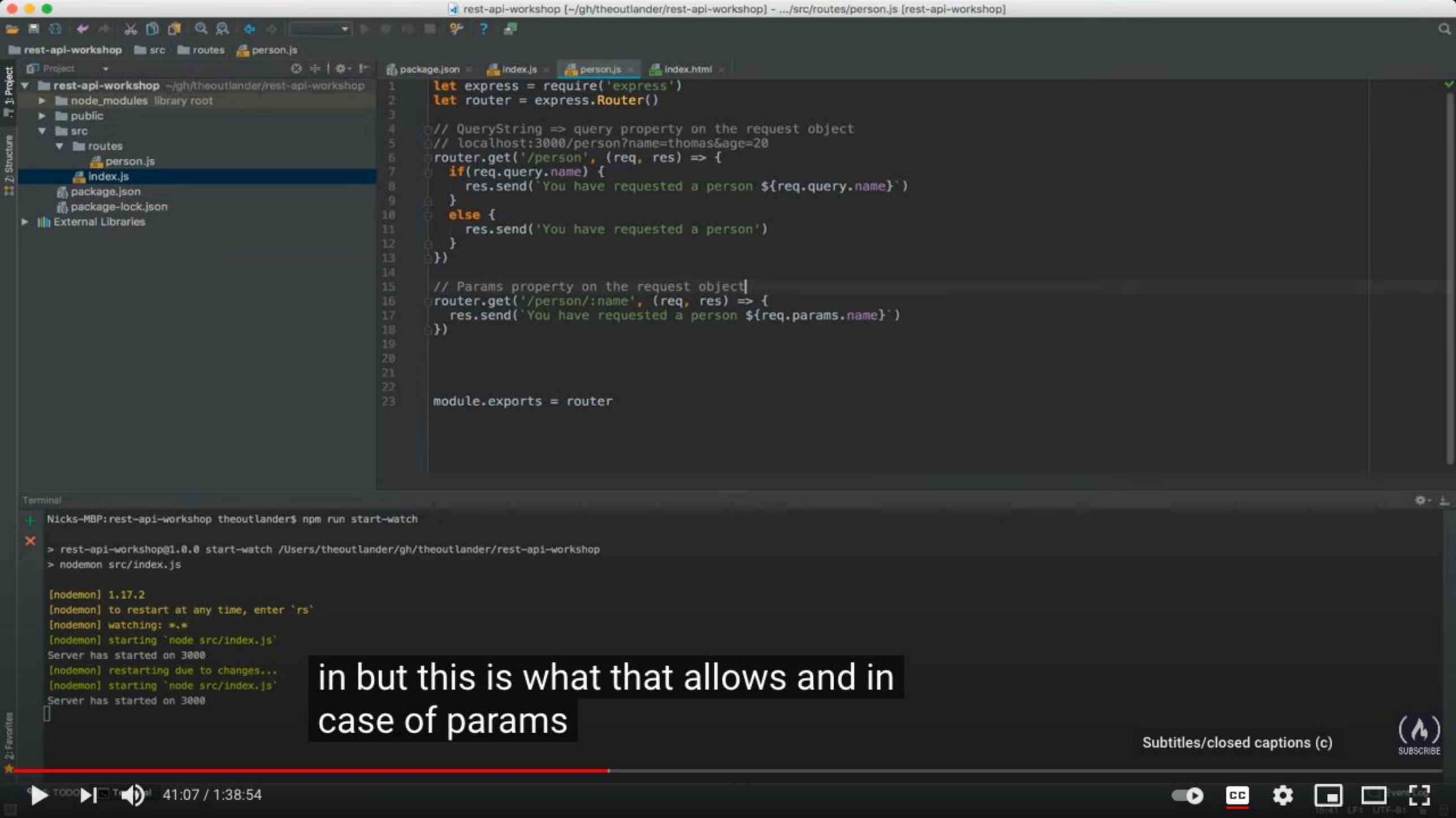

Build a REST API with Node,
localhost:3000/person
Nick
localhost:3000/person
You have requested a person
folders these are things you can look up
in the documentation as you get
Subtitles/closed captions (c)
SUBSCRIBE
36:07 / 1:38:54

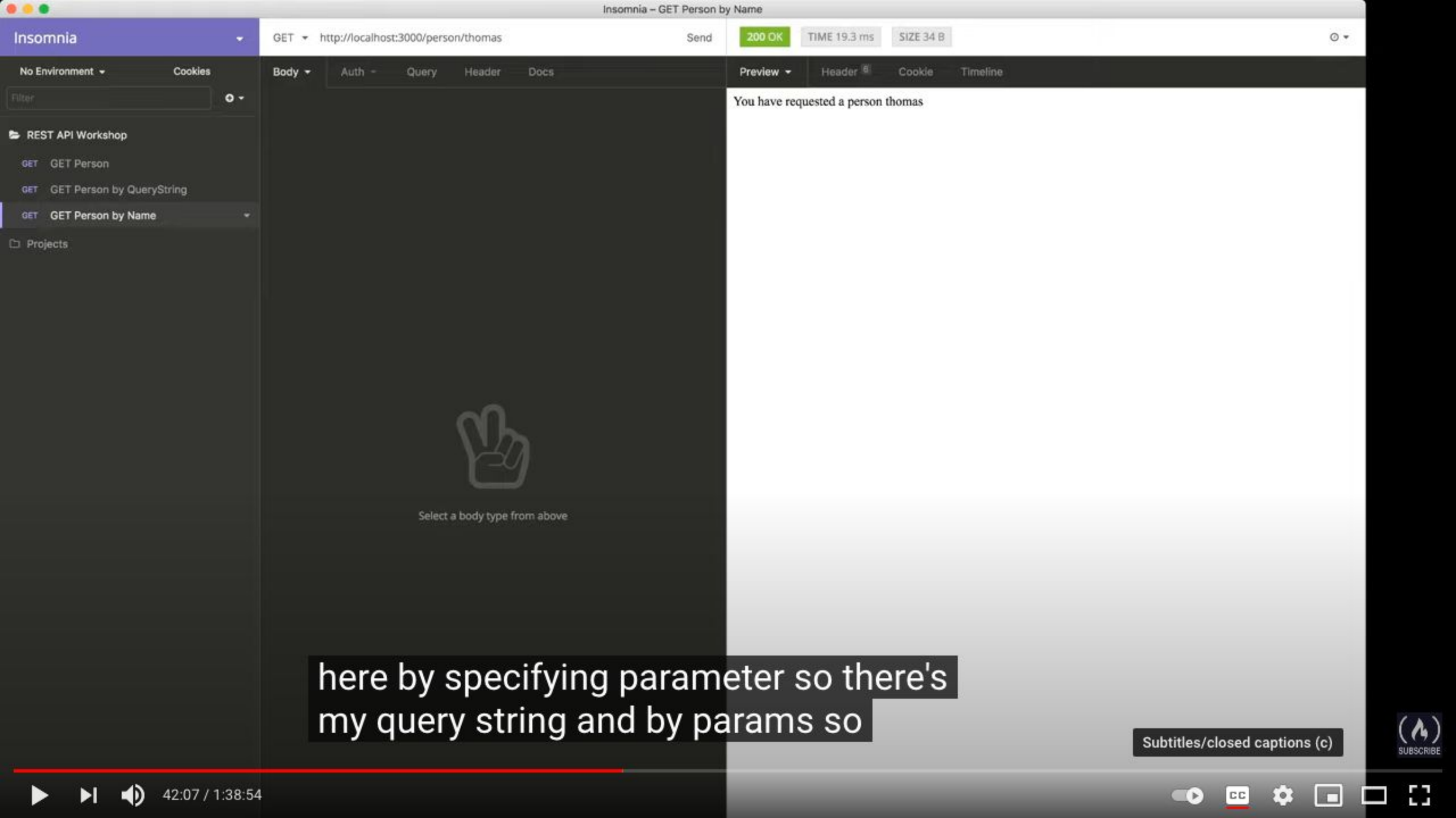

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/person.js [rest-api-workshop]
rest-api-workshop src routes person.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
src
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json index.js person.js index.html
let express = require('express')
let router = express.Router()
router.get('/person', (req, res) => {
res.send('You have requested a person')
})
router.get('/person/:name', (req, res) => {
res.send('You have requested a person')
})
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
mapped to a variable so let's call it a
name and we
Subtitles/closed captions (c)
SUBSCRIBE
37:07 / 1:38:54


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
package.json
index.js
person.js
index.html
let express = require('express')
let router = express.Router()
router.get('/person', (req, res) => {
res.send('You have requested a person')
})
router.get('/person/:name', (req, res) => {
res.send(`You have requested a person ${req.params.name}`)
})
module.exports = router
callback for get()
Terminal
Nicks-MBP:rest-api-workshop theoutlander$ 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
notice that our server automatically
restarted so
Subtitles/closed captions (c)
SUBSCRIBE
38:07 / 1:38:54


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
package.json
index.js
person.js
index.html
let express = require('express')
let router = express.Router()
router.get('/person', (req, res) => {
res.send('You have requested a person')
})
router.get('/person/:name', (req, res) => {
res.send(`You have requested a person ${req.params.name}`)
})
module.exports = router
Terminal
Nicks-MBP:rest-api-workshop theoutlander$ 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
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
Server has started on 3000
the other thing we can do is so this here is
Subtitles/closed captions (c)
SUBSCRIBE
39:07 / 1:38:54


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
package.json
index.js
person.js
index.html
let express = require('express')
let router = express.Router()
// QueryString => query property on the request object
router.get('/person', (req, res) => {
if(req.query.name) {
}
else
{
res.send('You have requested a person')
}
})
// Params property on the request object
router.get('/person/:name', (req, res) => {
res.send(`You have requested a person ${req.params.name}`)
})
module.exports = router
callback for get()
Terminal
Nicks-MBP:rest-api-workshop theoutlander$ 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
will do something there else we're going
to handle it like we were originally so
Subtitles/closed captions (c)
SUBSCRIBE
40:07 / 1:38:54


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
package.json
index.js
person.js
index.html
let express = require('express')
let router = express.Router()
// QueryString => query property on the request object
// localhost:3000/person?name=thomas&age=20
router.get('/person', (req, res) => {
if(req.query.name) {
res.send(`You have requested a person ${req.query.name}`)
}
else {
res.send('You have requested a person')
}
})
// Params property on the request object
router.get('/person/:name', (req, res) => {
res.send(`You have requested a person ${req.params.name}`)
})
module.exports = router
Terminal
Nicks-MBP:rest-api-workshop theoutlander$ 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
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
Server has started on 3000
in but this is what that allows and in case of params
Subtitles/closed captions (c)
SUBSCRIBE
41:07 / 1:38:54


Insomnia – GET Person by Name
Insomnia
GET http://localhost:3000/person/thomas
Send
200 OK
TIME 19.3 ms
SIZE 34 B
No Environment
Cookies
Filter
REST API Workshop
GET GET Person
GET GET Person by QueryString
GET GET Person by Name
Projects
Body
Auth
Query
Header
Docs
Preview
Header 6
Cookie
Timeline
You have requested a person thomas
Select a body type from above
here by specifying parameter so there's
my query string and by params so
Subtitles/closed captions (c)
SUBSCRIBE
42:07 / 1:38:54

# Middleware

Functions that execute serially

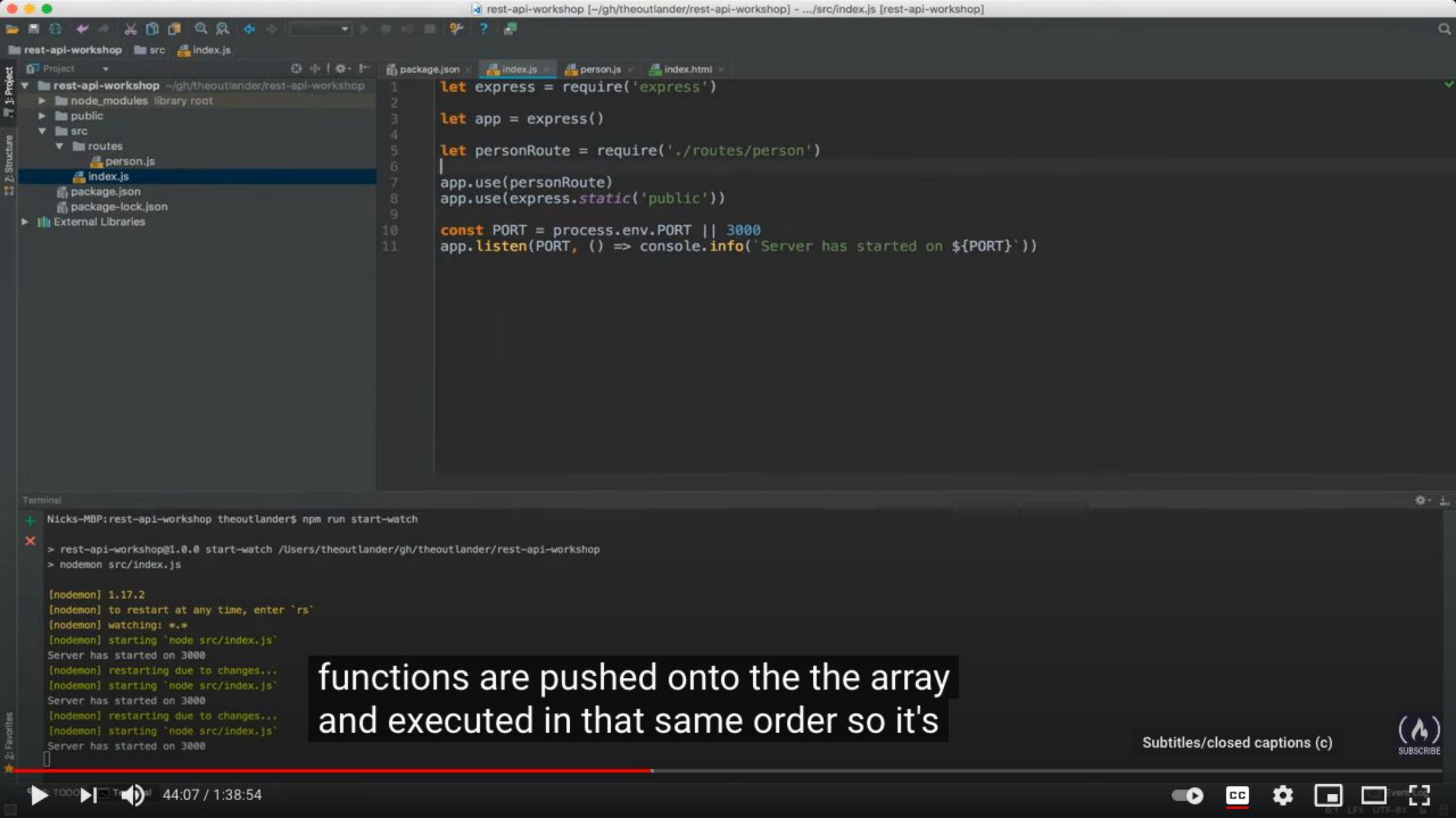
- Access to request/response objects and the next middleware function in the pipeline
- Execute any code
- Make changes to the request and the response objects
- End the request-response cycle
- Call the next middleware function in the stack

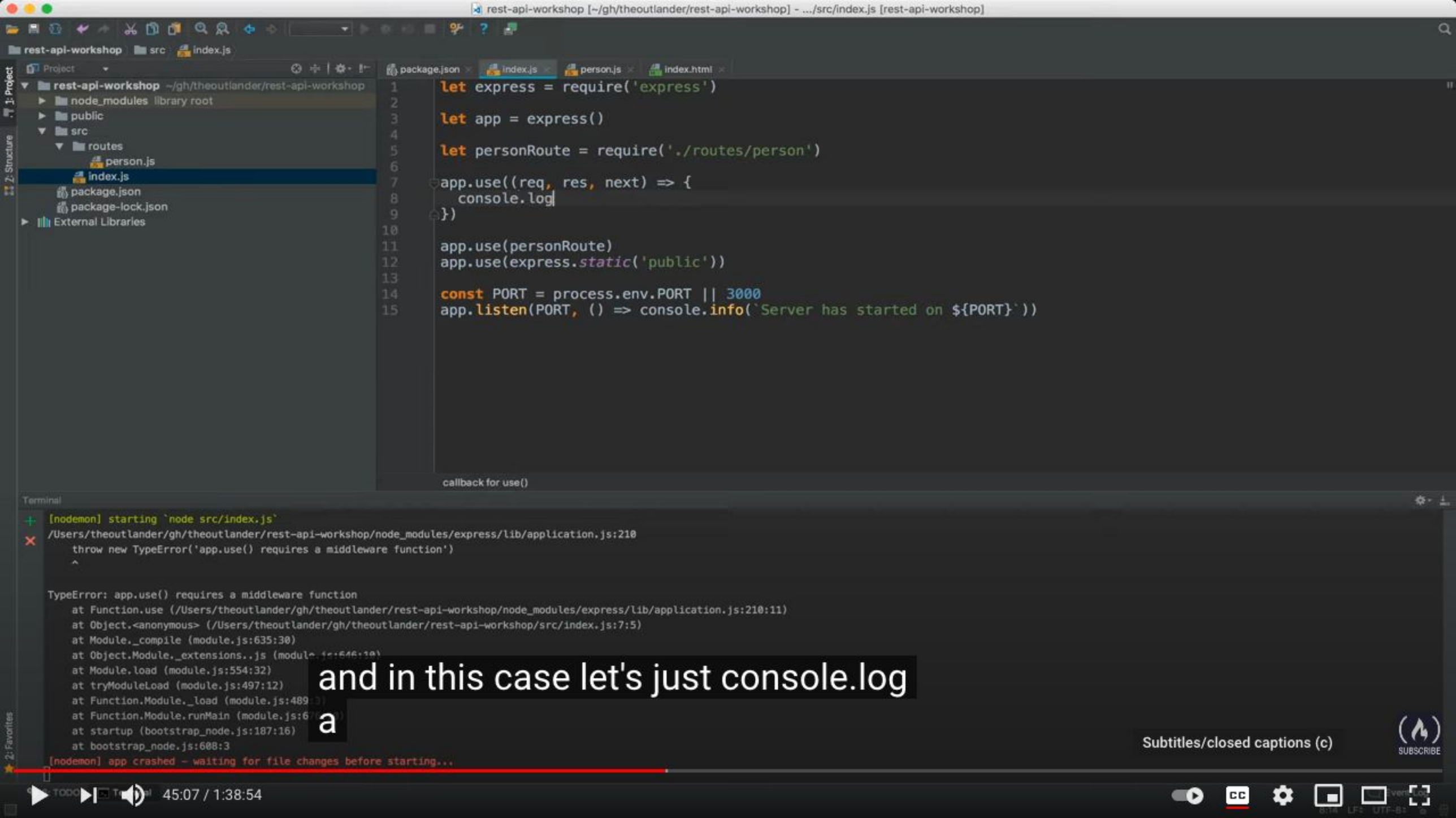
done calls into the next function it has the ability to also modify

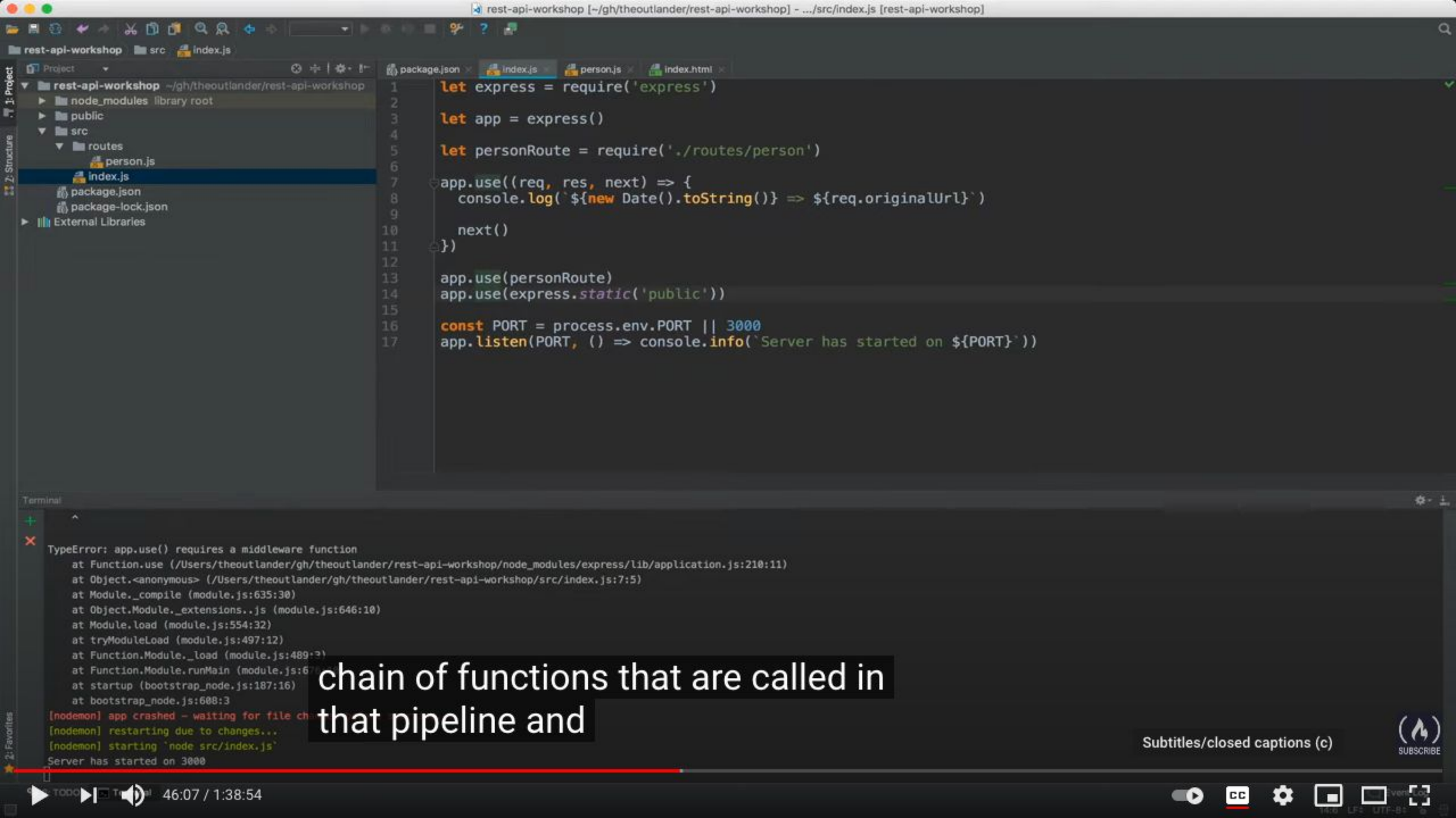
Subtitles/closed captions (c)

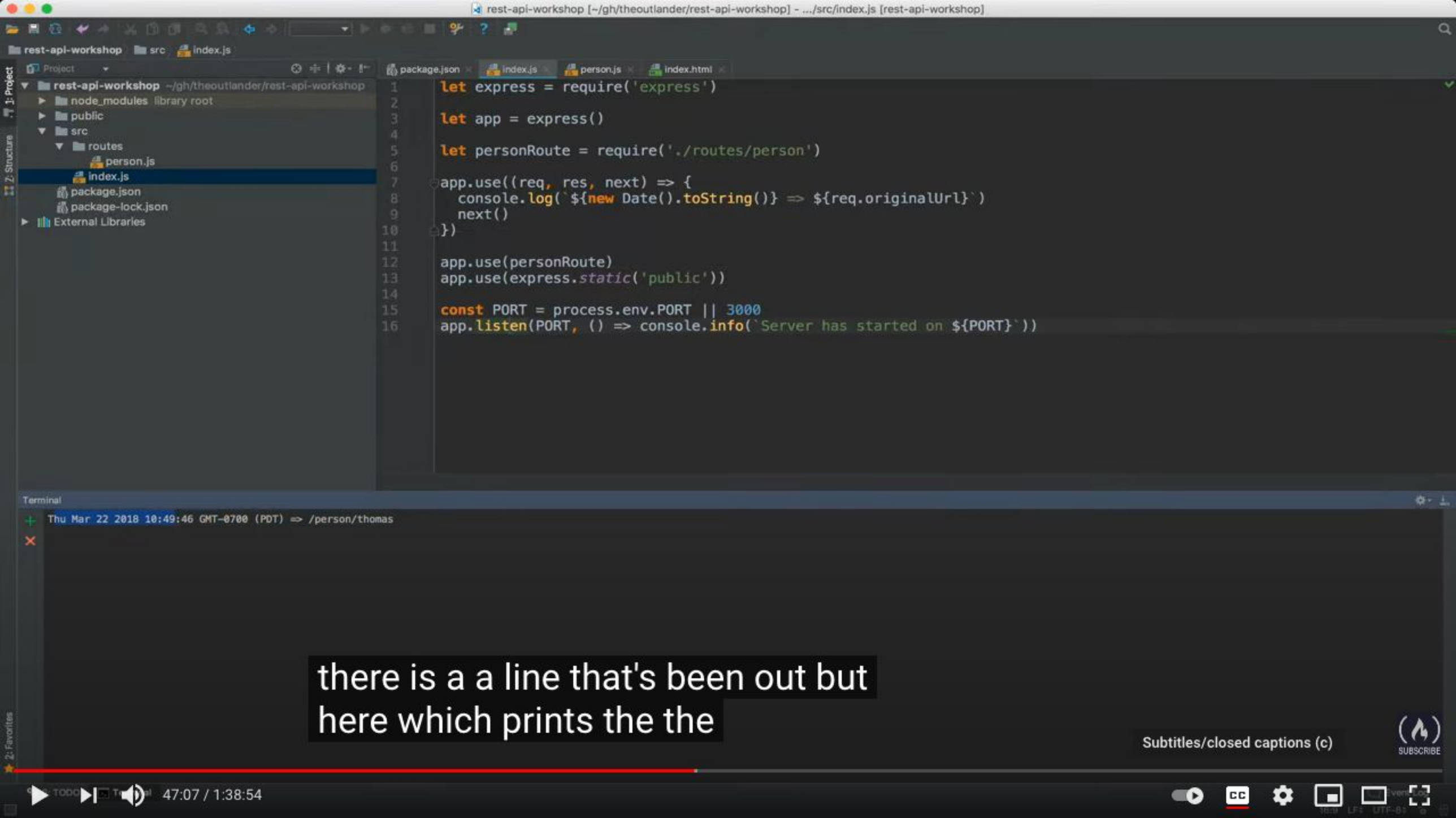
SUBSCRIBE

43:07 / 1:38:54


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(personRoute)
8 app.use(express.static('public'))
9
10 const PORT = process.env.PORT || 3000
11 app.listen(PORT, () => console.info(`Server has started on ${PORT}`))
Terminal
Nicks-MBP:rest-api-workshop theoutlander$ 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
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
Server has started on 3000
functions are pushed onto the the array
and executed in that same order so it's
Subtitles/closed captions (c)
SUBSCRIBE
44:07 / 1:38:54


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
let express = require('express')
let app = express()
let personRoute = require('./routes/person')
app.use((req, res, next) => {
console.log
})
app.use(personRoute)
app.use(express.static('public'))
const PORT = process.env.PORT || 3000
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 Module.load (module.js:554:32)
at tryModuleLoad (module.js:497:12)
at startup (bootstrap_node.js:187:16)
at bootstrap_node.js:608:3
[nodemon] app crashed - waiting for file changes before starting...
and in this case let's just console.log
a
Subtitles/closed captions (c)
SUBSCRIBE
45:07 / 1:38:54


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
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'))
const PORT = process.env.PORT || 3000
app.listen(PORT, () => console.info(`Server has started on ${PORT}`))
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: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:6
at startup (bootstrap_node.js:187:16)
at bootstrap_node.js:608:3
[nodemon] app crashed - waiting for file ch
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
Server has started on 3000
chain of functions that are called in
that pipeline and
Subtitles/closed captions (c)
SUBSCRIBE
46:07 / 1:38:54


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
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'))
const PORT = process.env.PORT || 3000
app.listen(PORT, () => console.info(`Server has started on ${PORT}`))
Terminal
Thu Mar 22 2018 10:49:46 GMT-0700 (PDT) => /person/thomas
there is a a line that's been out but
here which prints the the
Subtitles/closed captions (c)
SUBSCRIBE
47:07 / 1:38:54

# Error Pages

Custom Handlers

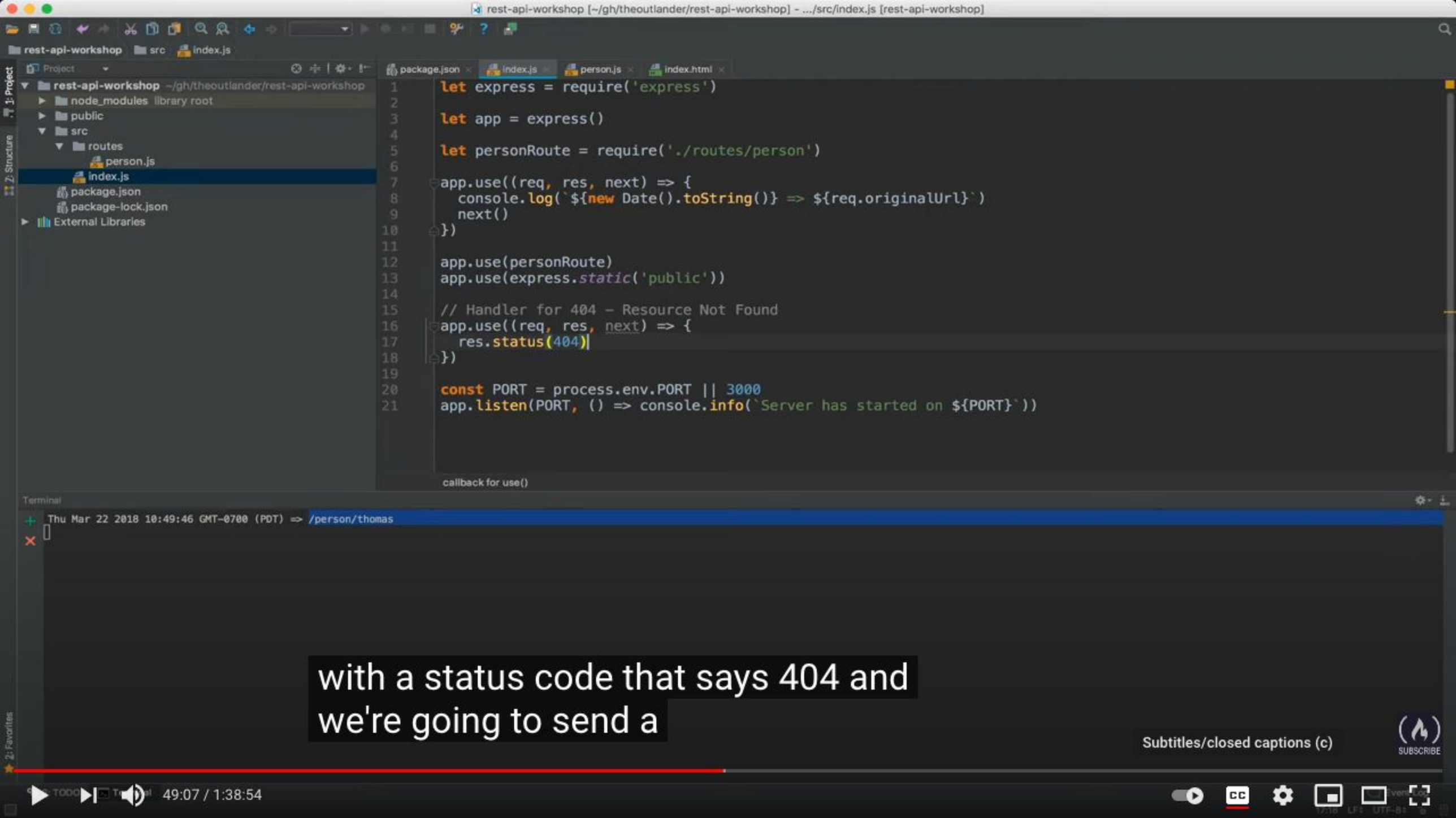
- 404 Not Found Handler
- 500 Internal Server Error

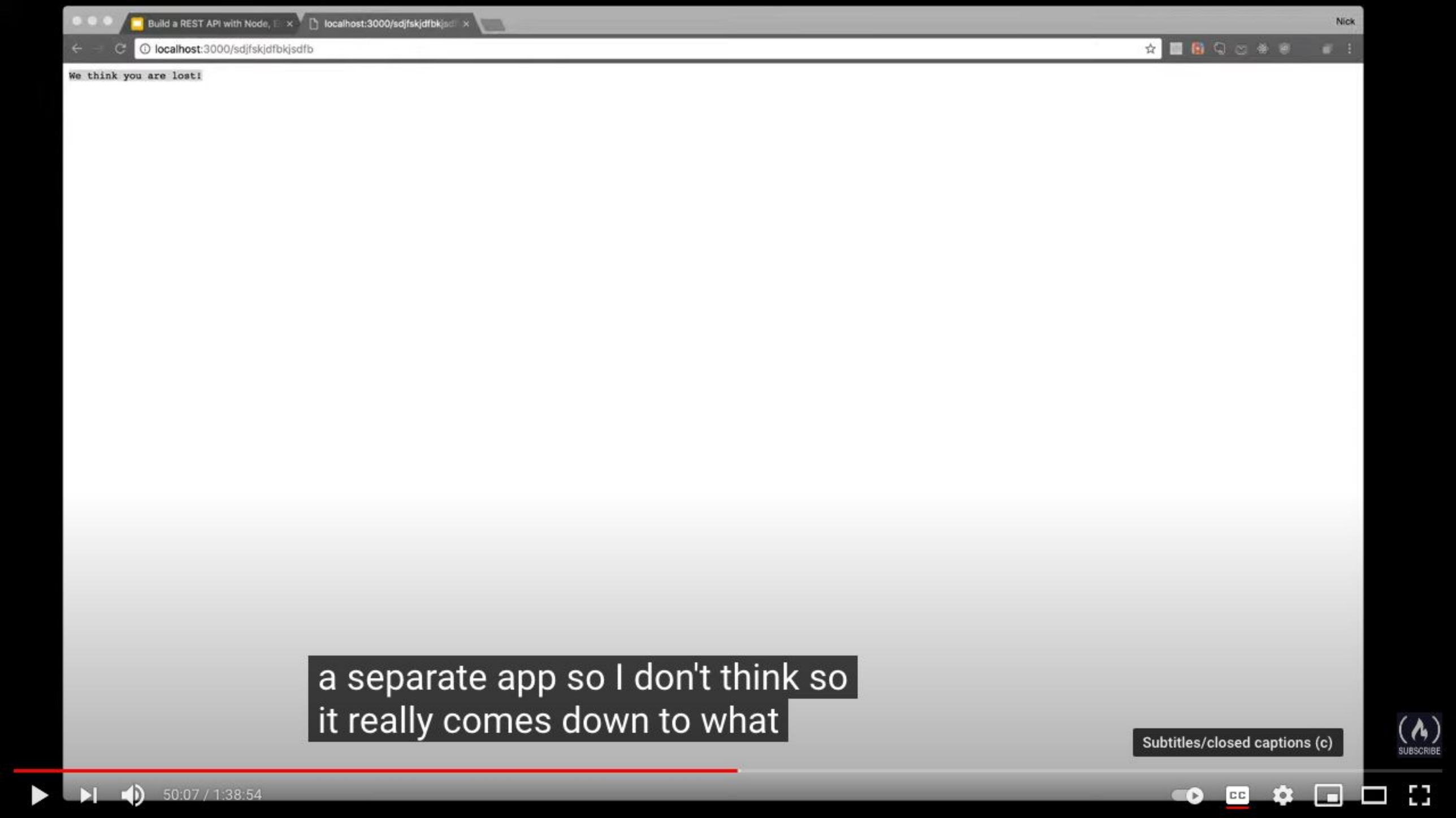
you how to do respond with a specific file

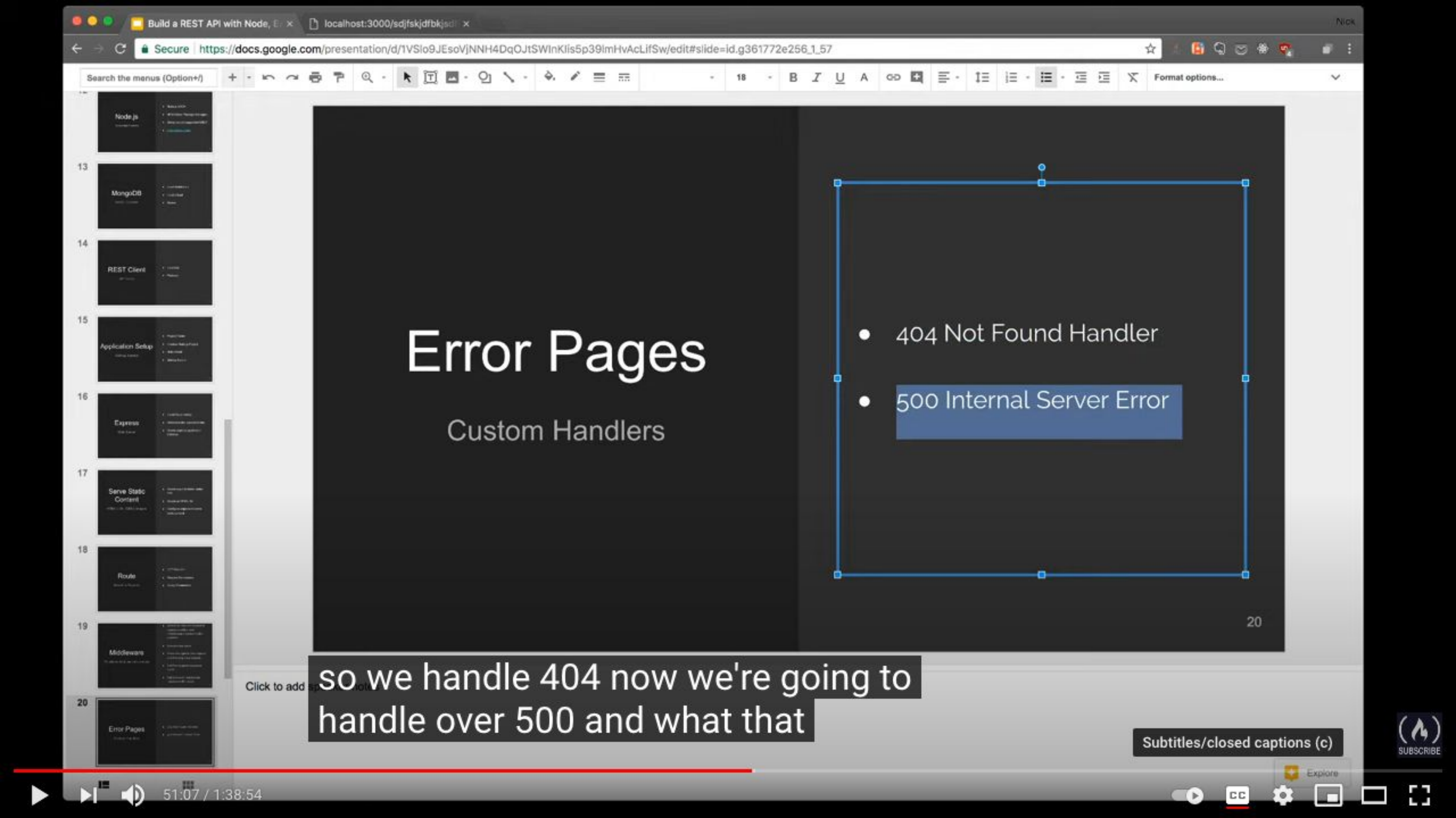
Subtitles/closed captions (c)

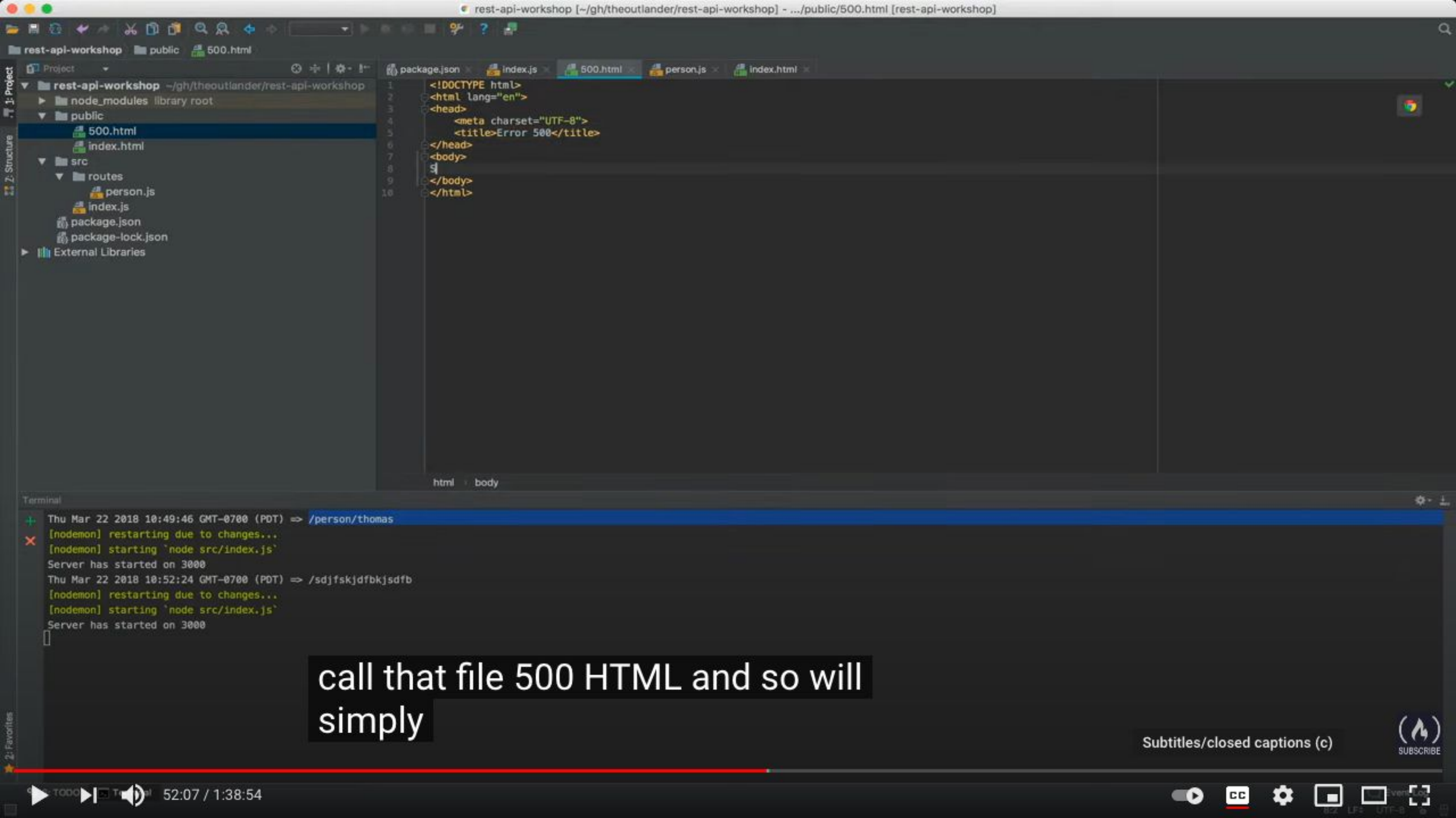
SUBSCRIBE

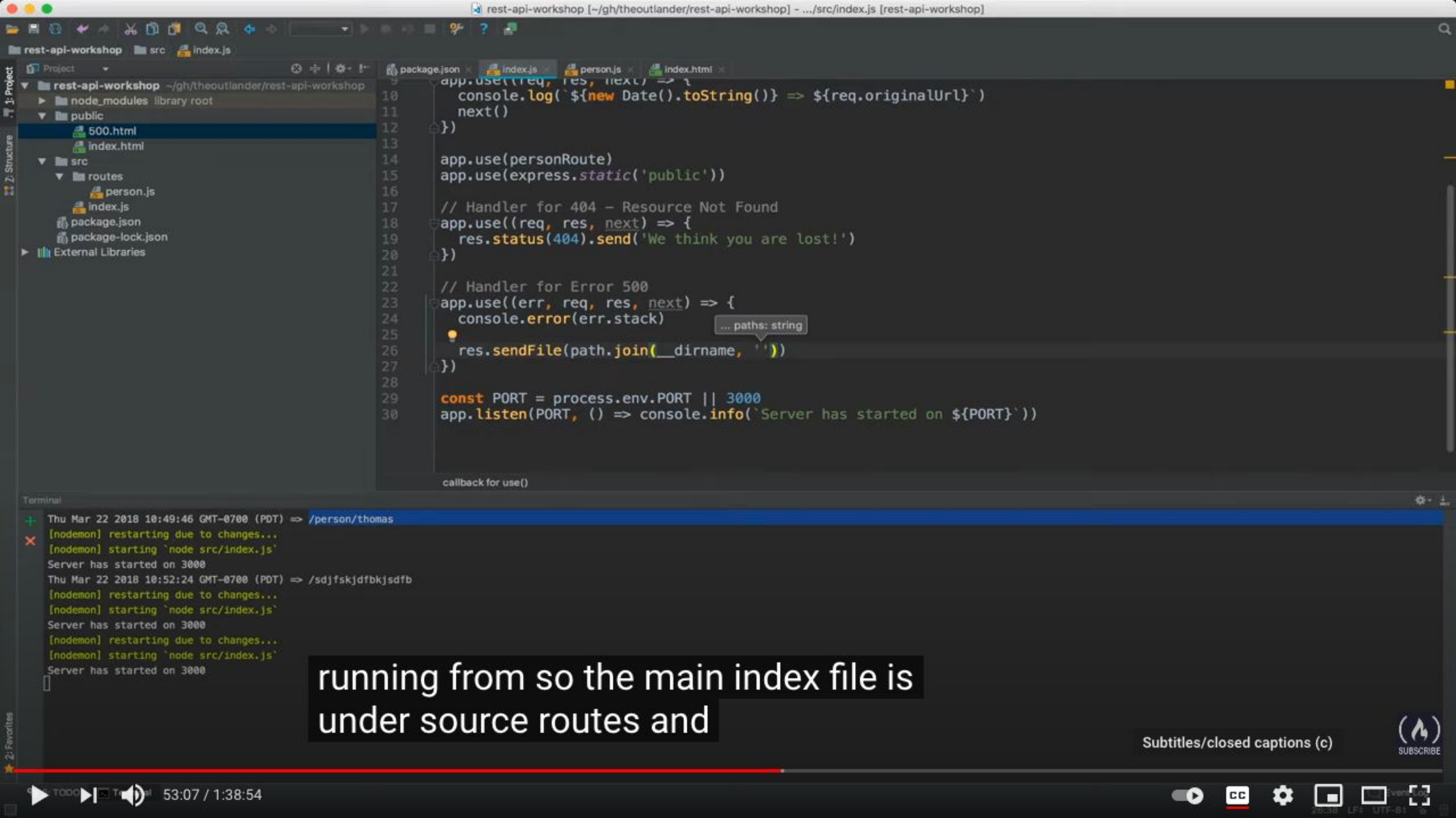
48:07 / 1:38:54

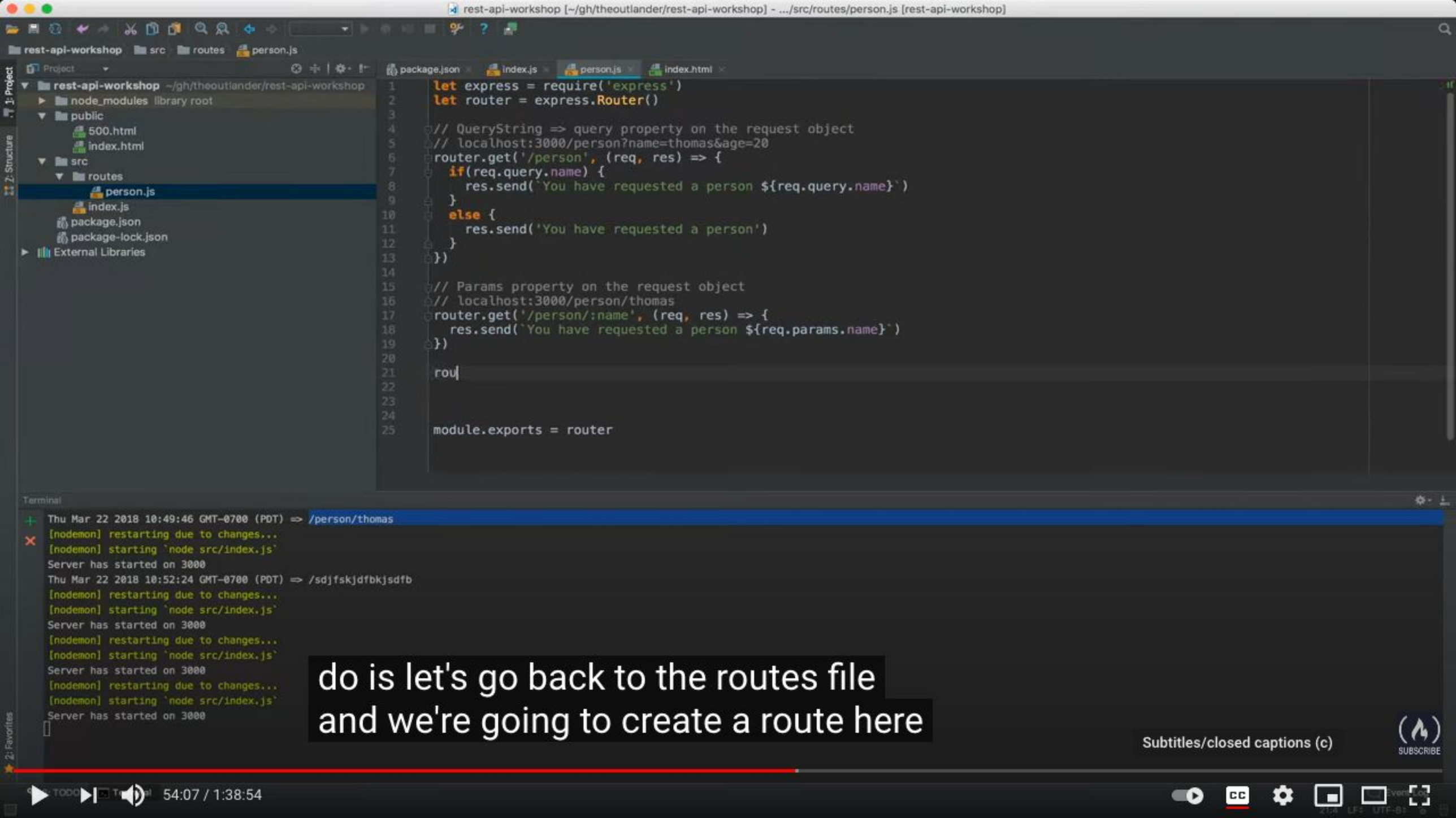

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/index.js [rest-api-workshop]
rest-api-workshop
src
index.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
src
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
person.js
index.html
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
with a status code that says 404 and
we're going to send a
Subtitles/closed captions (c)
SUBSCRIBE
49:07 / 1:38:54

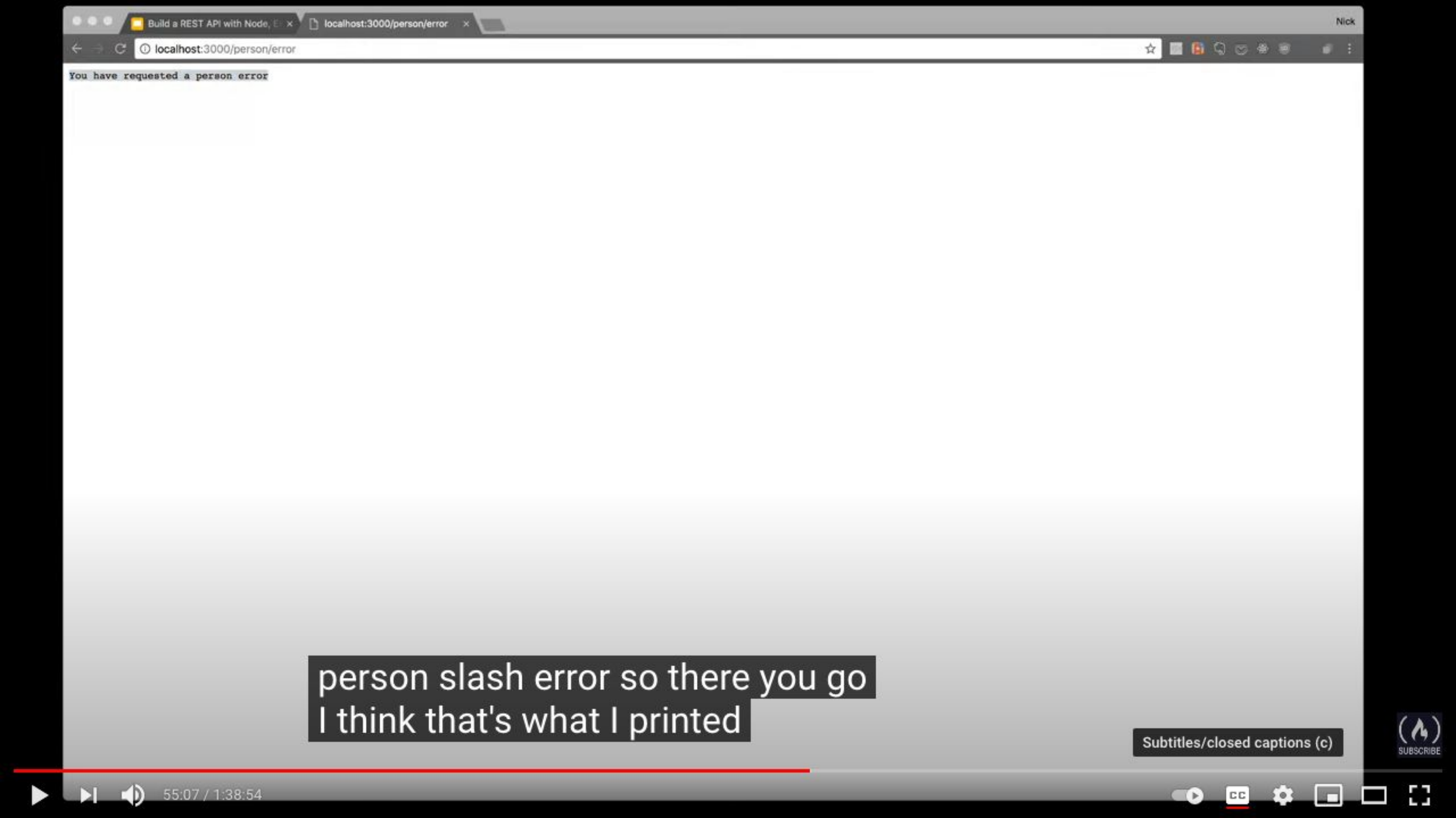

Build a REST API with Node,
localhost:3000/sdjfskjdfbkjsdf
Nick
localhost:3000/sdjfskjdfbkjsdfb
We think you are lost!
a separate app so I don't think so
it really comes down to what
Subtitles/closed captions (c)
SUBSCRIBE
50:07 / 1:38:54

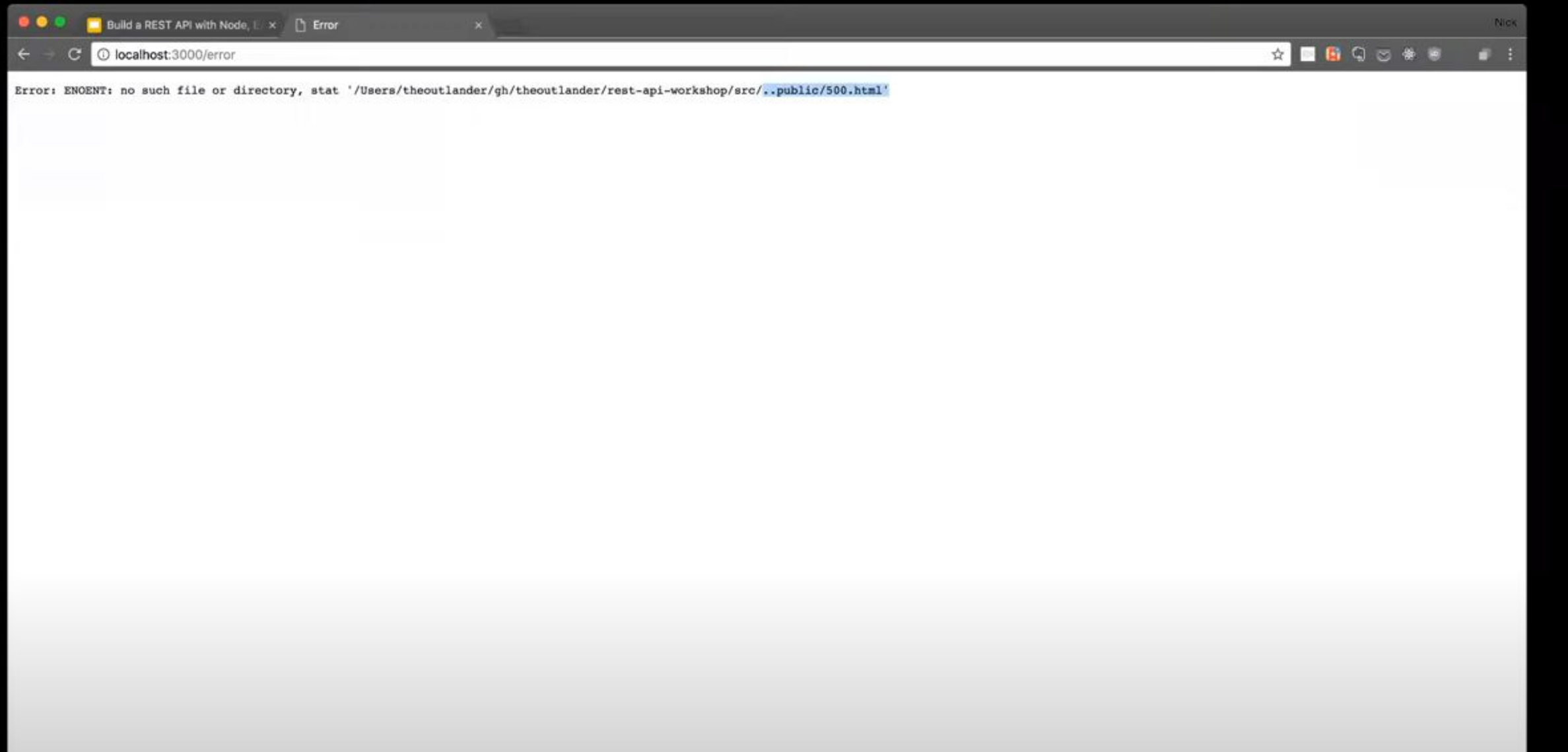

Build a REST API with Node,
localhost:3000/sdjfskjdfbkjsdf
Secure
https://docs.google.com/presentation/d/1VSlo9JEsoVjNNH4DqOJtSWInKlis5p39lmHvAcLifSw/edit#slide=id.g361772e256_1_57
Search the menus (Option+/)
Format options...
Node.js
13 MongoDB
14 REST Client
15 Application Setup
16 Express
17 Serve Static Content
18 Route
19 Middleware
20 Error Pages
Error Pages
Custom Handlers
404 Not Found Handler
500 Internal Server Error
20
Click to add
so we handle 404 now we're going to
handle over 500 and what that
Subtitles/closed captions (c)
SUBSCRIBE
Explore
51:07 / 1:38:54


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
package.json
index.js
500.html
person.js
index.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Error 500</title>
</head>
<body>
s
</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
call that file 500 HTML and so will simply
Subtitles/closed captions (c)
SUBSCRIBE
52:07 / 1:38:54


rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/index.js [rest-api-workshop]
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
package.json
index.js
person.js
index.html
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)
... paths: string
res.sendFile(path.join(__dirname, ''))
})
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
[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
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
Server has started on 3000
running from so the main index file is
under source routes and
Subtitles/closed captions (c)
SUBSCRIBE
53:07 / 1:38:54


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
500.html
index.html
src
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
person.js
index.html
let express = require('express')
let router = express.Router()
// QueryString => query property on the request object
// localhost:3000/person?name=thomas&age=20
router.get('/person', (req, res) => {
if(req.query.name) {
res.send(`You have requested a person ${req.query.name}`)
}
else {
res.send('You have requested a person')
}
})
// Params property on the request object
// localhost:3000/person/thomas
router.get('/person/:name', (req, res) => {
res.send(`You have requested a person ${req.params.name}`)
})
rou
module.exports = router
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
[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
do is let's go back to the routes file
and we're going to create a route here
Subtitles/closed captions (c)
SUBSCRIBE
54:07 / 1:38:54


Build a REST API with Node,
localhost:3000/person/error
Nick
localhost:3000/person/error
You have requested a person error
person slash error so there you go
I think that's what I printed
Subtitles/closed captions (c)
SUBSCRIBE
55:07 / 1:38:54

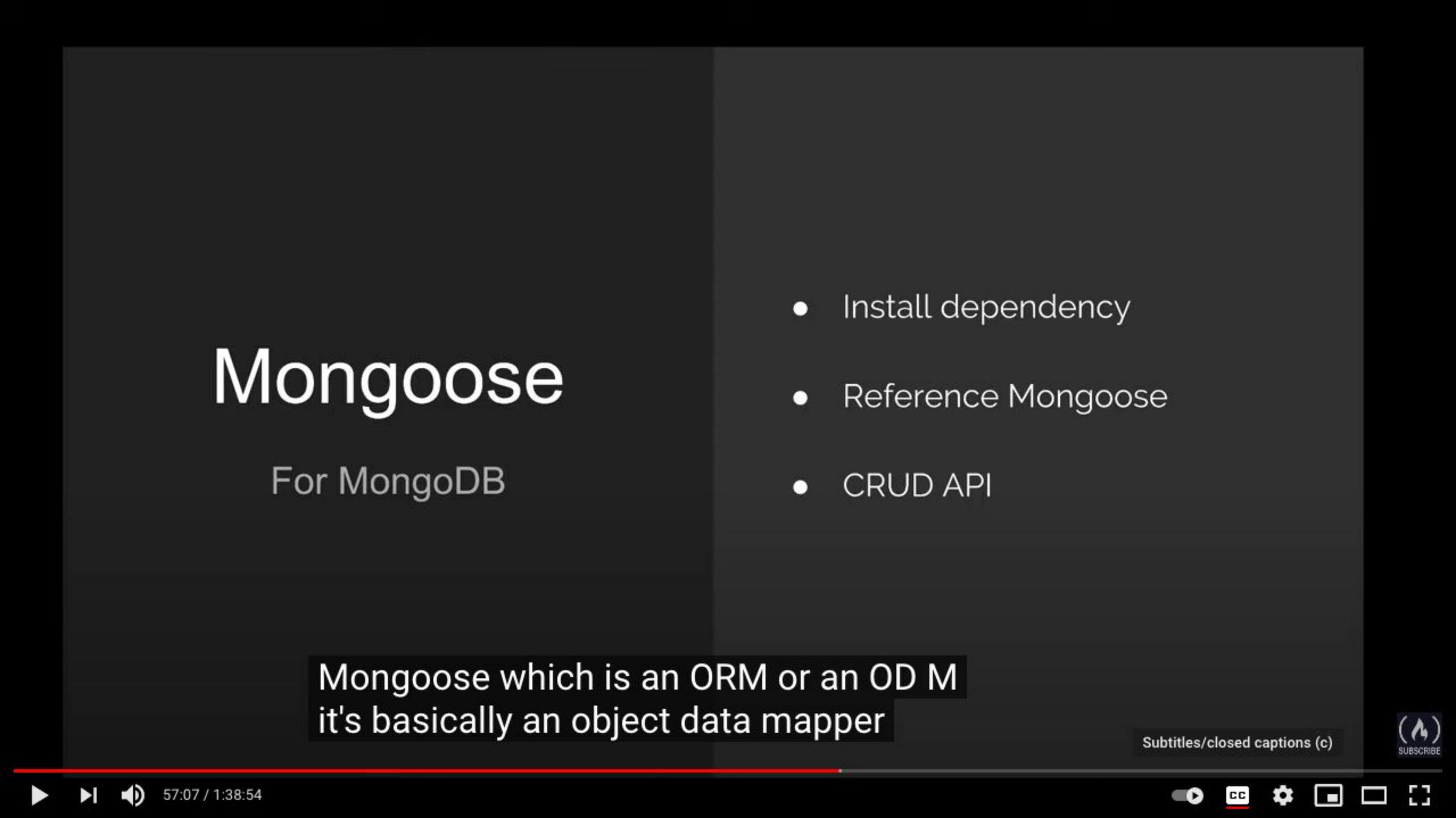


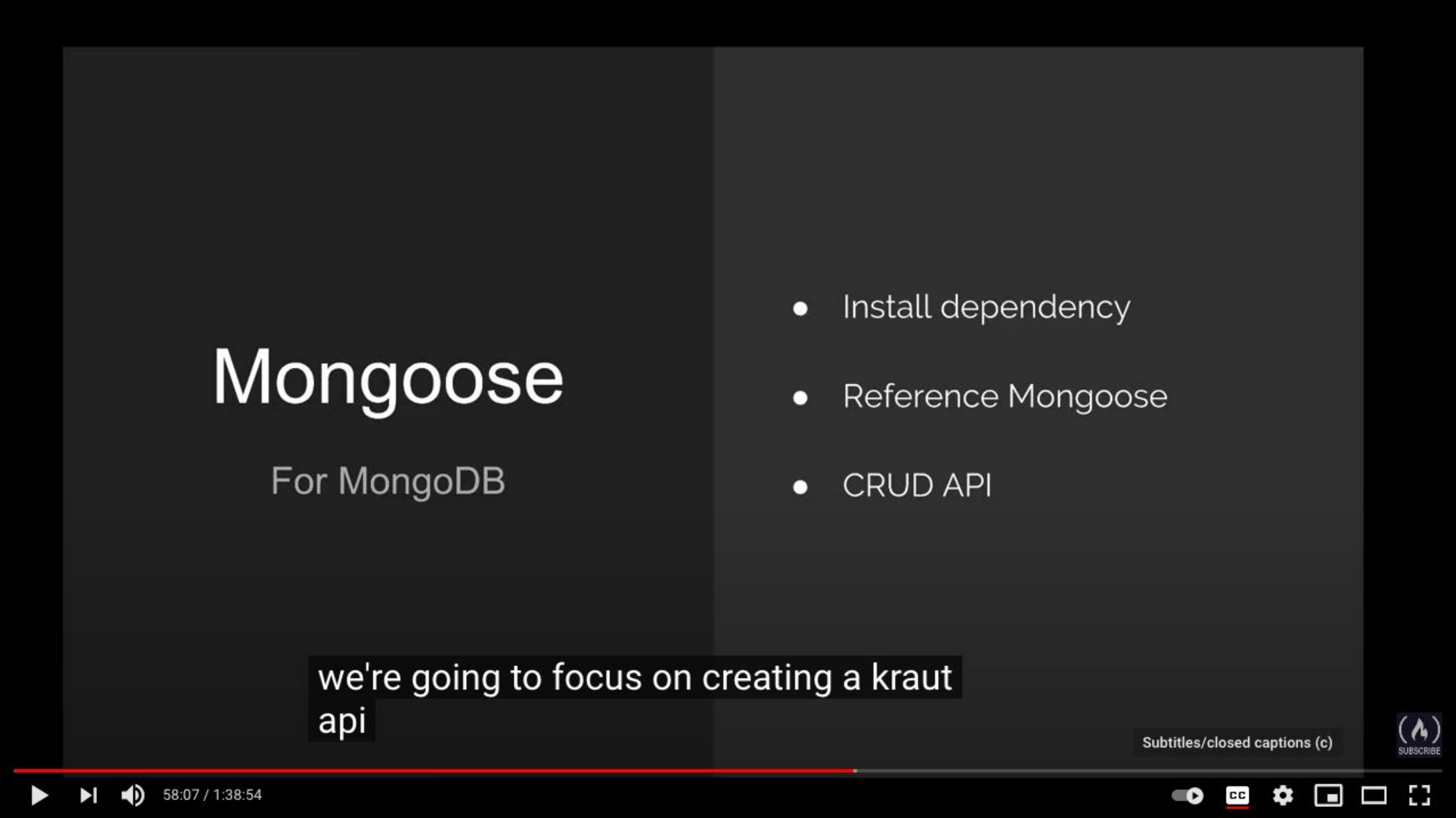

I need slash public so I find one at
HTML so let's retry that so going

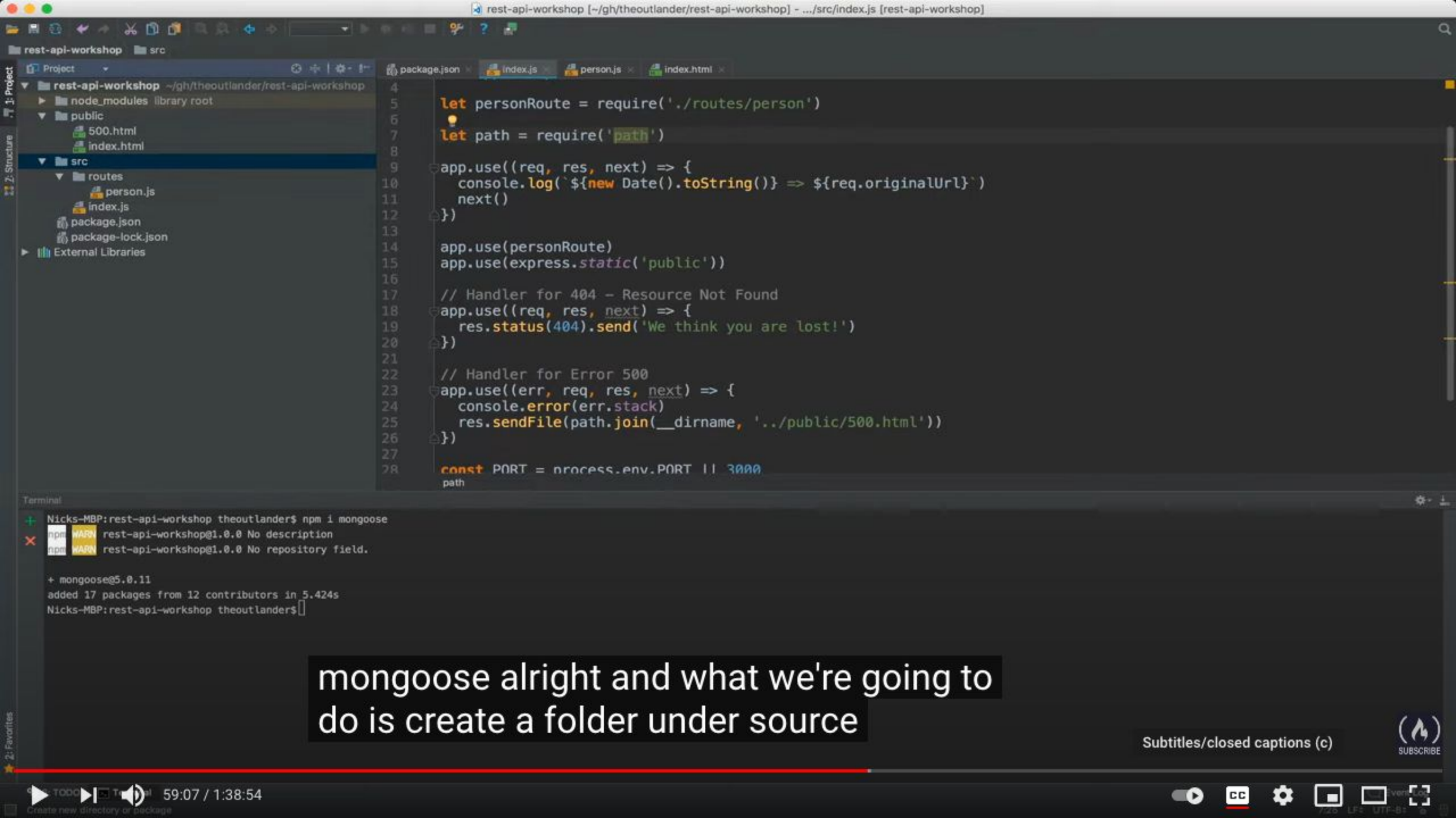
Subtitles/closed captions (c)

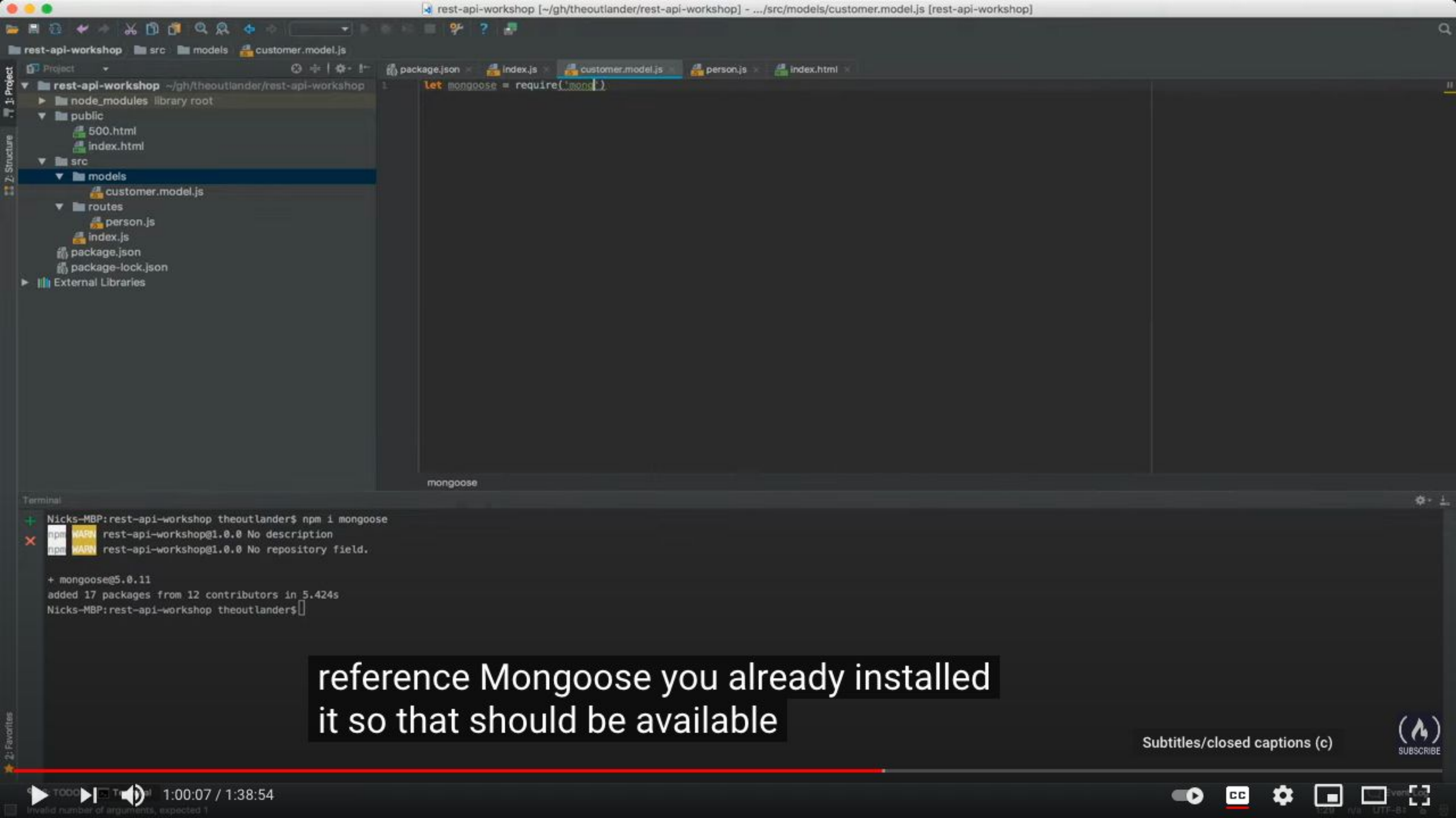
SUBSCRIBE

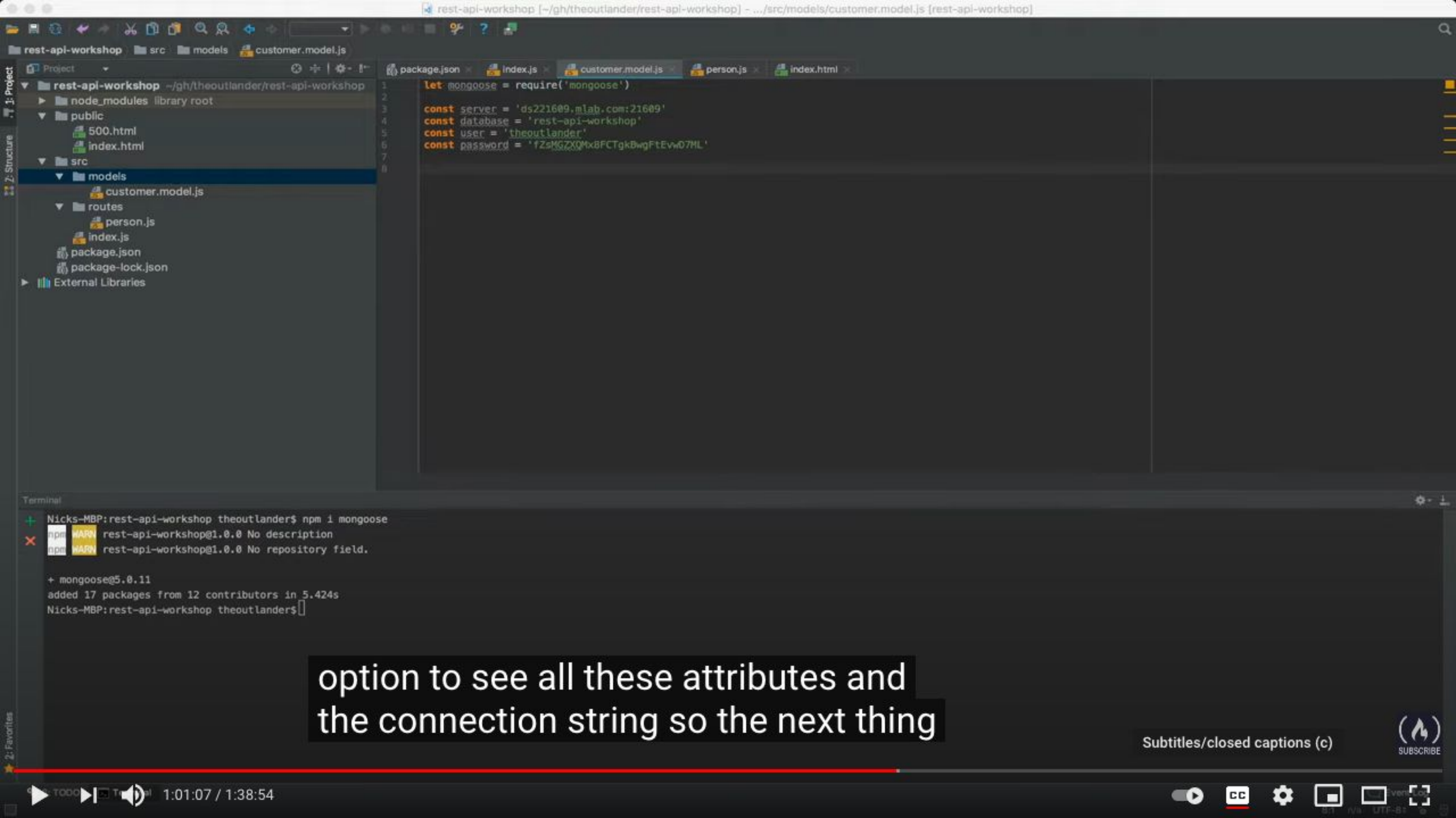
56:07 / 1:38:54

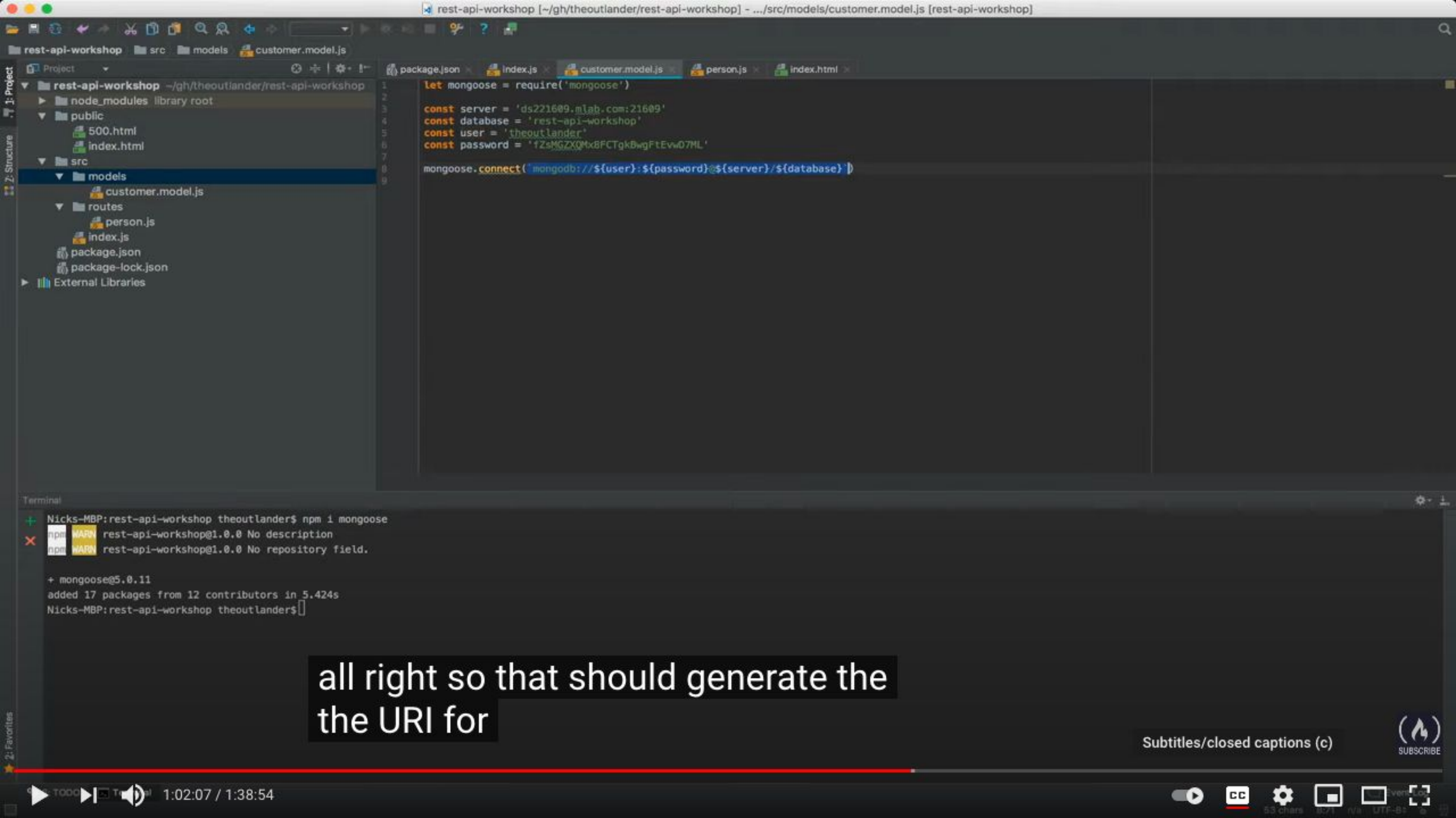

Mongoose
For MongoDB
Install dependency
Reference Mongoose
CRUD API
Mongoose which is an ORM or an OD M
it's basically an object data mapper
Subtitles/closed captions (c)
SUBSCRIBE
57:07 / 1:38:54

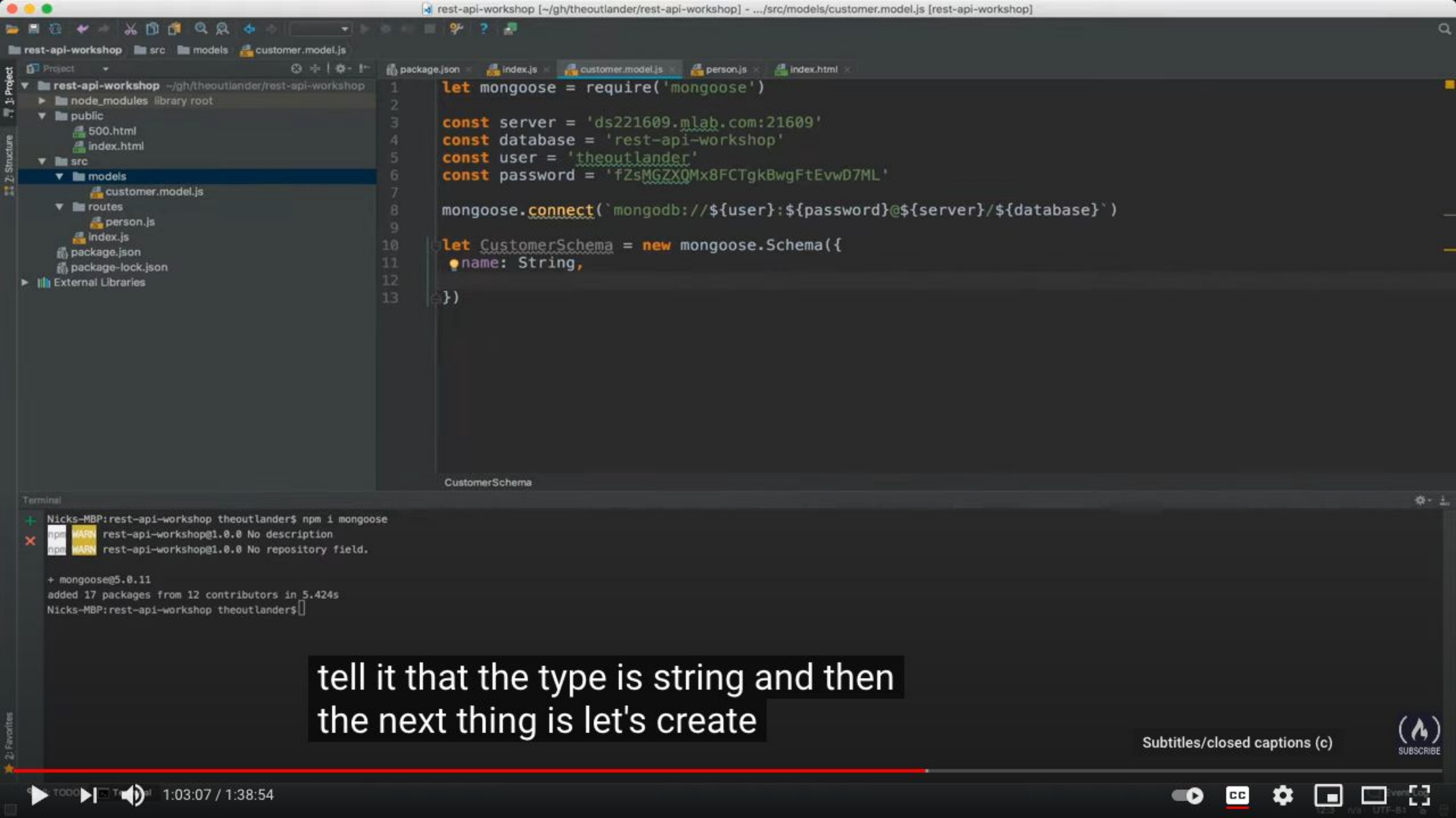

Mongoose
For MongoDB
Install dependency
Reference Mongoose
CRUD API
we're going to focus on creating a kraut
api
Subtitles/closed captions (c)
SUBSCRIBE
58:07 / 1:38:54

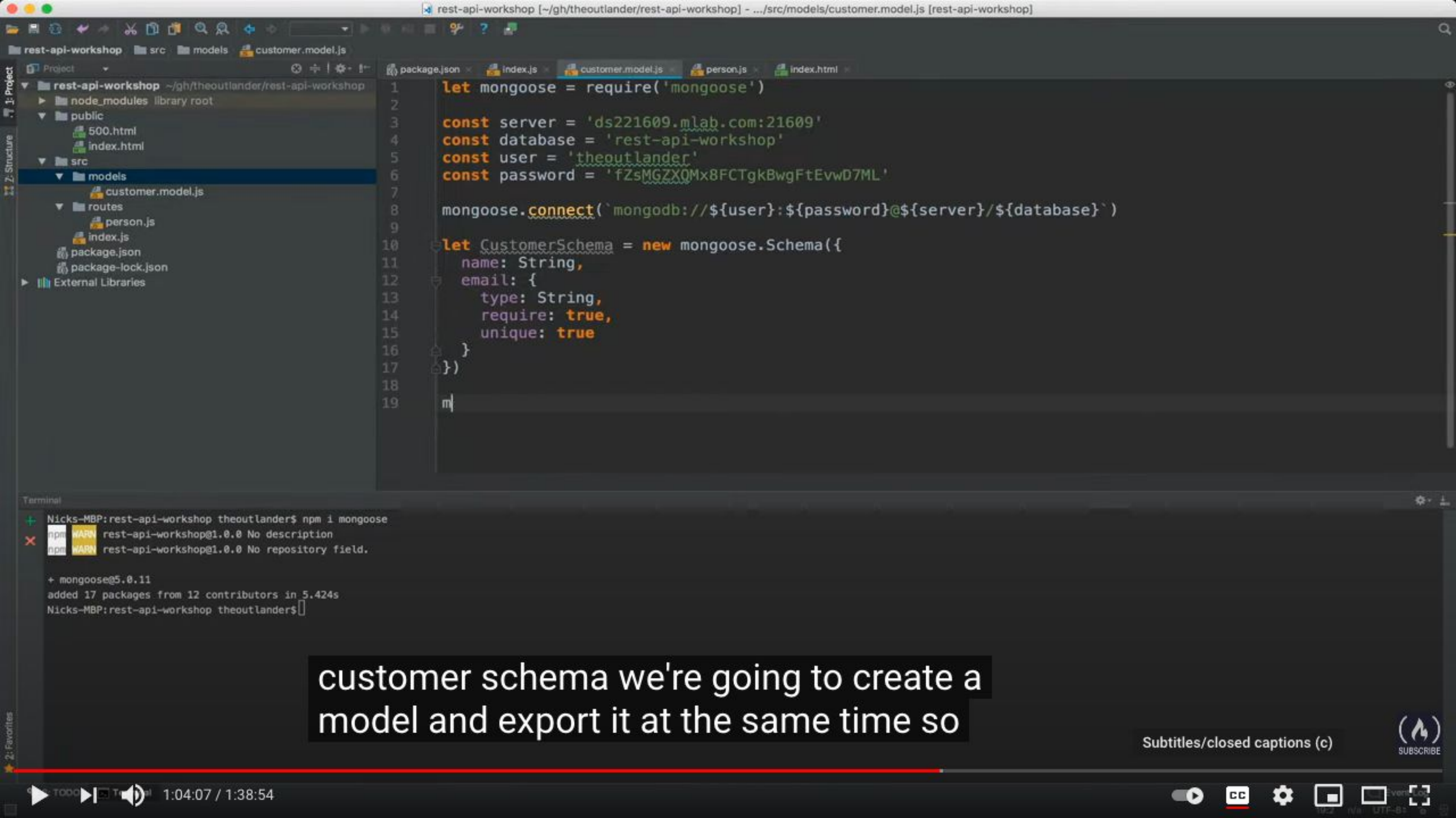

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/index.js [rest-api-workshop]
rest-api-workshop
src
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
person.js
index.html
let personRoute = require('./routes/person')
let path = require('path')
app.use((req, res, next) => {
console.log(`${new Date().toString()} => ${req.originalUrl}`)
next()
})
app.use(personRoute)
app.use(express.static('public'))
// Handler for 404 - Resource Not Found
app.use((req, res, next) => {
res.status(404).send('We think you are lost!')
})
// Handler for Error 500
app.use((err, req, res, next) => {
console.error(err.stack)
res.sendFile(path.join(__dirname, '../public/500.html'))
})
const PORT = process.env.PORT || 3000
path
Terminal
Nicks-MBP:rest-api-workshop theoutlander$ 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$
mongoose alright and what we're going to
do is create a folder under source
Subtitles/closed captions (c)
SUBSCRIBE
59:07 / 1:38:54

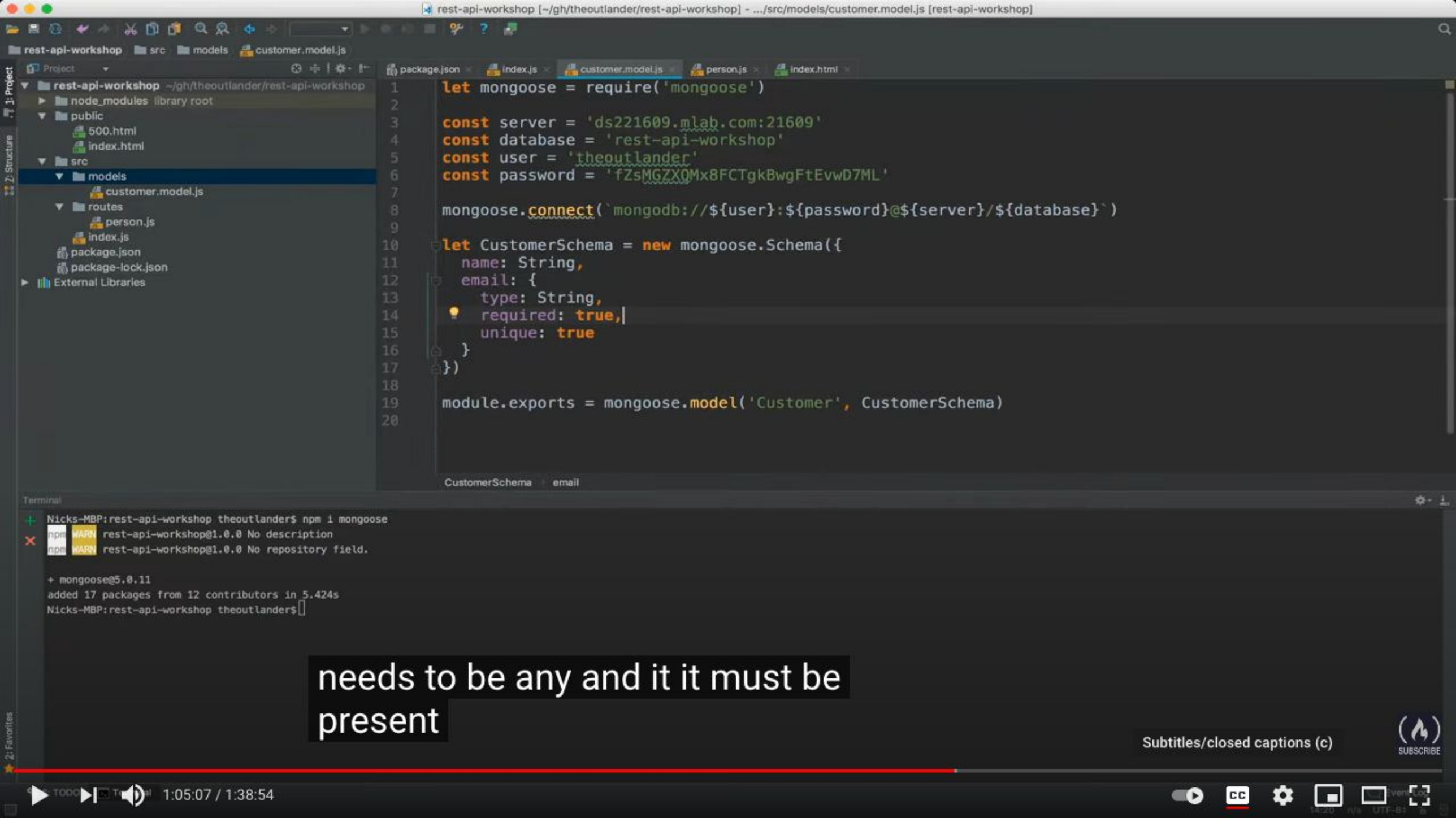

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/models/customer.model.js [rest-api-workshop]
rest-api-workshop
src
models
customer.model.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
person.js
index.html
let mongoose = require('mongo')
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$
reference Mongoose you already installed
it so that should be available
Subtitles/closed captions (c)
SUBSCRIBE
1:00:07 / 1:38:54

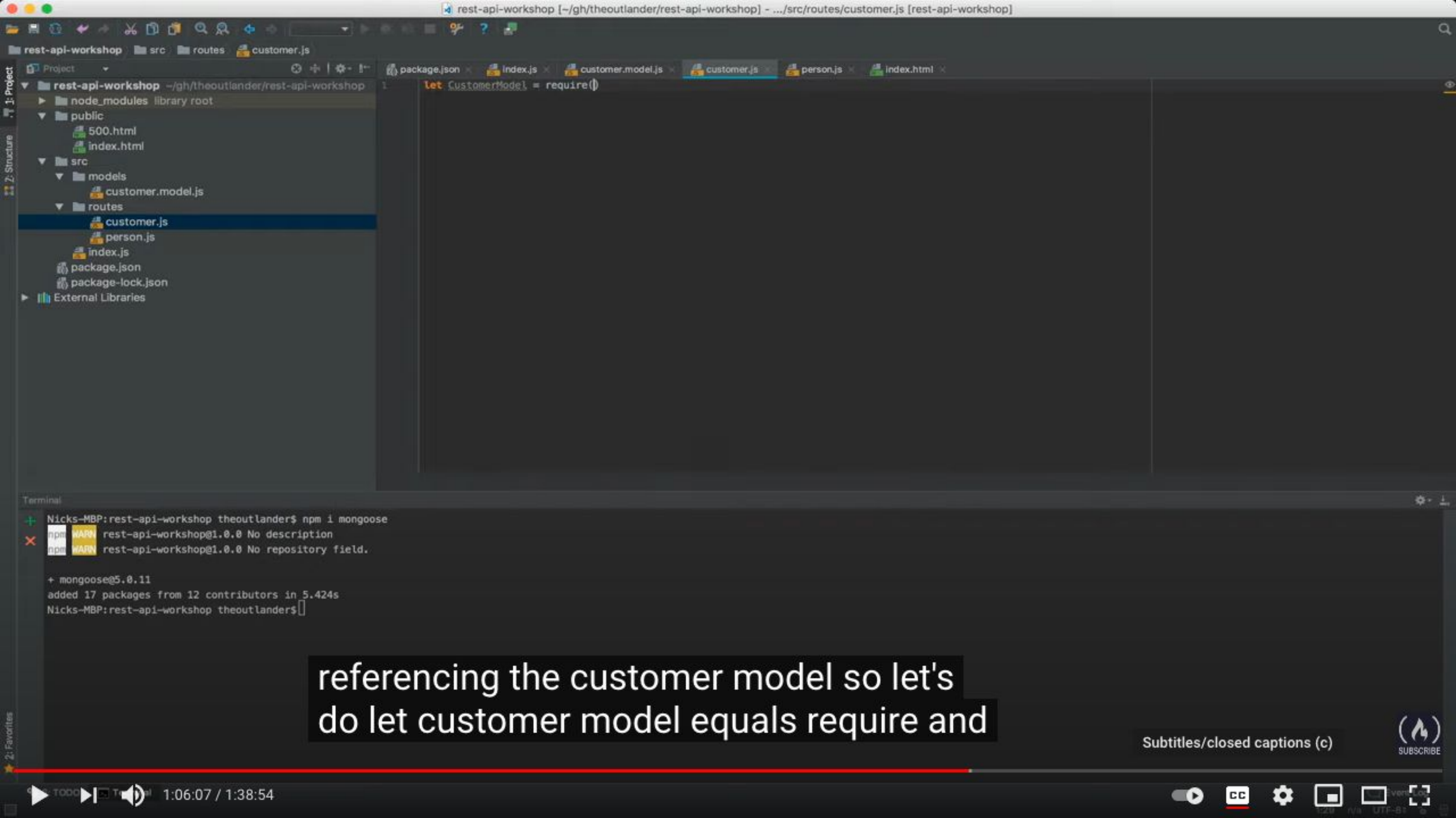

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/models/customer.model.js [rest-api-workshop]
rest-api-workshop
src
models
customer.model.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
person.js
index.html
let mongoose = require('mongoose')
const server = 'ds221609.mlab.com:21609'
const database = 'rest-api-workshop'
const user = 'theoutlander'
const password = 'fZsMGZXQMx8FCTgkBwgFtEvwD7ML'
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$
option to see all these attributes and
the connection string so the next thing
Subtitles/closed captions (c)
SUBSCRIBE
1:01:07 / 1:38:54

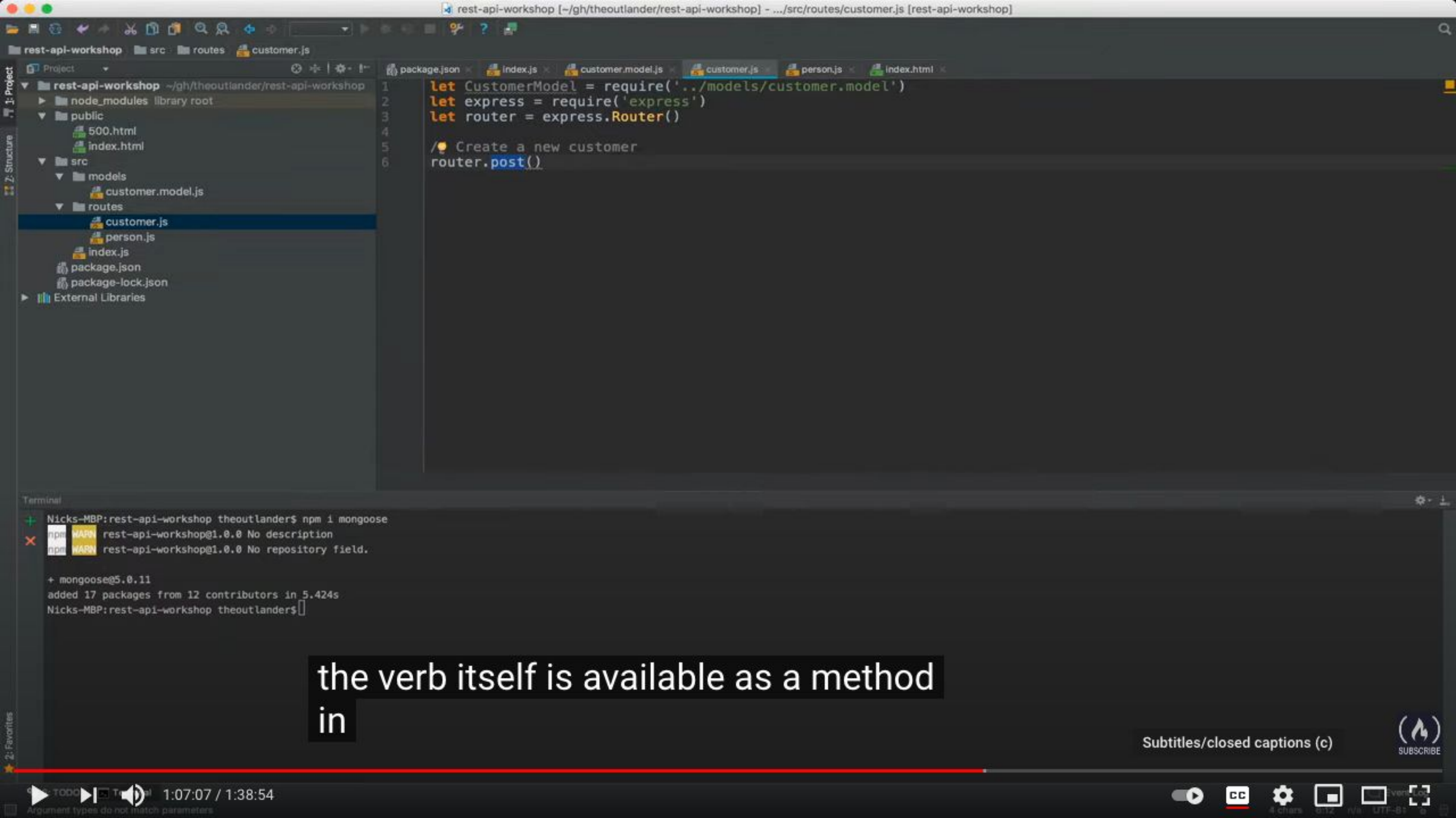

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/models/customer.model.js [rest-api-workshop]
rest-api-workshop src models customer.model.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
person.js
index.html
let mongoose = require('mongoose')
const server = 'ds221609.mlab.com:21609'
const database = 'rest-api-workshop'
const user = 'theoutlander'
const password = 'fZsMGZXQMx8FCTgkBwgFtEvwD7ML'
mongoose.connect(`mongodb://${user}:${password}@${server}/${database}`)
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$
all right so that should generate the
the URI for
Subtitles/closed captions (c)
SUBSCRIBE
1:02:07 / 1:38:54

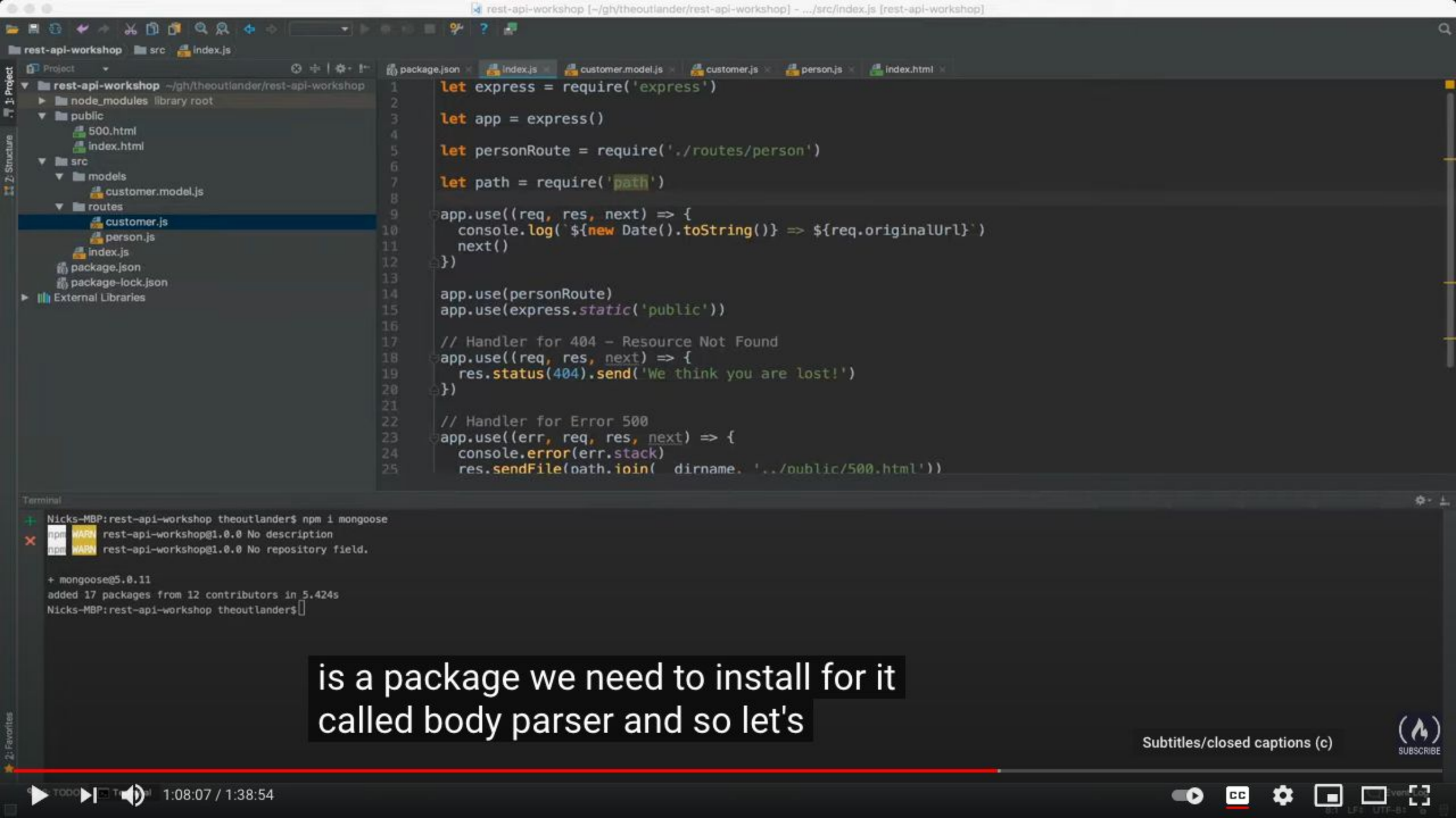

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/models/customer.model.js [rest-api-workshop]
rest-api-workshop
src
models
customer.model.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
person.js
index.html
let mongoose = require('mongoose')
const server = 'ds221609.mlab.com:21609'
const database = 'rest-api-workshop'
const user = 'theoutlander'
const password = 'fZsMGZXOMx8FCTgkBwgFtEvwD7ML'
mongoose.connect(`mongodb://${user}:${password}@${server}/${database}`)
let CustomerSchema = new mongoose.Schema({
name: String,
})
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$
tell it that the type is string and then
the next thing is let's create
Subtitles/closed captions (c)
SUBSCRIBE
1:03:07 / 1:38:54

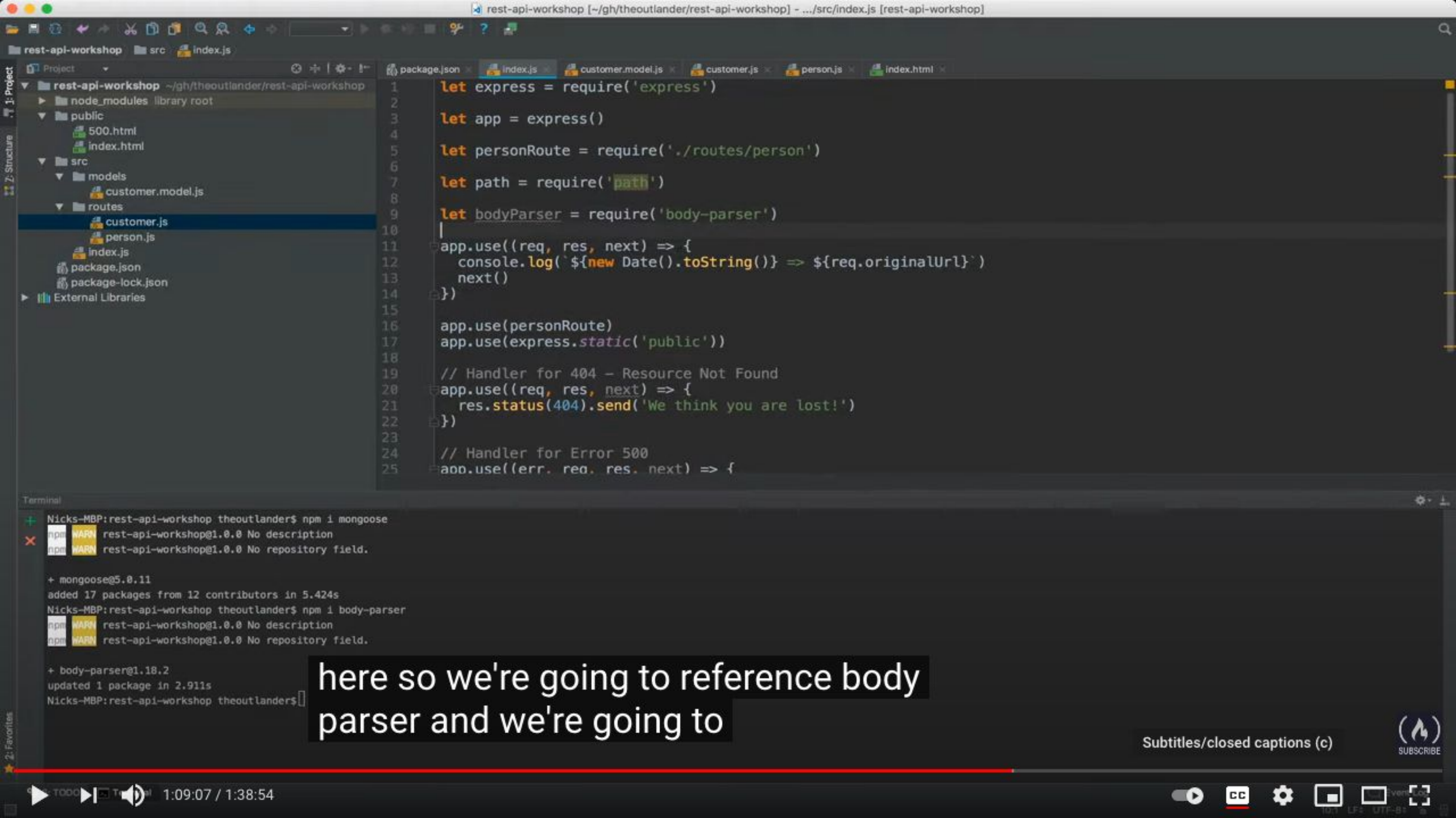

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/models/customer.model.js [rest-api-workshop]
rest-api-workshop
src
models
customer.model.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
person.js
index.html
let mongoose = require('mongoose')
const server = 'ds221609.mlab.com:21609'
const database = 'rest-api-workshop'
const user = 'theoutlander'
const password = 'fZsMGZXQMx8FCTgkBwgFtEvwD7ML'
mongoose.connect(`mongodb://${user}:${password}@${server}/${database}`)
let CustomerSchema = new mongoose.Schema({
name: String,
email: {
type: String,
require: true,
unique: true
}
})
m
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$
customer schema we're going to create a
model and export it at the same time so
Subtitles/closed captions (c)
SUBSCRIBE
1:04:07 / 1:38:54

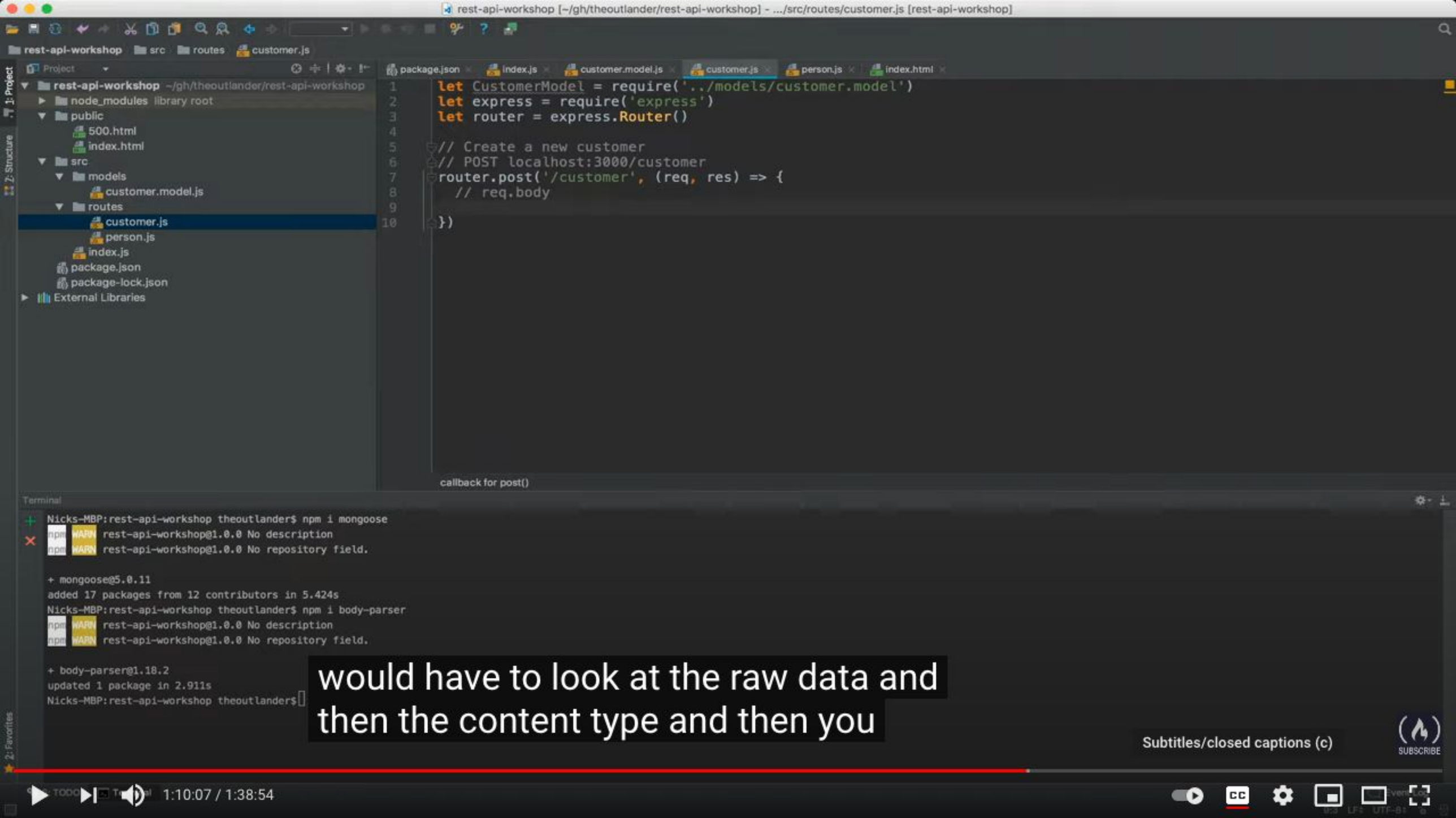

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/models/customer.model.js [rest-api-workshop]
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
person.js
index.html
let mongoose = require('mongoose')
const server = 'ds221609.mlab.com:21609'
const database = 'rest-api-workshop'
const user = 'theoutlander'
const password = 'fZsMGZXQMx8FCTgkBwgFtEvwD7ML'
mongoose.connect(`mongodb://${user}:${password}@${server}/${database}`)
let CustomerSchema = new mongoose.Schema({
name: String,
email: {
type: String,
required: true,
unique: true
}
})
module.exports = mongoose.model('Customer', CustomerSchema)
CustomerSchema
email
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$
needs to be any and it it must be present
Subtitles/closed captions (c)
SUBSCRIBE
1:05:07 / 1:38:54

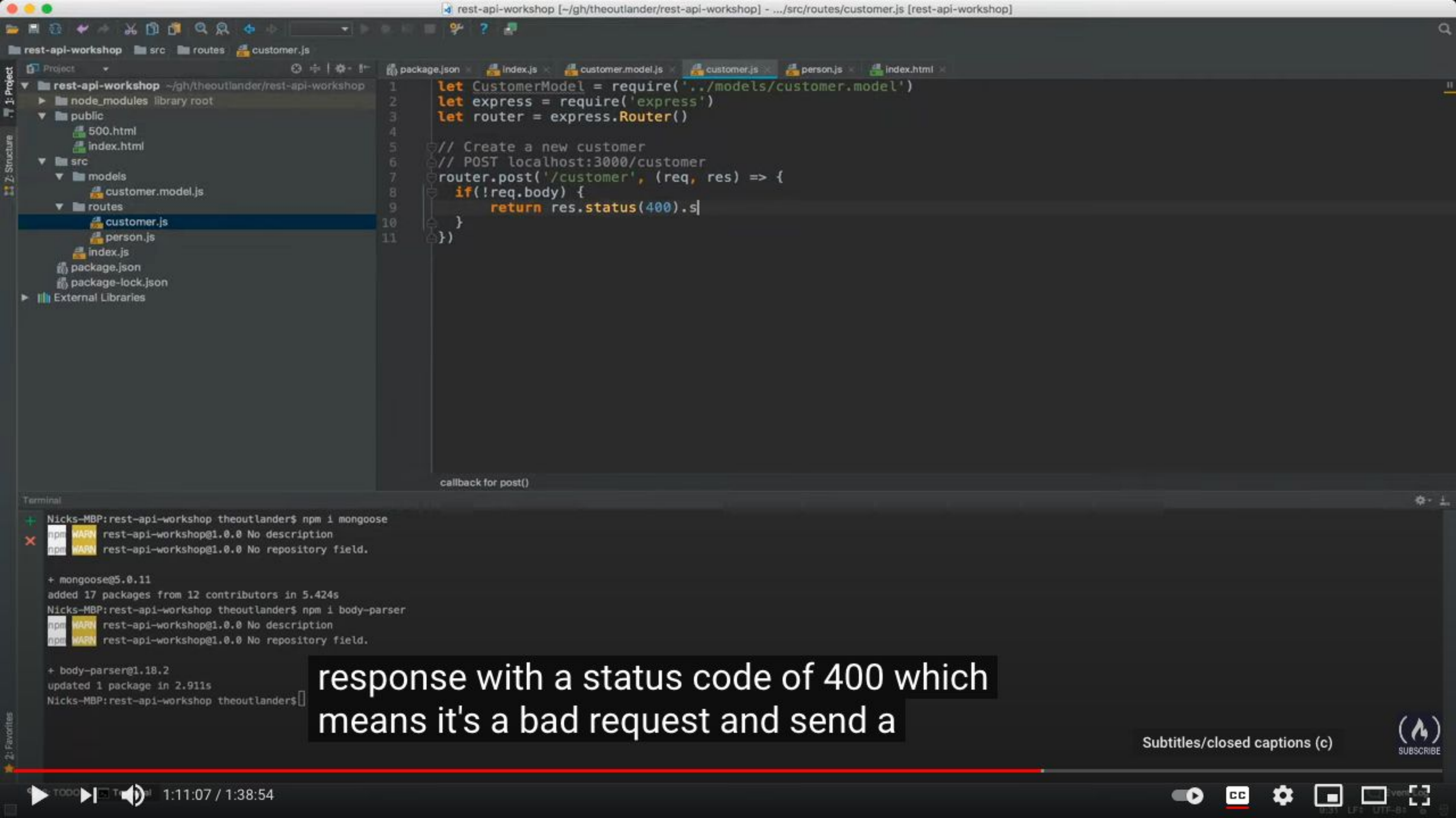

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop src routes customer.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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$
referencing the customer model so let's
do let customer model equals require and
Subtitles/closed captions (c)
SUBSCRIBE
1:06:07 / 1:38:54

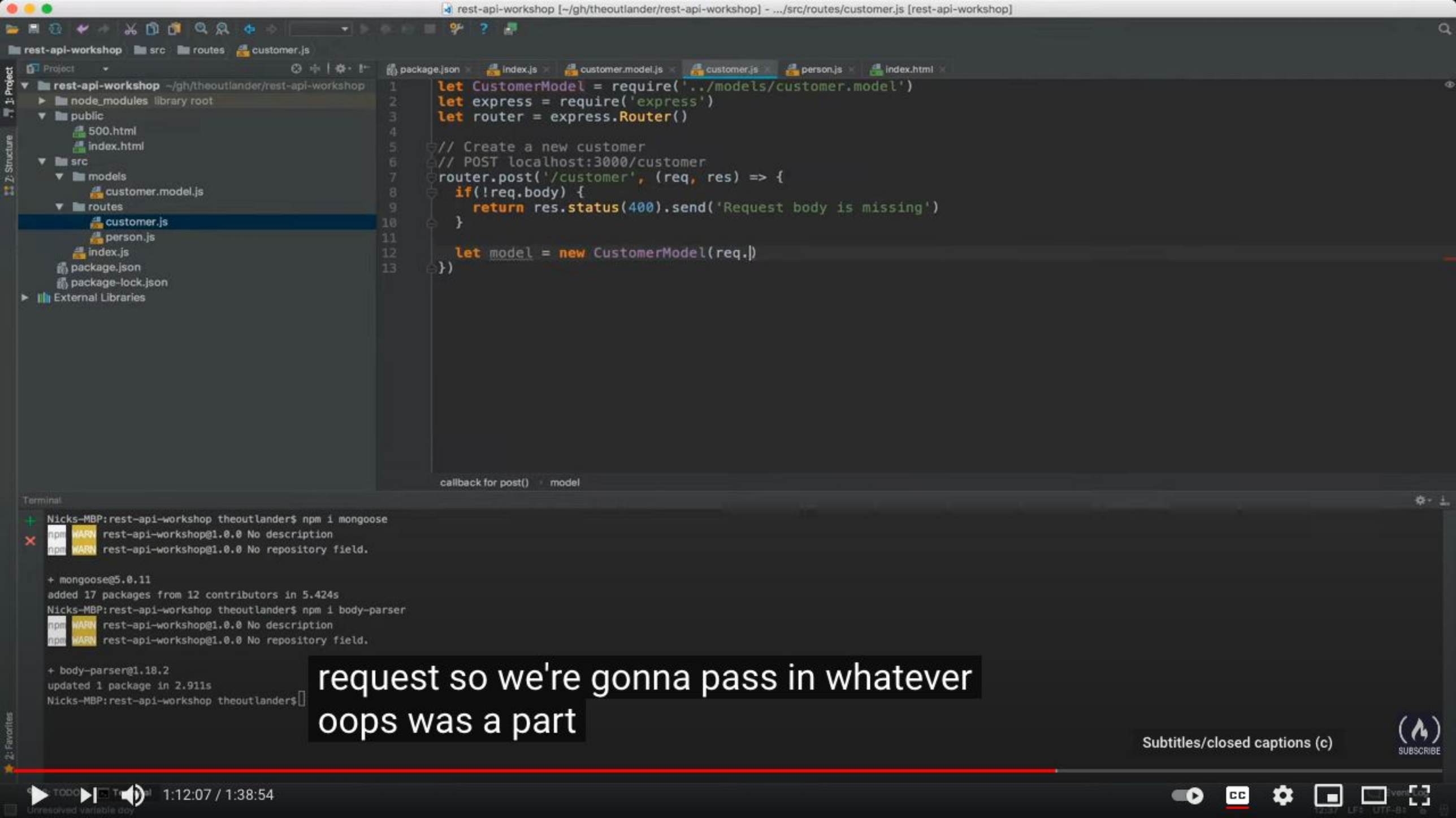

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop src routes customer.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
let CustomerModel = require('../models/customer.model')
let express = require('express')
let router = express.Router()
// Create a new customer
router.post()
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$
the verb itself is available as a method
in
Subtitles/closed captions (c)
SUBSCRIBE
1:07:07 / 1:38:54

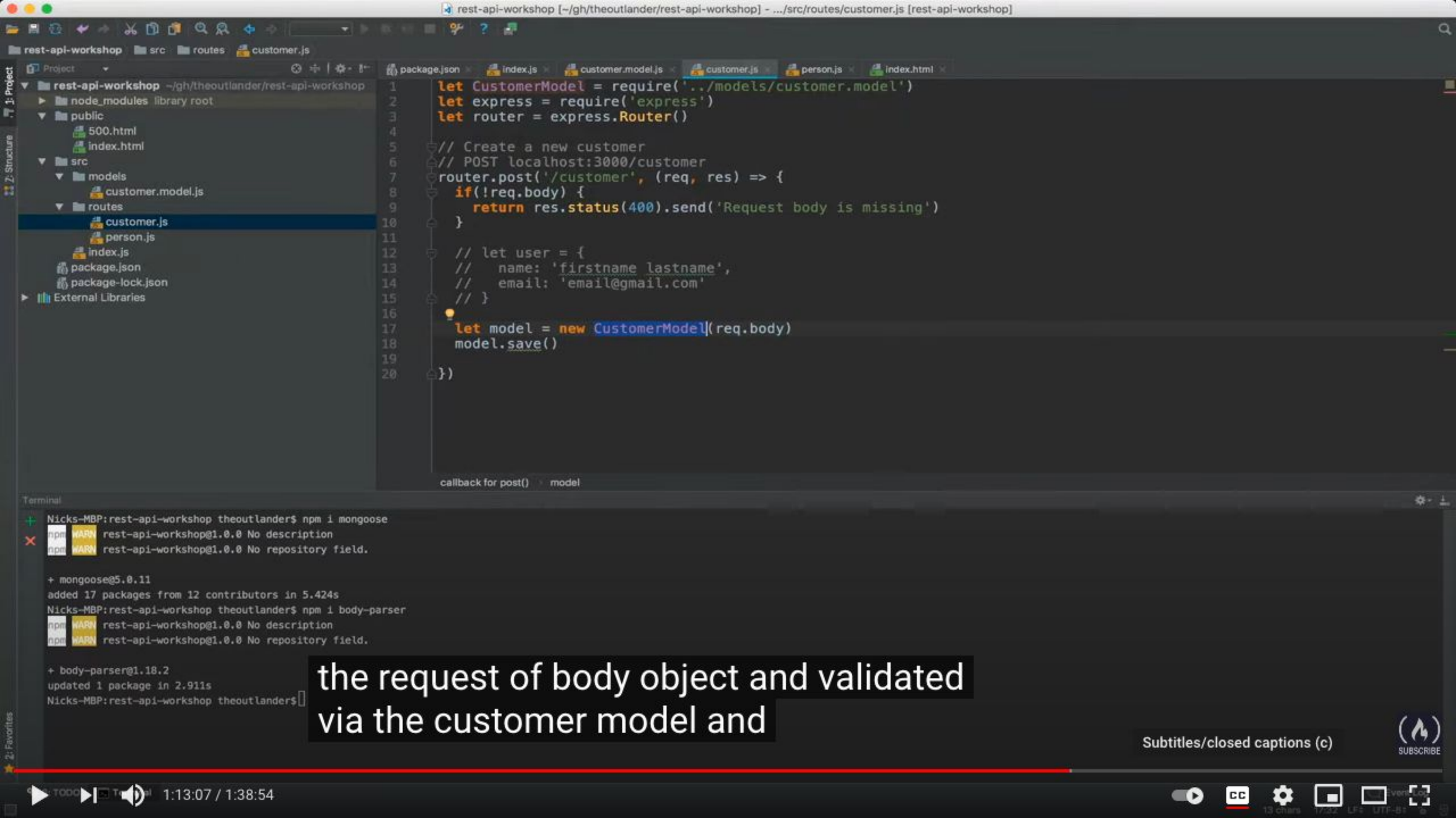

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/index.js [rest-api-workshop]
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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()} => ${req.originalUrl}`)
next()
})
app.use(personRoute)
app.use(express.static('public'))
// Handler for 404 - Resource Not Found
app.use((req, res, next) => {
res.status(404).send('We think you are lost!')
})
// Handler for Error 500
app.use((err, req, res, next) => {
console.error(err.stack)
Terminal
Nicks-MBP:rest-api-workshop theoutlander$ 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$
is a package we need to install for it
called body parser and so let's
Subtitles/closed captions (c)
SUBSCRIBE
1:08:07 / 1:38:54

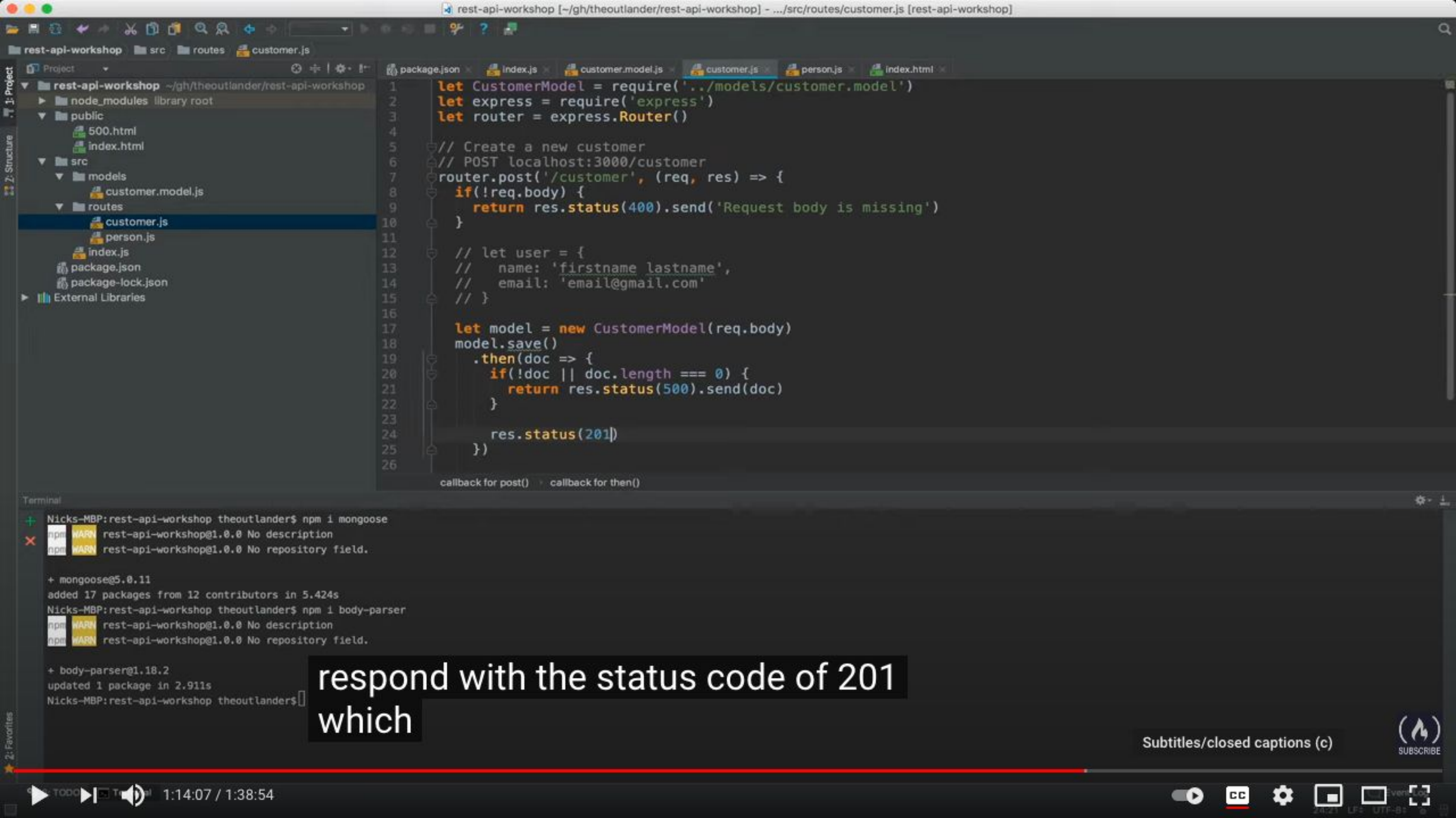

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/index.js [rest-api-workshop]
rest-api-workshop
src
index.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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()} => ${req.originalUrl}`)
next()
})
app.use(personRoute)
app.use(express.static('public'))
// Handler for 404 - Resource Not Found
app.use((req, res, next) => {
res.status(404).send('We think you are lost!')
})
// Handler for Error 500
Terminal
Nicks-MBP:rest-api-workshop theoutlander$ 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$
here so we're going to reference body parser and we're going to
Subtitles/closed captions (c)
SUBSCRIBE
1:09:07 / 1:38:54

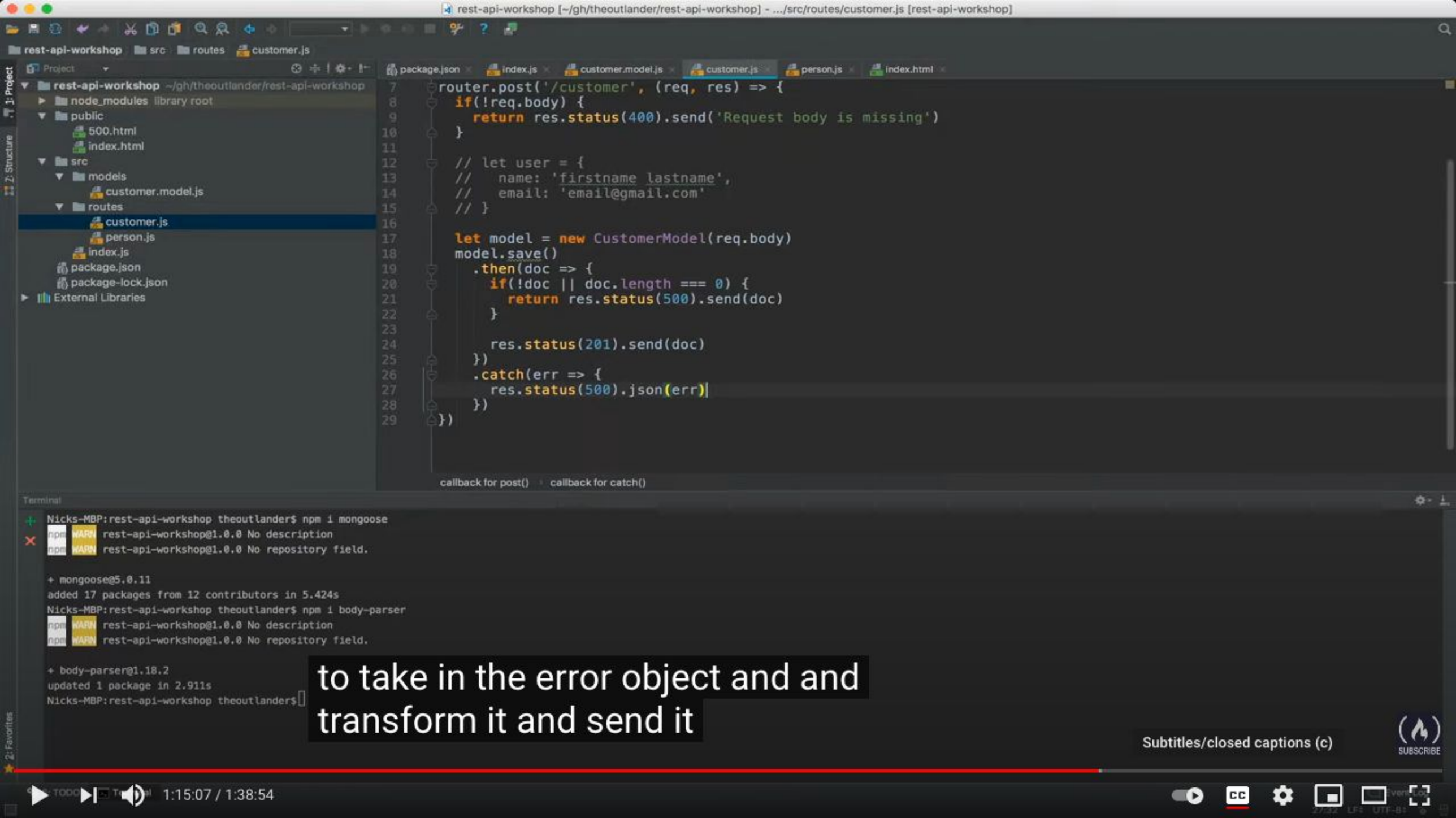

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
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) => {
// req.body
})
callback for post()
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$
would have to look at the raw data and
then the content type and then you
Subtitles/closed captions (c)
SUBSCRIBE
1:10:07 / 1:38:54

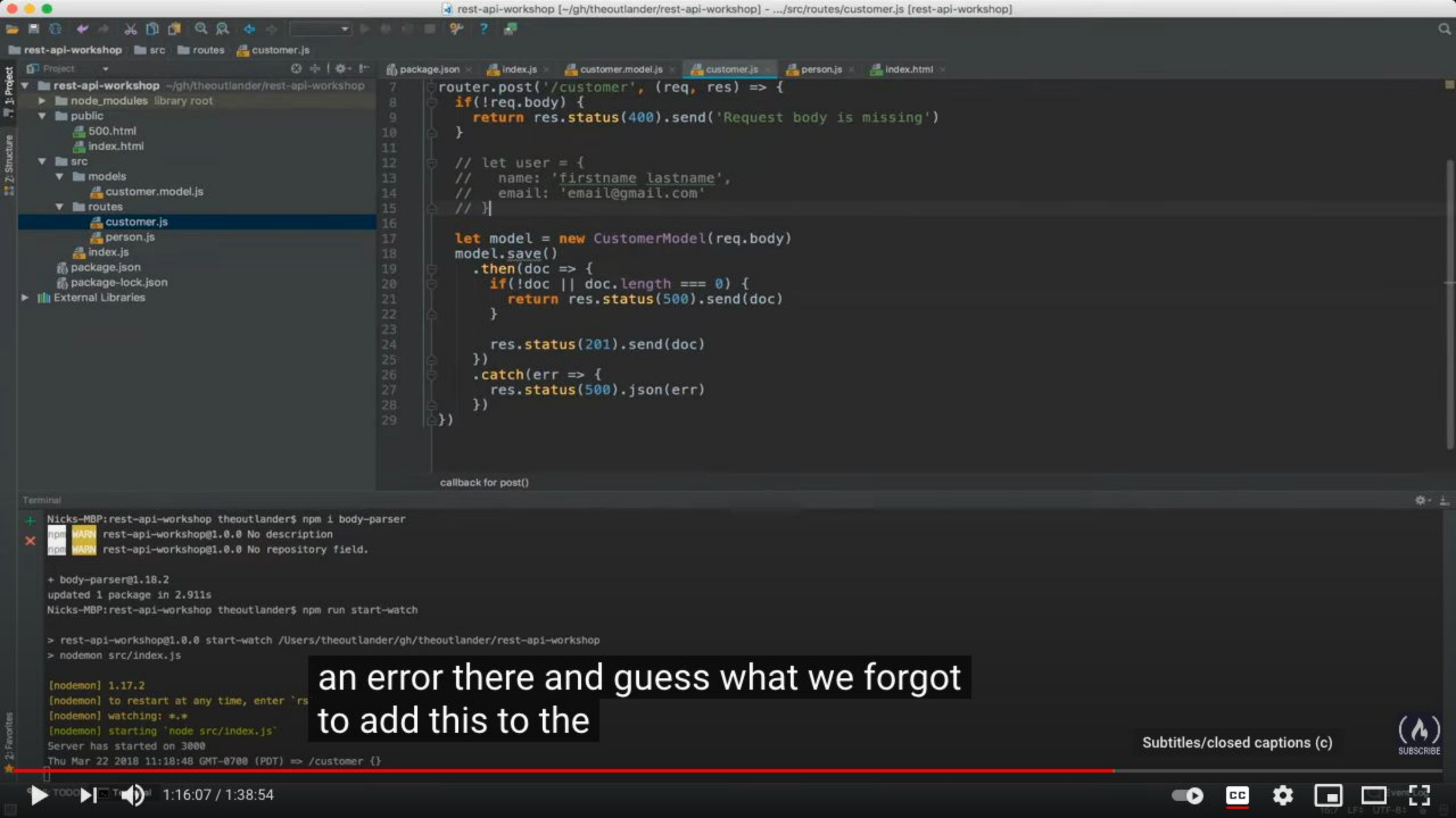

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop src routes customer.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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).s
}
})
callback for post()
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$
response with a status code of 400 which
means it's a bad request and send a
Subtitles/closed captions (c)
SUBSCRIBE
1:11:07 / 1:38:54

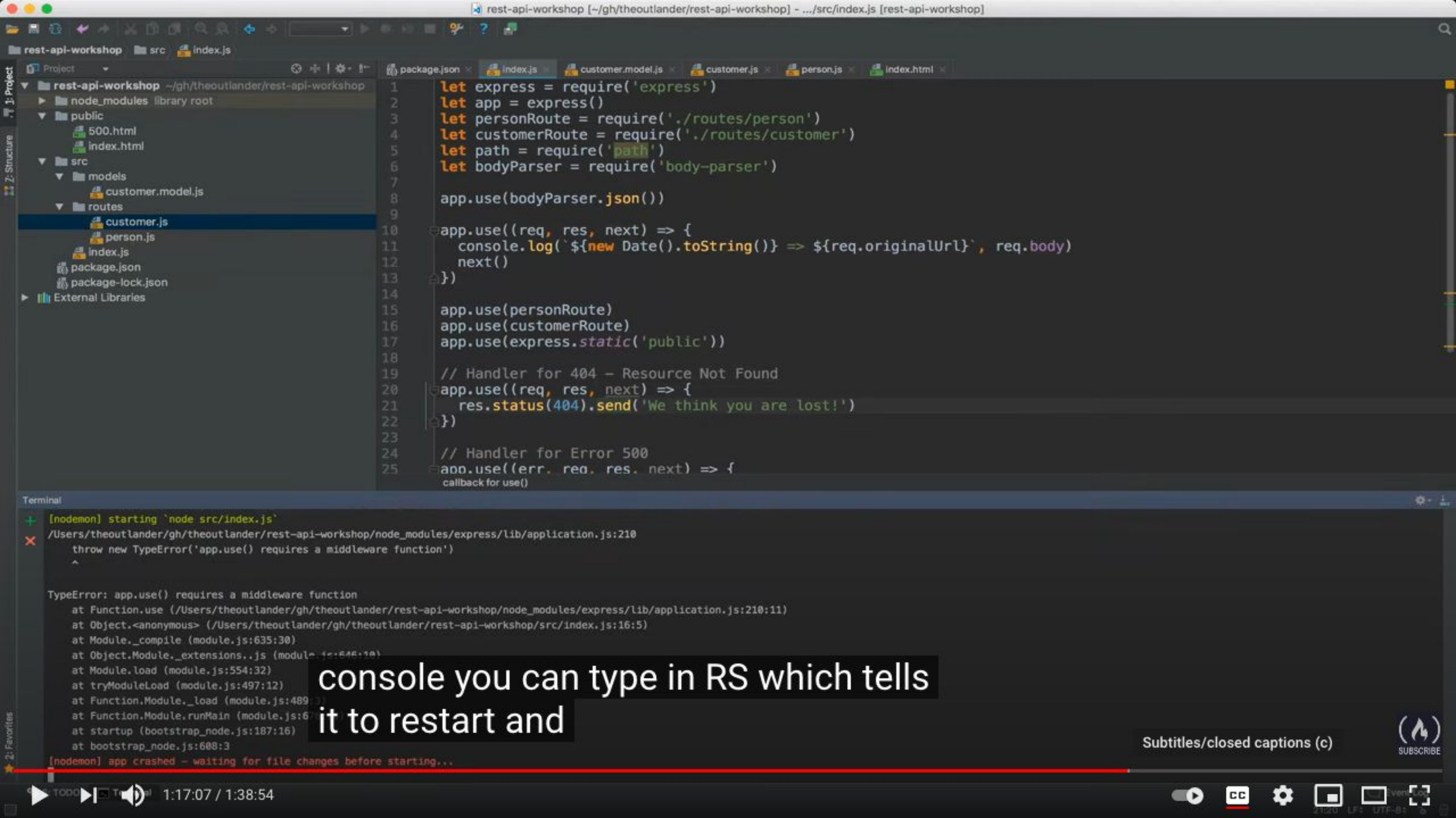

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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.)
})
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$
request so we're gonna pass in whatever
oops was a part
Subtitles/closed captions (c)
SUBSCRIBE
1:12:07 / 1:38:54

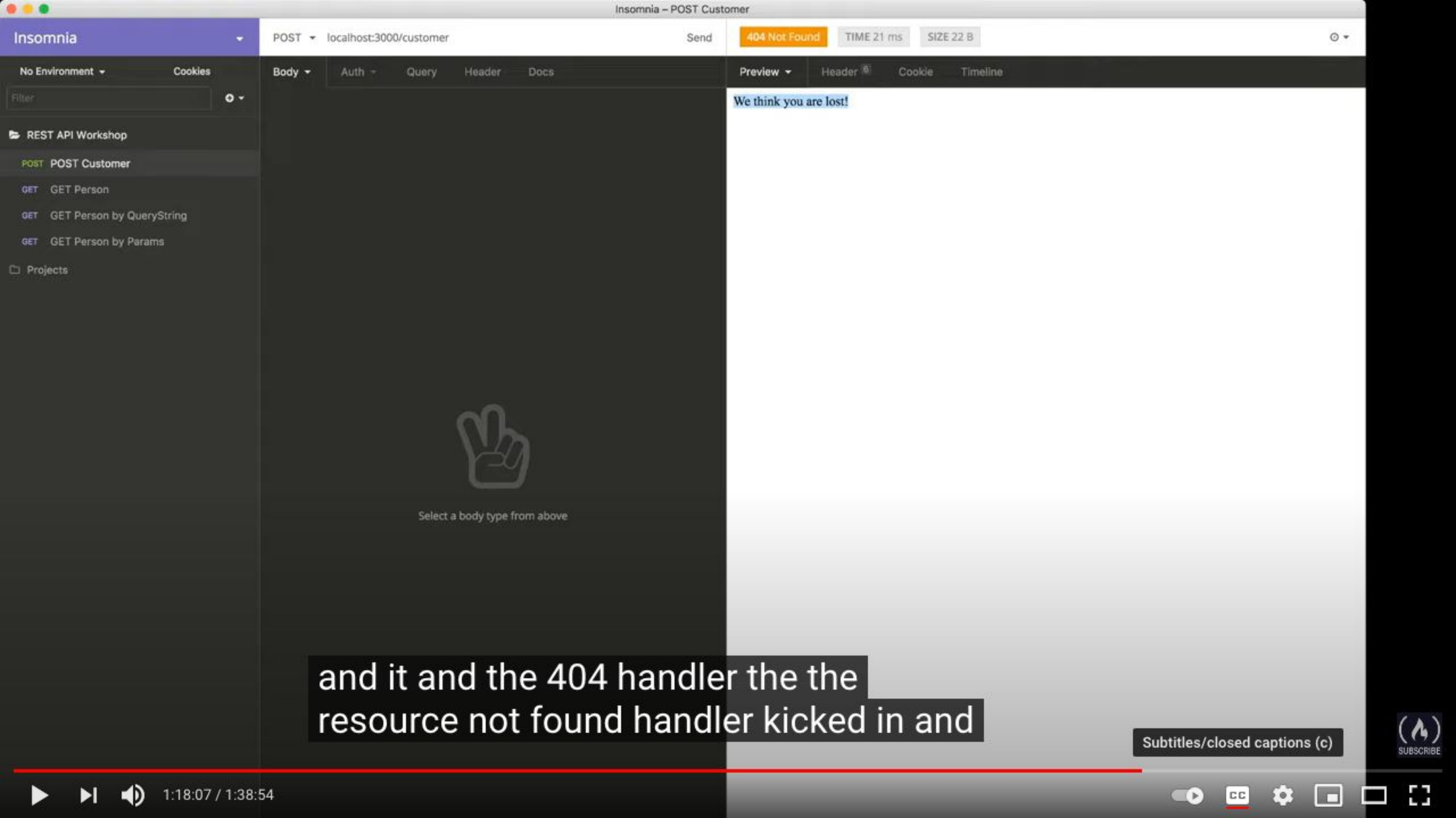

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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()
})
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 request of body object and validated
via the customer model and
Subtitles/closed captions (c)
SUBSCRIBE
1:13:07 / 1:38:54

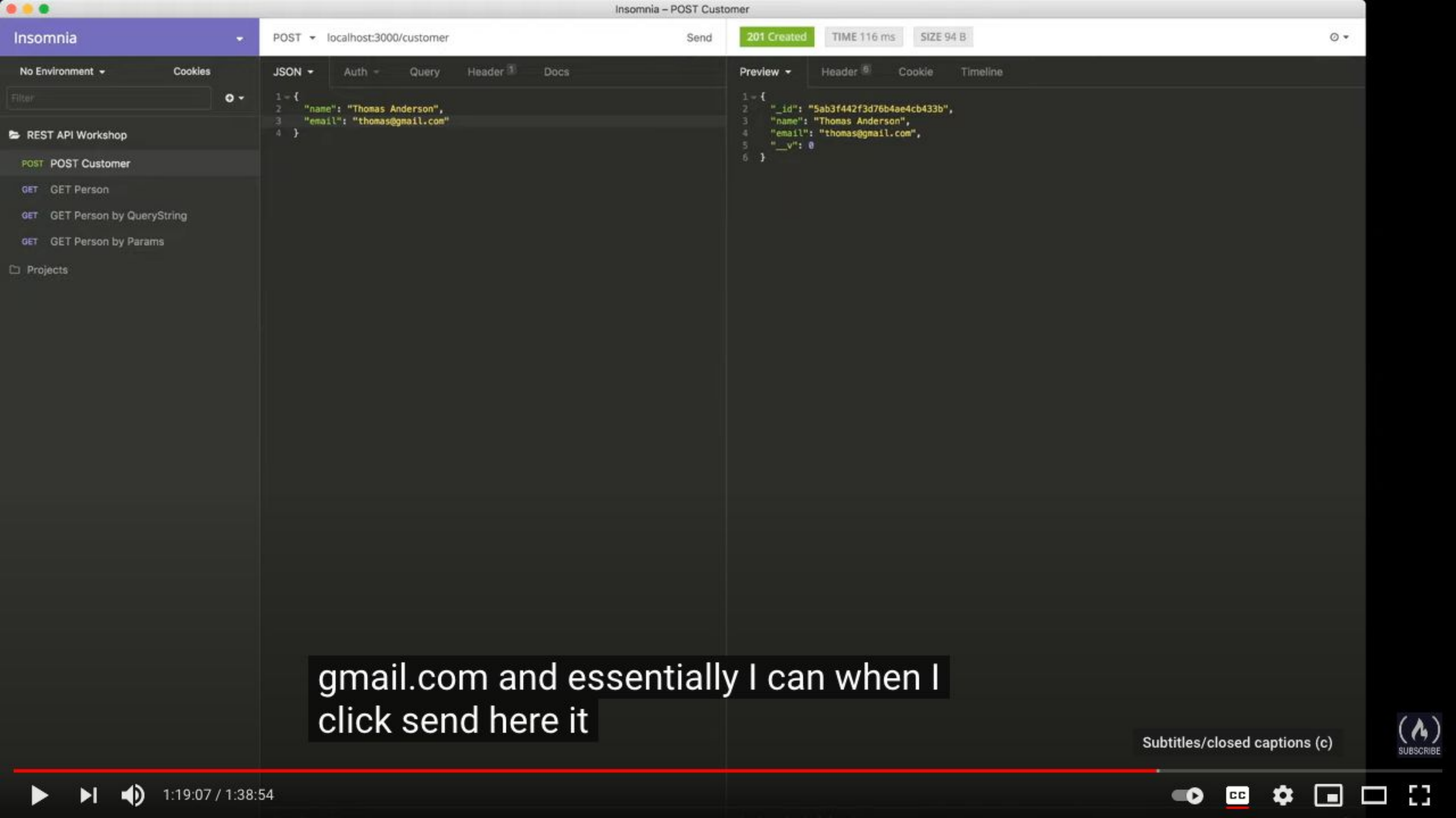

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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)
})
callback for post() callback for then()
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$
respond with the status code of 201
which
Subtitles/closed captions (c)
SUBSCRIBE
1:14:07 / 1:38:54

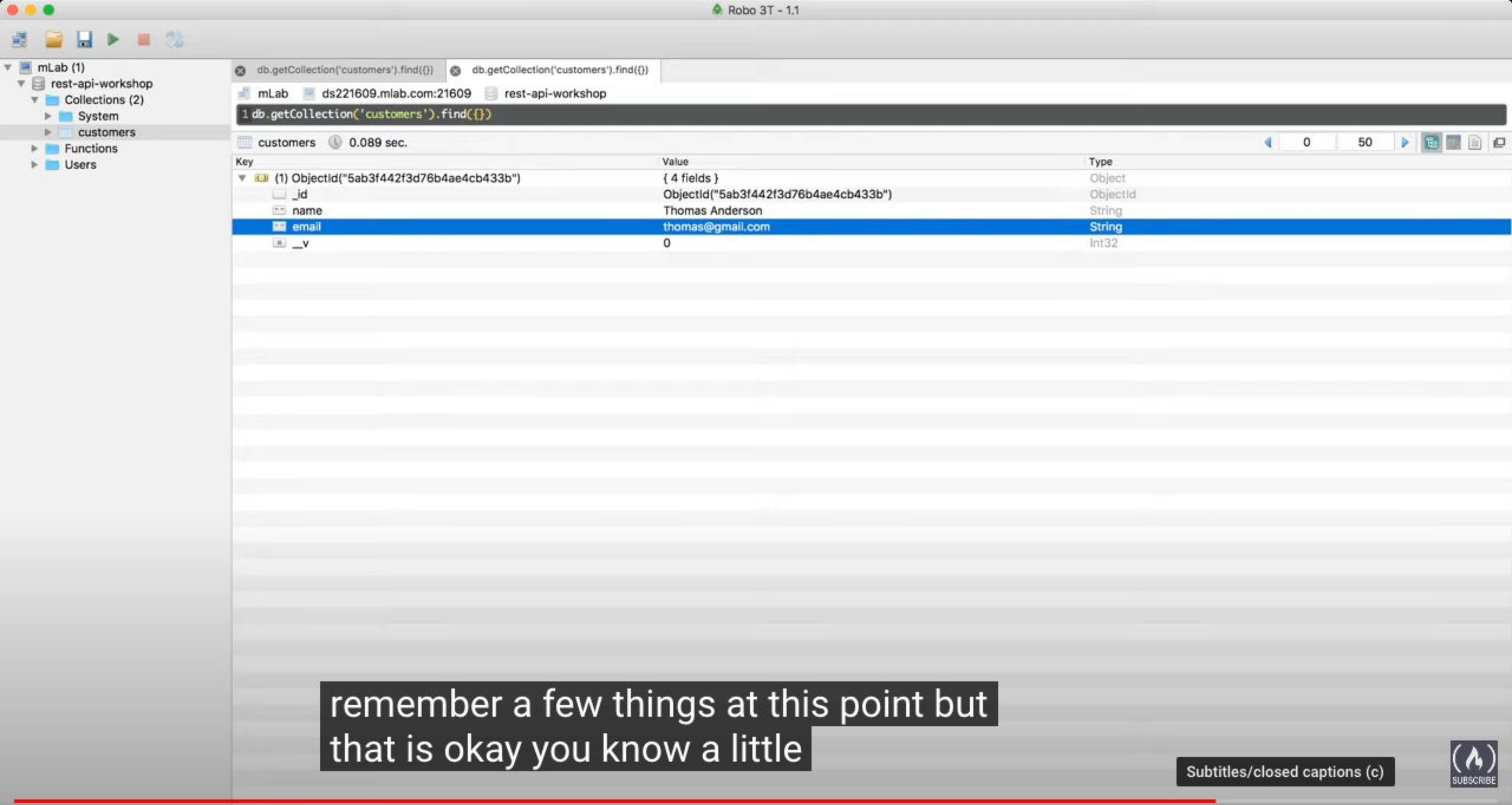

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop src routes customer.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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).send(doc)
})
.catch(err => {
res.status(500).json(err)
})
})
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@5.0.11
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$
to take in the error object and and transform it and send it
Subtitles/closed captions (c)
SUBSCRIBE
1:15:07 / 1:38:54

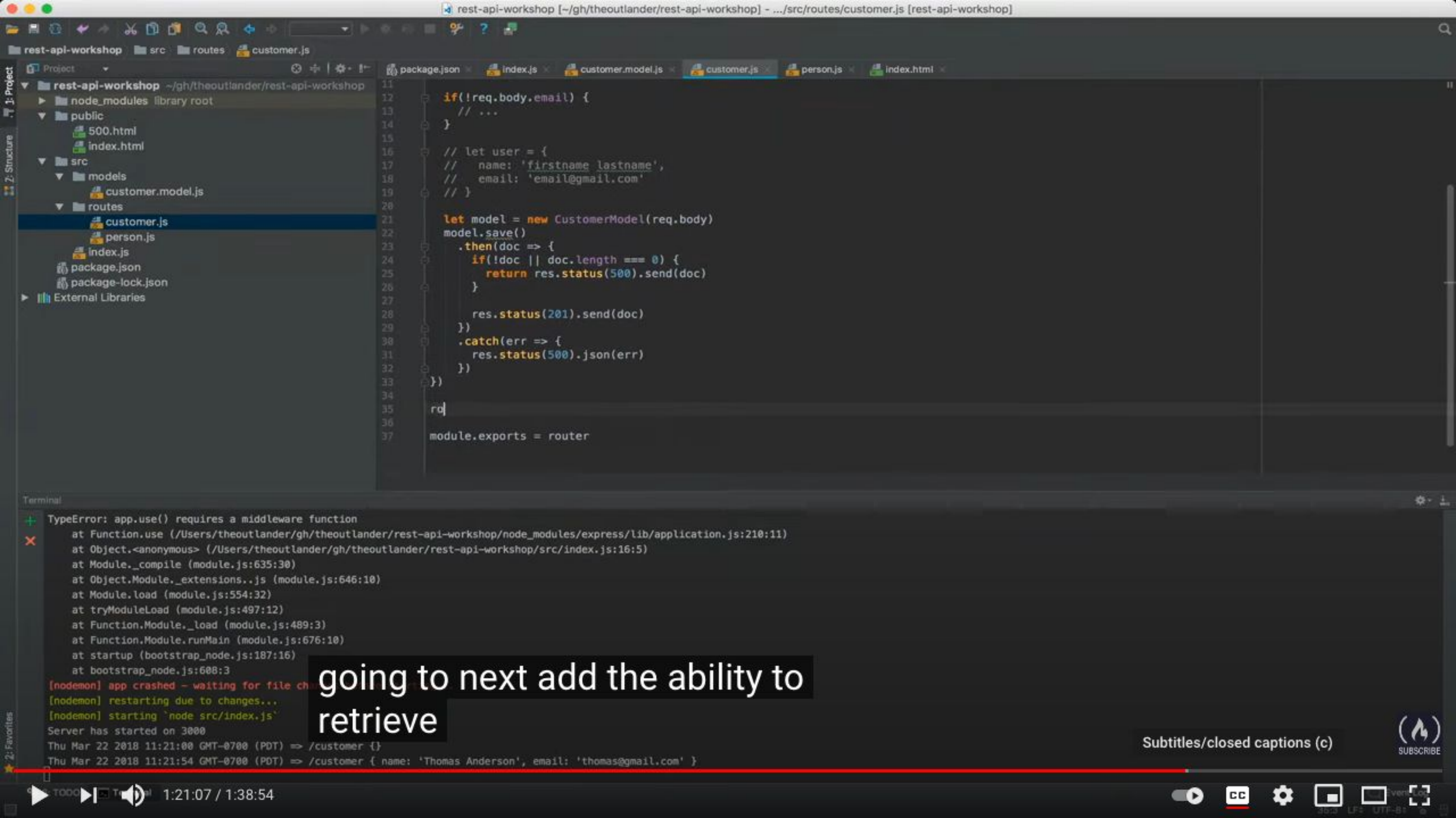

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop src routes customer.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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).send(doc)
})
.catch(err => {
res.status(500).json(err)
})
})
callback for post()
Terminal
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$ 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 {}
an error there and guess what we forgot
to add this to the
Subtitles/closed captions (c)
SUBSCRIBE
1:16:07 / 1:38:54

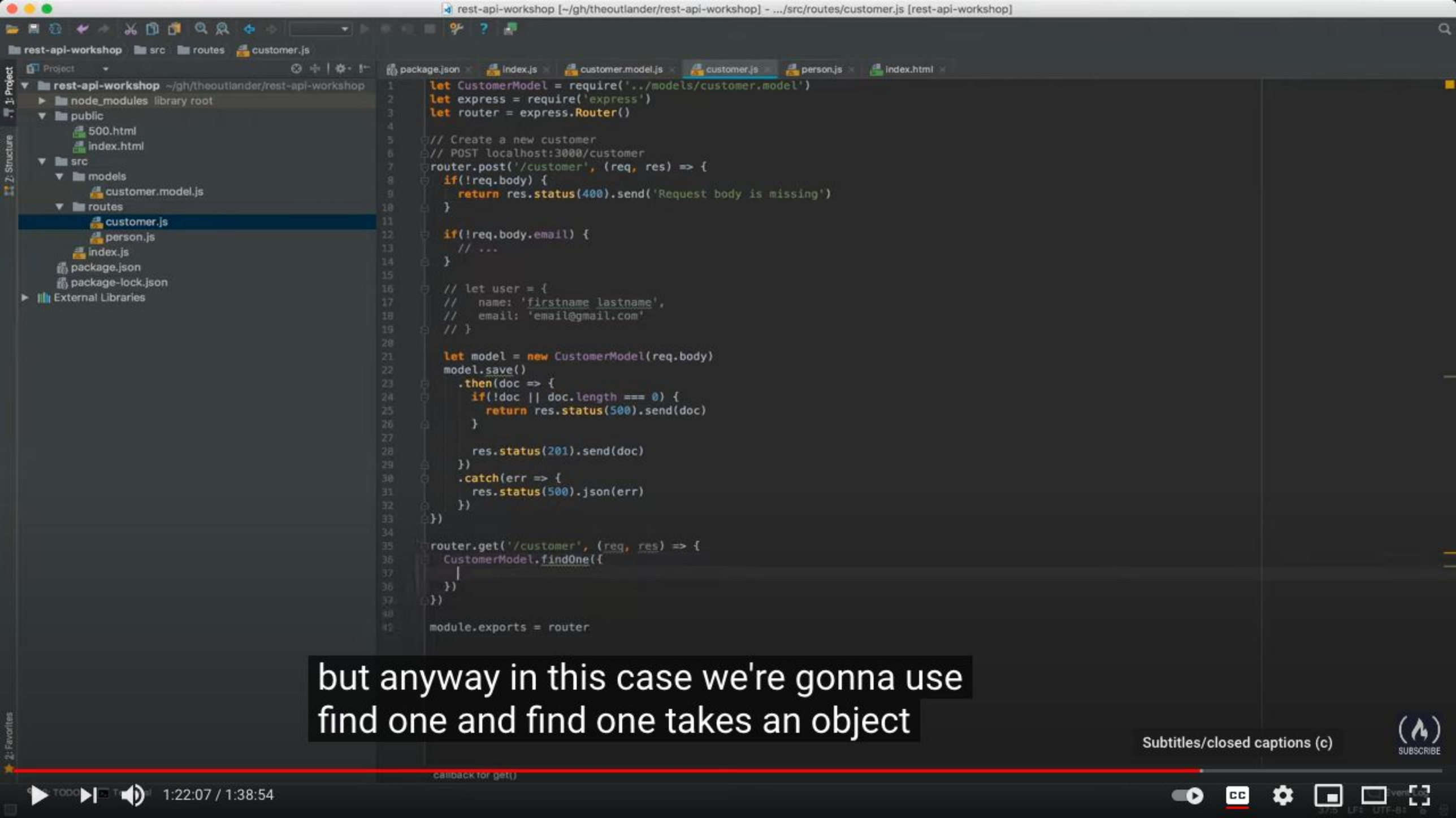

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/index.js [rest-api-workshop]
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
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) => {
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:16:5)
at Module._compile (module.js:635:30)
at Module.load (module.js:554:32)
at tryModuleLoad (module.js:497:12)
at Function.Module._load (module.js:489
at Function.Module.runMain (module.js:6
at startup (bootstrap_node.js:187:16)
at bootstrap_node.js:608:3
[nodemon] app crashed - waiting for file changes before starting...
console you can type in RS which tells
it to restart and
Subtitles/closed captions (c)
SUBSCRIBE
1:17:07 / 1:38:54

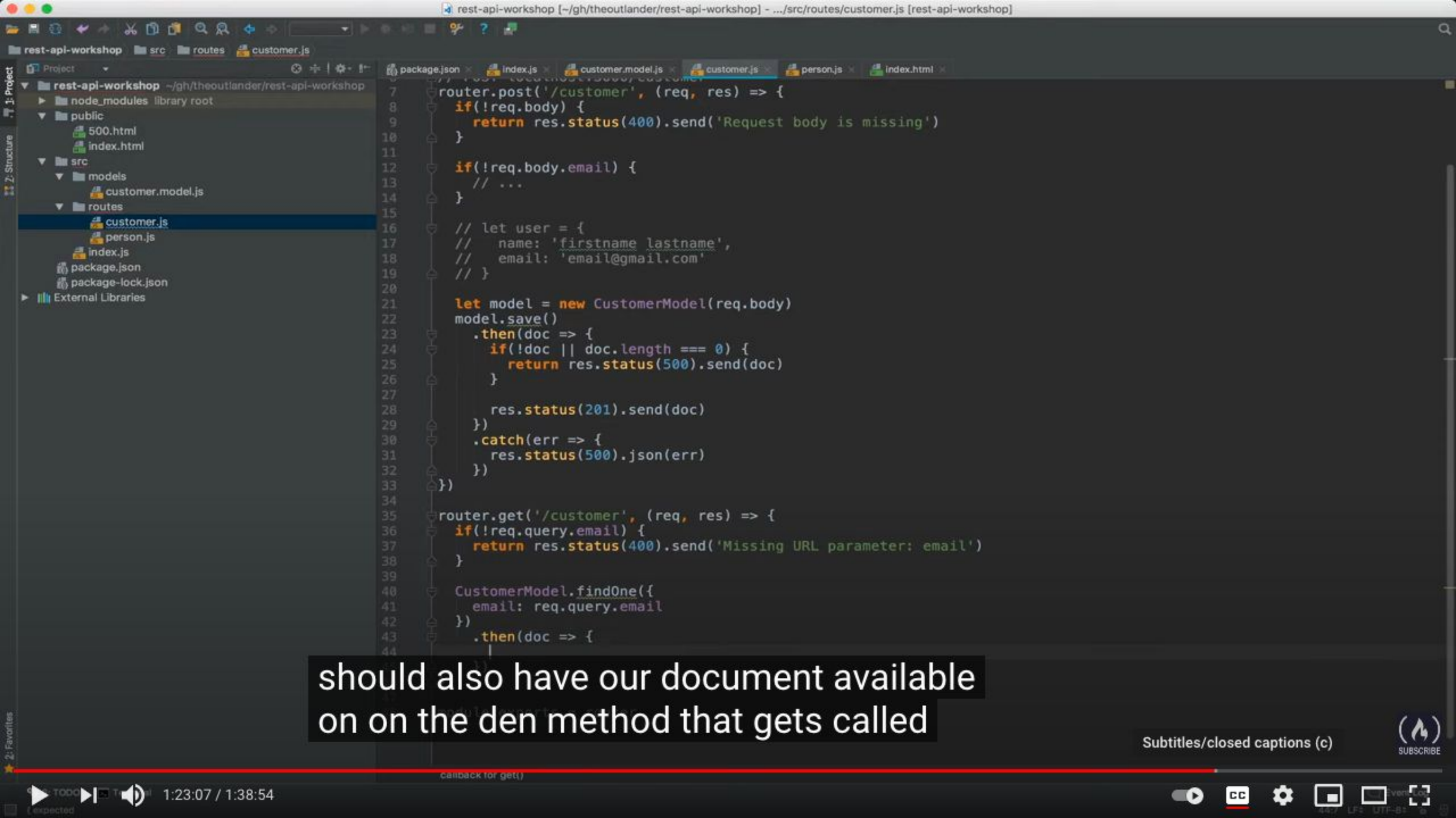

Insomnia – POST Customer
Insomnia
POST
localhost:3000/customer
Send
404 Not Found
TIME 21 ms
SIZE 22 B
No Environment
Cookies
Filter
REST API Workshop
POST POST Customer
GET GET Person
GET GET Person by QueryString
GET GET Person by Params
Projects
Body
Auth
Query
Header
Docs
Preview
Header 6
Cookie
Timeline
We think you are lost!
Select a body type from above
and it and the 404 handler the the
resource not found handler kicked in and
Subtitles/closed captions (c)
SUBSCRIBE
1:18:07 / 1:38:54

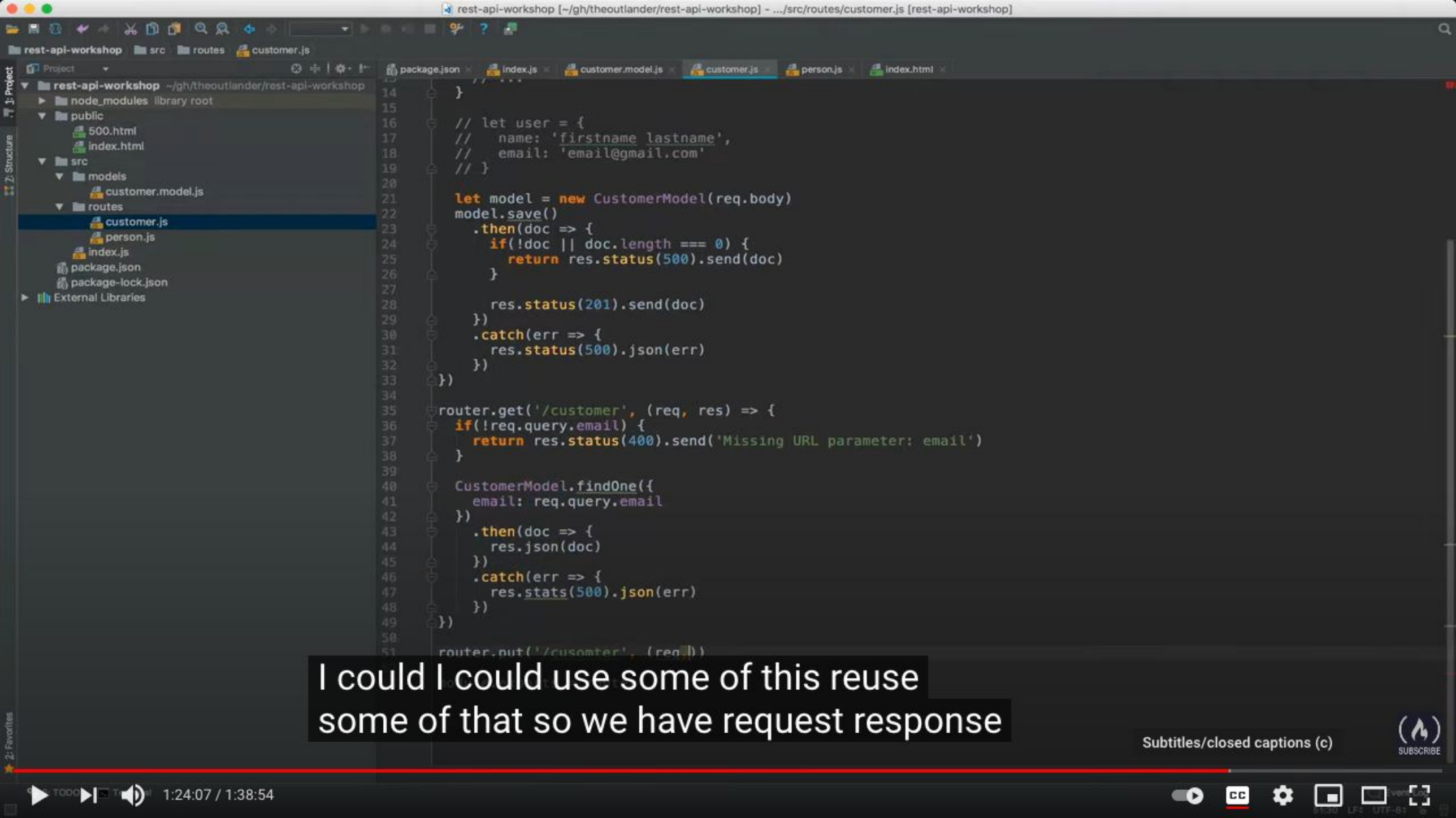

Insomnia – POST Customer
Insomnia
POST
localhost:3000/customer
Send
201 Created
TIME 116 ms
SIZE 94 B
No Environment
Cookies
Filter
REST API Workshop
POST POST Customer
GET GET Person
GET GET Person by QueryString
GET GET Person by Params
Projects
JSON
Auth
Query
Header 1
Docs
{
"name": "Thomas Anderson",
"email": "thomas@gmail.com"
}
Preview
Header 6
Cookie
Timeline
{
"_id": "5ab3f442f3d76b4ae4cb433b",
"name": "Thomas Anderson",
"email": "thomas@gmail.com",
"__v": 0
}
gmail.com and essentially I can when I
click send here it
Subtitles/closed captions (c)
SUBSCRIBE
1:19:07 / 1:38:54

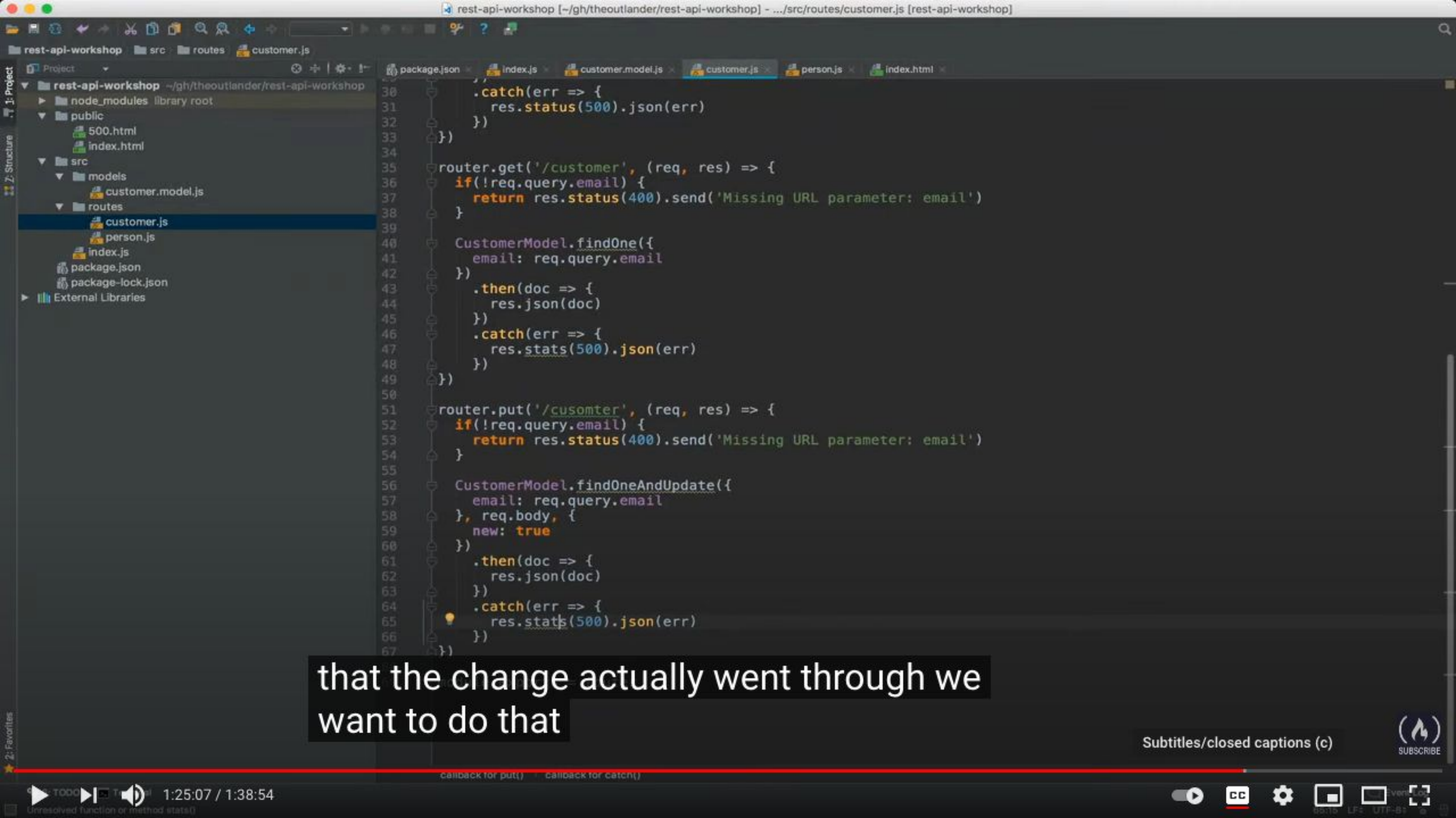

Robo 3T - 1.1
mLab (1)
rest-api-workshop
Collections (2)
System
customers
Functions
Users
db.getCollection('customers').find({})
db.getCollection('customers').find({})
mLab
ds221609.mlab.com:21609
rest-api-workshop
1 db.getCollection('customers').find({})
customers
0.089 sec.
0
50
Key
Value
Type
(1) ObjectId("5ab3f442f3d76b4ae4cb433b")
{ 4 fields }
Object
_id
ObjectId("5ab3f442f3d76b4ae4cb433b")
ObjectId
name
Thomas Anderson
String
email
thomas@gmail.com
String
__v
0
Int32
remember a few things at this point but
that is okay you know a little
Subtitles/closed captions (c)
SUBSCRIBE
1:20:07 / 1:38:54

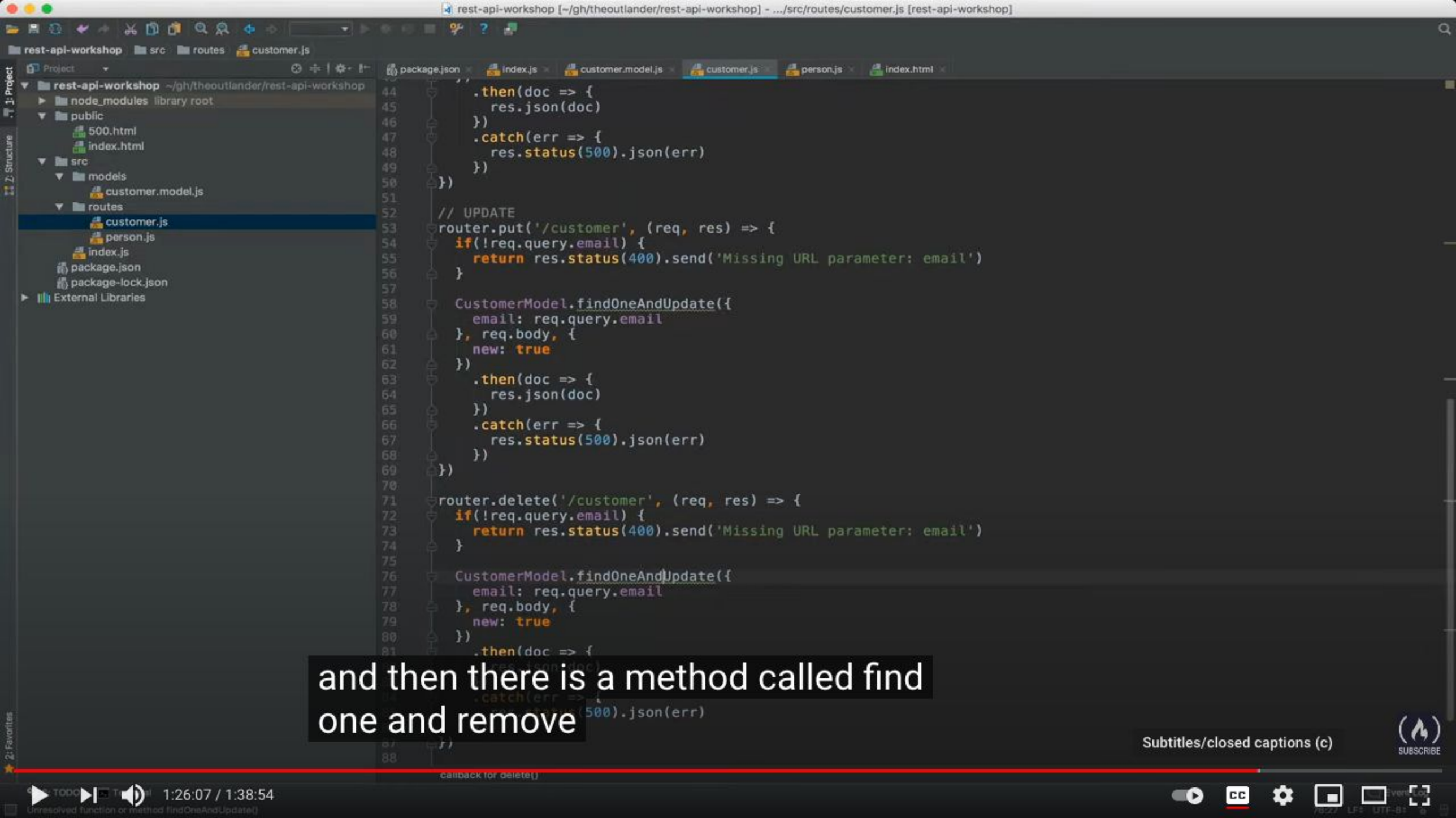

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop
src
routes
customer.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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)
})
})
ro
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 ch
[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' }
going to next add the ability to retrieve
Subtitles/closed captions (c)
SUBSCRIBE
1:21:07 / 1:38:54

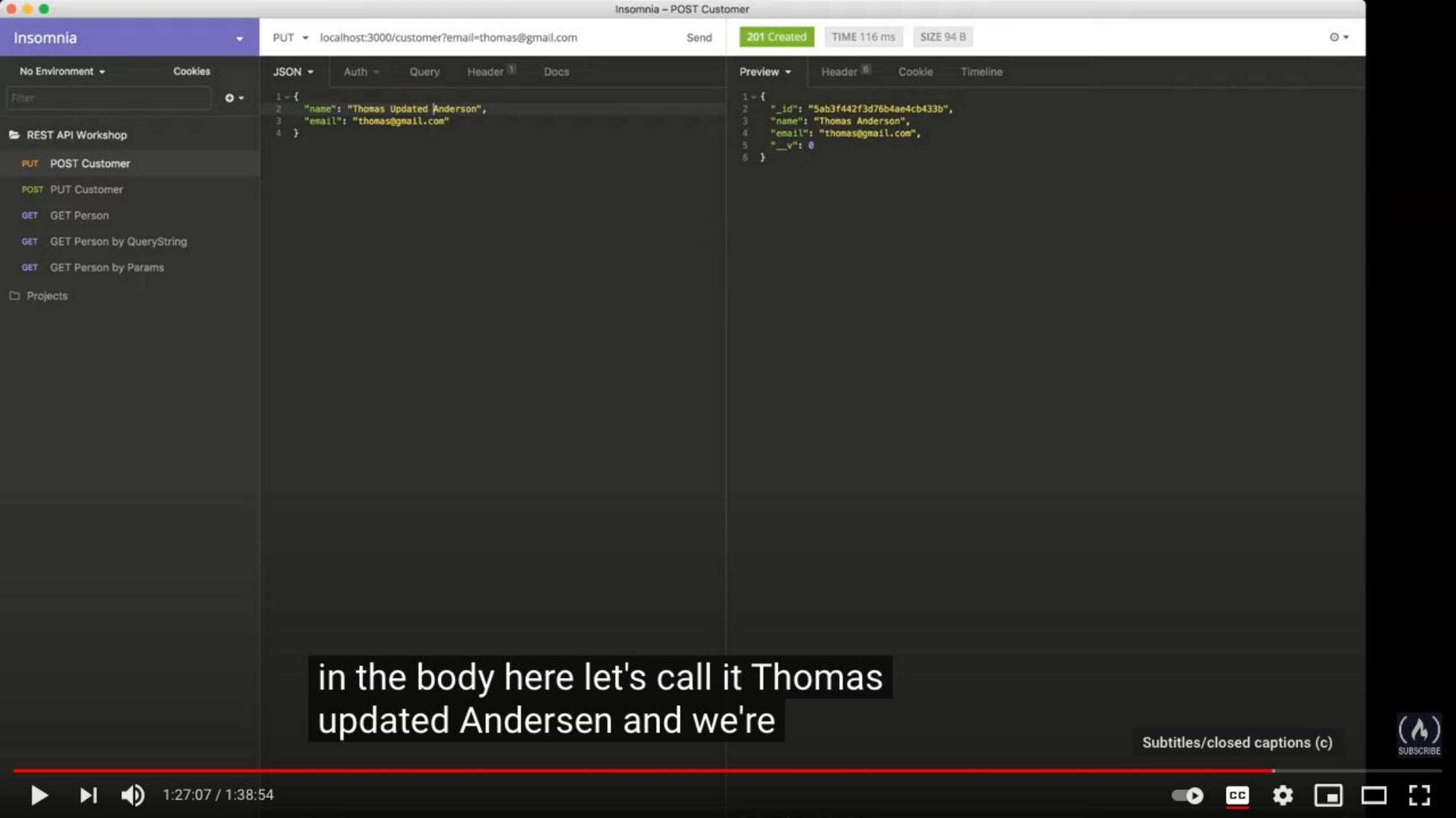

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop
src
routes
customer.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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')
}
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.get('/customer', (req, res) => {
CustomerModel.findOne({
})
})
module.exports = router
but anyway in this case we're gonna use
find one and find one takes an object
Subtitles/closed captions (c)
SUBSCRIBE
callback for get()
1:22:07 / 1:38:54

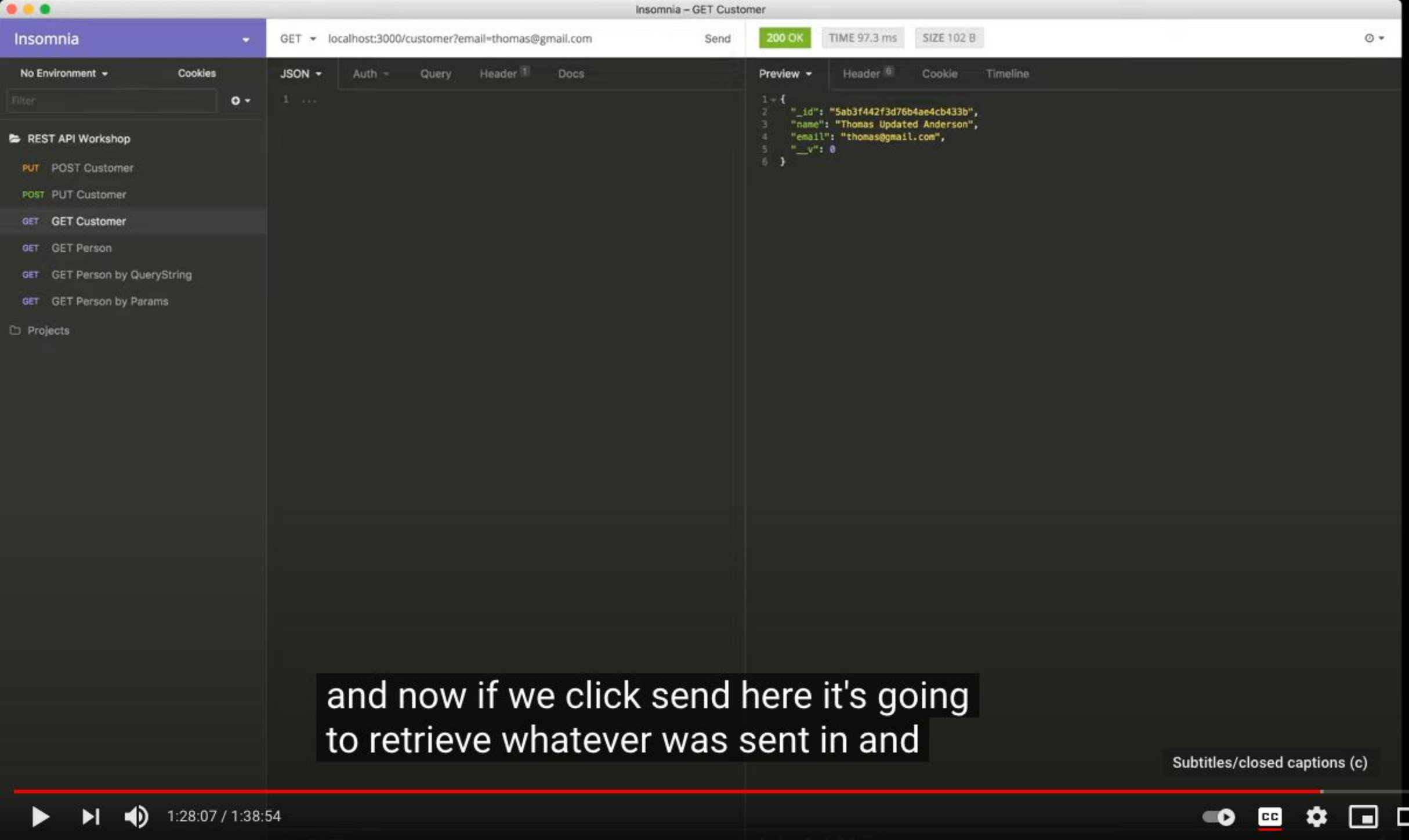

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
router.post('/customer', (req, res) => {
if(!req.body) {
return res.status(400).send('Request body is missing')
}
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.get('/customer', (req, res) => {
if(!req.query.email) {
return res.status(400).send('Missing URL parameter: email')
}
CustomerModel.findOne({
email: req.query.email
})
.then(doc => {
should also have our document available
on on the den method that gets called
Subtitles/closed captions (c)
SUBSCRIBE
1:23:07 / 1:38:54

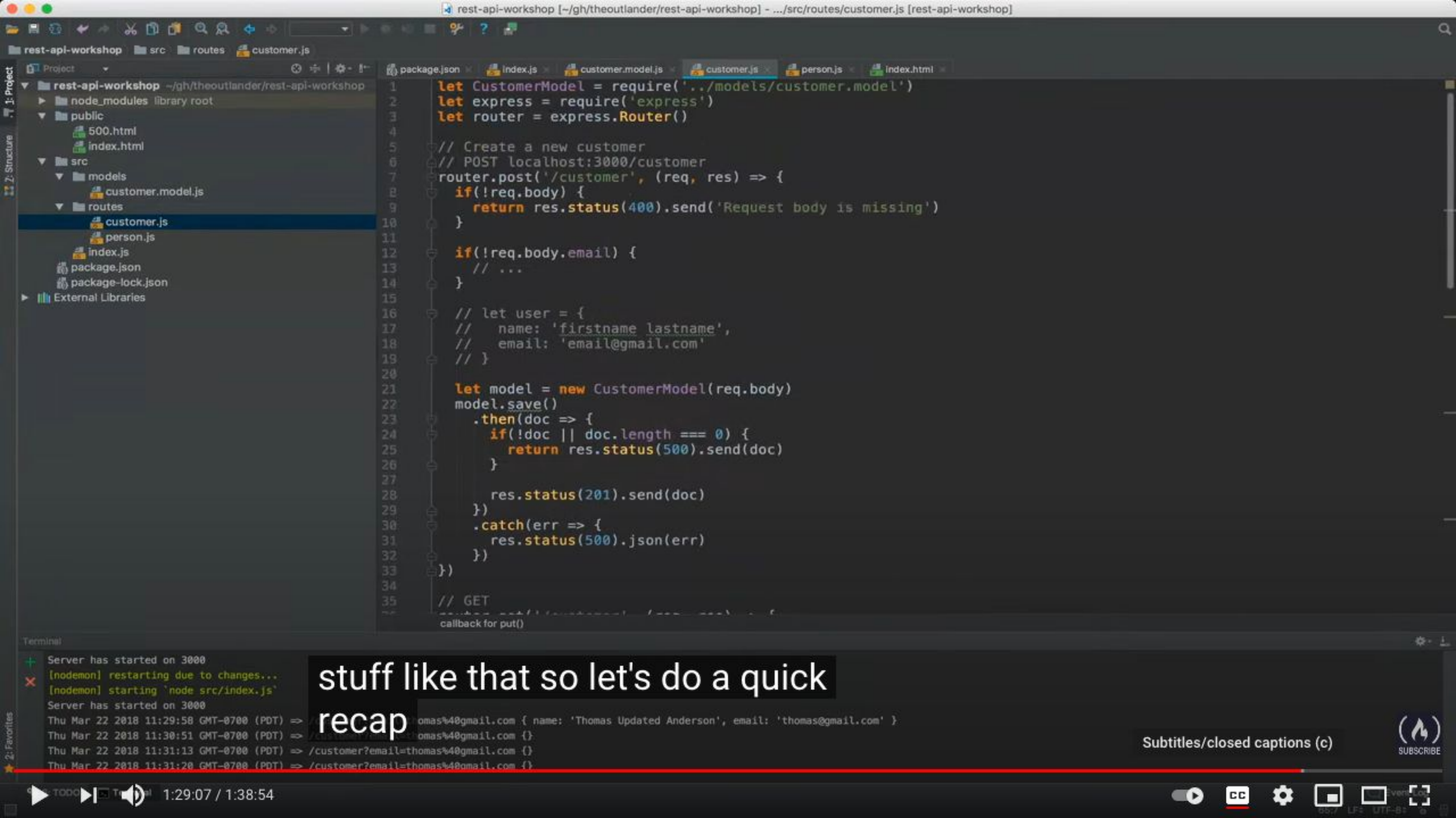

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
// 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.get('/customer', (req, res) => {
if(!req.query.email) {
return res.status(400).send('Missing URL parameter: email')
}
CustomerModel.findOne({
email: req.query.email
})
.then(doc => {
res.json(doc)
})
.catch(err => {
res.stats(500).json(err)
})
})
I could I could use some of this reuse
some of that so we have request response
Subtitles/closed captions (c)
SUBSCRIBE
1:24:07 / 1:38:54

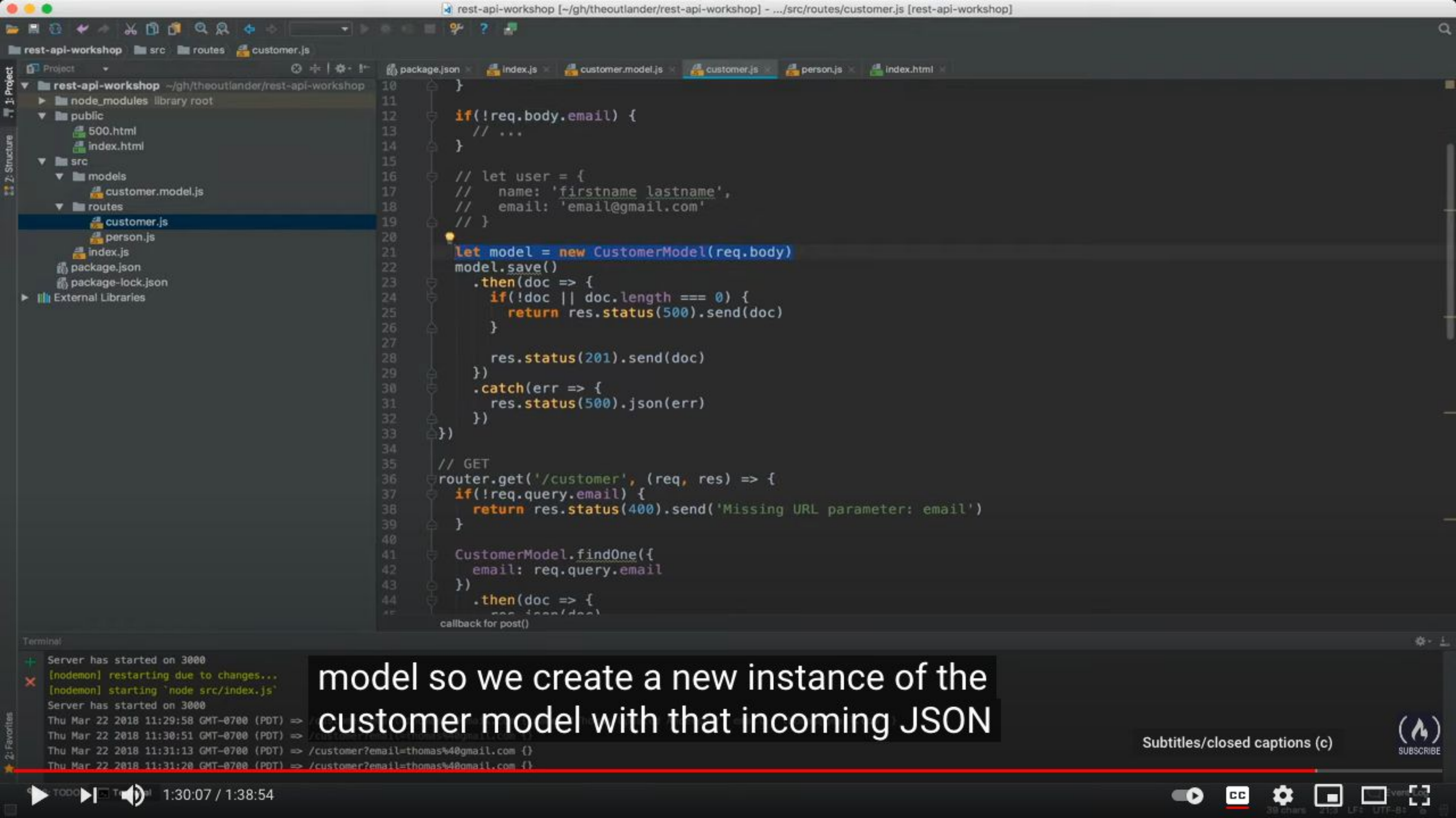

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop
src
routes
customer.js
package.json
index.js
customer.model.js
customer.js
person.js
index.html
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
.catch(err => {
res.status(500).json(err)
})
})
router.get('/customer', (req, res) => {
if(!req.query.email) {
return res.status(400).send('Missing URL parameter: email')
}
CustomerModel.findOne({
email: req.query.email
})
.then(doc => {
res.json(doc)
})
.catch(err => {
res.stats(500).json(err)
})
})
router.put('/cusomter', (req, res) => {
if(!req.query.email) {
return res.status(400).send('Missing URL parameter: email')
}
CustomerModel.findOneAndUpdate({
email: req.query.email
}, req.body, {
new: true
})
.then(doc => {
res.json(doc)
})
.catch(err => {
res.stats(500).json(err)
})
})
that the change actually went through we want to do that
Subtitles/closed captions (c)
SUBSCRIBE
callback for put() callback for catch()
1:25:07 / 1:38:54

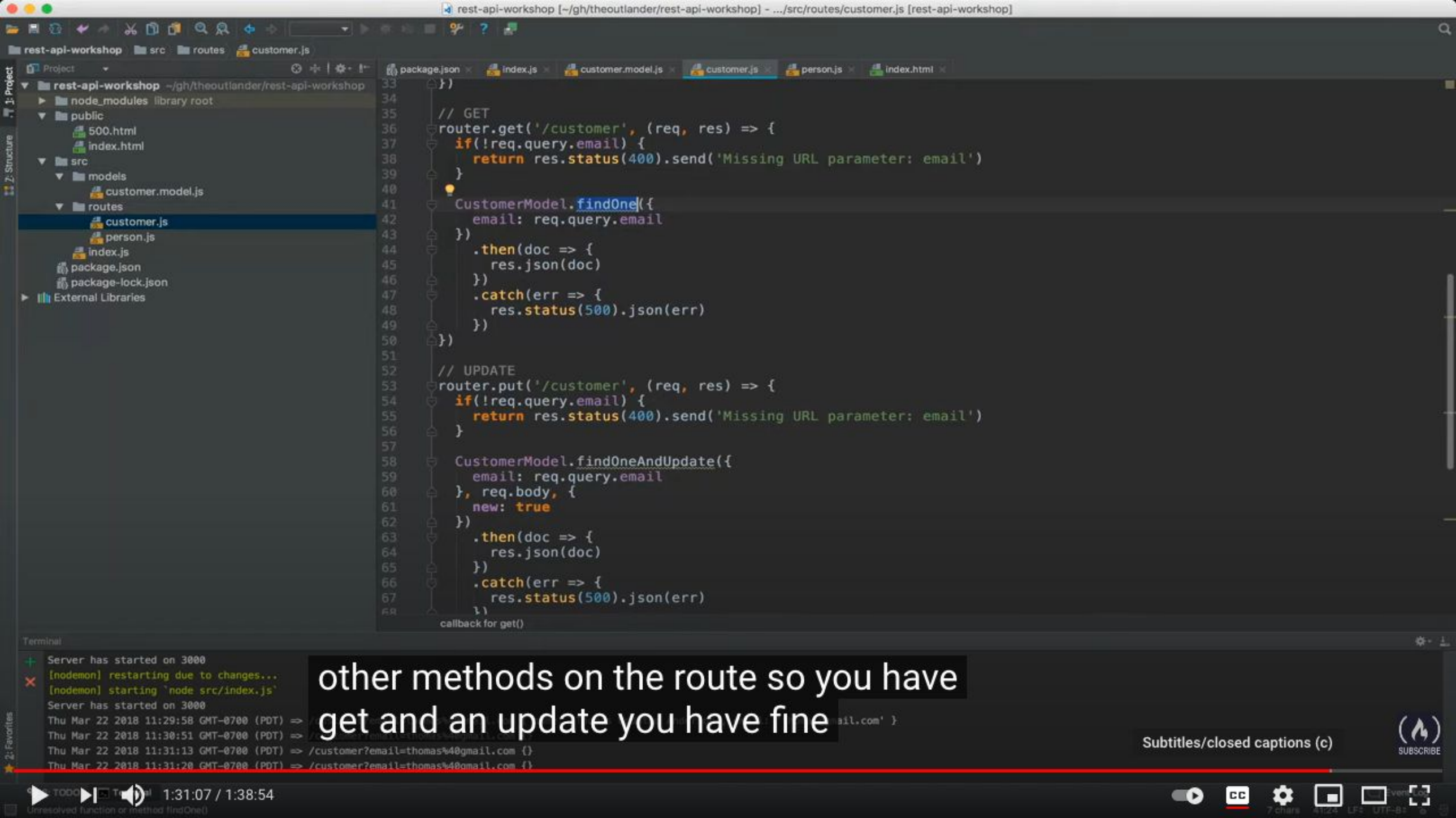

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop
src
routes
customer.js
package.json
index.js
customer.model.js
customer.js
person.js
index.html
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
.then(doc => {
res.json(doc)
})
.catch(err => {
res.status(500).json(err)
})
})
// UPDATE
router.put('/customer', (req, res) => {
if(!req.query.email) {
return res.status(400).send('Missing URL parameter: email')
}
CustomerModel.findOneAndUpdate({
email: req.query.email
}, req.body, {
new: true
})
.then(doc => {
res.json(doc)
})
.catch(err => {
res.status(500).json(err)
})
})
router.delete('/customer', (req, res) => {
if(!req.query.email) {
return res.status(400).send('Missing URL parameter: email')
}
CustomerModel.findOneAndUpdate({
email: req.query.email
}, req.body, {
new: true
})
.then(doc => {
and then there is a method called find one and remove
Subtitles/closed captions (c)
SUBSCRIBE
callback for delete()
1:26:07 / 1:38:54

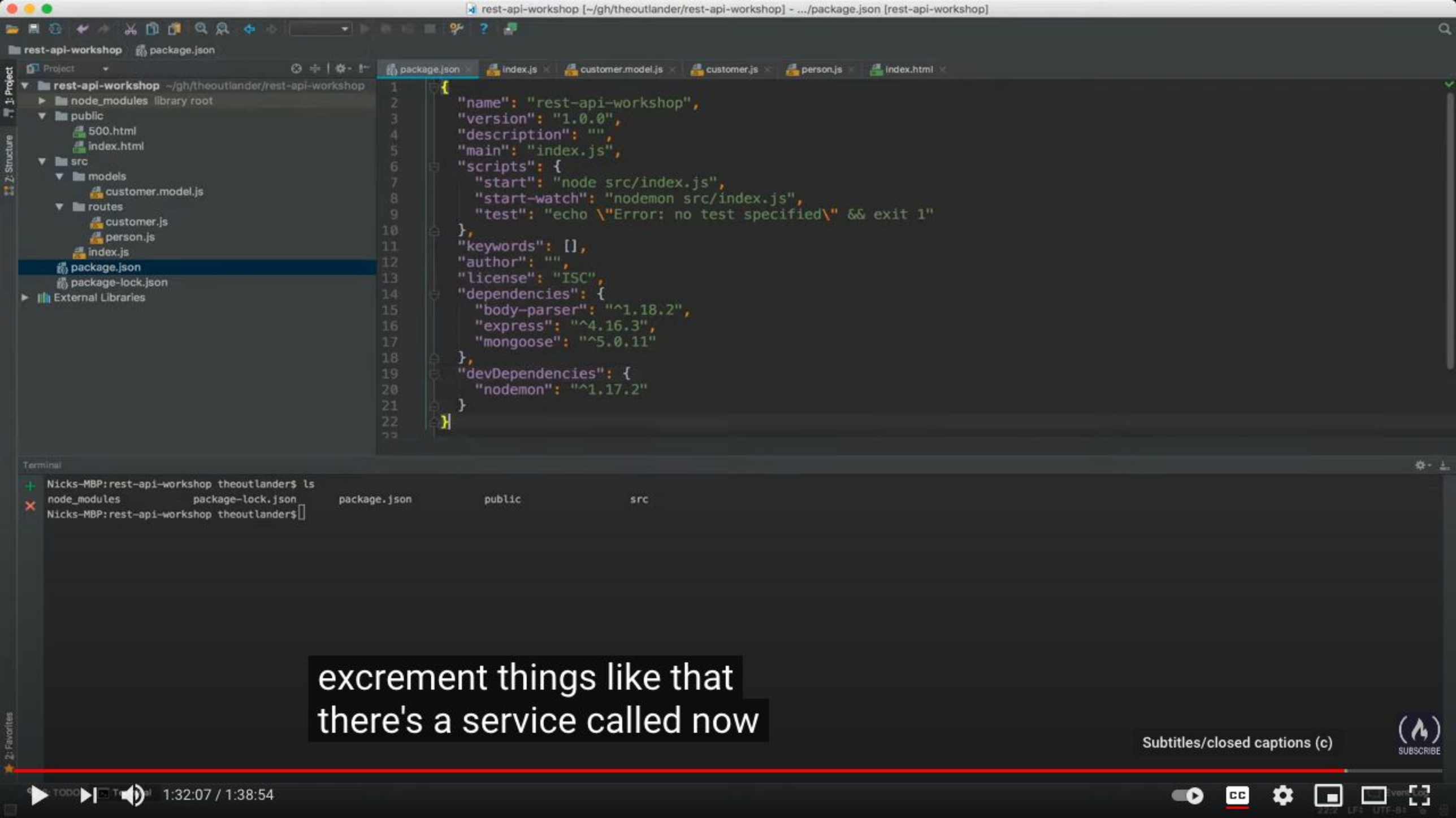

Insomnia – POST Customer
Insomnia
PUT
localhost:3000/customer?email=thomas@gmail.com
Send
201 Created
TIME 116 ms
SIZE 94 B
No Environment
Cookies
Filter
REST API Workshop
PUT POST Customer
POST PUT Customer
GET GET Person
GET GET Person by QueryString
GET GET Person by Params
Projects
JSON
Auth
Query
Header 1
Docs
{
"name": "Thomas Updated Anderson",
"email": "thomas@gmail.com"
}
Preview
Header 6
Cookie
Timeline
{
"_id": "5ab3f442f3d76b4ae4cb433b",
"name": "Thomas Anderson",
"email": "thomas@gmail.com",
"__v": 0
}
in the body here let's call it Thomas
updated Andersen and we're
Subtitles/closed captions (c)
SUBSCRIBE
1:27:07 / 1:38:54

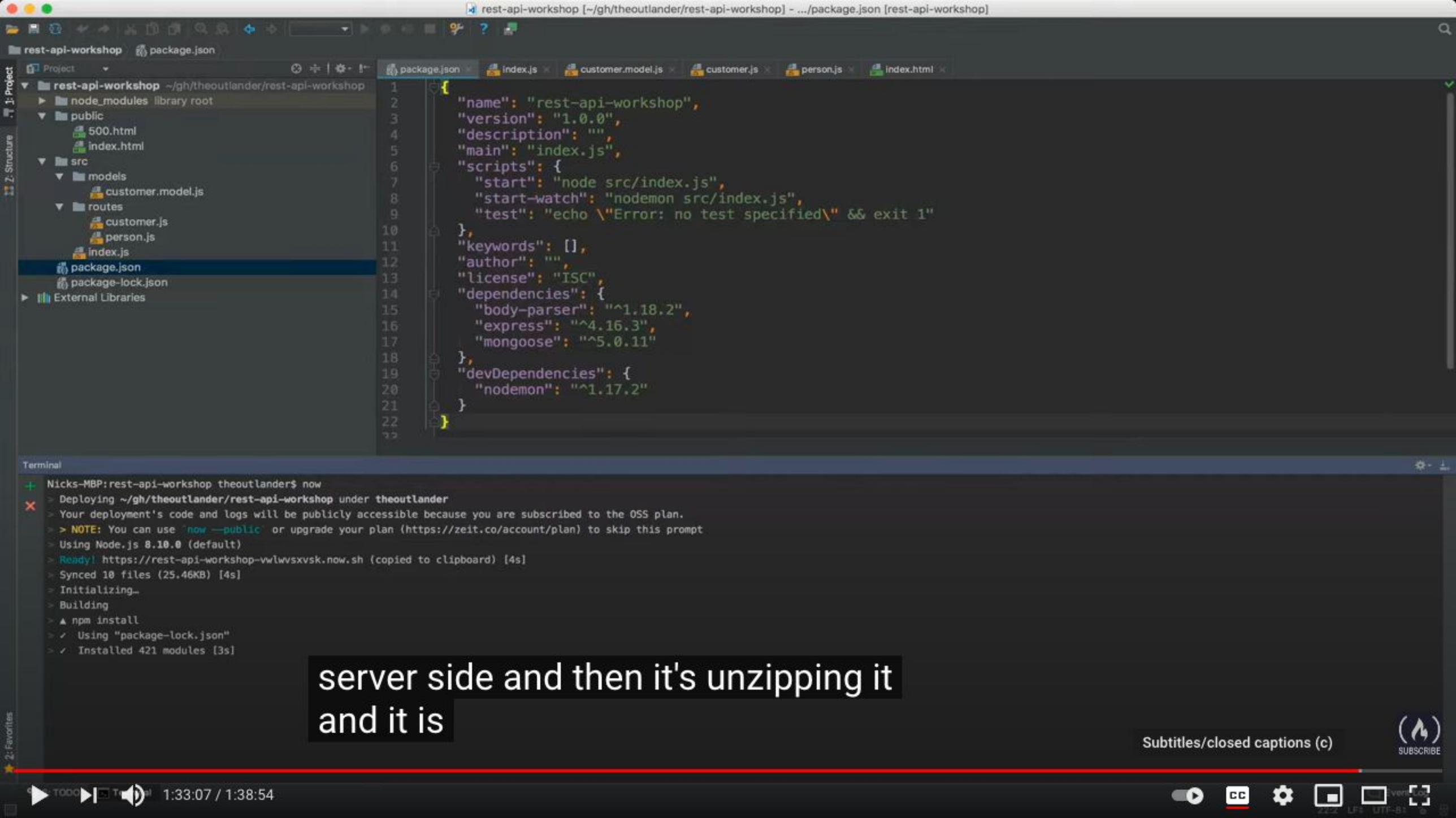

Insomnia – GET Customer
Insomnia
GET
localhost:3000/customer?email=thomas@gmail.com
Send
200 OK
TIME 97.3 ms
SIZE 102 B
No Environment
Cookies
Filter
REST API Workshop
PUT POST Customer
POST PUT Customer
GET GET Customer
GET GET Person
GET GET Person by QueryString
GET GET Person by Params
Projects
JSON
Auth
Query
Header 1
Docs
Preview
Header 6
Cookie
Timeline
{
"_id": "5ab3f442f3d76b4ae4cb433b",
"name": "Thomas Updated Anderson",
"email": "thomas@gmail.com",
"__v": 0
}
and now if we click send here it's going
to retrieve whatever was sent in and
Subtitles/closed captions (c)
SUBSCRIBE
1:28:07 / 1:38:54

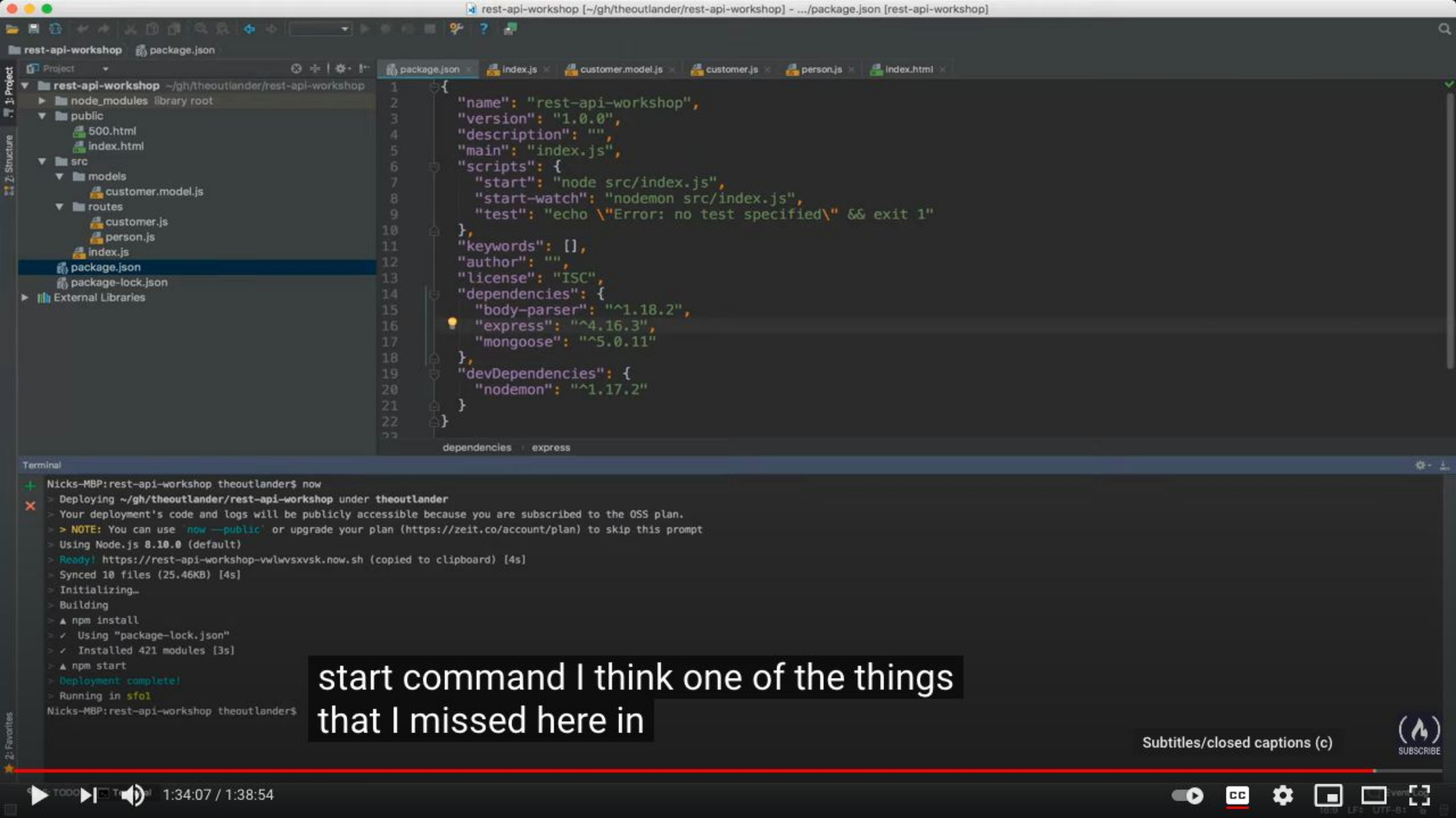

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop
src
routes
customer.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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')
}
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)
})
})
// GET
callback for put()
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) =>
omas%40gmail.com { name: 'Thomas Updated Anderson', email: 'thomas@gmail.com' }
Thu Mar 22 2018 11:30:51 GMT-0700 (PDT) =>
omas%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 {}
stuff like that so let's do a quick recap
Subtitles/closed captions (c)
SUBSCRIBE
1:29:07 / 1:38:54

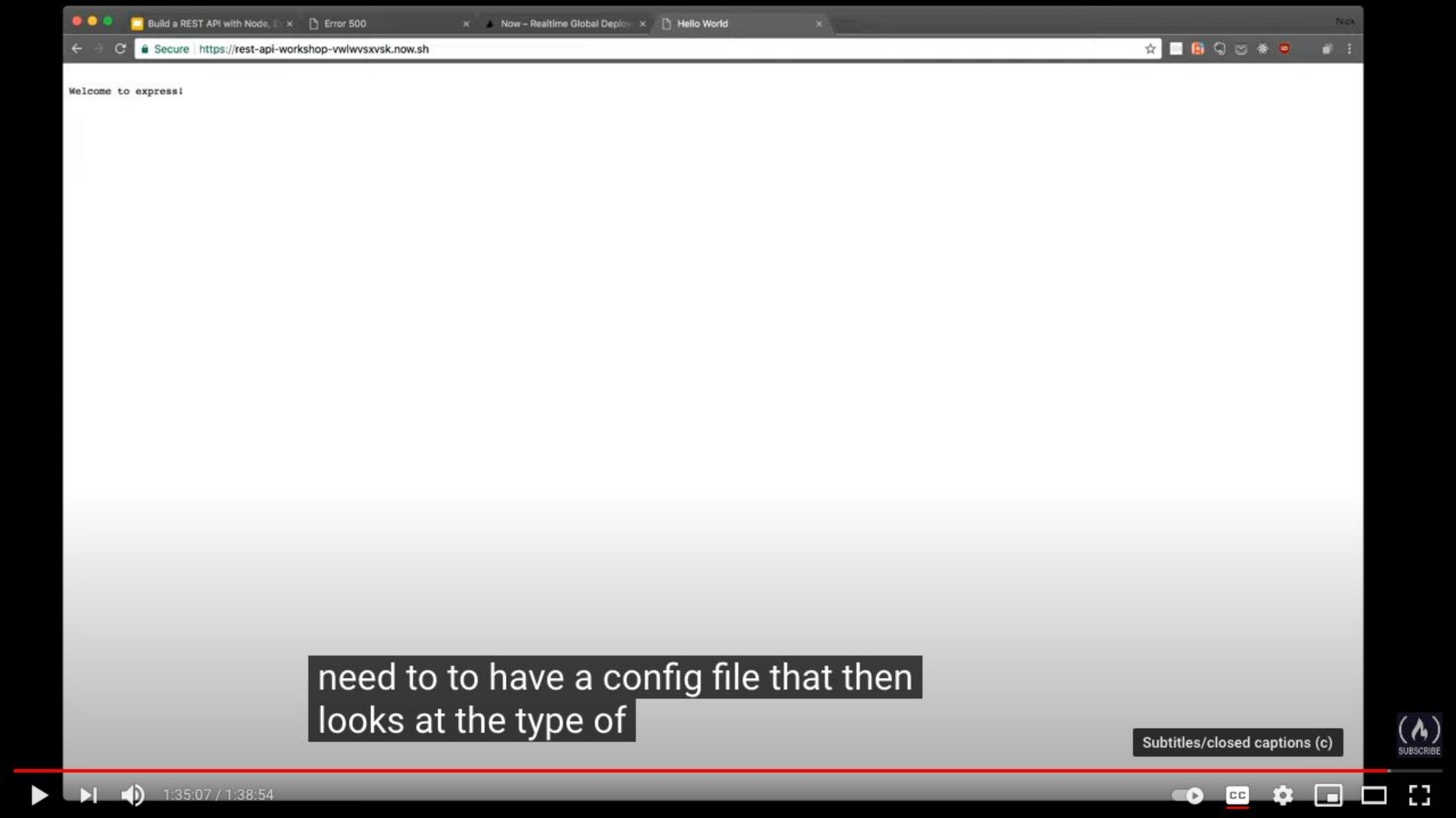

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop
src
routes
customer.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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 // 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 => {
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) =>
Thu Mar 22 2018 11:30:51 GMT-0700 (PDT) =>
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 {}
model so we create a new instance of the
customer model with that incoming JSON
Subtitles/closed captions (c)
SUBSCRIBE
1:30:07 / 1:38:54

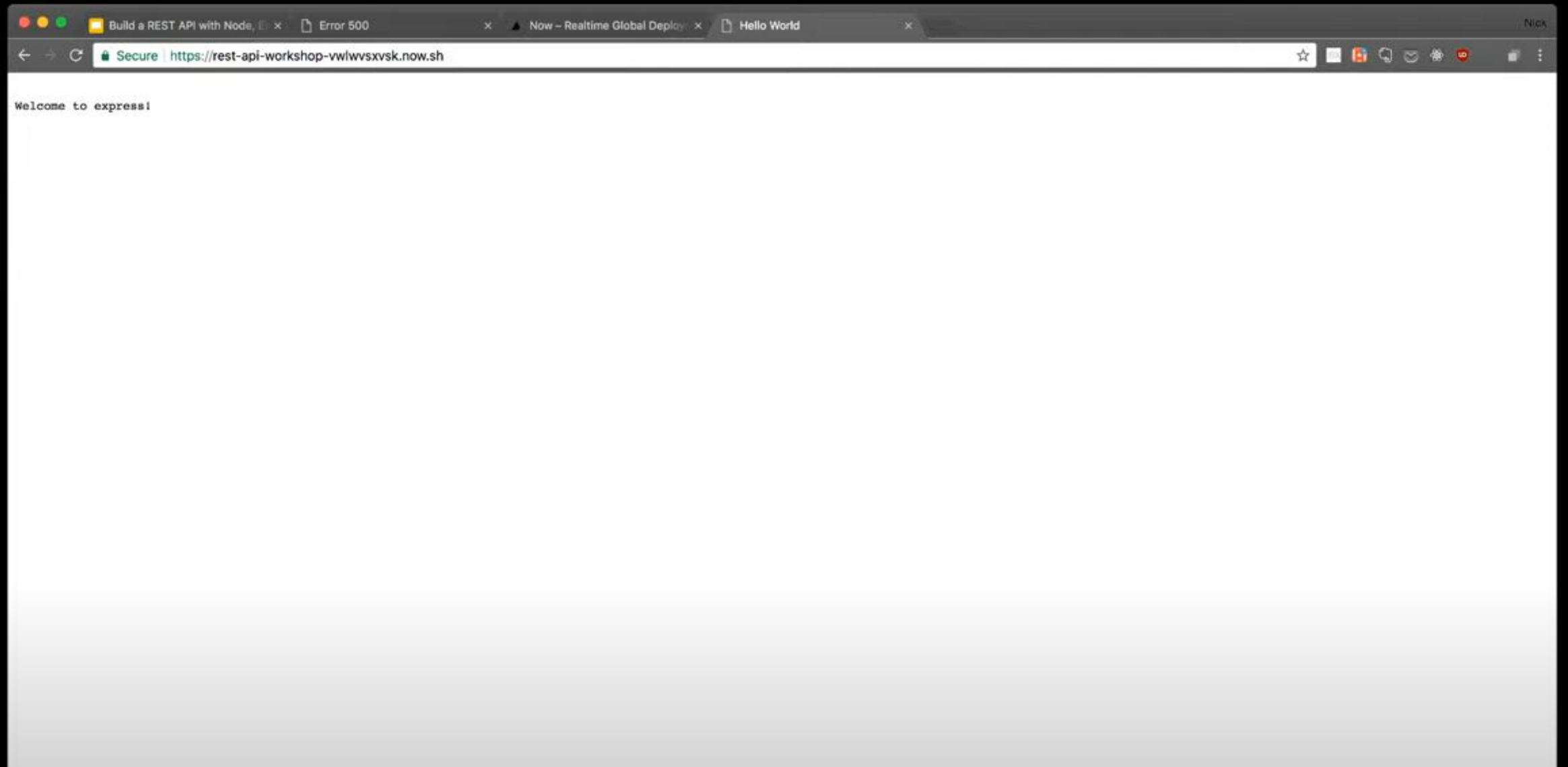

rest-api-workshop [~/gh/theoutlander/rest-api-workshop] - .../src/routes/customer.js [rest-api-workshop]
rest-api-workshop
src
routes
customer.js
Project
rest-api-workshop ~/gh/theoutlander/rest-api-workshop
node_modules library root
public
500.html
index.html
src
models
customer.model.js
routes
customer.js
person.js
index.js
package.json
package-lock.json
External Libraries
package.json
index.js
customer.model.js
customer.js
person.js
index.html
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)
callback for get()
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) =>
Thu Mar 22 2018 11:30:51 GMT-0700 (PDT) =>
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 {}
other methods on the route so you have
get and an update you have fine
Subtitles/closed captions (c)
SUBSCRIBE
1:31:07 / 1:38:54


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
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
index.js
customer.model.js
customer.js
person.js
index.html
{
"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"
}
}
Terminal
Nicks-MBP:rest-api-workshop theoutlander$ ls
node_modules package-lock.json package.json public src
Nicks-MBP:rest-api-workshop theoutlander$
excrement things like that
there's a service called now
Subtitles/closed captions (c)
SUBSCRIBE
1:32:07 / 1:38:54


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
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
index.js
customer.model.js
customer.js
person.js
index.html
{
"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"
}
}
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]
server side and then it's unzipping it
and it is
Subtitles/closed captions (c)
SUBSCRIBE
1:33:07 / 1:38:54


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
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
index.js
customer.model.js
customer.js
person.js
index.html
{
"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"
}
}
dependencies
express
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]
▲ npm start
Deployment complete!
Running in sfo1
Nicks-MBP:rest-api-workshop theoutlander$
start command I think one of the things
that I missed here in
Subtitles/closed captions (c)
SUBSCRIBE
1:34:07 / 1:38:54


Build a REST API with Node,
Error 500
Now – Realtime Global Deploy
Hello World
Nick
Secure
https://rest-api-workshop-vwlwvsxvsk.now.sh
Welcome to express!
need to to have a config file that then
looks at the type of
Subtitles/closed captions (c)
SUBSCRIBE
1:35:07 / 1:38:54




models out and add specific methods on
it to do more complex operations

Subtitles/closed captions (c)

SUBSCRIBE

1:36:07 / 1:38:54

Build a REST API with Node, | Error 500 | Now – Realtime Global Deploy | Hello World | Introduction to Mongoose for | How to build your own React b

Secure https://medium.freecodecamp.org/how-to-build-your-own-react-boilerplate-2f8cbbeb9b3f

freeCodeCamp(🔥) Writer

HOME DEV DESIGN DATA | LEARN TO CODE FOR FREE

Edit

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!*

people start out with using some existing boilerplate like create

## Git Repository: Setup

2.1K 5

Next story
Insights from Stack Overflow

Subtitles/closed captions (c)

SUBSCRIBE

1:37:07 / 1:38:54