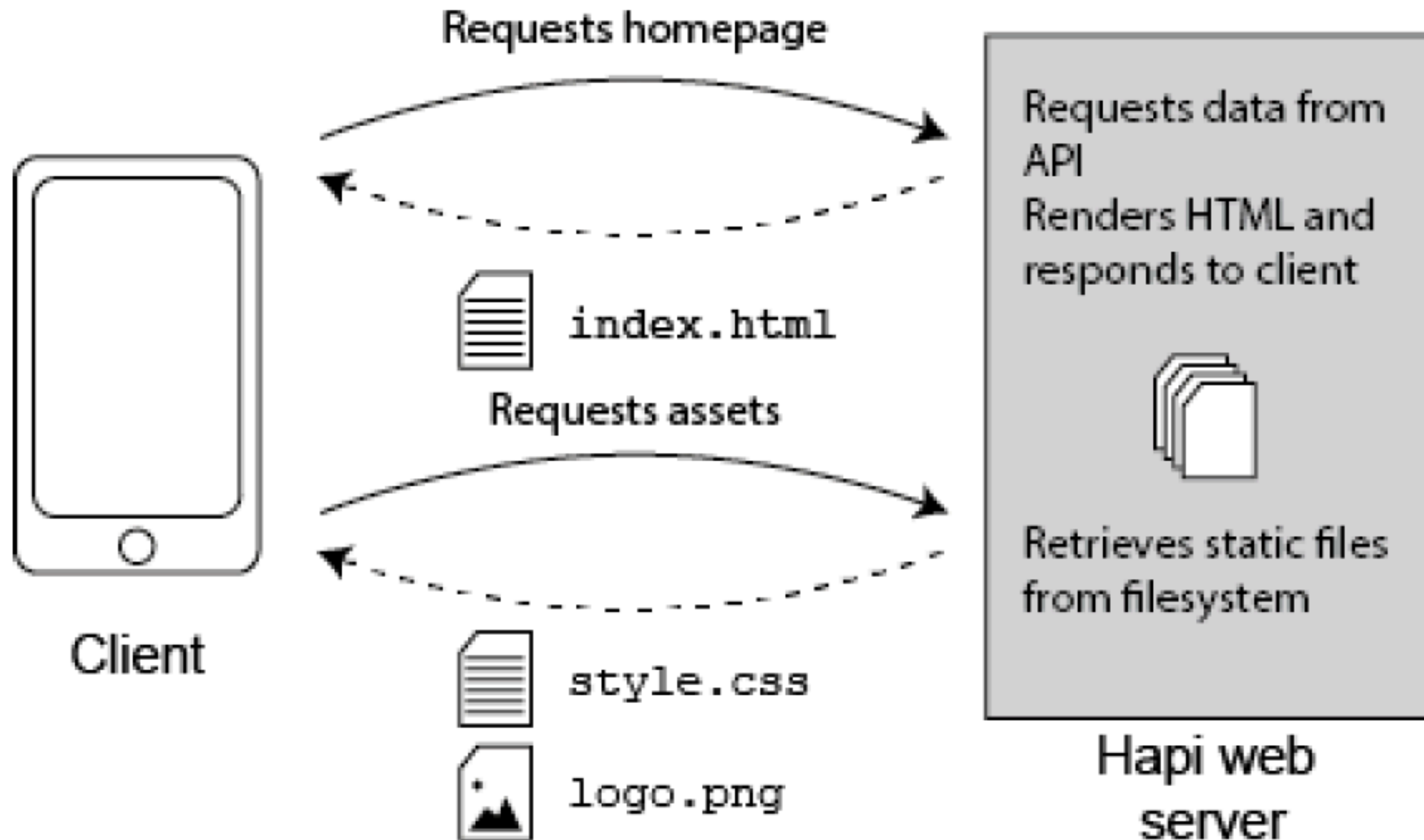


Views

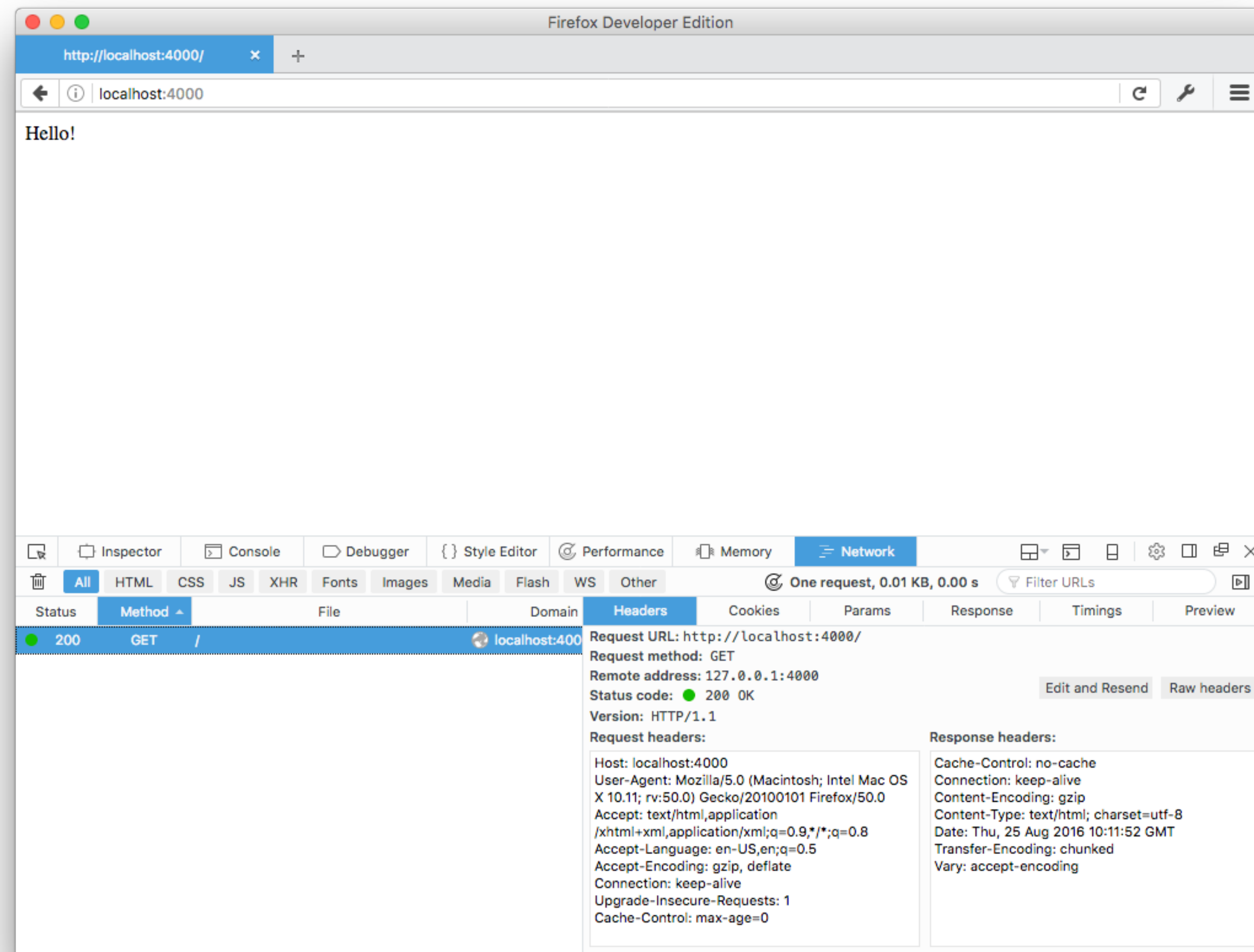
Application Structure



return

- Responds to the browser with a simple string.

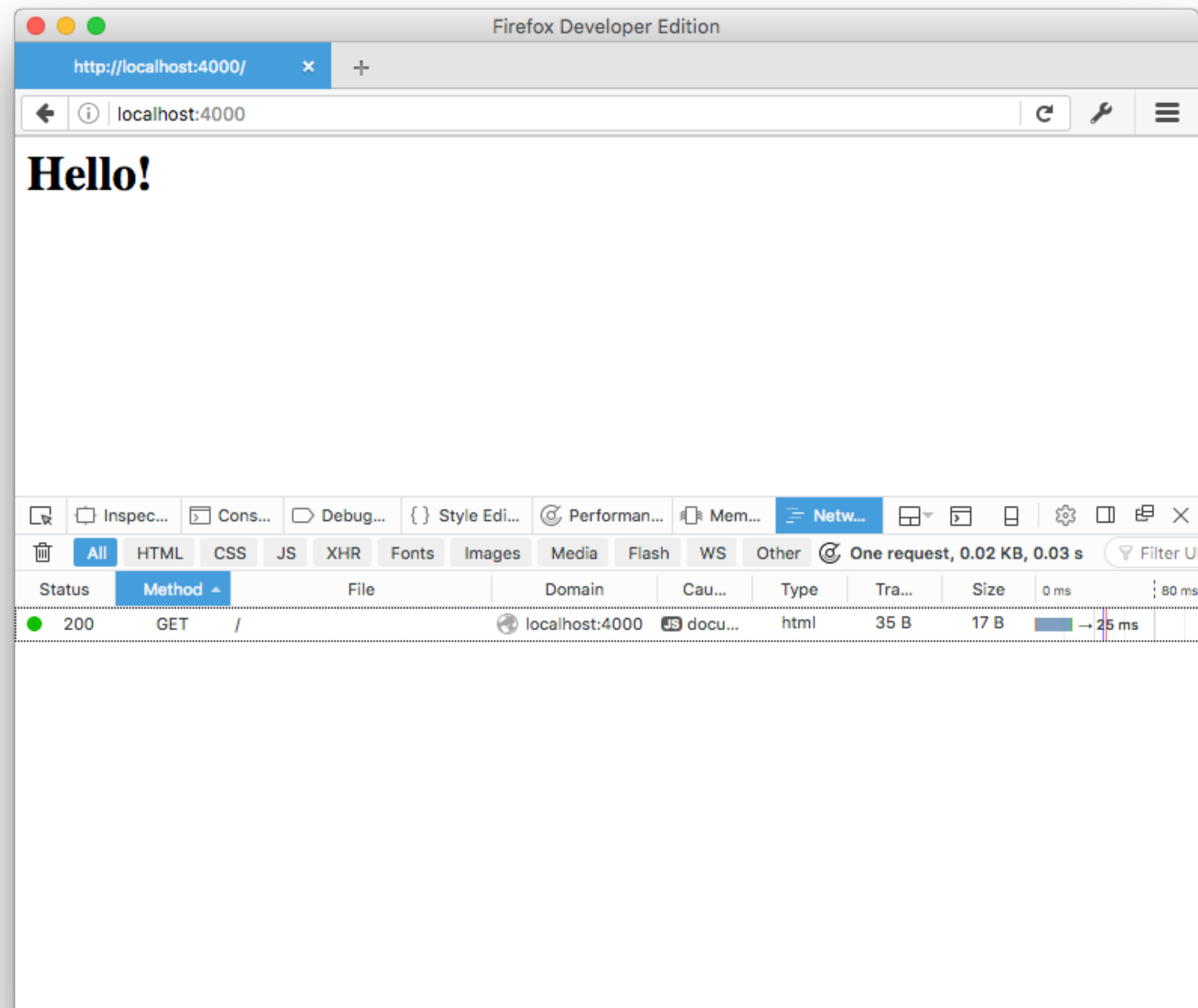
```
exports.index = {  
  handler: function(request, h) {  
    return 'Hello!';  
  }  
};
```



Return html

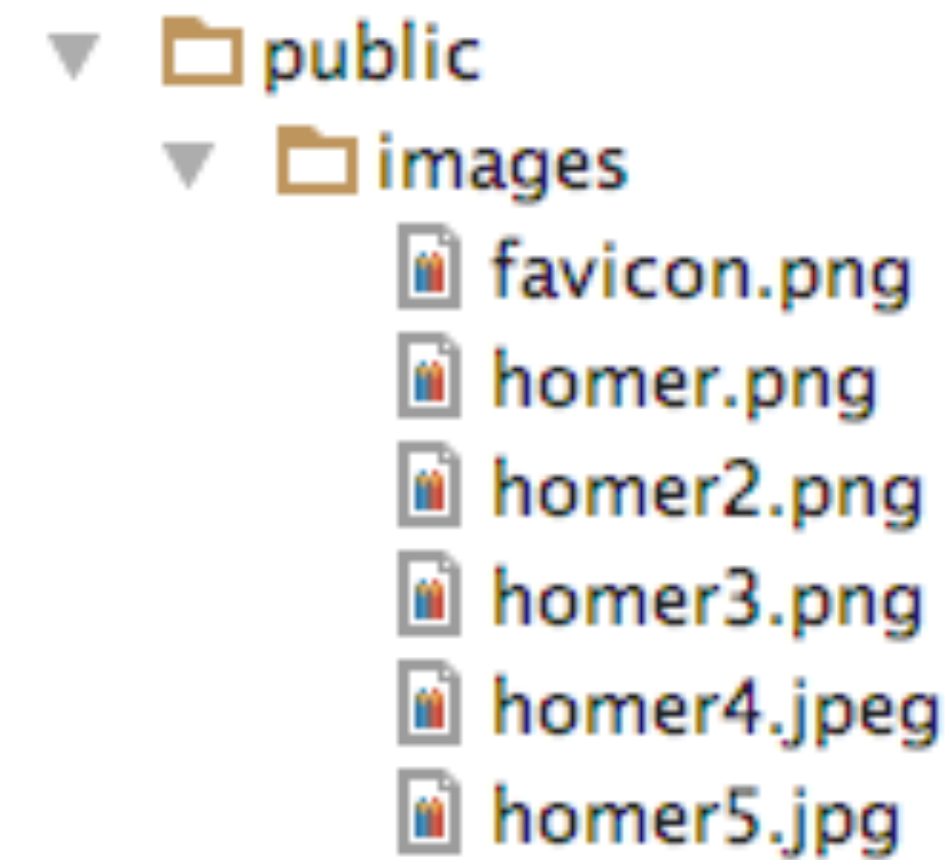
- In order to render web pages we could pass html content
- This would become very unwieldy and unmaintainable

```
exports.index = {  
  handler: function (request, h) {  
    return('<h1> Hello! </h1>');  
  }  
};
```



Static Assets

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



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

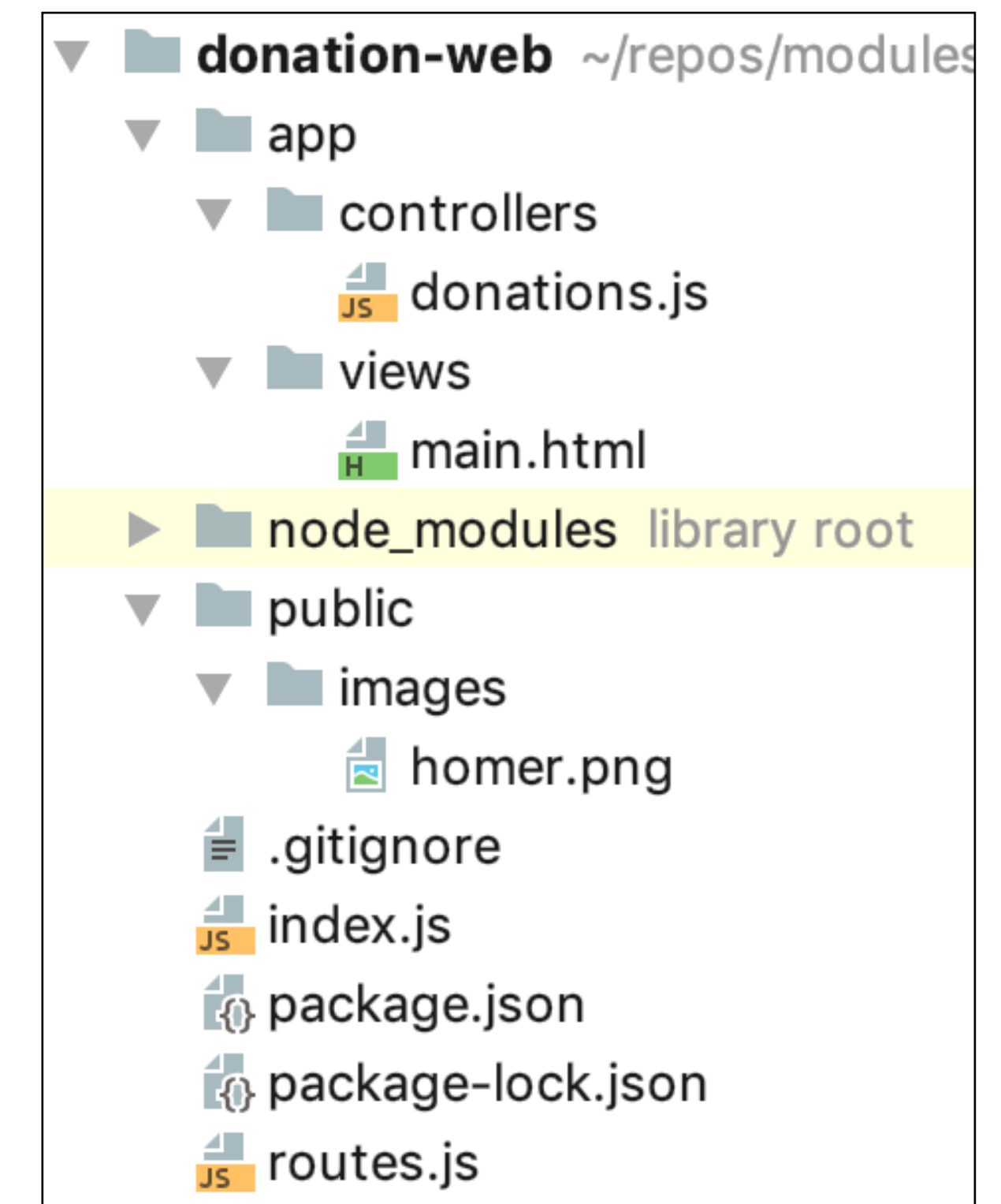
controllers will contain handlers

views will contain html templates

public images, stylesheets..

index.js is the application entry point

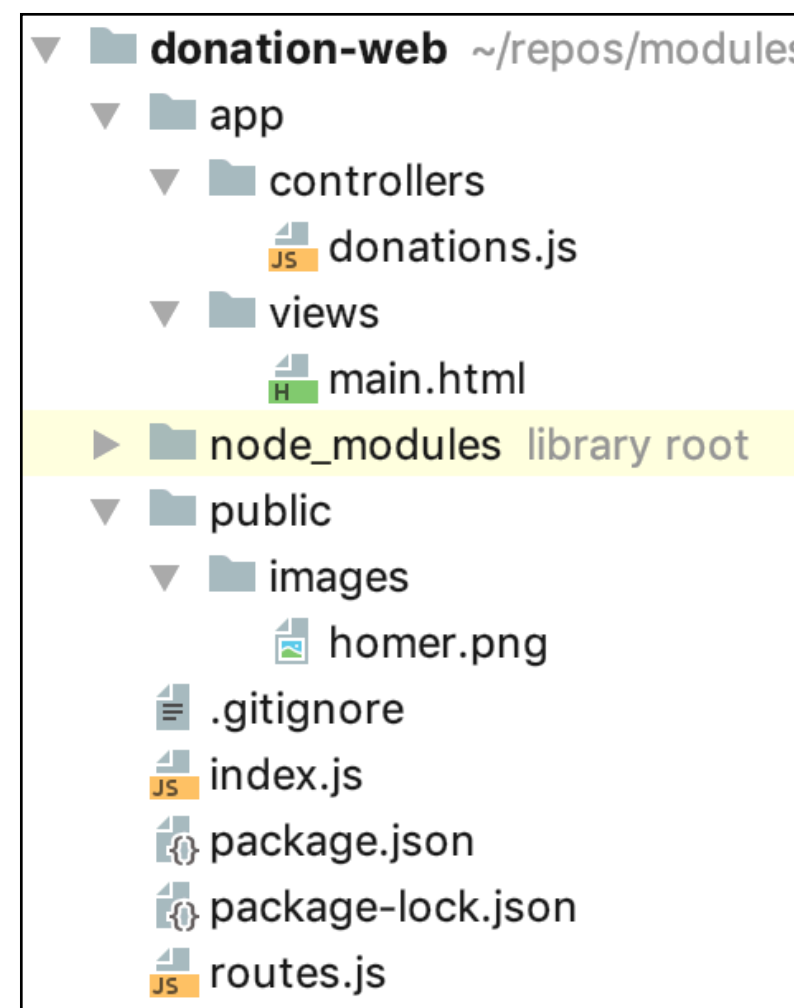
routes.js define all application routes



Main Application Entrypoint

index.js

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



```
'use strict';

const Hapi = require('hapi');

const server = Hapi.server({
  port: 3000,
  host: 'localhost'
});

async function init() {
  server.route(require('./routes'));
  await server.start();
  console.log(`Server running at: ${server.info.uri}`);
}

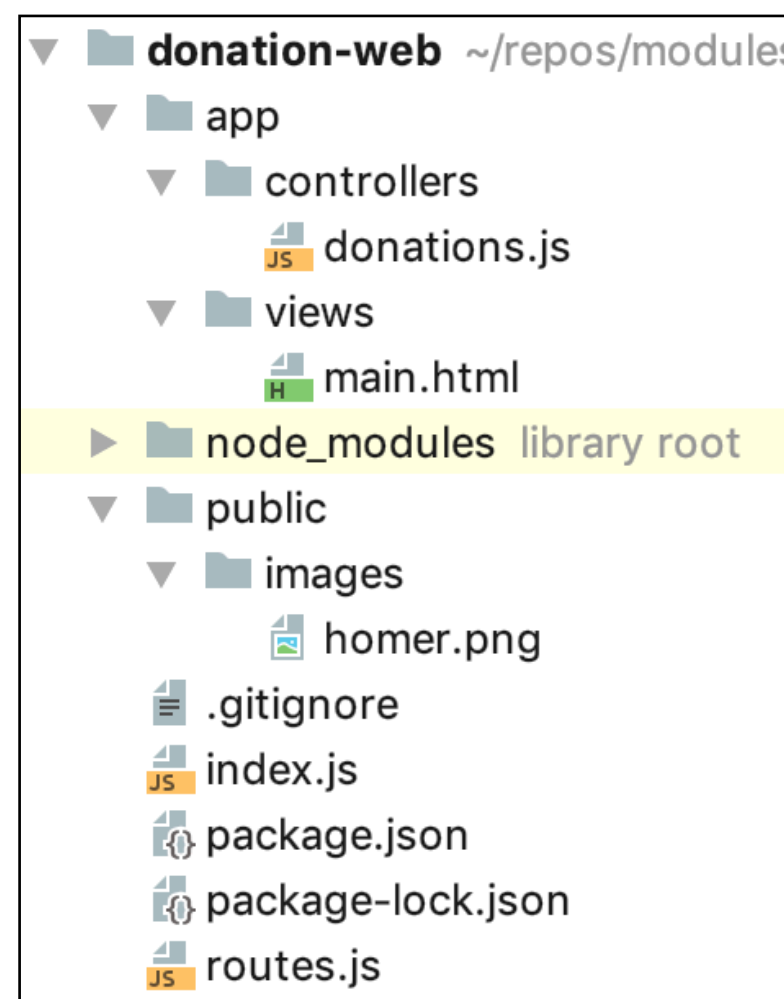
process.on('unhandledRejection', err => {
  console.log(err);
  process.exit(1);
});

init();
```

\$ npm install inert

Routes + Controller

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



routes.js

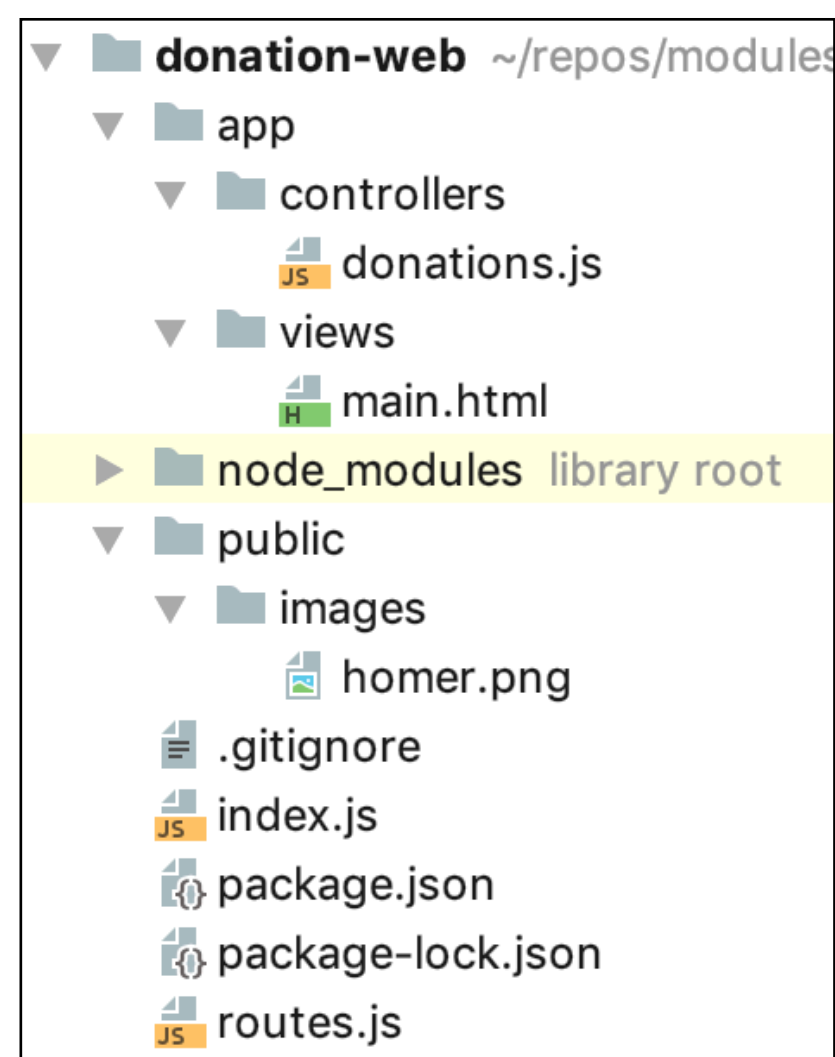
```
const Donations = require('./app/controllers/donations');  
  
module.exports = [  
  { method: 'GET', path: '/', config: Donations.home },  
];
```

app/controllers/donation.js

```
'use strict';  
  
const Donations = {  
  index: {  
    handler: async function(request, h) {  
      return h.file('./app/views/main.html');  
    }  
  }  
};  
  
module.exports = Donations;
```



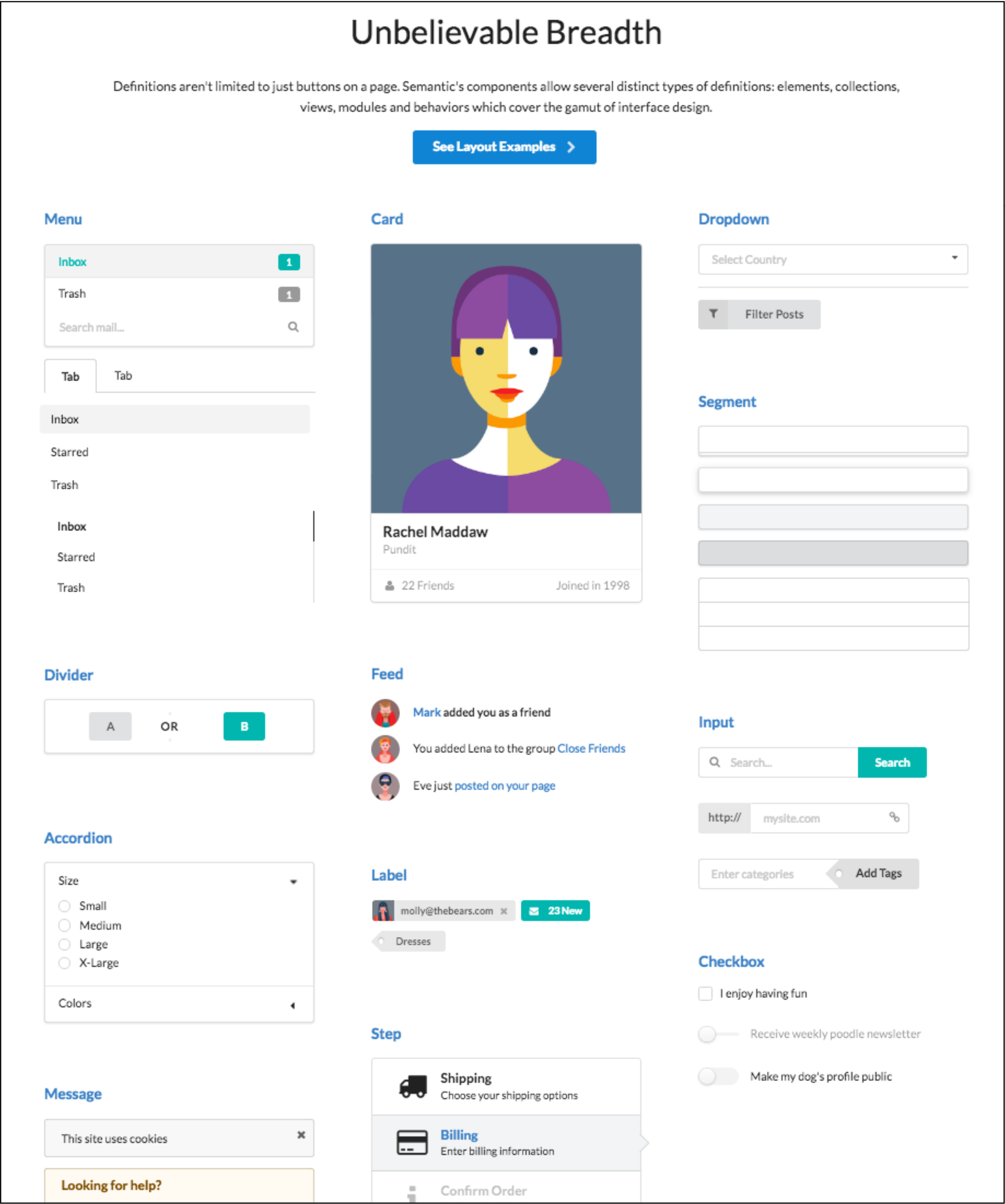
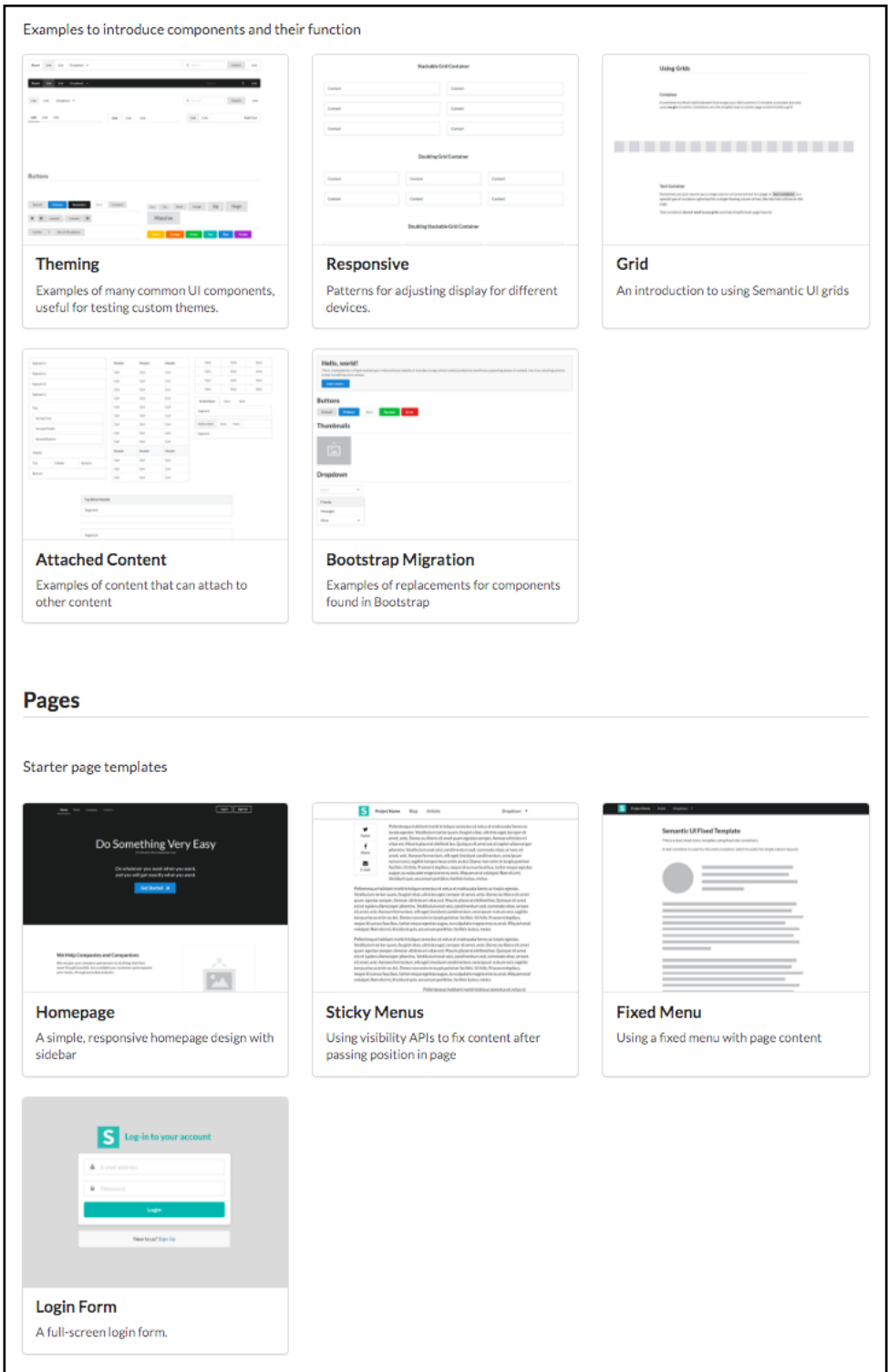
```
<!DOCTYPE html>
<html>
  <head>
    <title>Donations</title>
    <meta charset="UTF-8" />
    <script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
    <script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.4.1/semantic.min.js"></script>
    <link rel="stylesheet" media="screen" href="//cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.4.1/semantic.min.css" />
  </head>
  <body>
    <section class="ui container">
      <section class="ui stacked segment">
        <div class="ui grid">
          <aside class="six wide column"></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>
```

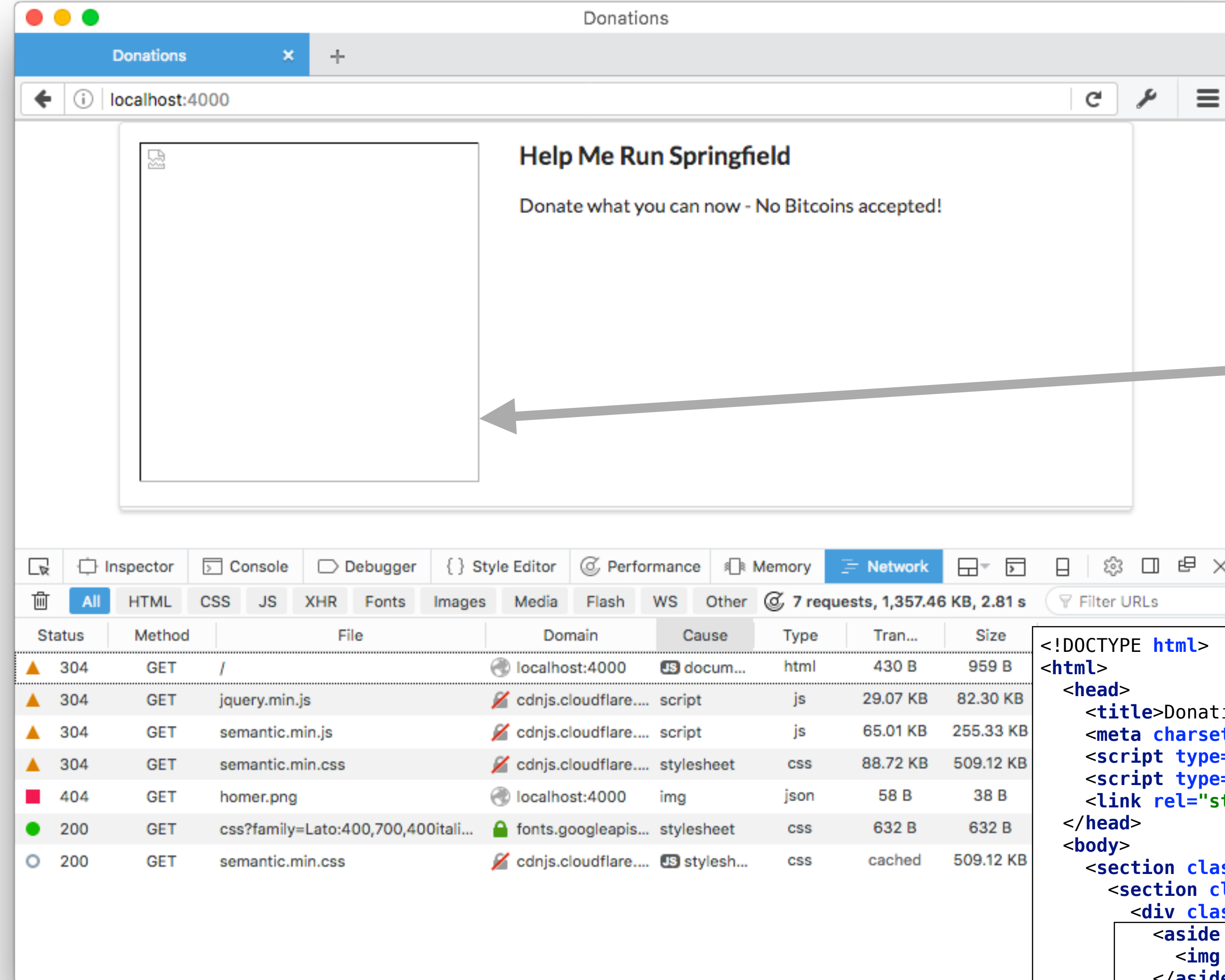


- Simple, static web page.
- Loads Semantic UI Stylesheet from CDN.

Semantic UI

- Rich, themeable, attractive layouts & UI components





- Static resource not served

```
<!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.m
    <script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.1.6/sem
    <link rel="stylesheet" media="screen" href="//cdnjs.cloudflare.com/ajax/libs/semantic-ui/2
  </head>
  <body>
    <section class="ui container">
      <section class="ui stacked segment">
        <div class="ui grid">
          <aside class="six wide column">
            
          </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>
```


Plugins Example

<https://github.com/hapijs/inert>

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 adding a `h.file()` method to the [toolkit](#), which can respond with file based resources.

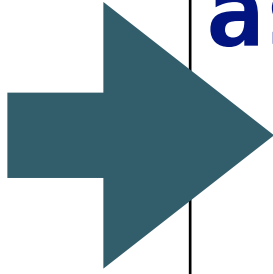
Features

- 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

- [Examples](#)
 - [Static file server](#)
 - [Serving a single file](#)
 - [Customized file response](#)
- [Usage](#)
 - [Server options](#)
 - `h.file(path, [options])`
 - [The file handler](#)
 - [The directory handler](#)
 - [Errors](#)

Plugin Configuration & Registration



```
...
async function init() {
  await server.register(require('inert'));
  server.route(require('./routes'));
  await server.start();
  console.log(`Server running at: ${server.info.uri}`);
}
...
```

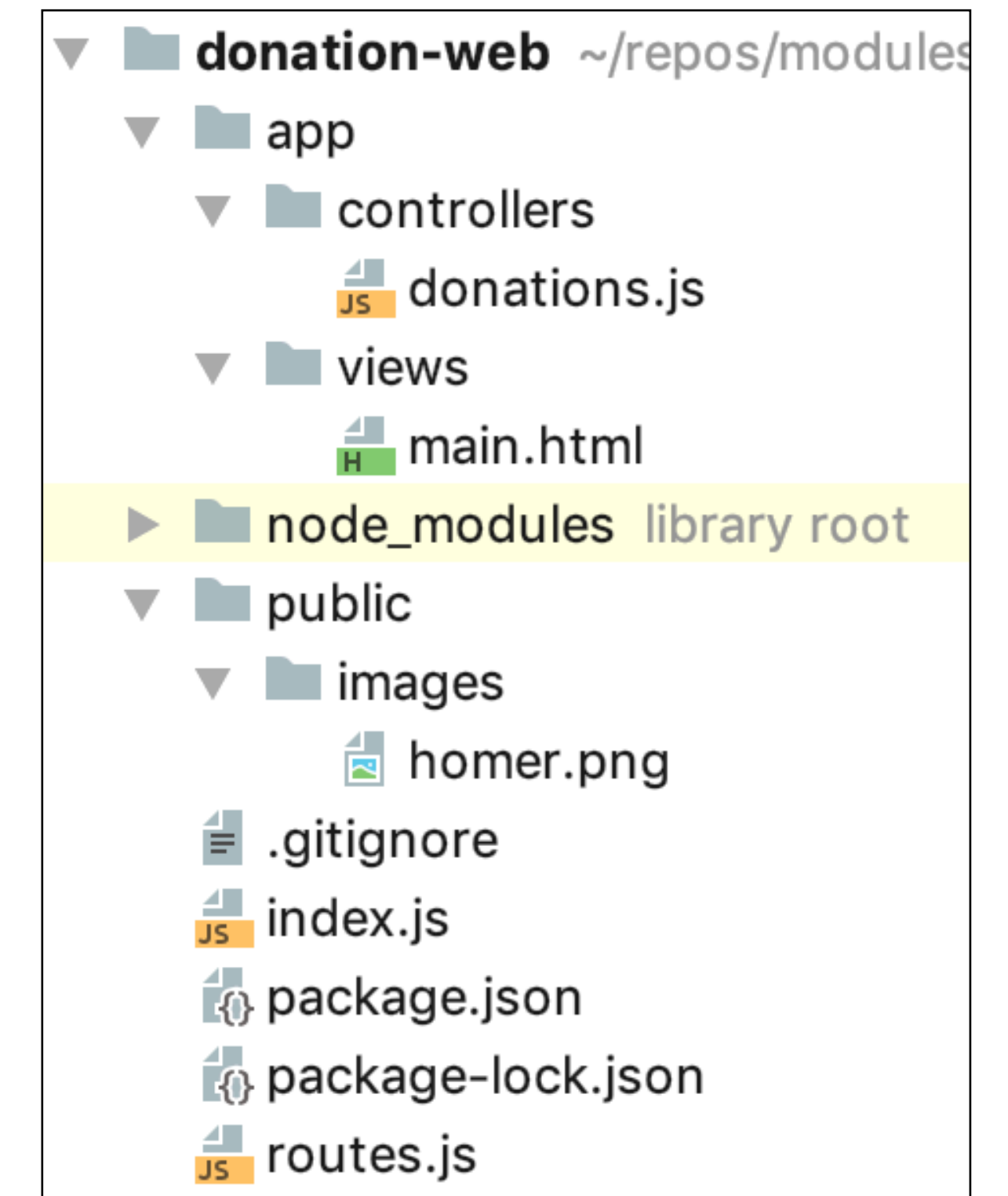
Static Assets

routes.js

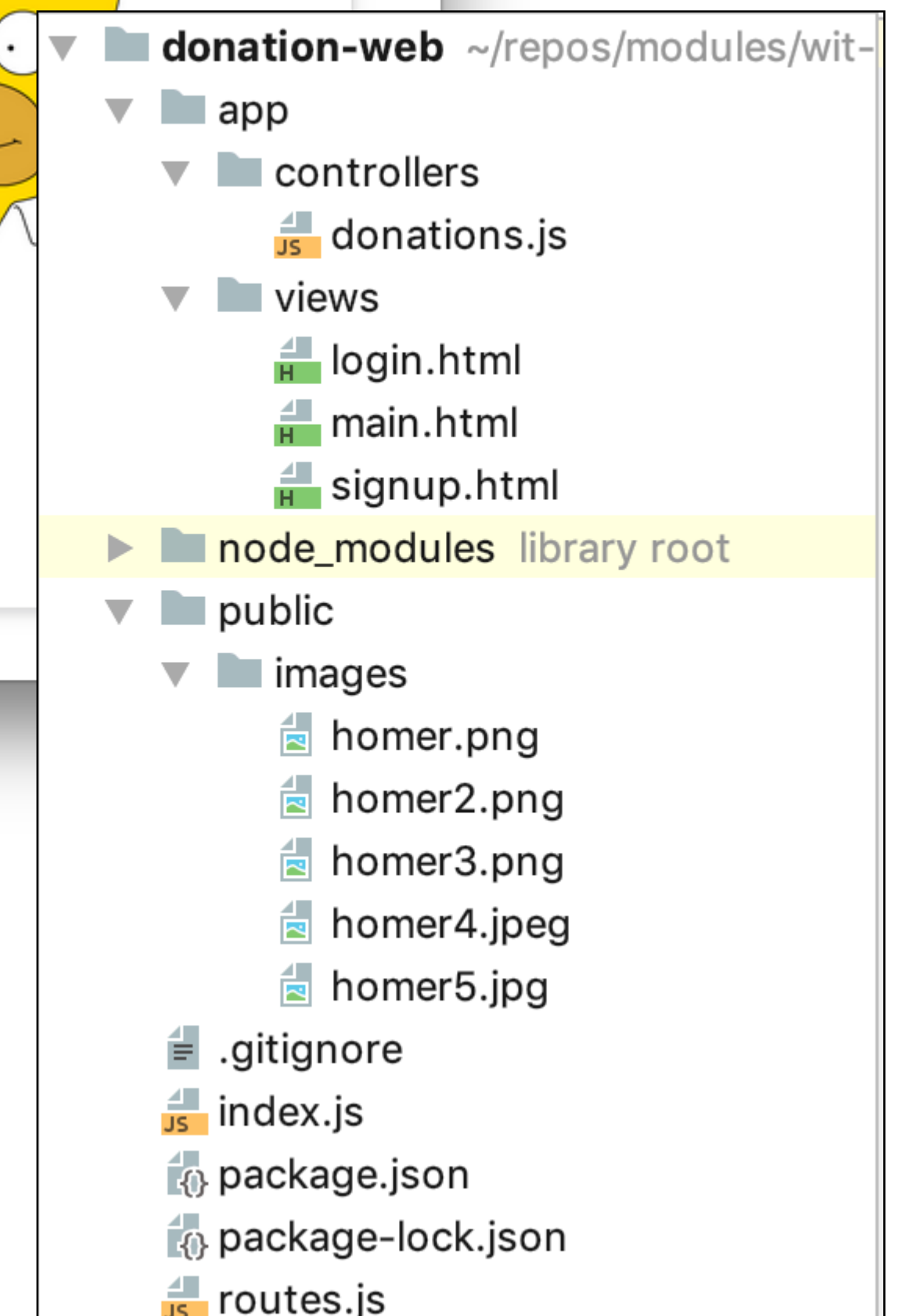
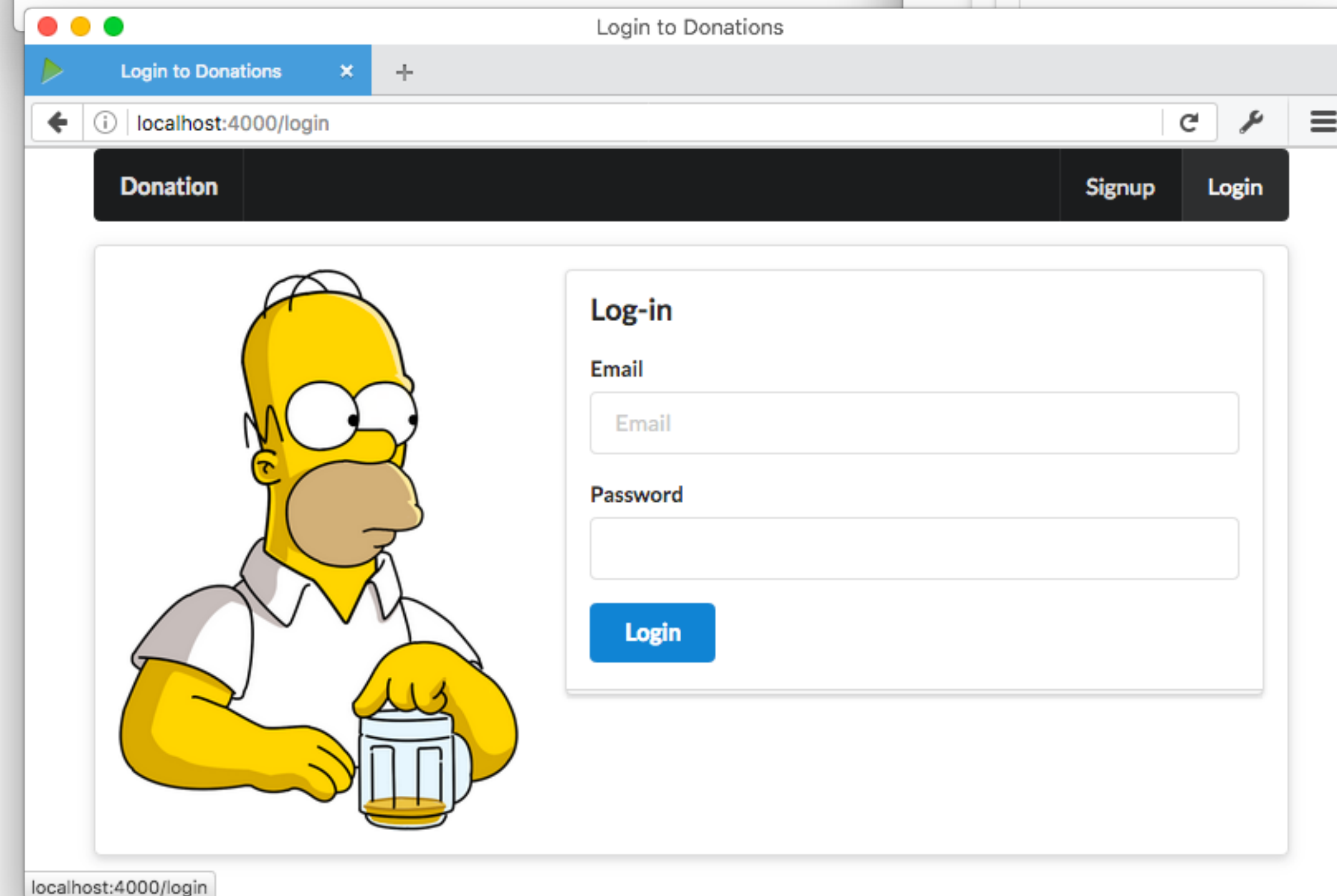
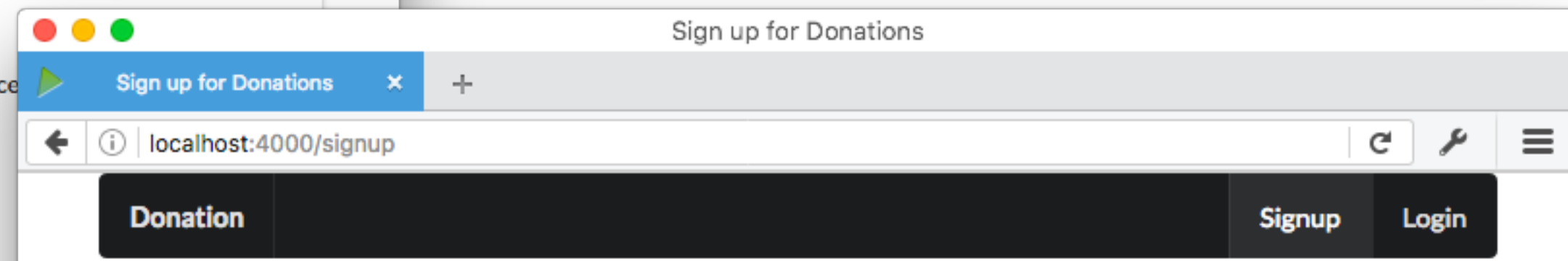
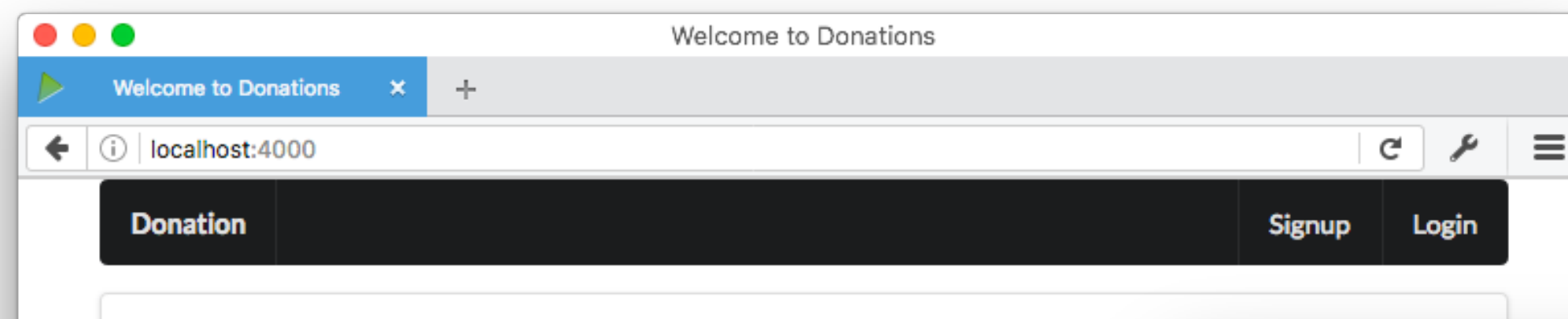
```
const Donations = require('./app/controllers/donations');

module.exports = [
  { method: 'GET', path: '/', config: Donations.index },
  {
    method: 'GET',
    path: '/{param*}',
    handler: {
      directory: {
        path: './public'
      }
    }
  }
];
```

- additional route + handler



Skeleton App UI




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

module.exports = [
  { method: 'GET', path: '/', config: Donations.index },
  { method: 'GET', path: '/signup', config: Donations.signup },
  { method: 'GET', path: '/login', config: Donations.login },
  {
    method: 'GET',
    path: '/{param*}',
    handler: {
      directory: {
        path: './public'
      }
    }
  }
];
```

```
'use strict';

const Donations = {
  index: {
    handler: function(request, h) {
      return h.file('./app/views/main.html');
    }
  },
  signup: {
    handler: function(request, h) {
      return h.file('./app/views/signup.html');
    }
  },
  login: {
    handler: function(request, h) {
      return h.file('./app/views/login.html');
    }
  }
};

module.exports = Donations;
```

Project Structure

