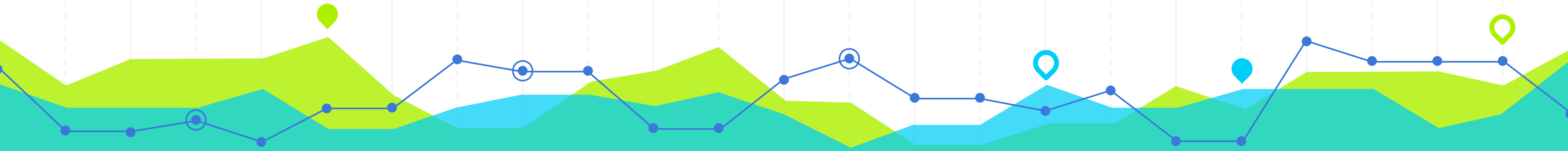


Node.JS als webserver

Express en Jade

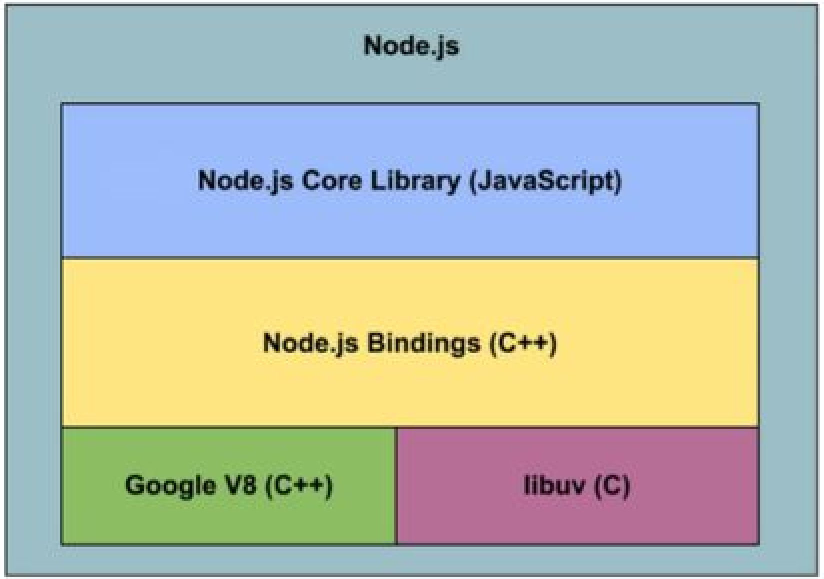
INHOUD

- Node.JS
- NPM
- Express
- Jade template
- Socket.IO

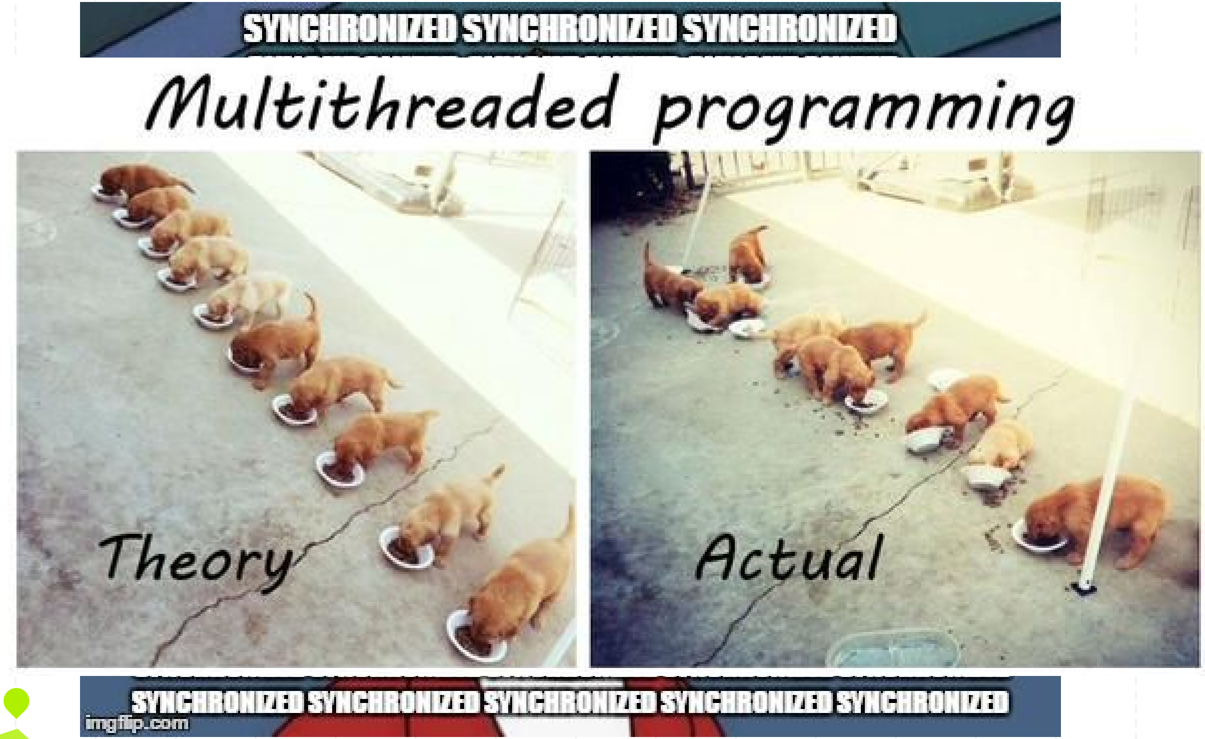
JavaScript, V8, Event-driven, Non-blocking I/O

NPM, Dependency Hell, Spaghetti code (callbacks).

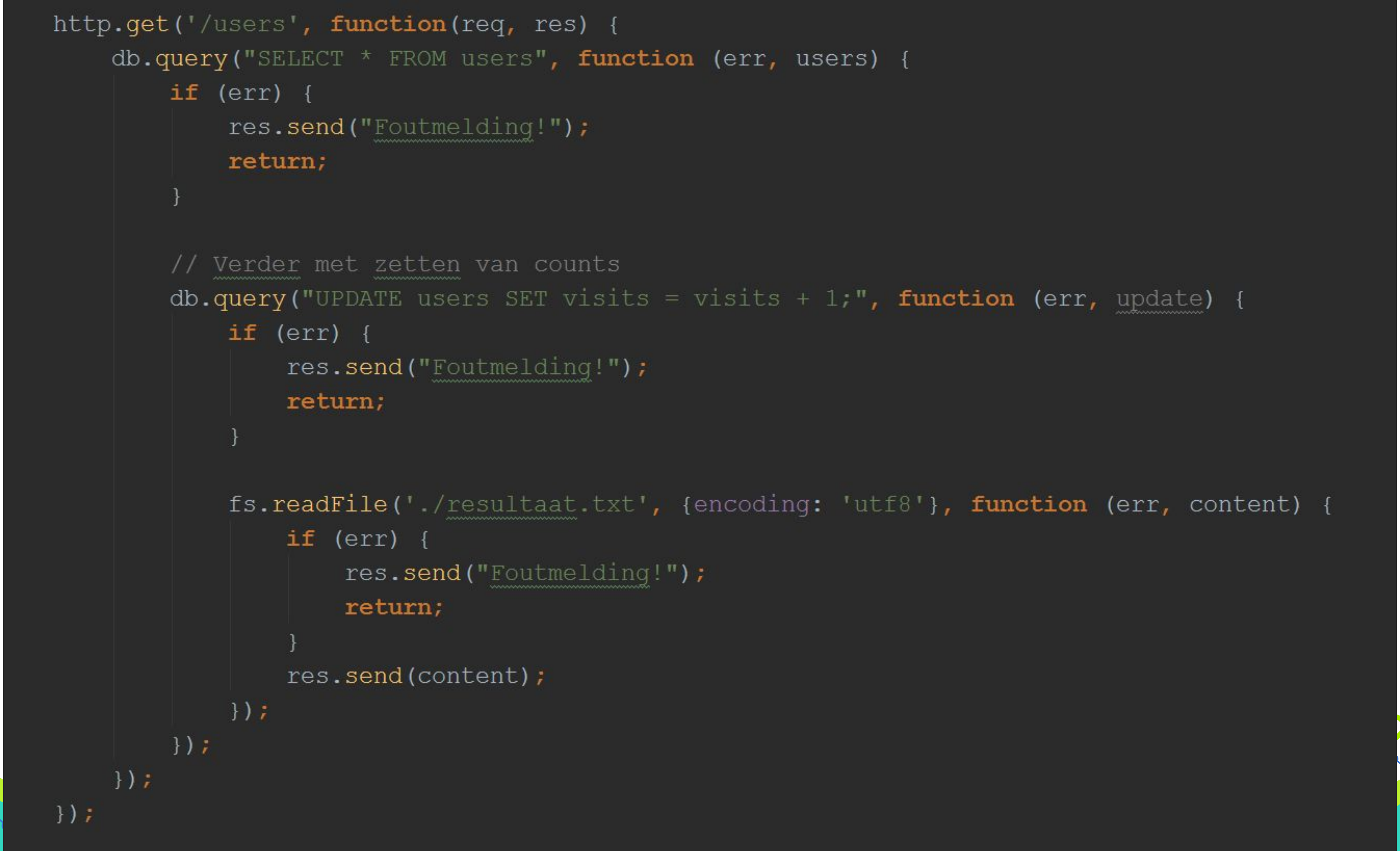
JavaScript, V8 Engine en NodeJS API, De opbouw



Single Thread, waarom!


SYNCHRONIZED SYNCHRONIZED SYNCHRONIZED
Multithreaded programming
Theory
Actual
SYNCHRONIZED SYNCHRONIZED SYNCHRONIZED SYNCHRONIZED SYNCHRONIZED
imgflip.com

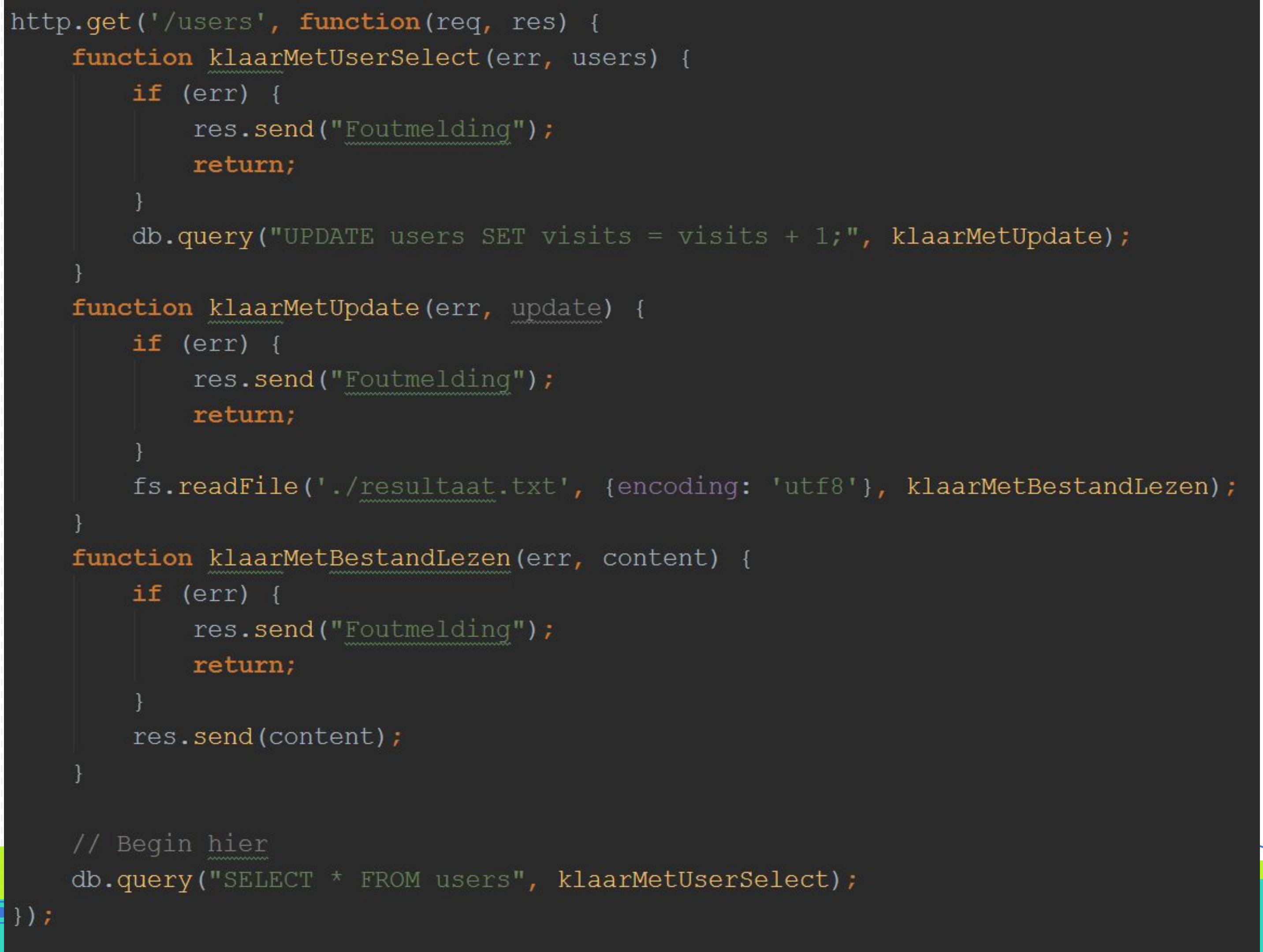
Async / Non blocking IO



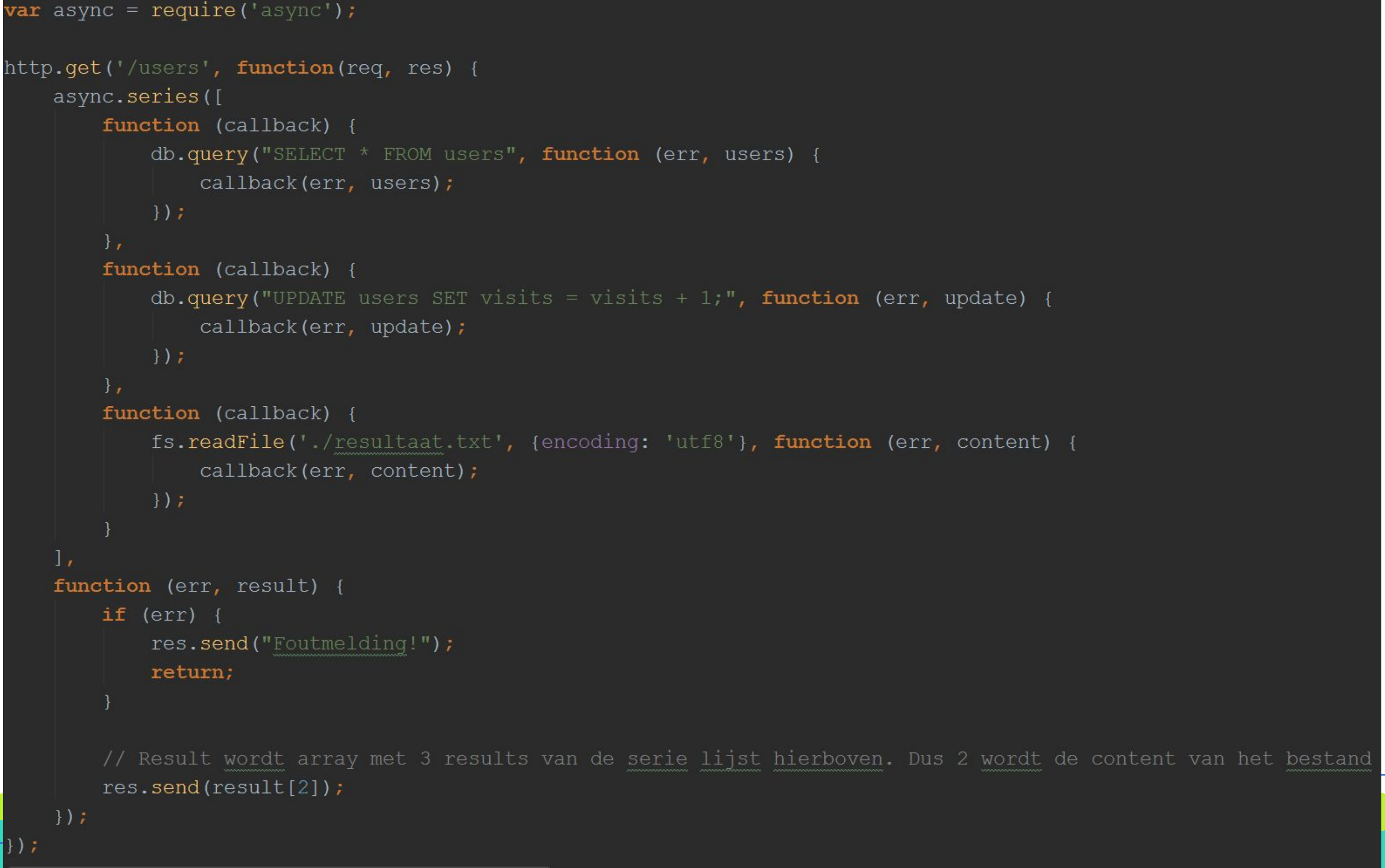
```
http.get('/users', function(req, res) {
    db.query("SELECT * FROM users", function (err, users) {
        if (err) {
            res.send("Foutmelding!");
            return;
        }

        // Verder met zetten van counts
        db.query("UPDATE users SET visits = visits + 1;", function (err, update) {
            if (err) {
                res.send("Foutmelding!");
                return;
            }

            fs.readFile('./resultaat.txt', {encoding: 'utf8'}, function (err, content) {
                if (err) {
                    res.send("Foutmelding!");
                    return;
                }
                res.send(content);
            });
        });
    });
});
```

Hoe dan?

- Verschillende functies (traditionele oplossing)
- Gebruik maken van een library (bijvoorbeeld async)
- Gebruik maken van de ES6 oplossing met Promises.

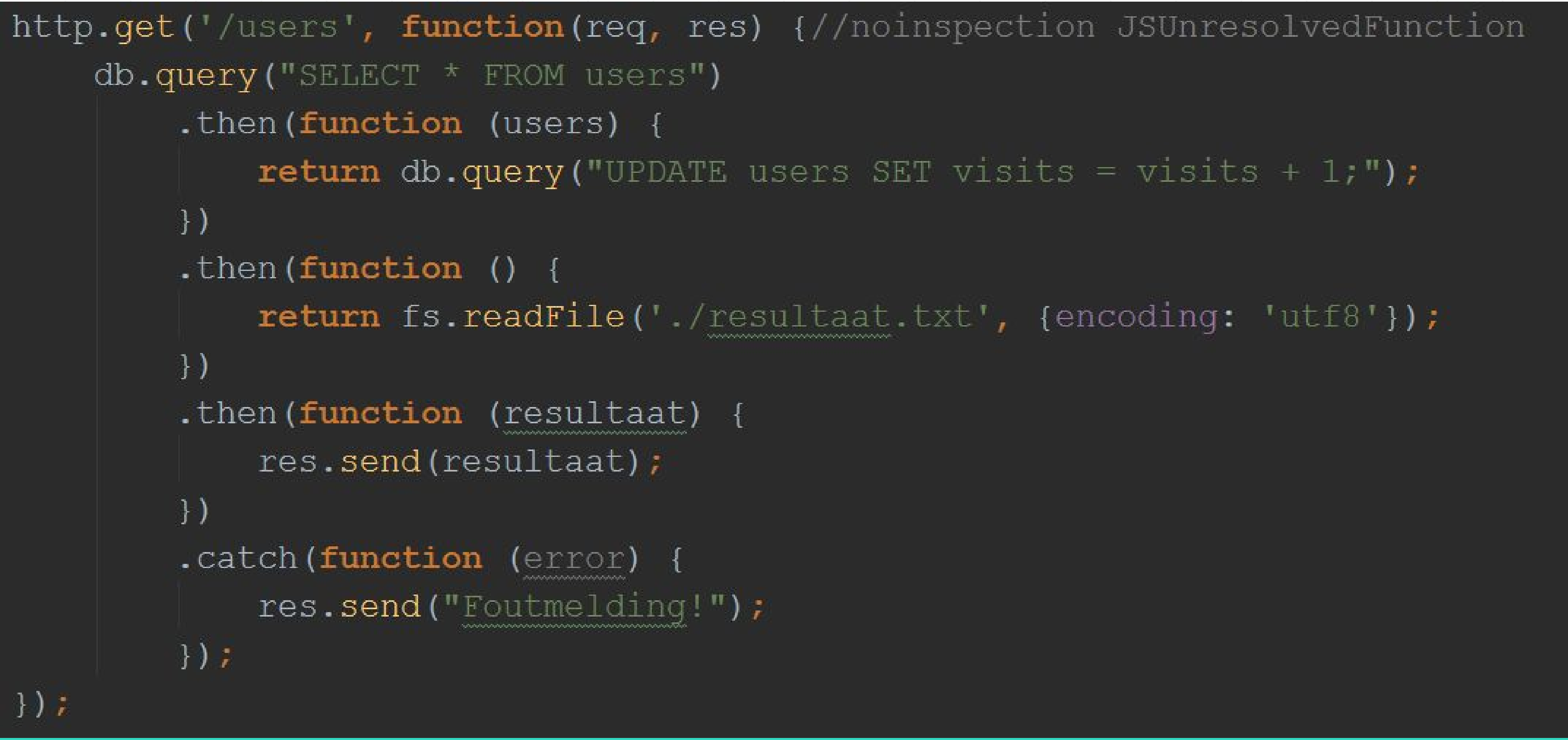


```
http.get('/users', function(req, res) {
    function klaarMetUserSelect(err, users) {
        if (err) {
            res.send("Foutmelding");
            return;
        }
        db.query("UPDATE users SET visits = visits + 1;", klaarMetUpdate);
    }
    function klaarMetUpdate(err, update) {
        if (err) {
            res.send("Foutmelding");
            return;
        }
        fs.readFile('./resultaat.txt', {encoding: 'utf8'}, klaarMetBestandLezen);
    }
    function klaarMetBestandLezen(err, content) {
        if (err) {
            res.send("Foutmelding");
            return;
        }
        res.send(content);
    }

    // Begin hier
    db.query("SELECT * FROM users", klaarMetUserSelect);
});
```

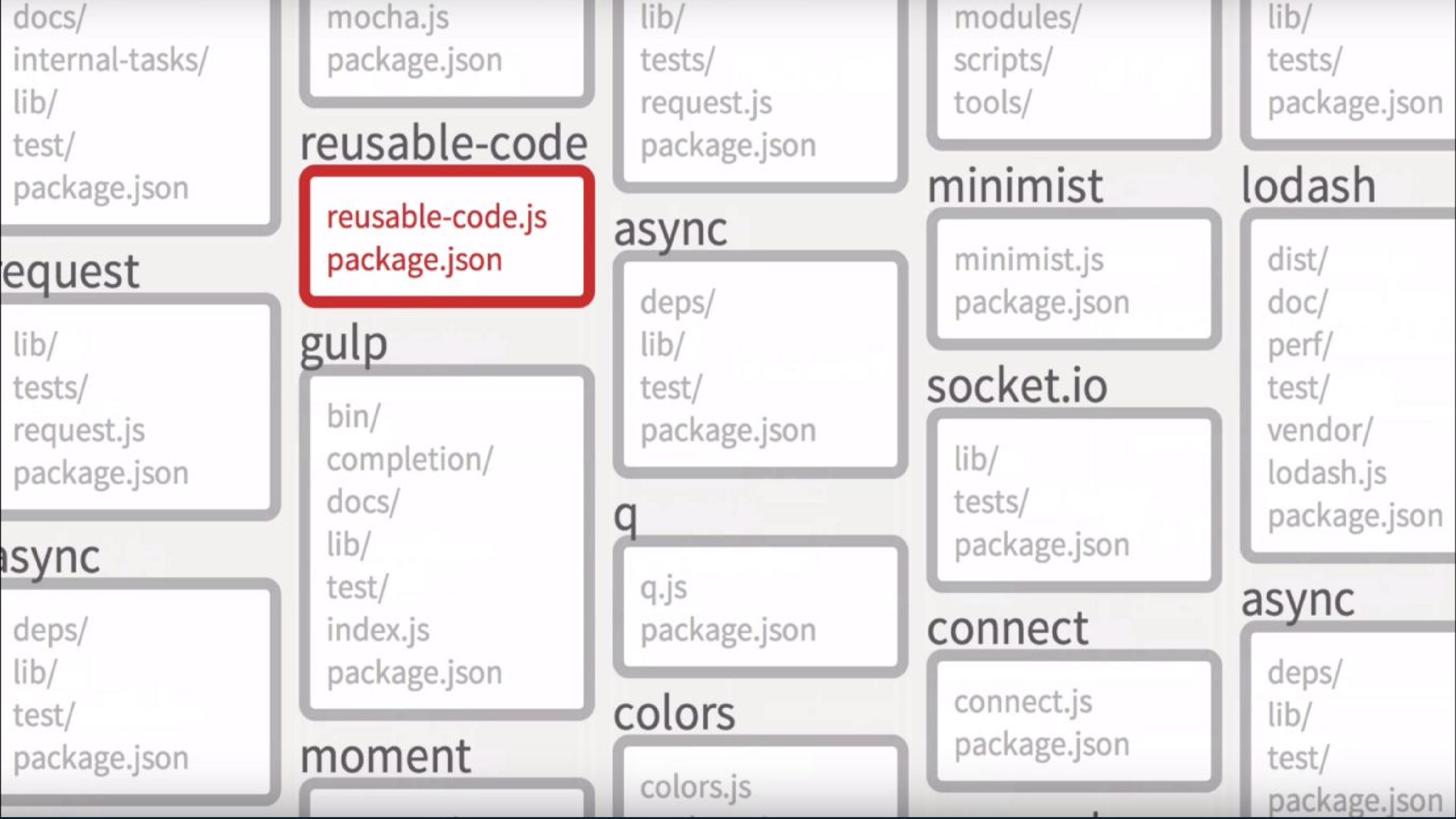


```
var async = require('async');

http.get('/users', function(req, res) {
    async.series([
        function (callback) {
            db.query("SELECT * FROM users", function (err, users) {
                callback(err, users);
            });
        },
        function (callback) {
            db.query("UPDATE users SET visits = visits + 1;", function (err, update) {
                callback(err, update);
            });
        },
        function (callback) {
            fs.readFile('./resultaat.txt', {encoding: 'utf8'}, function (err, content) {
                callback(err, content);
            });
        }
    ],
    function (err, result) {
        if (err) {
            res.send("Foutmelding!");
            return;
        }

        // Result wordt array met 3 results van de serie lijst hierboven. Dus 2 wordt de content van het bestand
        res.send(result[2]);
    });
});
```

Promise (ES6)



```
http.get('/users', function(req, res) {//noinspection JSUnresolvedFunction
    db.query("SELECT * FROM users")
        .then(function (users) {
            return db.query("UPDATE users SET visits = visits + 1;");
        })
        .then(function () {
            return fs.readFile('./resultaat.txt', {encoding: 'utf8'});
        })
        .then(function (resultaat) {
            res.send(resultaat);
        })
        .catch(function (error) {
            res.send("Foutmelding!");
        });
});
```



docs/
internal-tasks/
lib/
test/
package.json

equest

lib/
tests/
request.js
package.json

async

deps/
lib/
test/
package.json

mocha.js
package.json

reusable-code

reusable-code.js
package.json

gulp

bin/
completion/
docs/
lib/
test/
index.js
package.json

moment

lib/
tests/
request.js
package.json

async

deps/
lib/
test/
package.json

q

q.js
package.json

colors

colors.js

modules/
scripts/
tools/

minimist

minimist.js
package.json

socket.io

lib/
tests/
package.json

connect

connect.js
package.json

lib/
tests/
package.json

lodash

dist/
doc/
perf/
test/
vendor/
lodash.js
package.json

async

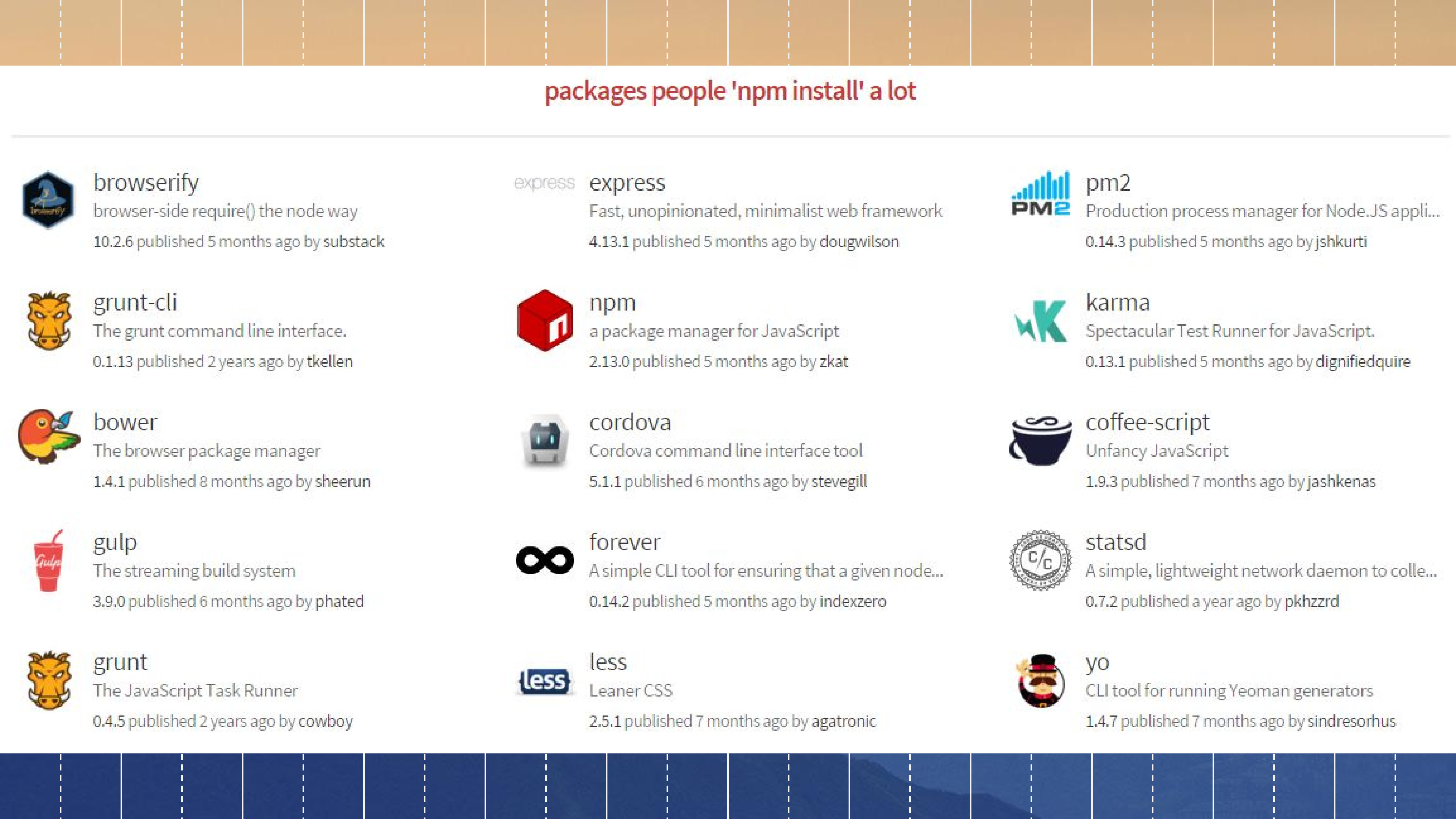
deps/
lib/
test/
package.json

NPM

npm install express --save

215.916

NPM Packages



packages people 'npm install' a lot

browserify
browser-side require() the node way
10.2.6 published 5 months ago by substack

grunt-cli
The grunt command line interface.
0.1.13 published 2 years ago by tkellen

bower
The browser package manager
1.4.1 published 8 months ago by sheerun

gulp
The streaming build system
3.9.0 published 6 months ago by phated

grunt
The JavaScript Task Runner
0.4.5 published 2 years ago by cowboy

express
Fast, unopinionated, minimalist web framework
4.13.1 published 5 months ago by dougwilson

npm
a package manager for JavaScript
2.13.0 published 5 months ago by zkat

cordova
Cordova command line interface tool
5.1.1 published 6 months ago by stevegill

forever
A simple CLI tool for ensuring that a given node...
0.14.2 published 5 months ago by indexzero

less
Leaner CSS
2.5.1 published 7 months ago by agatronic

pm2
Production process manager for Node.JS appli...
0.14.3 published 5 months ago by jshkurti

karma
Spectacular Test Runner for JavaScript.
0.13.1 published 5 months ago by dignifiedquire

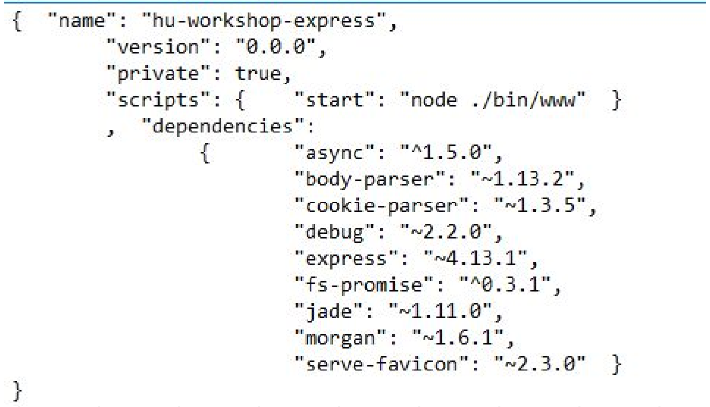
coffee-script
Unfancy JavaScript
1.9.3 published 7 months ago by jashkenas

statsd
A simple, lightweight network daemon to colle...
0.7.2 published a year ago by pkhzzrd

yo
CLI tool for running Yeoman generators
1.4.7 published 7 months ago by sindresorhus

Node Package Manager - Setup

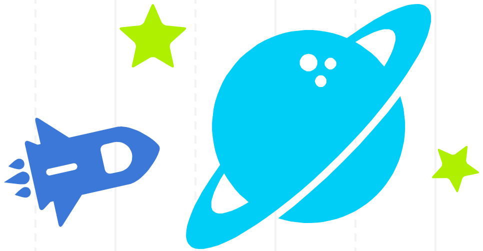
Package.Json



```
{  "name": "hu-workshop-express",
         "version": "0.0.0",
         "private": true,
         "scripts": {    "start": "node ./bin/www"  }
         ,  "dependencies":
                 {        "async": "^1.5.0",
                          "body-parser": "~1.13.2",
                          "cookie-parser": "~1.3.5",
                          "debug": "~2.2.0",
                          "express": "~4.13.1",
                          "fs-promise": "^0.3.1",
                          "jade": "~1.11.0",
                          "morgan": "~1.6.1",
                          "serve-favicon": "~2.3.0"  }
}
```

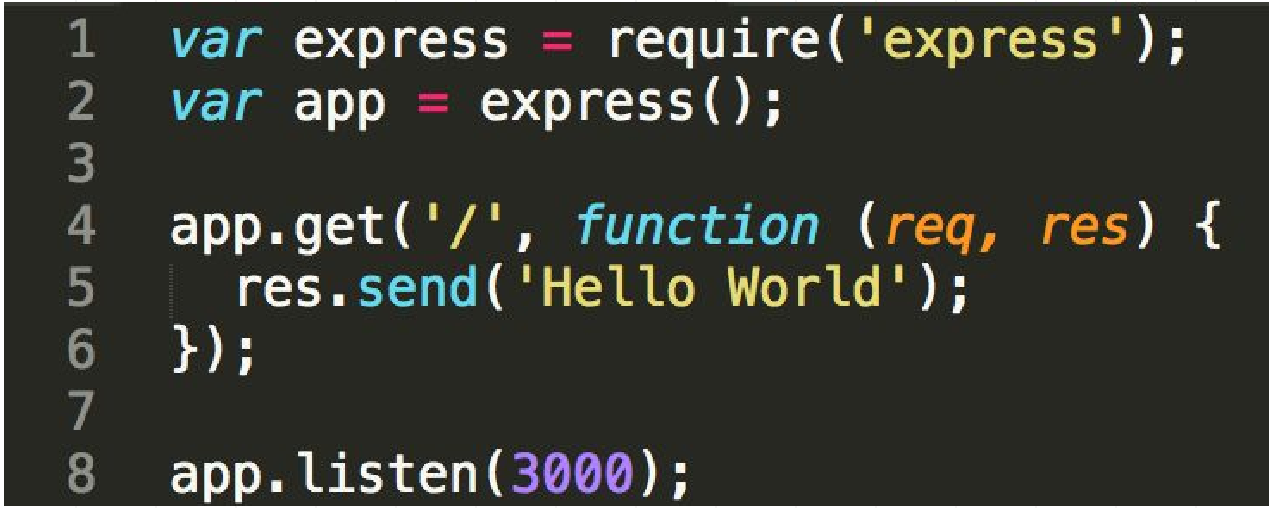
Cmd commands

npm install	Project Root
npm install express	Install a package locally
npm install -g express	Install a package globally
npm publish code	Upload your code!



EXPRESS

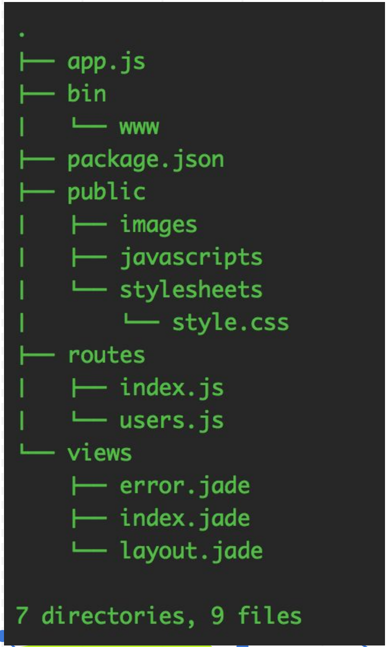
HELLO WORLD



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

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

app.listen(3000);
```

Express application generator

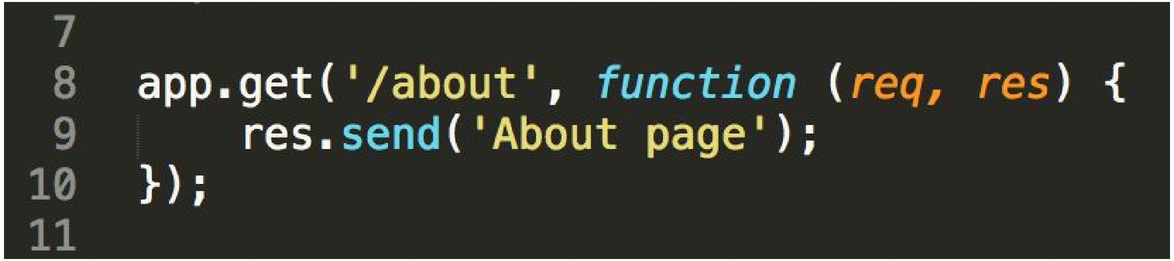
```
$ npm install express-generator -g

$ express testapp

$ cd myapp

$ npm install
```



```
.
├── app.js
├── bin
│   └── www
├── package.json
├── public
│   ├── images
│   ├── javascripts
│   └── stylesheets
│       └── style.css
├── routes
│   ├── index.js
│   └── users.js
└── views
    ├── error.jade
    ├── index.jade
    └── layout.jade

7 directories, 9 files
```

- Applicatie skelet
- Middleware
- Mogelijkheden

Basic routing



```
app.get('/about', function (req, res) {
    res.send('About page');
});
```

- app
- get
- path
- handler

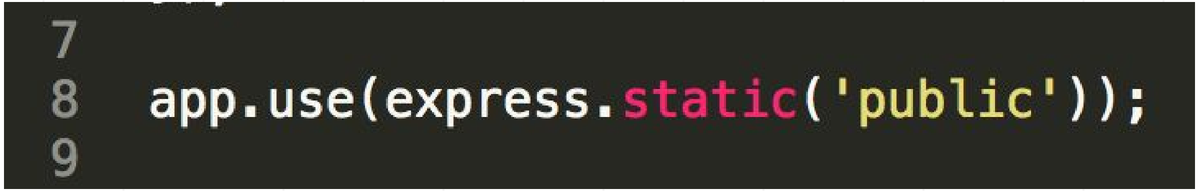
app.get

app.post

app.put

app.delete

Static files



```
app.use(express.static('public'));
```

- Middleware
- Files from filesystem

WebSockets

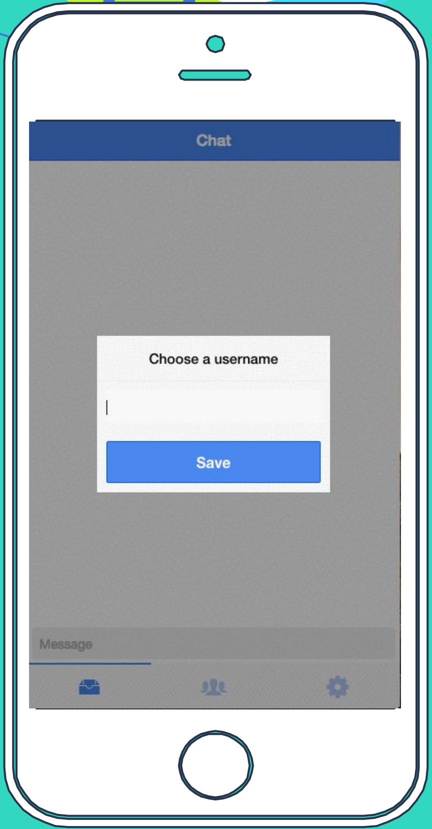
Socket.io

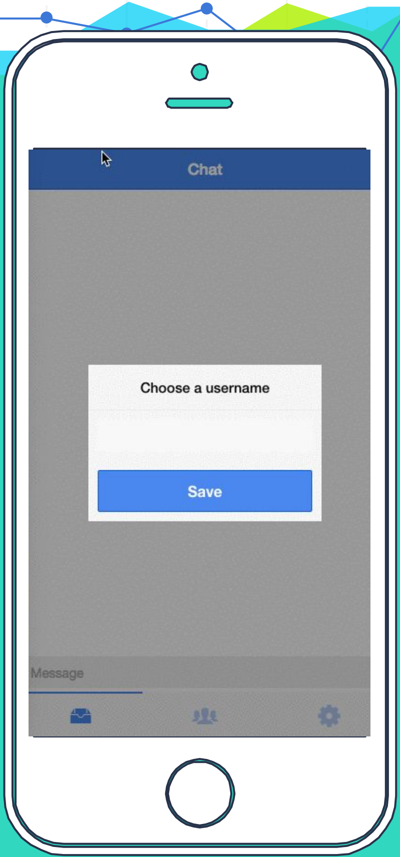
Real-time Analyses

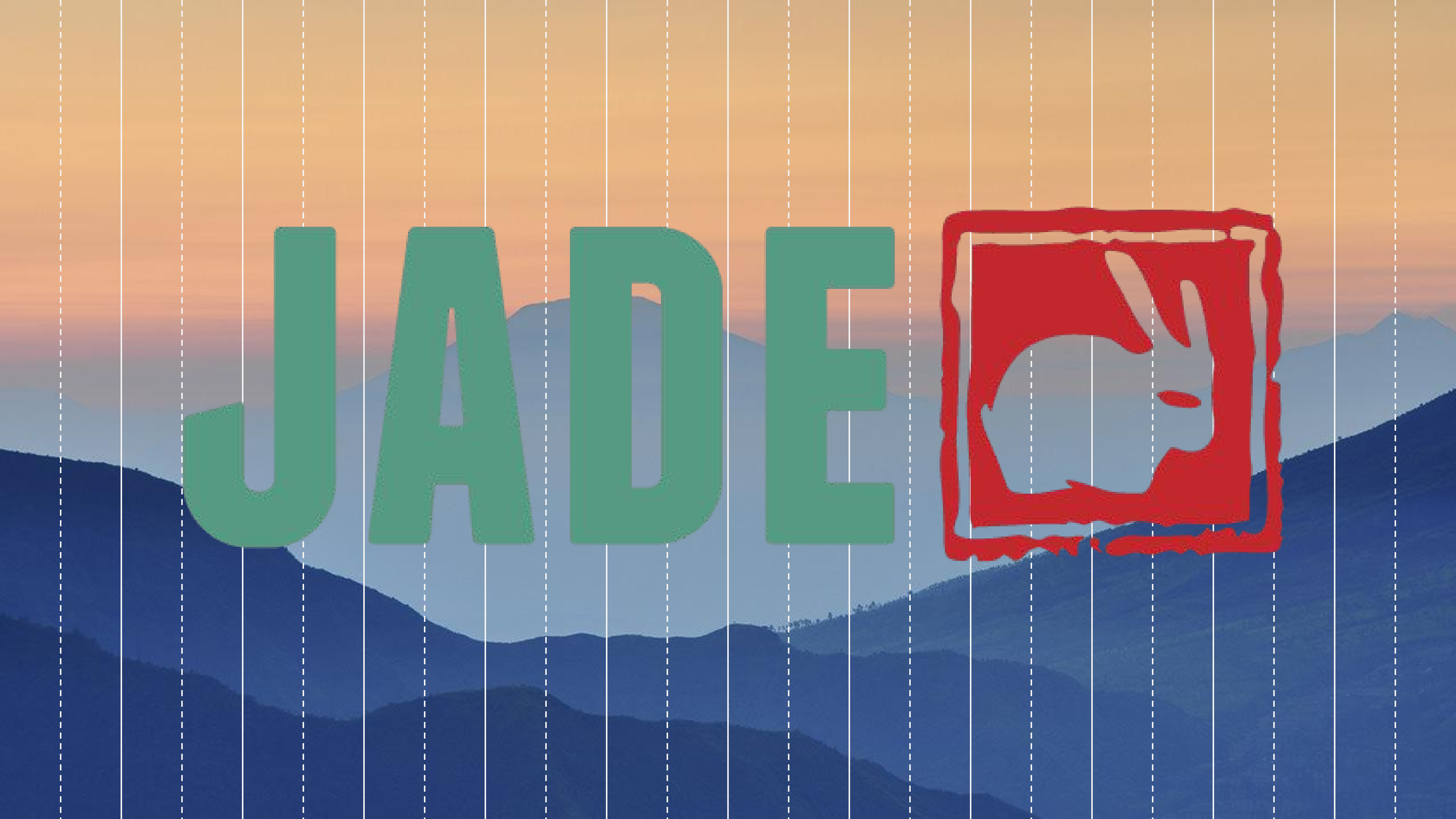
Chat Systeem

0110 Binary Streaming

Voorbeeld chat systeem



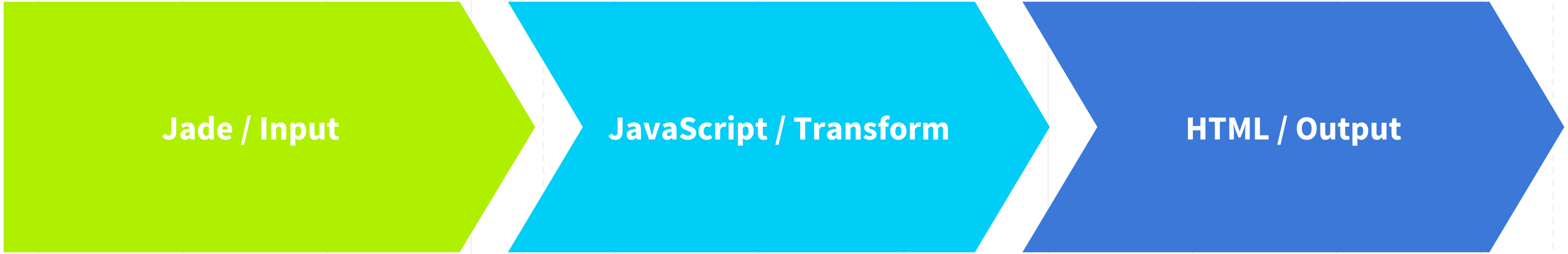



JADE

Wat is het?

- Front end framework
- Gebaseerd op Haml
- Template Engine

Process

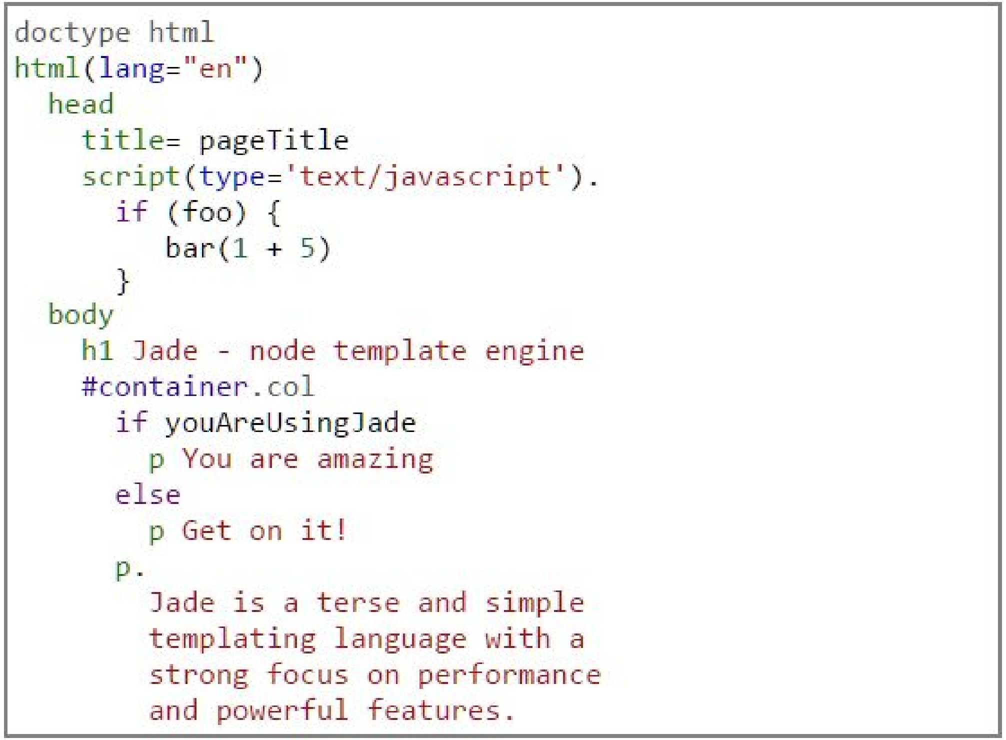


Jade / Input

JavaScript / Transform

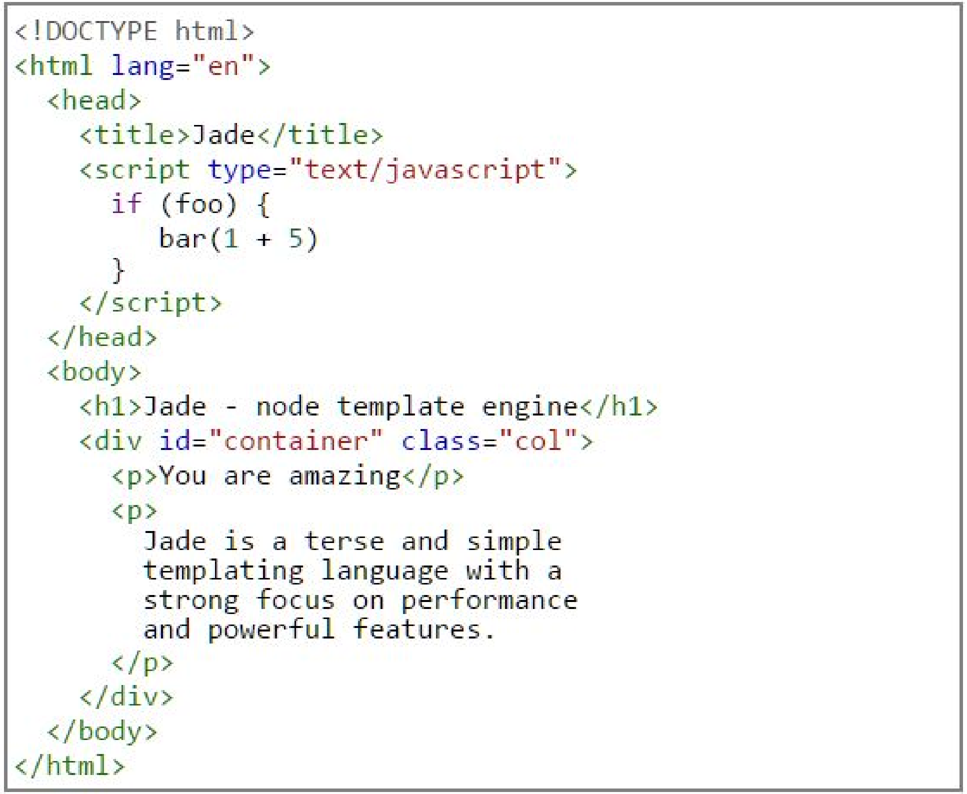
HTML / Output

Voorbeeld (Input)



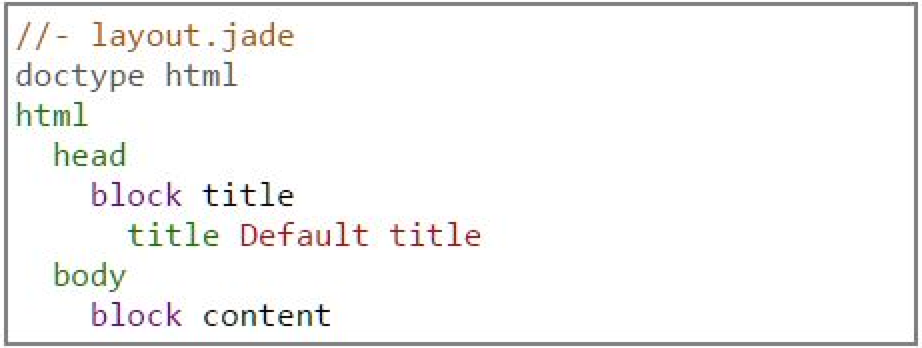
```
doctype html
html(lang="en")
  head
    title= pageTitle
    script(type='text/javascript').
      if (foo) {
         bar(1 + 5)
      }
  body
    h1 Jade - node template engine
    #container.col
      if youAreUsingJade
        p You are amazing
      else
        p Get on it!
      p.
        Jade is a terse and simple
        templating language with a
        strong focus on performance
        and powerful features.
```

Voorbeeld (Output)

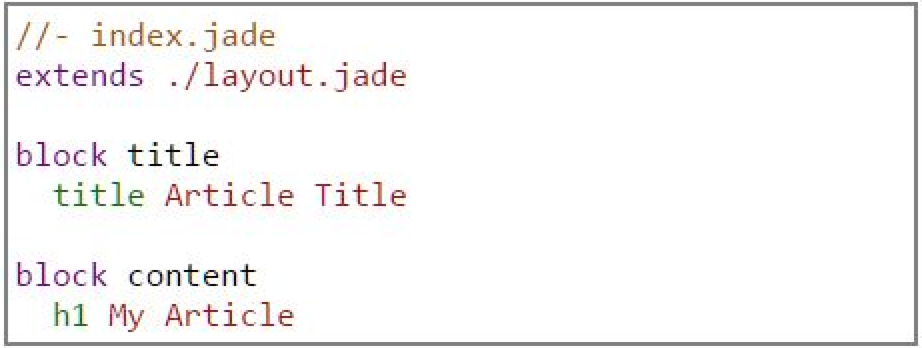


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Jade</title>
    <script type="text/javascript">
      if (foo) {
         bar(1 + 5)
      }
    </script>
  </head>
  <body>
    <h1>Jade - node template engine</h1>
    <div id="container" class="col">
      <p>You are amazing</p>
      <p>
        Jade is a terse and simple
        templating language with a
        strong focus on performance
        and powerful features.
      </p>
    </div>
  </body>
</html>
```

Extend (Input)

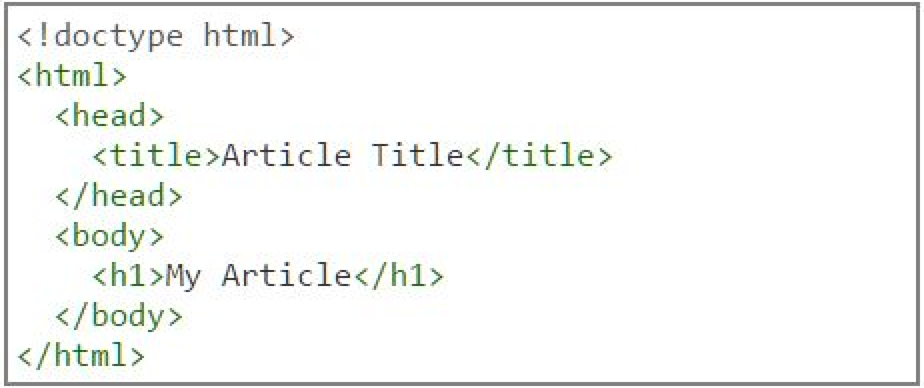


```
//- layout.jade
doctype html
html
  head
    block title
      title Default title
  body
    block content
```



```
//- index.jade
extends ./layout.jade

block title
  title Article Title

block content
  h1 My Article
```

Extend (Output)



```
<!doctype html>
<html>
  <head>
    <title>Article Title</title>
  </head>
  <body>
    <h1>My Article</h1>
  </body>
</html>
```

Voordelen

- Ondersteund dynamische code
- Makkelijk “code reuse”
- Makkelijk te gebruiken (vooral met Node.JS)

There is no place like 127.0.0.1

Gratis Node.js hosting providers

heroku

OPENSHIFT

Koding

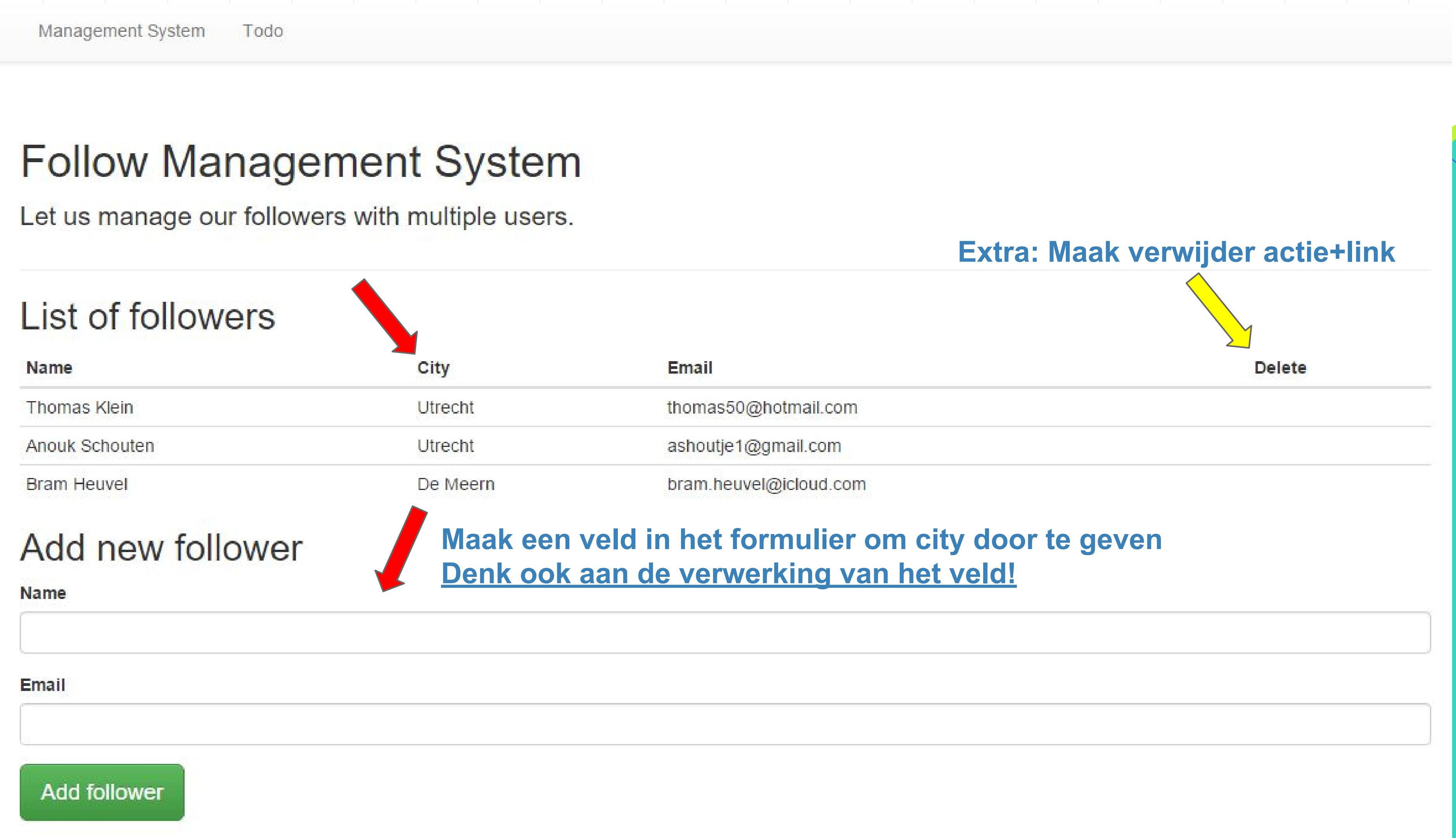
Demo applicatie, Todo list

Opdracht

1. Download NodeJS 4 of hoger.
2. Download Project Sources:

   http://mooie.link/hu-node

3. Open command prompt/terminal in je uitgepakte map
4. Voer uit: ‘npm update’
5. Open in je IDE/Editor



Follow Management System

Let us manage our followers with multiple users.

Extra: Maak verwijder actie+link

List of followers

Name	City	Email	Delete
Thomas Klein	Utrecht	thomas50@hotmail.com	
Anouk Schouten	Utrecht	ashoutje1@gmail.com	
Bram Heuvel	De Meern	bram.heuvel@icloud.com	

Add new follower

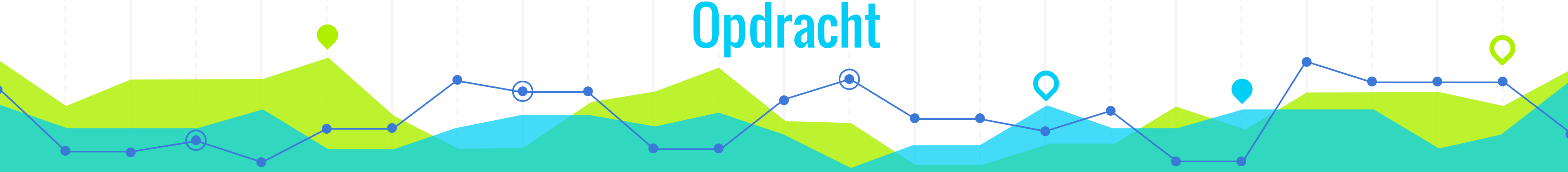
Maak een veld in het formulier om city door te geven
Denk ook aan de verwerking van het veld!

Name:

Email:

```
storage['Thomas50@hotmail.com'] = {
    name: 'Thomas Klein',
    city: 'Utrecht'
};

storage['AShoutje1@gmail.com'] = {
    name: 'Anouk Schouten',
    city: 'Utrecht'
};

storage['bram.heuvel@icloud.com'] = {
    name: 'Bram Heuvel',
    city: 'De Meern'
};
```

Opdracht



De opdracht:

- Maak een extra form element voor de waarde ‘city’
- Zorg dat deze ook in de controller verwerkt wordt

Extra uitdaging:

- Maak een delete link in de tabel

Opdracht

Cheatsheet:

- Uitvoeren van de server: ‘npm start’ in root. http://localhost:3000/
- Express: function (req, res) is je callback
    - body parameters: ‘var iets = req.body.iets;’

  http://expressjs.com/en/api.html

- Kijk naar dia’s, zie de pdf in de sources.
- Google is uw vriend.
- Vragen stellen.

THANKS!

Any questions?

CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash

PRESENTATION DESIGN

This presentation uses the following typographies and colors:

- Titles: **Oswald**
- Body copy: **Source Sans Pro**

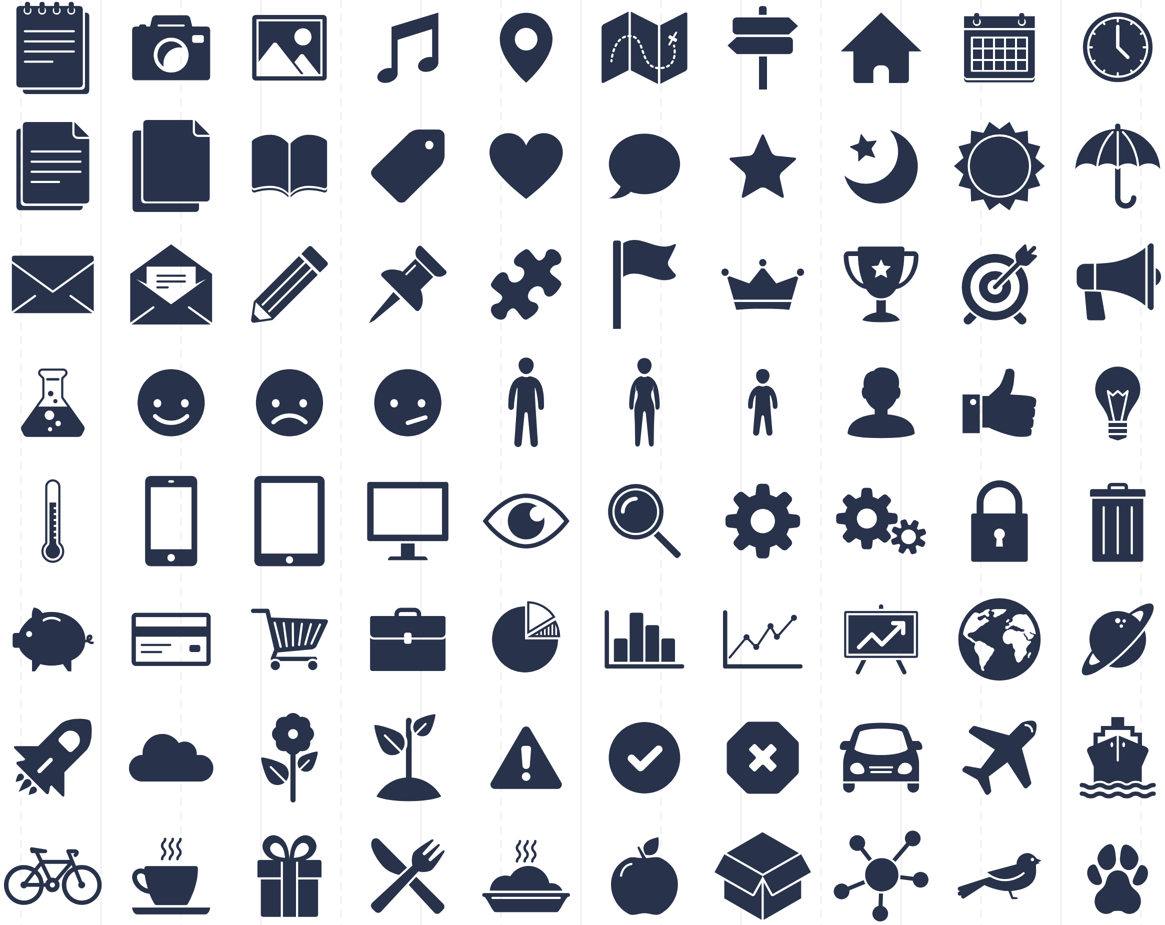
You can download the fonts on this page:

https://www.google.com/fonts#UsePlace:use/Collection:Source+Sans+Pro:400,700|Oswald:400,700

Click on the “arrow button” that appears on the top right 🔗 ⬇

Sky blue **#00cef6** / Bright green **#aff000** / Blue **#3c78d8** / Dark blue **#28324a**

You don’t need to keep this slide in your presentation. It’s only here to serve you as a design guide if you need to create new slides or download the fonts to edit the presentation in PowerPoint®



SlidesCarnival icons are editable shapes.

This means that you can:

- Resize them without losing quality.
- Change fill color and opacity.
- Change line color, width and style.

Isn’t that nice? :)

Examples: