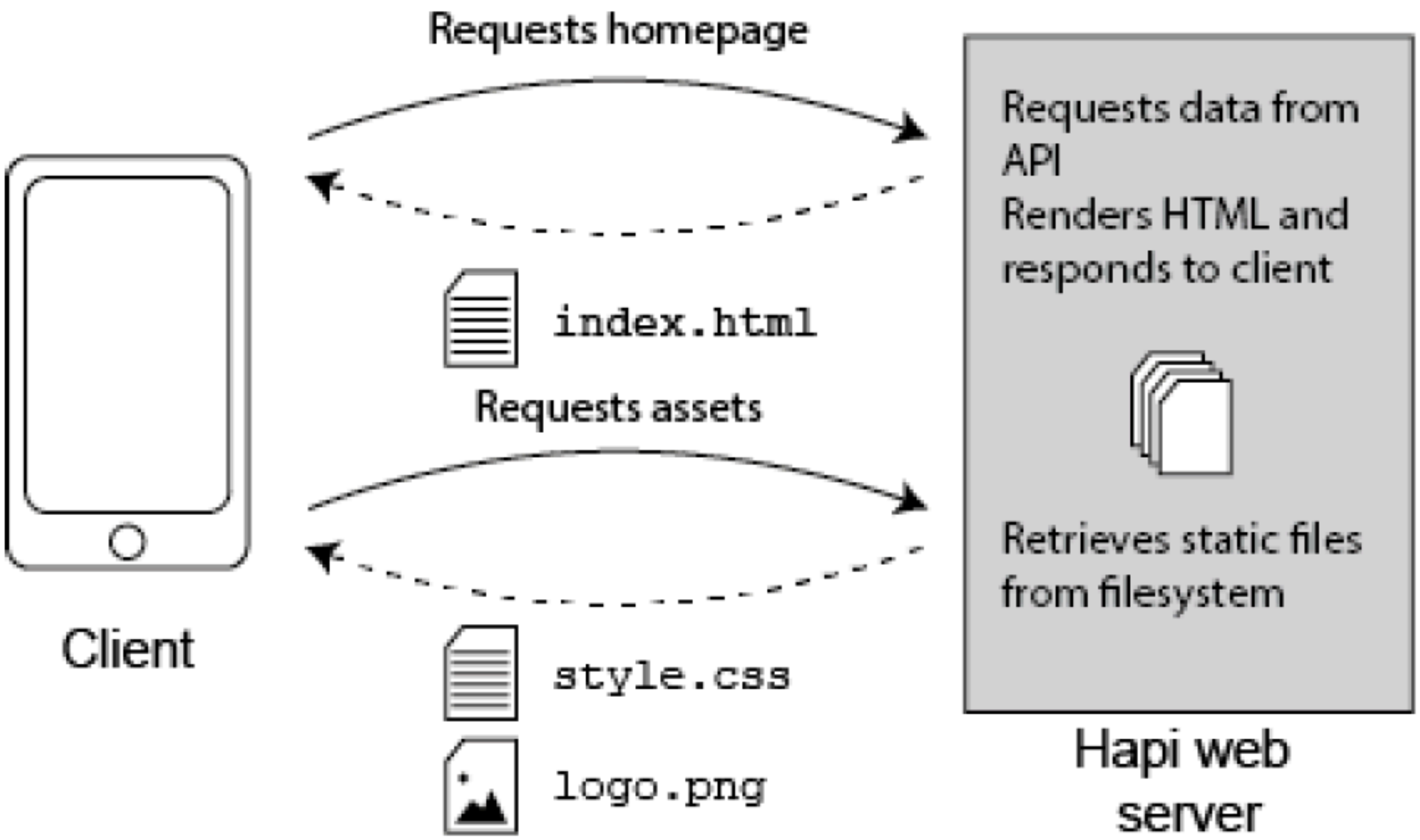
Views

Application Structure

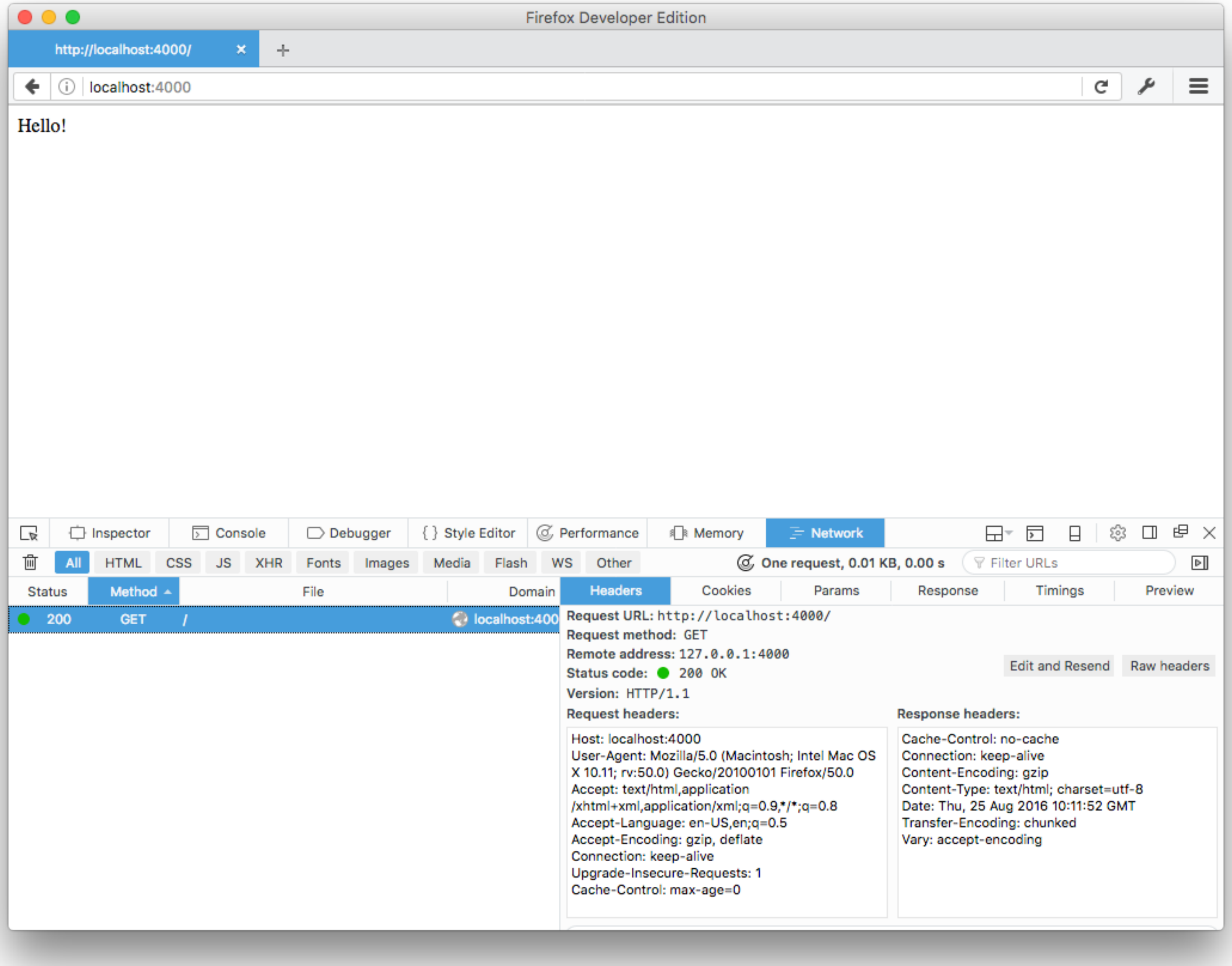


reply

- reply responds to the browser with a simple string.

controller.js

```
exports.index = {

  handler: function (request, reply) {
    reply('Hello!');
  }

};
```



Hello!

One request, 0.01 KB, 0.00 s

Status	Method	File	Domain
● 200	GET	/	localhost:400

```
Request URL: http://localhost:4000/
Request method: GET
Remote address: 127.0.0.1:4000
Status code: ● 200 OK
Version: HTTP/1.1
```

Request headers:

```
Host: localhost:4000
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS
X 10.11; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application
/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response headers:

```
Cache-Control: no-cache
Connection: keep-alive
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
Date: Thu, 25 Aug 2016 10:11:52 GMT
Transfer-Encoding: chunked
Vary: accept-encoding
```

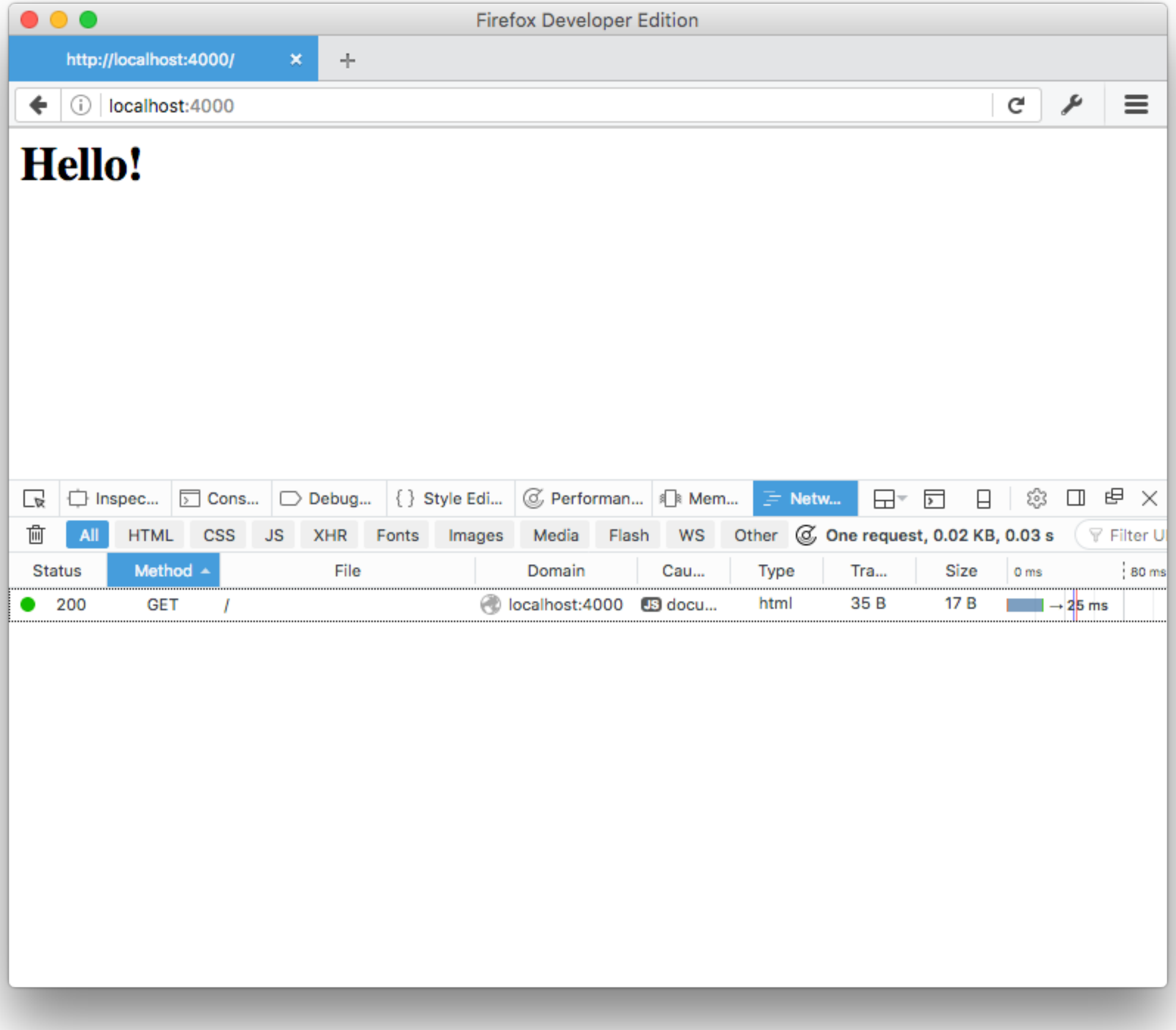
reply

- In order to render web pages we could pass html content

- This would become very unwieldy and unmaintainable

controller.js

```
exports.index = {

  handler: function (request, reply) {
    reply(‘<h1> Hello! </h1>’);
  }

};
```

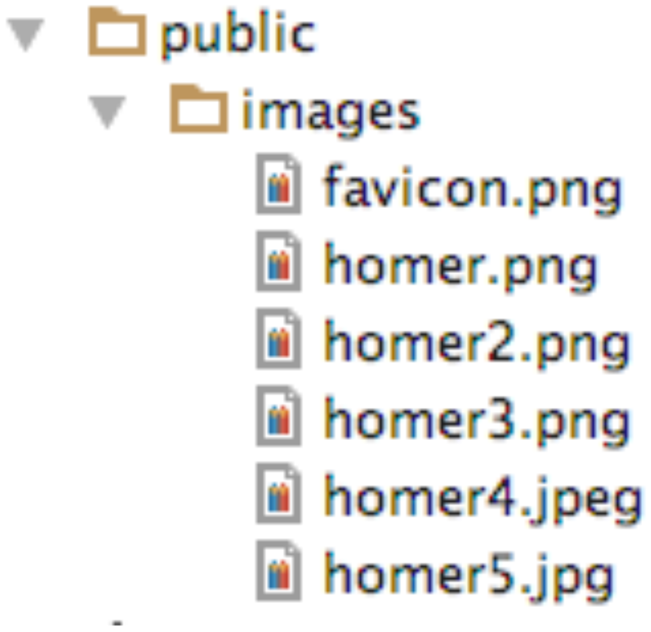


Hello!

```
Status  Method  File  Domain          Cau...   Type  Tra...  Size  0 ms       80 ms
200     GET     /     localhost:4000  docu...  html  35 B    17 B  → 25 ms
```

Static Assets

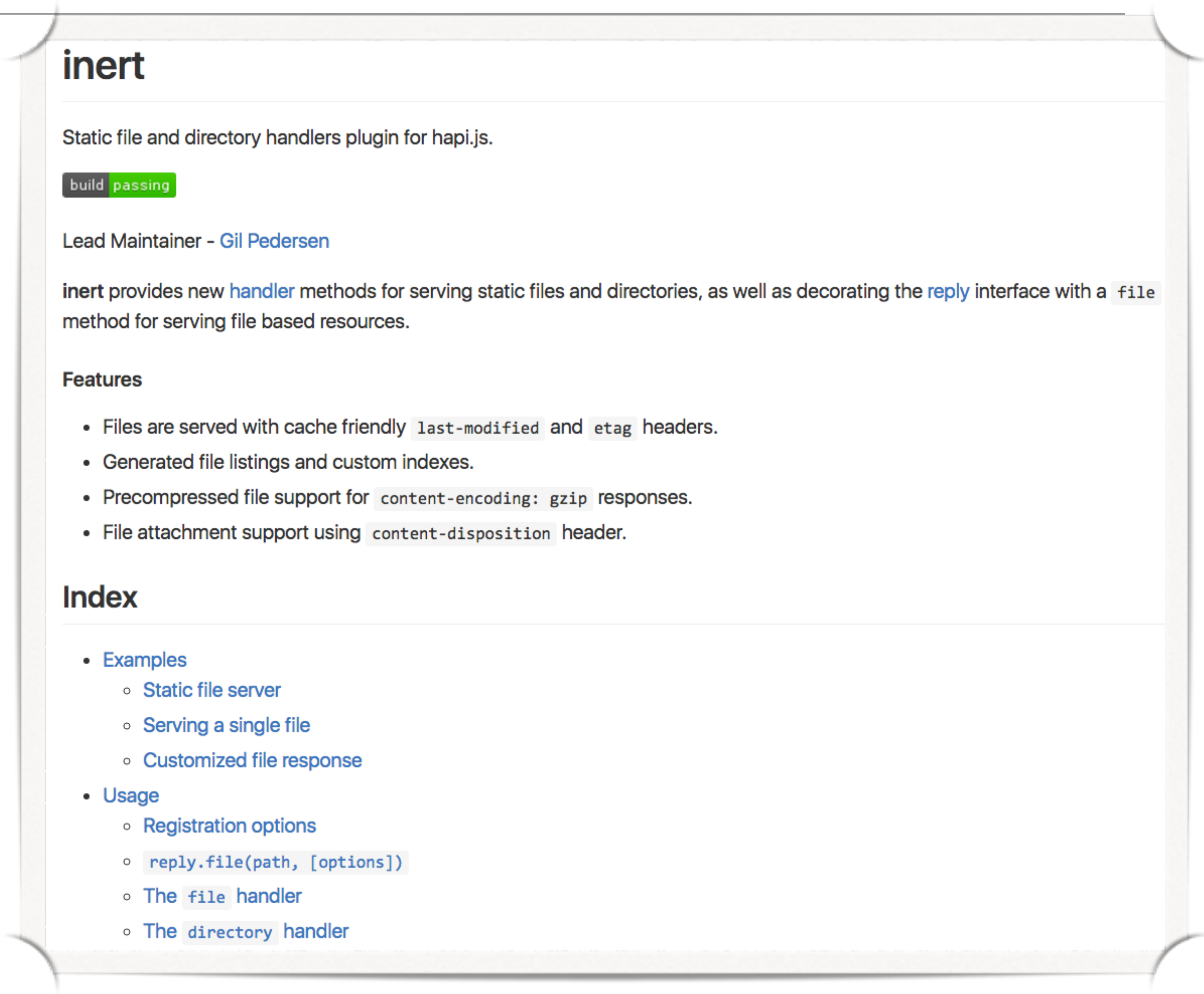
- Additionally, serving folders of static content (images, css etc..) might require an extensive range or routes + handlers



```
public
  images
    favicon.png
    homer.png
    homer2.png
    homer3.png
    homer4.jpeg
    homer5.jpg
```

inert plugin: https://github.com/hapijs/inert

- This plugin provides simple file serving features for hapi



inert

Static file and directory handlers plugin for hapi.js.

build passing

Lead Maintainer - Gil Pedersen

inert provides new handler methods for serving static files and directories, as well as decorating the reply interface with a `file` method for serving file based resources.

Features

- Files are served with cache friendly `last-modified` and `etag` headers.
- Generated file listings and custom indexes.
- Precompressed file support for `content-encoding: gzip` responses.
- File attachment support using `content-disposition` header.

Index

https://github.com/hapijs/inert

Restructure the Application Project Structure

- As we are about to grow a more substantial projects, we take care to lay out a sustainable folder structure.

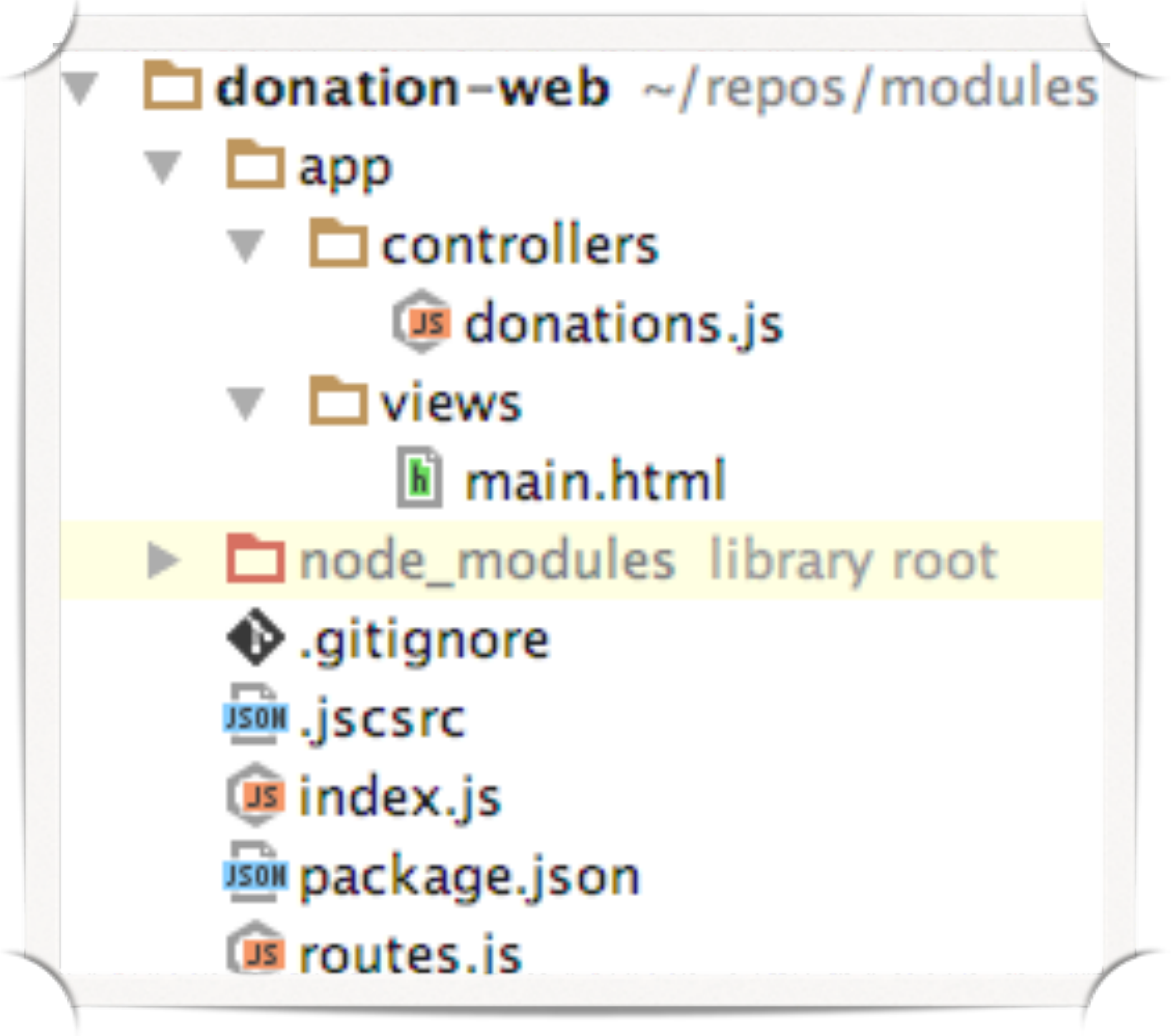
app contents bulk of application sources

controllers will contain handlers

views will contain html templates

index.js is the application entry point

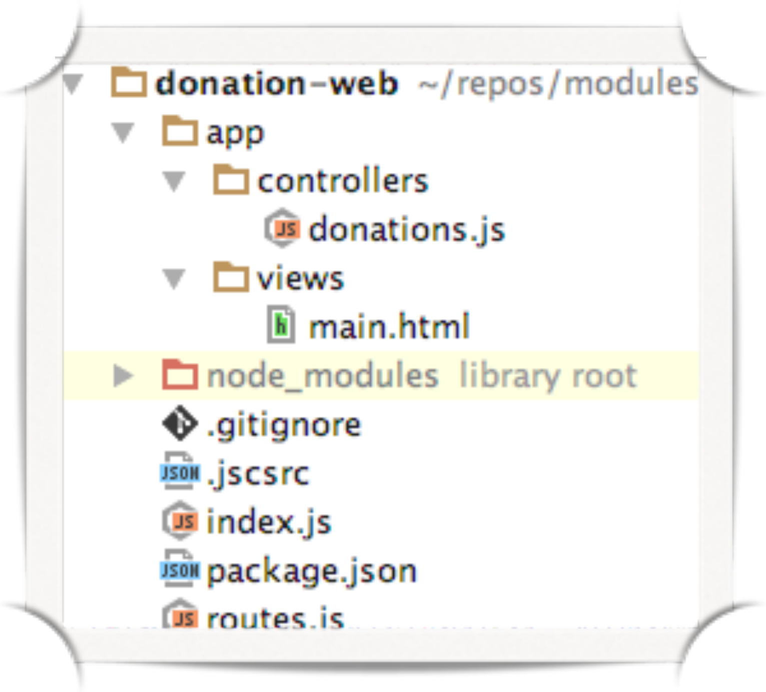
routes.js define all application routes



```
▼ donation-web ~/repos/modules
  ▼ app
    ▼ controllers
        donations.js
    ▼ views
        main.html
  ▶ node_modules library root
    .gitignore
    .jscsrc
    index.js
    package.json
    routes.js
```

Main Application Entrypoint

- Plugin must be registered, then if successful:
  - Routes included
  - Server started



```
donation-web ~/repos/modules
  app
    controllers
      donations.js
    views
      main.html
  node_modules library root
  .gitignore
  .jscsrc
  index.js
  package.json
  routes.js
```

index.js

```
'use strict';

const Hapi = require('hapi');

var server = new Hapi.Server();
server.connection({ port: process.env.PORT || 4000 });

server.register(require('inert'), err => {

  if (err) {
    throw err;
  }

  server.route(require('./routes'));
  server.start((err) => {
    if (err) {
      throw err;
    }

    console.log('Server listening at:', server.info.uri);
  });
});
```

```
$ npm install inert -save
```

Routes + Controller

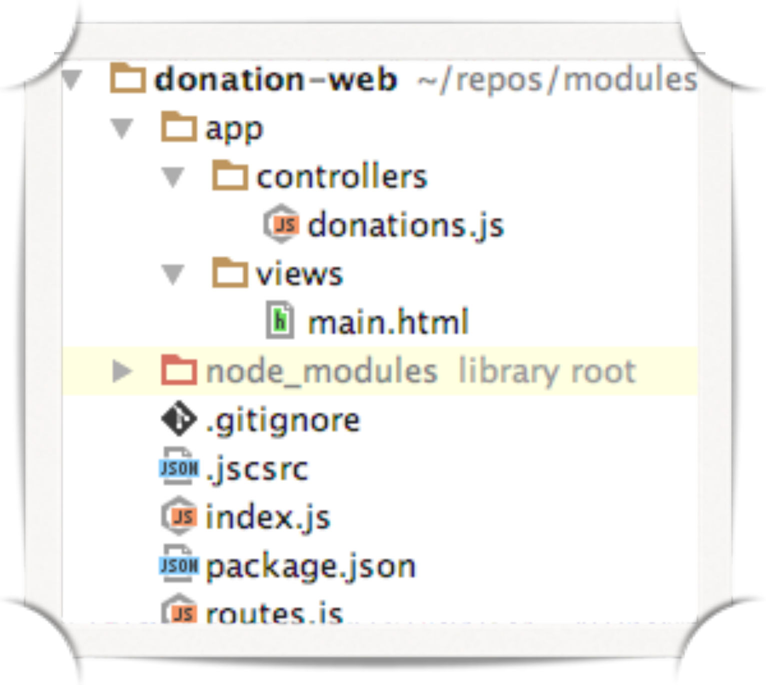
- Route recognises a single pattern: ‘/‘.
- The handler replies will a single file - which will be rendered in the client browser

routes.js

```
const Donations = require('./app/controllers/donations');

module.exports = [

  { method: 'GET', path: '/', config: Donations.home },

];
```

app/controllers/donation.js

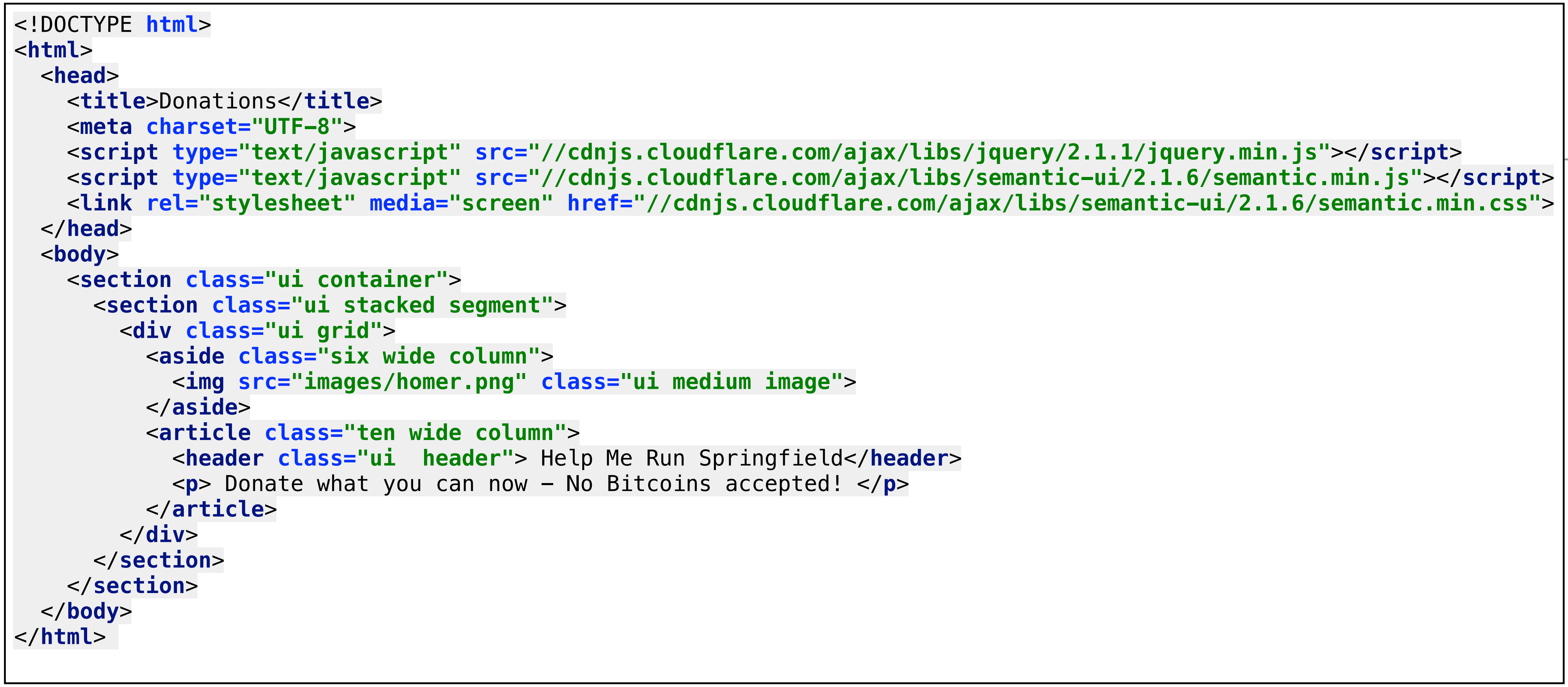
```
'use strict';

exports.home = {

  handler: (request, reply) => {
    reply.file('./app/views/main.html');
  },

};
```



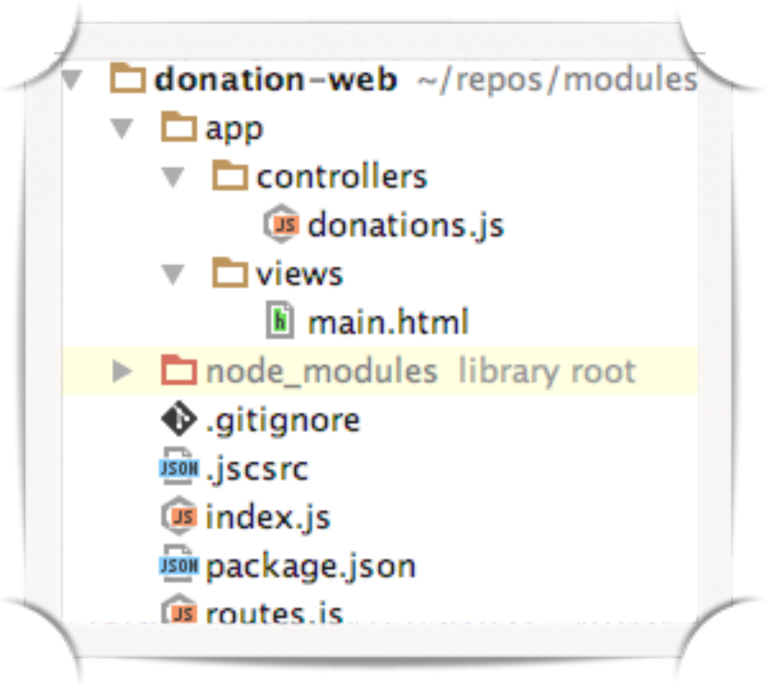
```
▼ donation-web ~/repos/modules
  ▼ app
    ▼ controllers
        donations.js
    ▼ views
        main.html
  ▶ node_modules library root
    .gitignore
    .jscsrc
    index.js
    package.json
    routes.js
```

app/views/main.html

view



```
<!DOCTYPE html>
<html>
  <head>
    <title>Donations</title>
    <meta charset="UTF-8">
    <script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
    <script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.1.6/semantic.min.js"></script>
    <link rel="stylesheet" media="screen" href="//cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.1.6/semantic.min.css">
  </head>
  <body>
    <section class="ui container">
      <section class="ui stacked segment">
        <div class="ui grid">
          <aside class="six wide column">
            <img src="images/homer.png" class="ui medium image">
          </aside>
          <article class="ten wide column">
            <header class="ui  header"> Help Me Run Springfield</header>
            <p> Donate what you can now - No Bitcoins accepted! </p>
          </article>
        </div>
      </section>
    </section>
  </body>
</html>
```

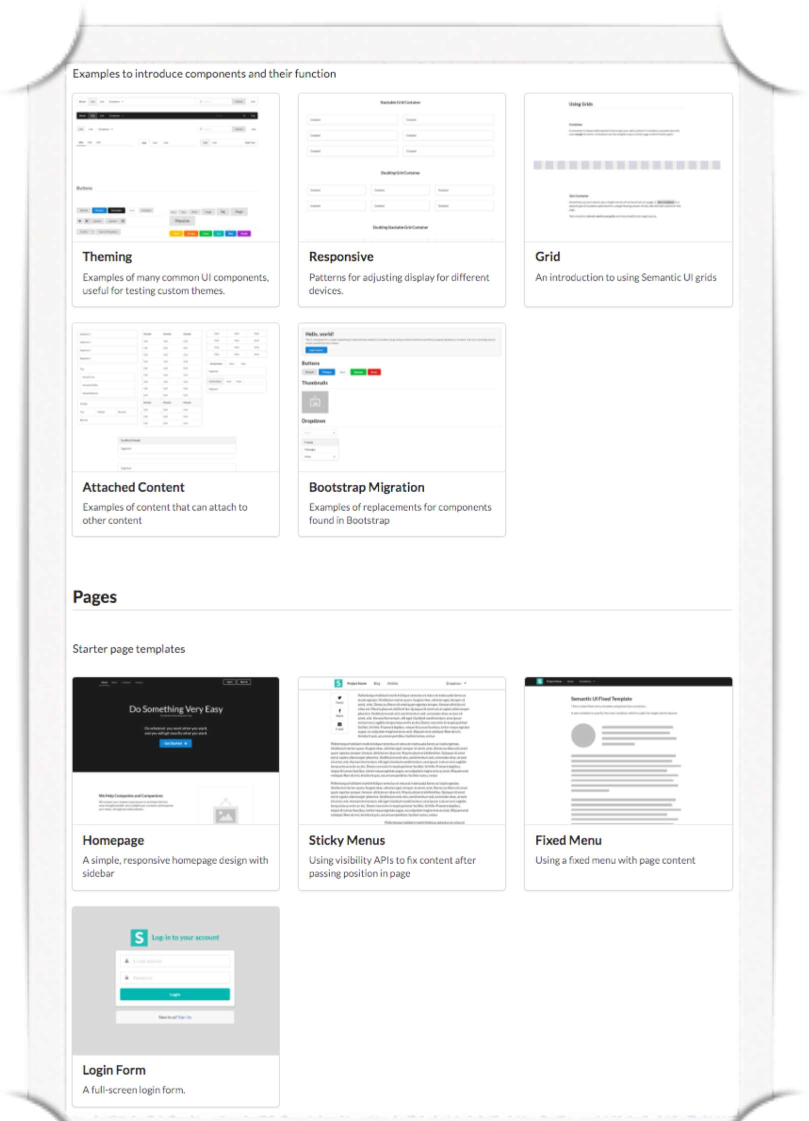


```
donation-web ~/repos/modules
  app
    controllers
      donations.js
    views
      main.html
  node_modules library root
  .gitignore
  .jscsrc
  index.js
  package.json
  routes.js
```

- Simple, static web page.
- Loads Semantic UI Stylesheet from CDN.

Semantic UI

- Rich, themeable, attractive layouts & UI components



Examples to introduce components and their function

Theming
Examples of many common UI components, useful for testing custom themes.

Responsive
Patterns for adjusting display for different devices.

Grid
An introduction to using Semantic UI grids

Attached Content
Examples of content that can attach to other content

Bootstrap Migration
Examples of replacements for components found in Bootstrap

Pages

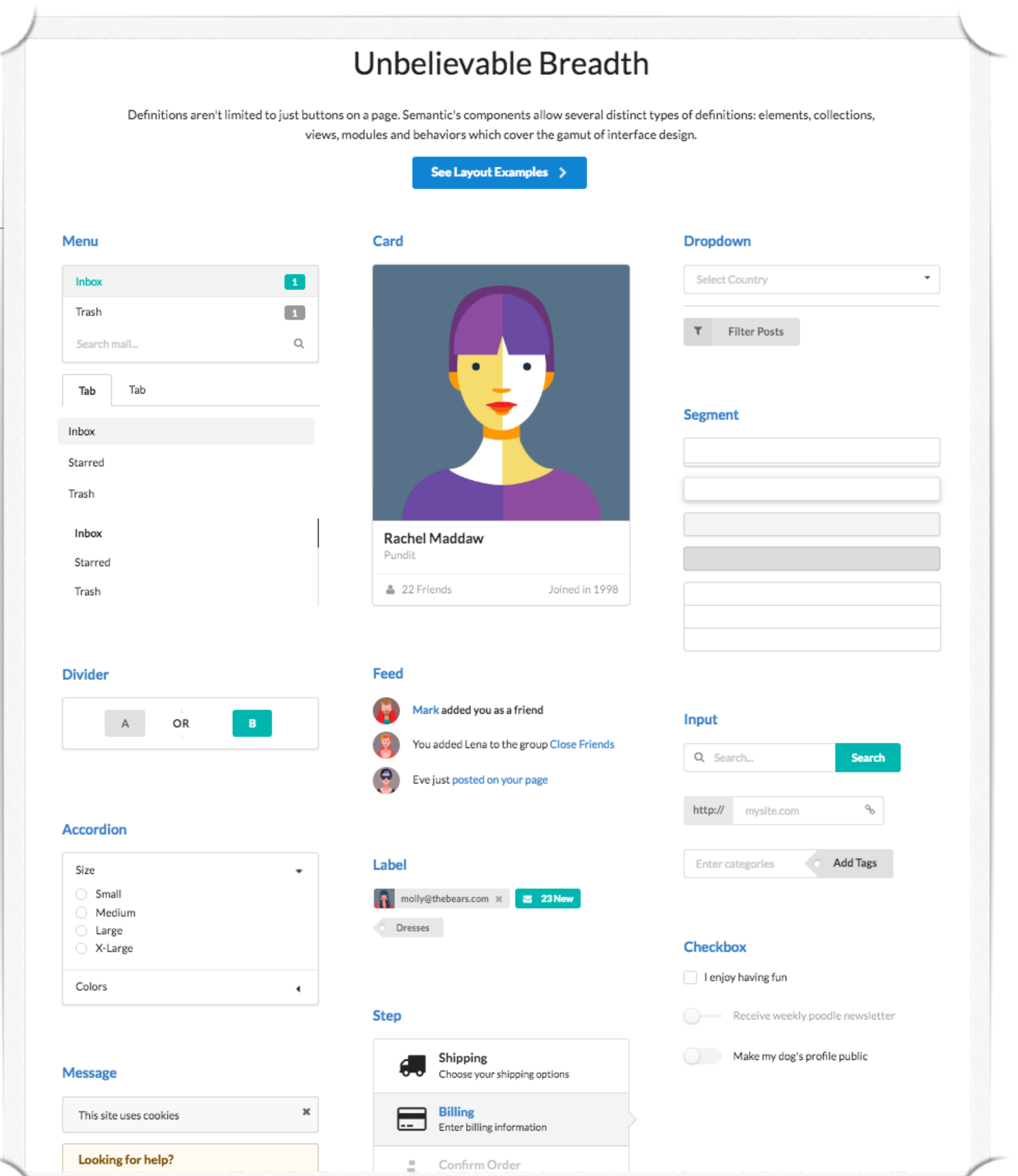
Starter page templates

Homepage
A simple, responsive homepage design with sidebar

Sticky Menus
Using visibility APIs to fix content after passing position in page

Fixed Menu
Using a fixed menu with page content

Login Form
A full-screen login form.



Unbelievable Breadth

Definitions aren't limited to just buttons on a page. Semantic's components allow several distinct types of definitions: elements, collections, views, modules and behaviors which cover the gamut of interface design.

Menu

Card

Rachel Maddaw
Pundit
22 Friends
Joined in 1998

Dropdown

Segment

Divider

Feed

Mark added you as a friend

You added Lena to the group Close Friends

Eve just posted on your page

Input

Accordion

Size
○ Small
○ Medium
○ Large
○ X-Large

Colors

Label

Checkbox

□ I enjoy having fun

Receive weekly poodle newsletter

Make my dog's profile public

Step

Shipping
Choose your shipping options

Billing
Enter billing information

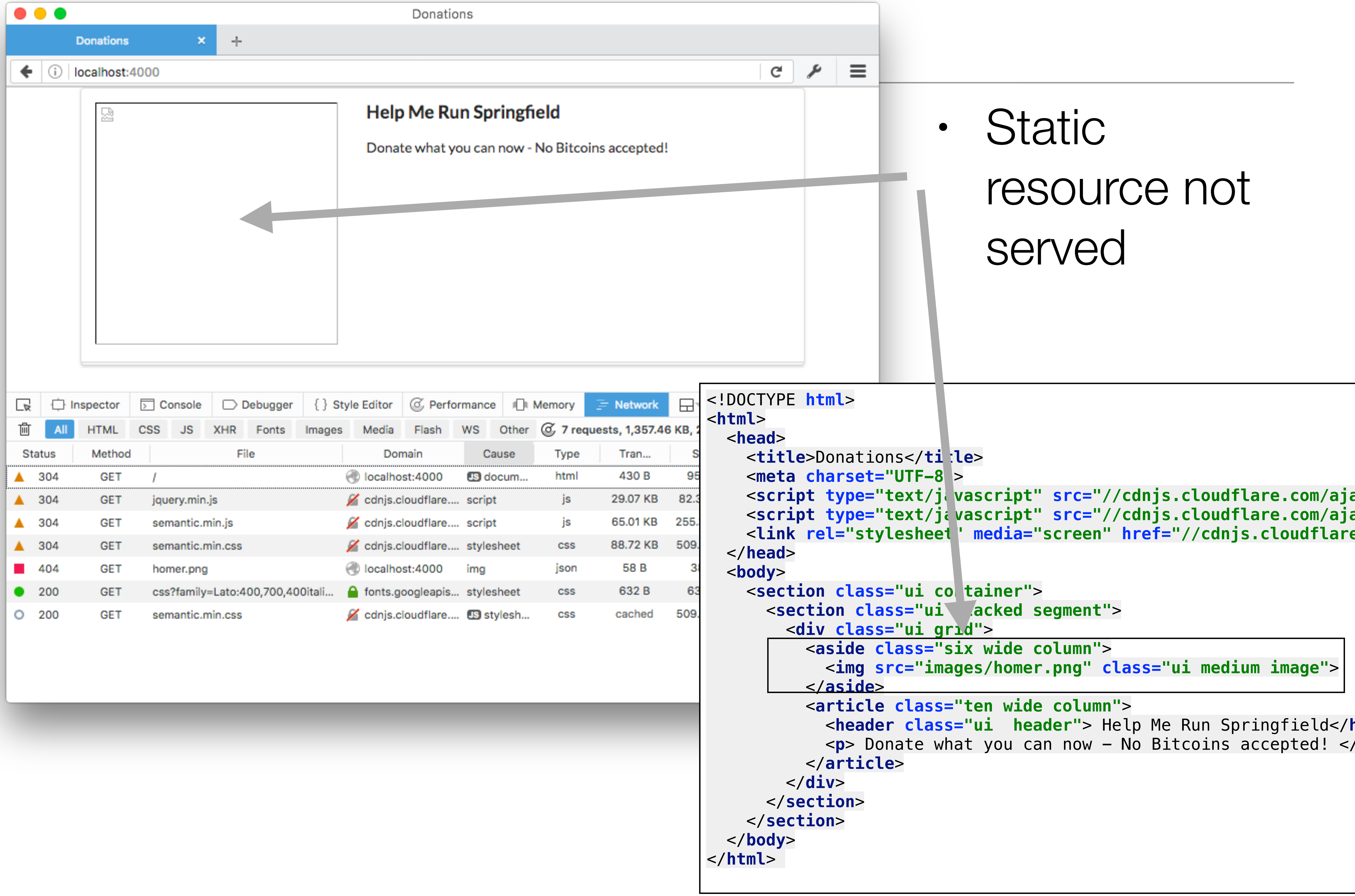
Confirm Order

Message

This site uses cookies ×

Looking for help?

- Static resource not served



Help Me Run Springfield

Donate what you can now - No Bitcoins accepted!

7 requests, 1,357.46 KB, 2

Status	Method	File	Domain	Cause	Type	Tran...	S
▲ 304	GET	/	localhost:4000	JS docum...	html	430 B	95
▲ 304	GET	jquery.min.js	cdnjs.cloudflare....	script	js	29.07 KB	82.3
▲ 304	GET	semantic.min.js	cdnjs.cloudflare....	script	js	65.01 KB	255.
▲ 304	GET	semantic.min.css	cdnjs.cloudflare....	stylesheet	css	88.72 KB	509.
■ 404	GET	homer.png	localhost:4000	img	json	58 B	3
● 200	GET	css?family=Lato:400,700,400itali...	fonts.googleapis...	stylesheet	css	632 B	63
○ 200	GET	semantic.min.css	cdnjs.cloudflare....	JS stylesh...	css	cached	509.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Donations</title>
    <meta charset="UTF-8">
    <script type="text/javascript" src="//cdnjs.cloudflare.com/aja
    <script type="text/javascript" src="//cdnjs.cloudflare.com/aja
    <link rel="stylesheet" media="screen" href="//cdnjs.cloudflare
  </head>
  <body>
    <section class="ui container">
      <section class="ui stacked segment">
        <div class="ui grid">
          <aside class="six wide column">
            <img src="images/homer.png" class="ui medium image">
          </aside>
          <article class="ten wide column">
            <header class="ui  header"> Help Me Run Springfield</h
            <p> Donate what you can now - No Bitcoins accepted! </
          </article>
        </div>
      </section>
    </section>
  </body>
</html>
```

Static Assets

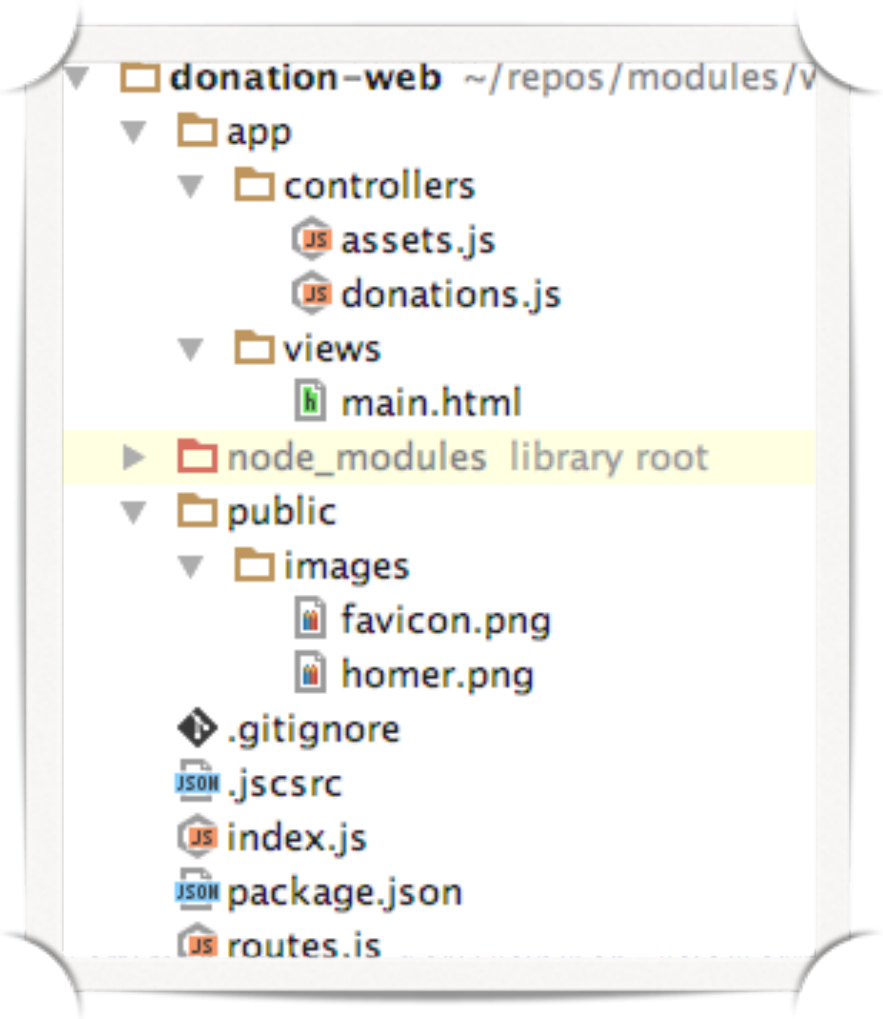
- additional route + handler

routes.js

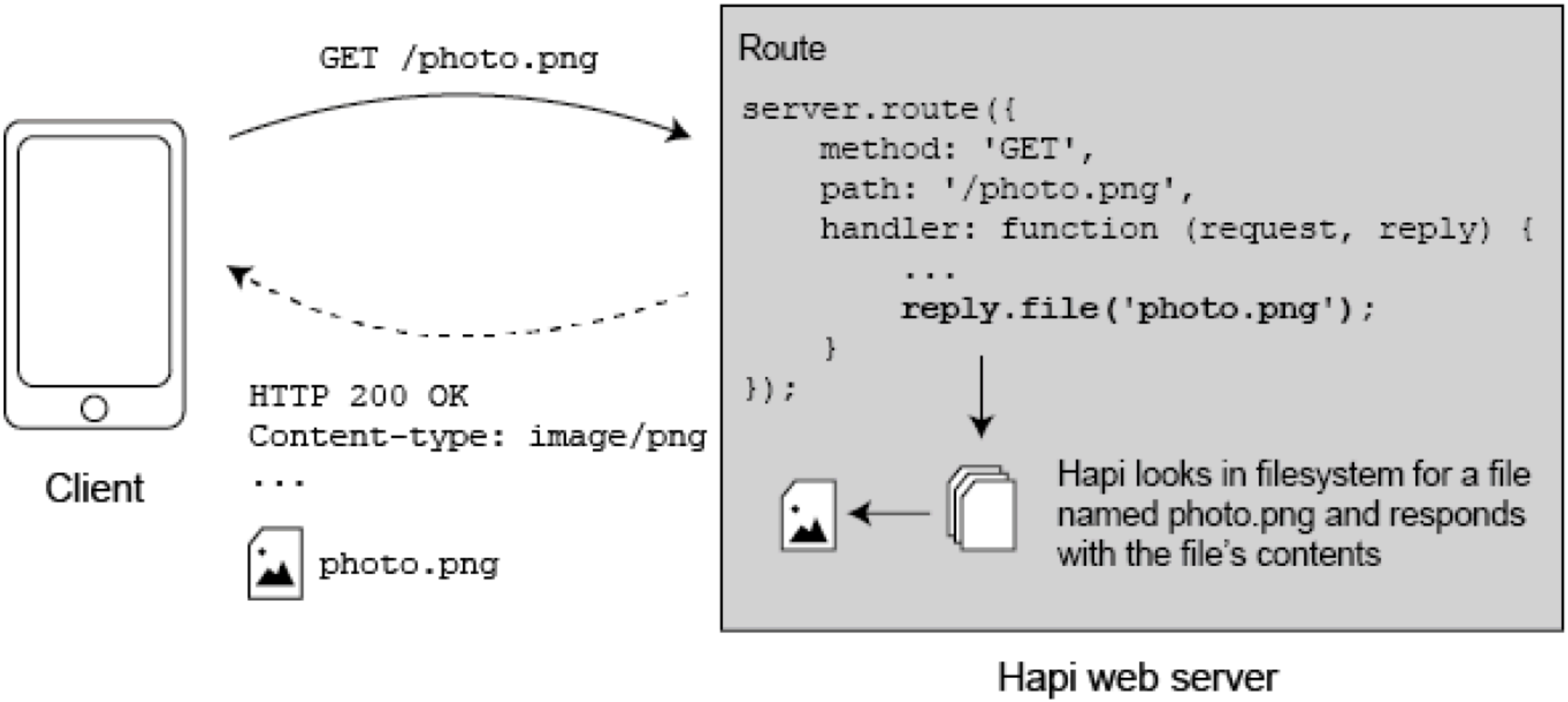
```
const Donations = require('./app/controllers/donations');
const Assets = require('./app/controllers/assets');

module.exports = [

  { method: 'GET', path: '/', config: Donations.home },

  {
    method: 'GET',
    path: '/{param*}',
    config: { auth: false },
    handler: Assets.servePublicDirectory,
  },

];
```

app/controllers/assets.js

```
'use strict';

exports.servePublicDirectory = {
  directory: {
    path: 'public',
  },

};
```



```
donation-web ~/repos/modules/v
  app
    controllers
      assets.js
      donations.js
    views
      main.html
  node_modules library root
  public
    images
      favicon.png
      homer.png
  .gitignore
  .jscsrc
  index.js
  package.json
  routes.js
```



GET /photo.png

Route

```
server.route({
    method: 'GET',
    path: '/photo.png',
    handler: function (request, reply) {
        ...
        reply.file('photo.png');
    }
});
```

Hapi looks in filesystem for a file named photo.png and responds with the file's contents

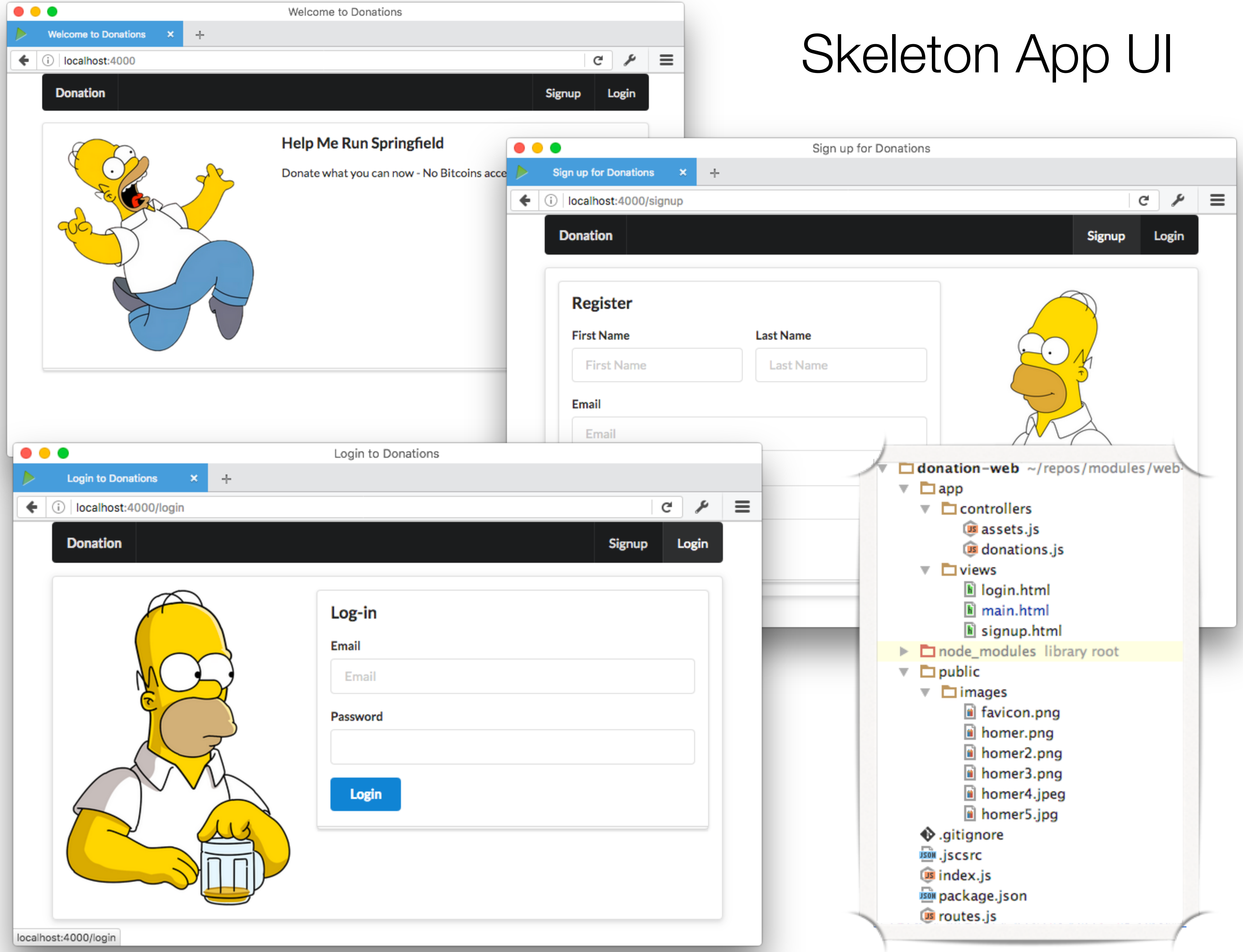
```
HTTP 200 OK
Content-type: image/png
...
```

photo.png

Client

Hapi web server

Skeleton App UI



Help Me Run Springfield

Donate what you can now - No Bitcoins acce

Register

First Name: First Name

Last Name: Last Name

Email: Email

Log-in

Email: Email

Password:

```
▼ donation-web ~/repos/modules/web-
  ▼ app
    ▼ controllers
        assets.js
        donations.js
    ▼ views
        login.html
        main.html
        signup.html
  ▶ node_modules library root
  ▼ public
    ▼ images
        favicon.png
        homer.png
        homer2.png
        homer3.png
        homer4.jpeg
        homer5.jpg
    .gitignore
    .jscsrc
    index.js
    package.json
    routes.js
```

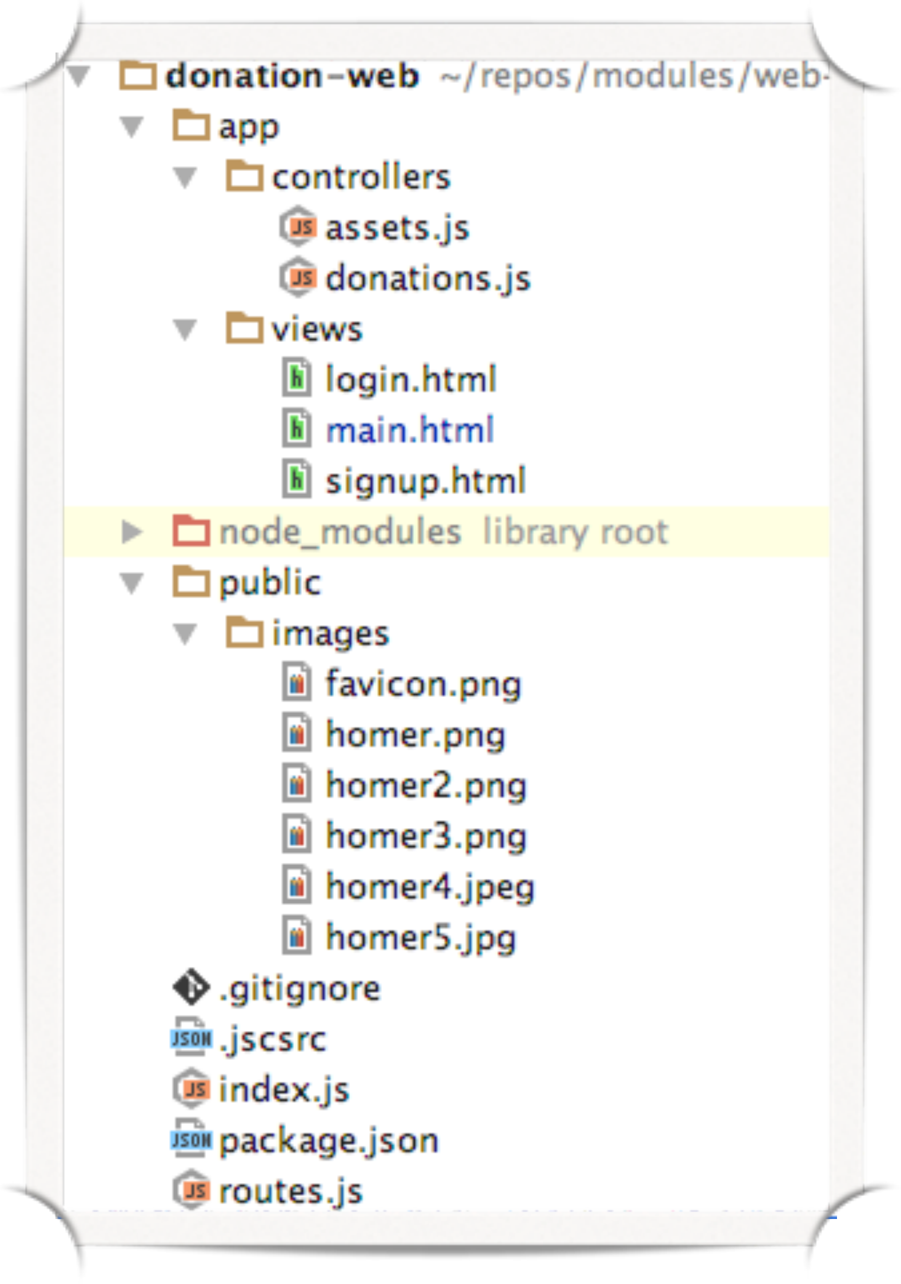
Project Structure

routes.js

```
const Donations = require('./app/controllers/donations');
const Assets = require('./app/controllers/assets');

module.exports = [

  { method: 'GET', path: '/', config: Donations.home },
  { method: 'GET', path: '/signup', config: Donations.signup },
  { method: 'GET', path: '/login', config: Donations.login },

  {
    method: 'GET',
    path: '/{param*}',
    config: { auth: false },
    handler: Assets.servePublicDirectory,
  },

];
```

app/controllers/donations.js

```
'use strict';

exports.home = {

  handler: (request, reply) => {
    reply.file('./app/views/main.html');
  },

};

exports.signup = {

  handler: (request, reply) => {
    reply.file('./app/views/signup.html');
  },

};

exports.login = {

  handler: (request, reply) => {
    reply.file('./app/views/login.html');
  },

};
```



```
donation-web ~/repos/modules/web-
  app
    controllers
      assets.js
      donations.js
    views
      login.html
      main.html
      signup.html
  node_modules library root
  public
    images
      favicon.png
      homer.png
      homer2.png
      homer3.png
      homer4.jpeg
      homer5.jpg
  .gitignore
  .jscsrc
  index.js
  package.json
  routes.js
```