Express.js (Node.js)

Studio Hyperdrive

Thomas Bormans
Jasper De Smet

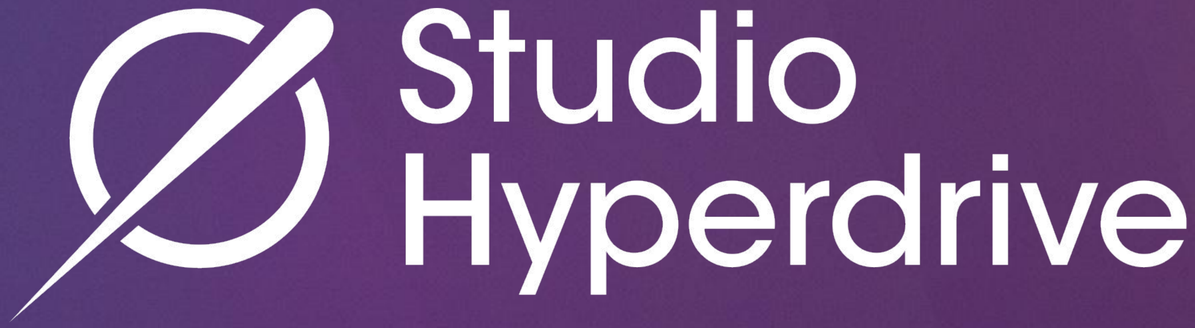
WHO ARE WE?

Thomas Bormans
@thomasbormans

Jasper De Smet
@jsprds

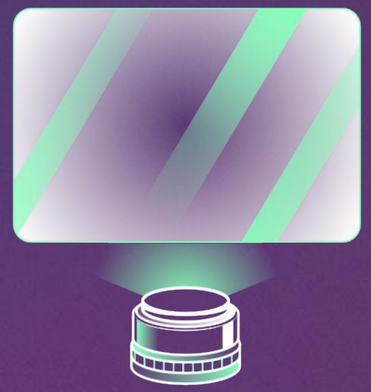
Studio Hyperdrive


PASSION LED US HERE


Studio
Hyperdrive

We ❤ JavaScript

Studio
Hyperdrive



Websites & Webapps

Angular, React, Vue, Node.js...

Studio Hyperdrive

Mobile Apps

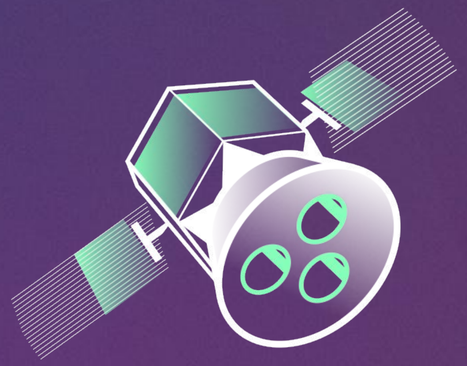
Progressive Webapps, Hybrid Apps (Cordova/Phonegap) and React Native

Studio Hyperdrive

Digital Experiences

WebGL, Css & Javascript Animations, WebXR, Chatbots

Studio
Hyperdrive



Business Engines

Node.js

Studio Hyperdrive

Consulting Services

Helping other teams!

Studio Hyperdrive

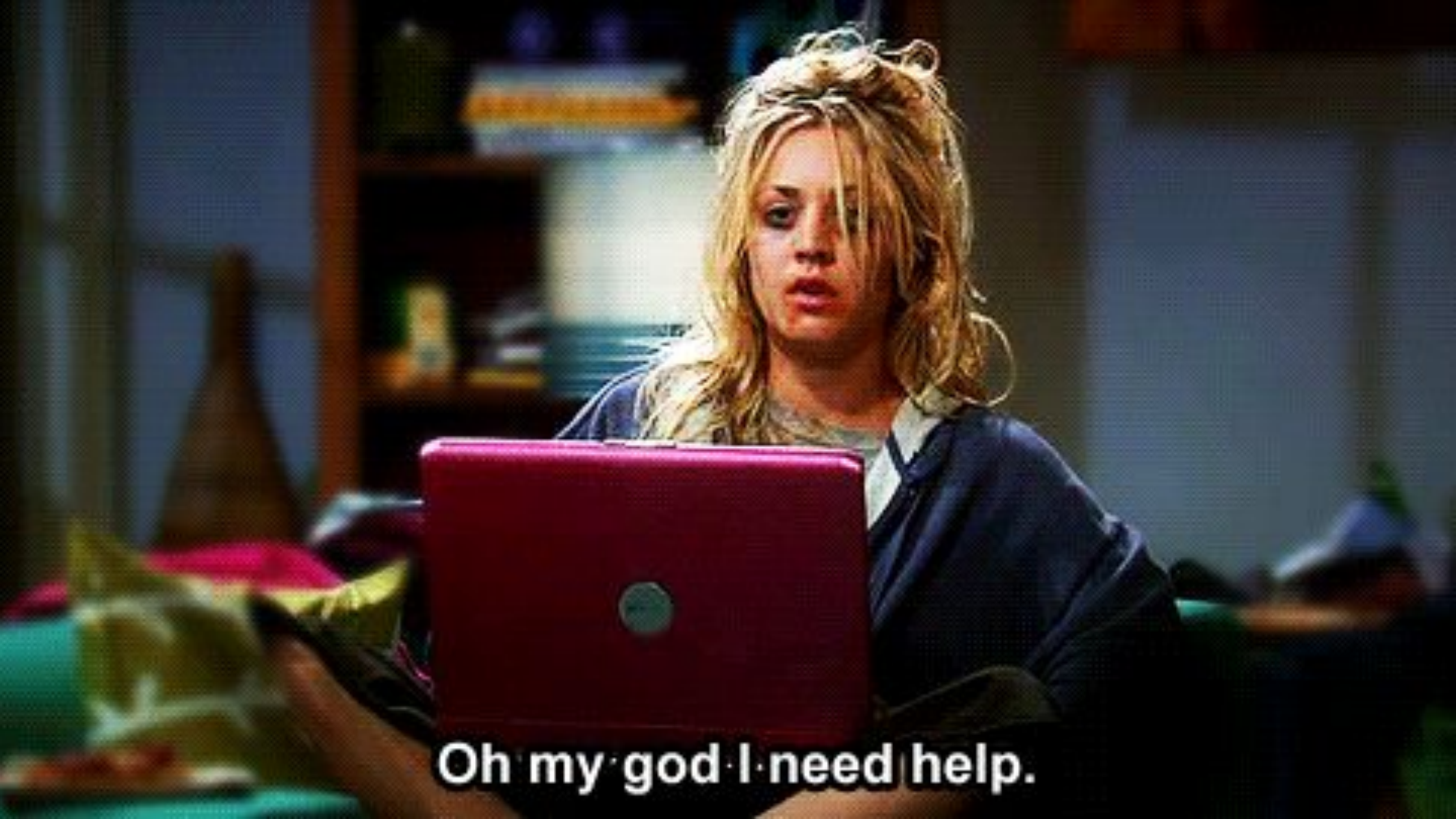
an introduction to Node.js and API design

Node.js

Express.js

REST API

Studio Hyperdrive


Oh my god I need help.

slides are (soon) available online
Github repository with (partial) solutions
questions? just ask!


NODE.JS
Studio
Hyperdrive

browser vs. server

PHP
.NET
Java
Python
...

Node.js vs. PHP

Studio Hyperdrive

Node.js server vs. PHP server

event driven
asynchronous
non-blocking
single-threaded
modular

low latency
chat apps
data/social streaming
online multiplayer
live-updates

LTS
latest
semver
nvm

LTS (10.15.1)

Studio
Hyperdrive


LET'S DO THIS.

the basics

Studio
Hyperdrive

Node.js

```
$ node --version
```

Node.js

```
$ node --version
v10.15.1
```

Node.js

```
$ node
```

Node.js

```
$ node
>
```

Node.js

```
$ node
> 1 + 1
```

Node.js

```
$ node
> 1 + 1
2
```

Node.js

```
> console.log('Hello World')
```

Node.js

```
> console.log('Hello World')
Hello World
undefined
```

exit/stuck?

ctrl + c (x2)

Node.js

```
>
(To exit, press ^C again or type .exit)
>
```

Use a file

Node.js

```
$ echo "console.log('Hello World')" >> hello.js
```

Node.js

```
$ echo "console.log('Hello World')" >> hello.js
$ cat hello.js
```

Node.js

```
$ echo "console.log('Hello World')" >> hello.js
$ cat hello.js
console.log('Hello World')
```

Node.js

```
$ echo "console.log('Hello World')" >> hello.js
$ cat hello.js
console.log('Hello World')
$ node hello.js
```

Node.js

```
$ echo "console.log('Hello World')" >> hello.js
$ cat hello.js
console.log('Hello World')
$ node hello.js
Hello World
```

visual studio code

handy tools

Studio Hyperdrive

https://code.visualstudio.com/


Studio
Hyperdrive

(a)sync

(a)sync

callback
promise
await

callback

```
console.log('1');
setTimeout(() => {
  console.log('2')
}, 500);
console.log('3');
```

callback

```
console.log('1');
setTimeout(() => console.log('2'), 500);
console.log('3');
```

callback

```
console.log('1');
setTimeout(() => console.log('2'), 500);
console.log('3');

1
3
2
```

promise

```
const test = () => {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      resolve('done');
    }, 500);
  });
};
```

promise

```
test()
  .then(() => {
    // Success
  })
  .catch((err) => {
    // Error
  });
```

await

```
const test = () => {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      resolve('done');
    }, 500);
  });
};
```

await

```
const init = async () => {
  const result = await test();
  console.log(result);
};
init();
```

modules (= multiple files)

index.js

logger.js

service.js

logger.js

```
module.exports = (message) => {
    console.log(message);
}
```

service.js

```
const logger = require('./logger');

module.exports.logHelloWorld = () => {
   logger('Hello World');
}
```

index.js

```
const service = require('./service');

service.logHelloWorld();
```

index.js

```
const { logHelloWorld } = require('./service');

logHelloWorld();
```

result

```
$ node index.js
Hello World
```

your turn

```
$ node index.js
Hello World
Hello Student
```

service.js

```
const logger = require('./logger');

…

module.exports.logHelloStudent = () => {
   logger('Hello Student');
}
```

index.js

```
const { logHelloWorld } = require('./service');
const { logHelloStudent } = require('./service');

logHelloWorld();
logHelloStudent();
```

index.js

```
const { logHelloWorld, logHelloStudent } = require('./service');

logHelloWorld();
logHelloStudent();
```

folders

Studio
Hyperdrive

folders

```
- services
  |
  |- service.js
```

folders

```
const service = require('./services/service');

service.logHelloWorld();
```

folders

```
- services
  |
  |- index.js
```

folders

```
const service = require('./services);

service.logHelloWorld();
```

Node.js ecosystem

package manager
command line interface (CLI)

npm

```
$ npm --version
```

npm

```
$ npm --version
6.4.1
```

npm

```
$ npm init
```

npm

```
$ npm init
This utility will walk you through...

See `npm help json` ...

Use `npm install <pkg>`....

Press ^C at any time to quit.
package name: (my-app)
```

npm

```
package name: (my-app)
version: (1.0.0)
```

npm

```
package name: (my-app)
version: (1.0.0)
description:
```

npm

```
package name: (my-app)
version: (1.0.0)
description:
entry point: (index.js)
```

npm

```
package name: (my-app)
version: (1.0.0)
description:
entry point: (index.js)
test command:
```

npm

```
package name: (my-app)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
```

npm

```
package name: (my-app)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
```

npm

```
package name: (my-app)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
```

npm

```
package name: (my-app)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
```

npm

```
{
  "name": "test",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

main file

Studio
Hyperdrive

npm

```
{
    ...
    "main": "index.js",
   ...
}
```

npm

```
$ node index.js
...

$ node .
...
```

scripts

npm

```
{
    ...
    "scripts": {
        "start": "...",
        "test": "...",
        "build": "...",
        "my-custom-script": "...",
    }
    ...
}
```

npm

```
$ npm start
…
$ npm build
…
$ npm test
…

$ npm run my-custom-script
…
$ npm run start
…
```

packages

npm

https://www.npmjs.com

2042 packages found

Sort Packages

Optimal

Popularity

Quality

Maintenance

Who's Hiring?

Tinder, **Red Badger**, **Airtable** and lots of other companies are hiring javascript developers.

See all 19 companies

passport

exact match

Simple, unobtrusive authentication for Node.js.

express connect auth authn authentication

jaredhanson published 0.4.0 • 2 years ago

passport-strategy

An abstract class implementing Passport's strategy API.

passport strategy

jaredhanson published 1.0.0 • 6 years ago

passport-jwt

Passport authentication strategy using JSON Web Tokens

Passport Strategy JSON Web Token JWT

themikenicholson published 4.0.0 • a year ago

passport-github

GitHub authentication strategy for Passport.

passport github auth authn authentication identity

jaredhanson published 1.1.0 • 3 years ago

passport-oauth1

OAuth 1.0 authentication strategy for Passport.

passport auth authn authentication authz authorization oauth

npm

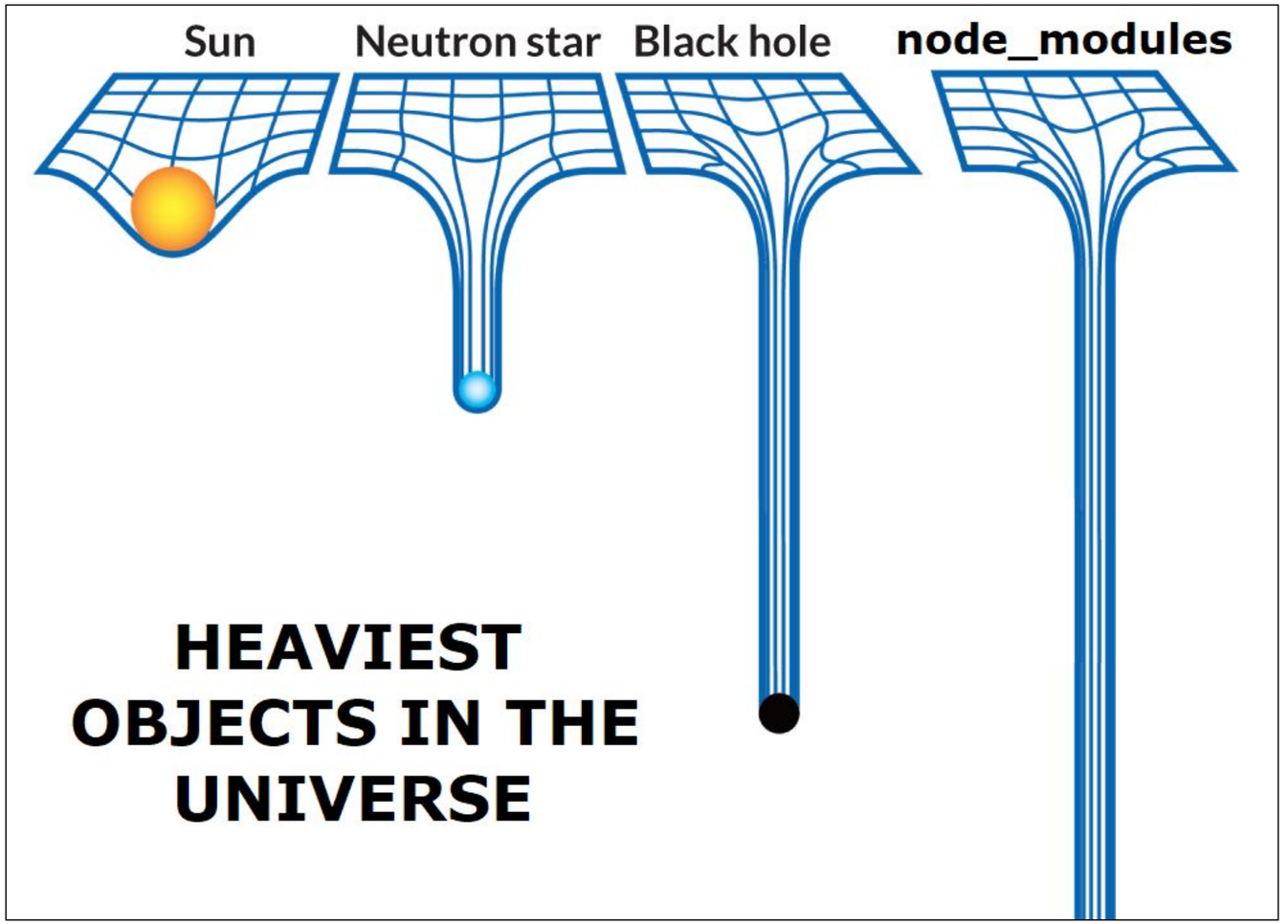
```
$ npm install [package-name]
$ npm install --save [package-name]
$ npm install --save-dev [package-name]
$ npm install --global [package-name]
```

npm

package.json vs. package-lock.json

npm

node_modules


Sun
Neutron star
Black hole
node_modules
HEAVIEST OBJECTS IN THE UNIVERSE

Express.js

Studio Hyperdrive

Express.js

Fast, unopinionated, minimalist web framework for Node.js

Express.js

Feathers, ItemsAPI, KeystoneJS, Kraken, LoopBack, MEAN, Sails, Hydra-Express, Blueprint, Locomotive, graphql-yoga, Express Gateway, Dinoloop, Kites, FoalTS, …

Express.js

koa

next generation web framework for node.js

Studio Hyperdrive

Express.js basics

Studio Hyperdrive

Express.js basics

webserver
routes
middleware

webserver

Studio Hyperdrive

webserver

apache, nginx, … nodejs

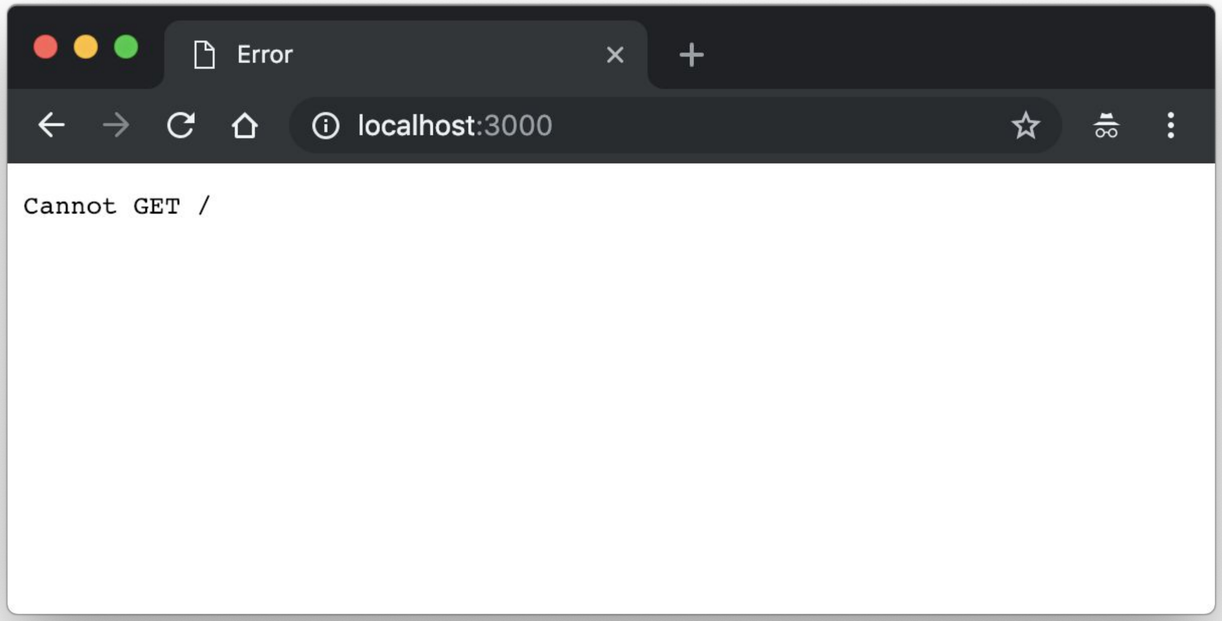
webserver

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

// config
const port = 3000;

app.listen(port, () => {
  console.log(`Example app listening on port ${port}!`)
});
```

webserver

```
$ node .
Example app listening on port 3000!
```

webserver



nodemon

handy tools

Studio Hyperdrive

nodemon

```
$ npm i --save-dev nodemon
+ nodemon@1.18.10
added 285 packages … in 7.83s
```

nodemon

```
...
"scripts": {
    "start": "./node_modules/.bin/nodemon",
    "test": "echo \"Error: no test specified\" && exit 1"
}
...
```

nodemon

./node_modules/**.bin**/nodemon

nodemon

```
...
"scripts": {
    "start": "nodemon",
    "test": "echo \"Error: no test specified\" && exit 1"
}
...
```

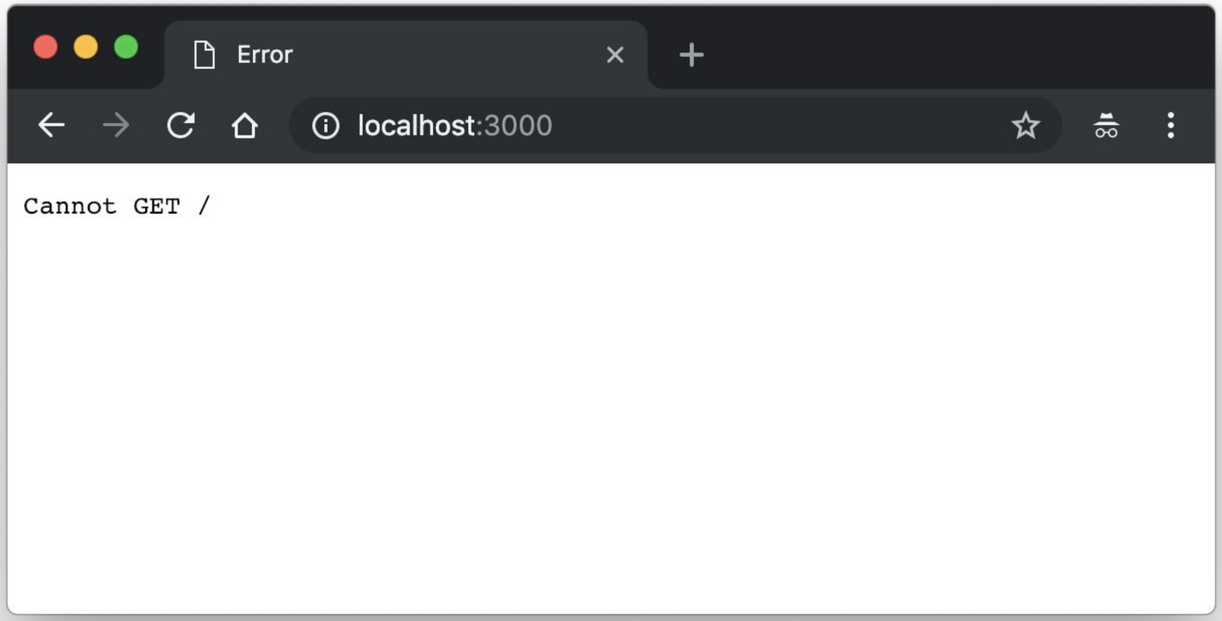
nodemon

```
$ npm start
[nodemon] 1.18.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node index.js`
Example app listening on port 3000!
```

nodemon

```
...
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Example app listening on port 3000!
```

nodemon



routes

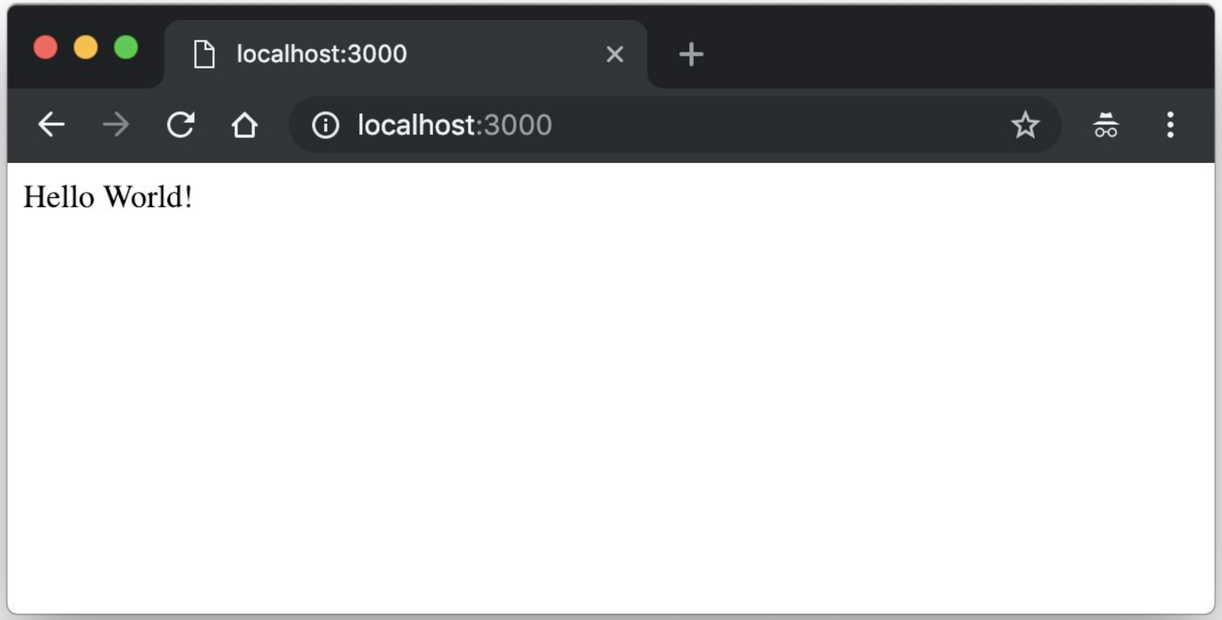
routes

```
app.get('/', (req, res) => {
  res.send('Hello World!')
});
```

routes

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

// config
const port = 3000;

app.get('/', (req, res) => {
    res.send('Hello World!')
});

app.listen(port, () => {
    console.log(`Example app listening on port ${port}!`)
});
```

nodemon



routes

```
(req, res) => {}
```

middleware

middleware

validation
authentication
parsing
mapping
fetching data for next part

middleware

```
const logger = (req, res, next) => {
  // Do something
  console.log('Hello world');
  next();
};
```

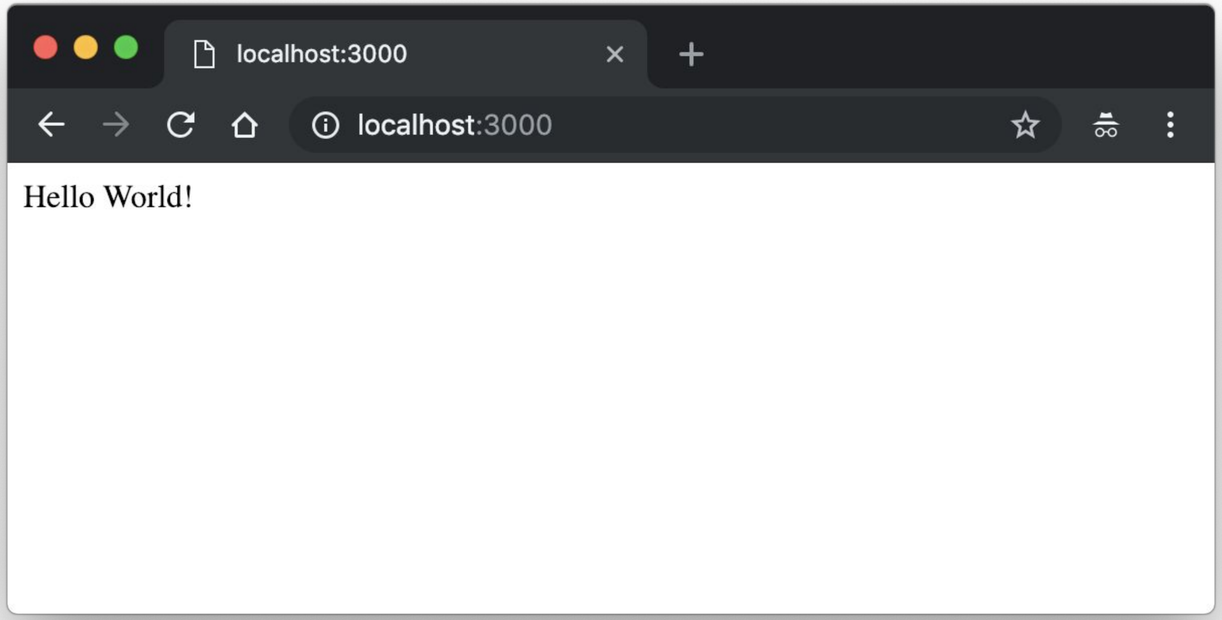
middleware

```
const logger = (req, res, next) => {
    console.log('Hello world');
    next();
};
app.get('/', logger, (req, res) => {
    res.send('Hello World!')
});
```

nodemon

```
…
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Example app listening on port 3000!
Hello world
```

nodemon



routes

```
(req, res, next) => {}
```

Express.js structure

Studio Hyperdrive

Express.js structure

helpers
services
controllers
templates

helpers

helpers

small & internally:
parse data
map data
format data

helpers

```
module.exports = (message) => {
    console.log(message);
}
```

services

Studio
Hyperdrive

services

connection with external services:
authentication with API
fetch data from API

services

```
const rp = require('request-promise');

module.exports.fetchAll = () => {
   return rp('https://google.com');
}
```

controllers

controllers

connection between routes and services/helpers
gets everything from anywhere and returns it to the client

controllers

```
module.exports.hello = (req, res) => {
  res.send('Hello World!')
};
```

controllers

```
const controller = require('./my-controller');
app.get('/', controller.hello);
```

template engines

template engines = views

views

template engines
pug
mustach
ejs

views

```
app.set("view engine", "ejs");
```

views

```
|...
|-middleware:
|---...
|-routes:
|---...
|-views:
|---index.ejs
|-app.js
```

views

```
app.set("views", "/views"));
```

views

```
res.render("index", {
  name: "Bob Ross"
});
```

```
<html>
  <body>
    <h1>Hello {{name}}!</h1>
  </body>
</html>
```

one-"click"-install

Studio
Hyperdrive

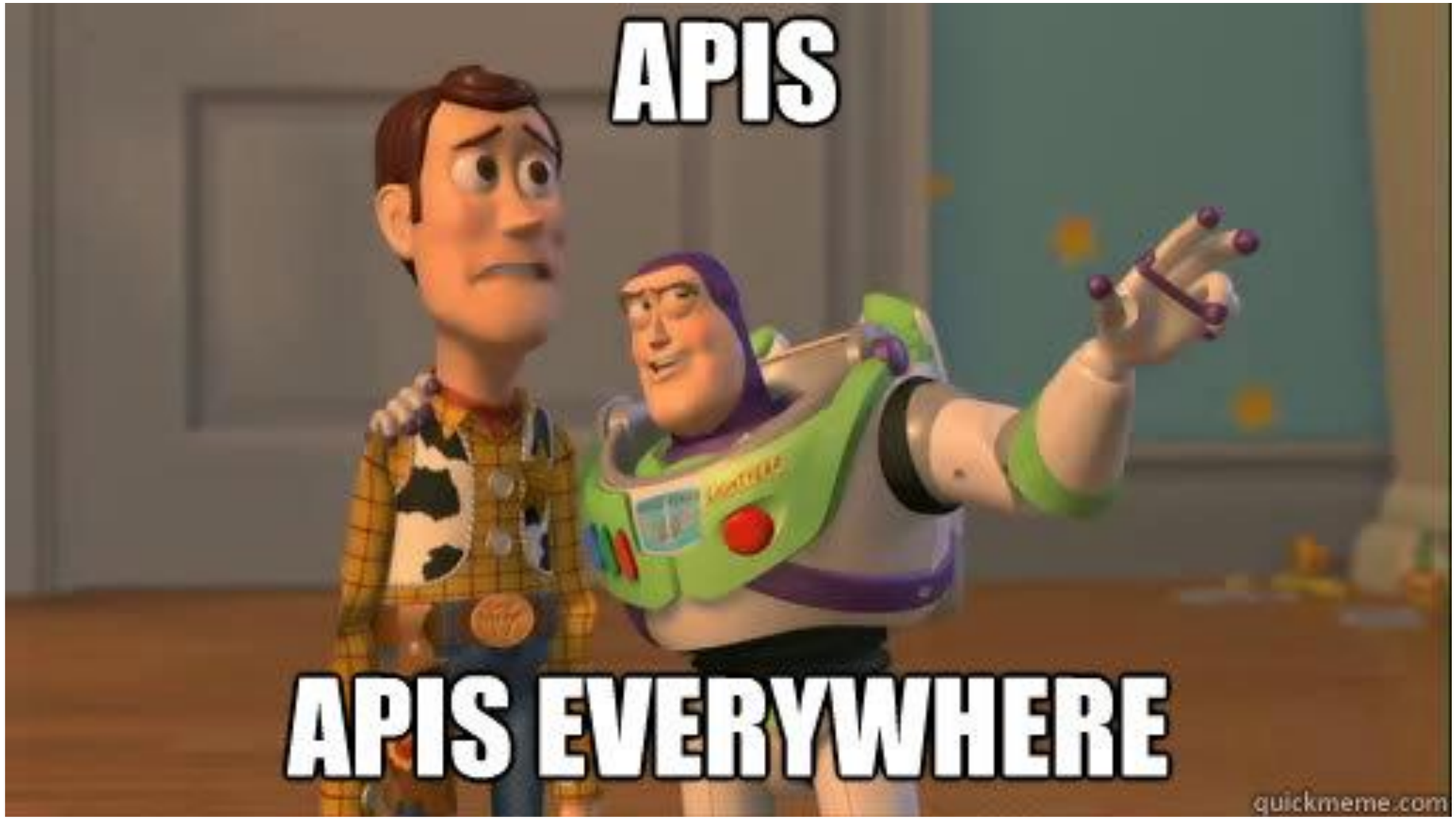
one-"click"-install

```
$ npm install express-generator --save-dev
$ ./node_modules/.bin/express --ejs
$ node app.js
```

one-"click"-install

basic Node.js set-up
you'll learn it along the way

But how do we use Express.js @studiohyperdrive?


APIS
APIS EVERYWHERE
quickmeme.com

REST

Representational State Transfer

R + (E) + **S** + **T**

Resources

Studio Hyperdrive

Resource = Entity = Object = Noun = Model

Resources

“a message”

“a user”

“a comment”

“a notification”

Resources

https://api.my-application.be

Resources

http://localhost:3000

Resources

https://api.my-application.be**/my-resource**

Resources

https://api.my-application.be**/v1/my-resource**

Resources

/my-resource

Resources

“a message”

Resources

/messages

Resources

plural, lowercase, hyphens, nouns, never a verb, …

Resources

https://opensource.zalando.com/restful-api-guidelines/

Resources

/User

/blogPosts

/create_message

Resources

/User

/blogPosts

/create_message

Resources

/users

/blogPosts

/create_message

Resources

/users

/blog-posts

/create_message

Resources

/users

/blog-posts

/messages

HTTP Verbs

Studio Hyperdrive

HTTP verbs

GET
POST
PUT
PATCH
DELETE

HTTP verbs

```
app.get('/', (req, res) => {});
app.post('/', (req, res) => {});
app.put('/', (req, res) => {});
app.patch('/', (req, res) => {});
app.delete('/', (req, res) => {});
```

HTTP verbs

GET = GET

POST = CREATE

PUT = UPDATE

PATCH = UPDATE PARTIALLY

DELETE = REMOVE

Resources

“a message”

Resources

/messages

Resources

Get all messages?
Get one message by id?
Create a new message?
Update a message?
Remove a message?

Resources

Get all messages? → GET
Get one message by id?
Create a new message?
Update a message?
Remove a message?

Resources

Get all messages? → GET
Get one message by id? → GET
Create a new message?
Update a message?
Remove a message?

Resources

Get all messages? → GET
Get one message by id? → GET
Create a new message? → POST
Update a message?
Remove a message?

Resources

Get all messages? → GET
Get one message by id? → GET
Create a new message? → POST
Update a message? → PUT
Remove a message?

Resources

Get all messages? → GET
Get one message by id? → GET
Create a new message? → POST
Update a message? → PUT
Remove a message? → DELETE

Resources

Get all? → GET

GET /messages

Get one by id? → GET

GET /messages

2x GET → /messages?

Resources

Get all? → GET

GET /messages

Get one by id? → GET

GET /messages/{id}

Resources

GET /messages
GET /messages/{id}
POST /messages
PUT /messages/{id}
DELETE /messages/{id}

HTTP verbs

Browser = GET

postman

handy tools

Studio Hyperdrive

https://www.getpostman.com

Let’s go!

Studio
Hyperdrive


Oh, oh, one more thing
before I forget ...

TALENT@STUDIOHYPERDRIVE.BE

Studio
Hyperdrive

Let’s go!

Studio
Hyperdrive

Let’s go!

https://github.com/studiohyperdrive/gastles_arteveldehogeschool_2019-03

Let’s go!

$ git clone

Let’s go!

$ git checkout **feature/start**

Let’s go!

https://github.com/studiohyperdrive/gastles_arteveldehogeschool_2019-03

Let’s go!

https://gastles-2019-03.hyperdrive.studio

Let’s go!

https://gastles-2019-03.hyperdrive.studio/docs


Go up and never stop

Studio
Hyperdrive