
http-framework

A web framework based purely on `require('http')`

Table of Contents *generated with DocToc*

- http-framework
 - Motivation
 - [Modules][20]
 - ★ For a complete list of [Modules check out the wiki][20]
 - [Documentation][26]
 - ★ [HTTP workshop][27]
 - ★ module docs
 - ★ rest of the modules
 - [Examples][17]
 - ★ [express](#)[18] inspired examples
 - ★ [hapi](#)[24] inspired examples
 - ★ [connect](#)[25] inspired examples
 - Todo
 - Installation
 - Contributors
 - MIT Licenced

Motivation

`require('http')` is a web framework. All you need are a few small modules that do one thing well when you are building your web application route handlers.

This module aggregates and documents (with examples) a selection of small modules that can be used to achieve this goal.

It's recommended you check out the:

- the wiki of modules
- the documentation
- the examples

I do not recommend you use this “framework”. You should check out the small modules and use them directly. Use the list of examples here for inspiration.

http-framework is an OPEN Open Source Project, see the Contributing section to find out what this means.

Modules

Check out the modules folder for an example of small modules you can combine to build your own custom “framework”

See the [package.json](#) dependencies hash for an example of many small modules used in the examples folder of this project.

For a complete list of Modules check out the wiki

Documentation

HTTP workshop

For a tutorial / workshop see [http-works](#)

module docs

- [config-chain](#)
- [cookies](#)
- [corsify](#)
- [csrf-lite](#)
- [filed](#)
- [form-body](#)
- [send-json](#)
- [send-html](#)

rest of the modules

```
1 // TODO
```

For now see the examples folder

Examples

express inspired examples

These examples are clones of [express](#) examples demonstrating how to author web apps without frameworks.

- **auth** An example demonstrating how to login and authenticate users
- **auth-helpers** An example demonstrating how you can restrict certain routes in your apps to only be accessed by certain types of users
- **content-negotiation** An example demonstrating how you can return different types of content based on what the user asks for in his Accept header.
- **cookie-sessions** An example of storing session information in a users cookie
- **cookies** An example of setting a cookie to track a user
- **cors** An example of adding cors support to your HTTP server
- **downloads** An example of allowing users to download files from your server
- **error** An example of handling errors in your HTTP server
- **error-pages** An example of rendering custom 500 and 404 pages in your web server
- **expose-data-to-client** An example of putting server side state into your template so that it can be accessed from browser javascript
- **hello-world** A simple hello world example
- **multipart** An example demonstrating file uploads and saving them to disk temporarily.
- **mvc** An over engineered example of how to structure a slightly larger web application. Includes configuration, modularity and databases. Take ideas from it, please do not copy paste it
- **online** An example of using redis and the [onl](#)ine module to track user presence
- **route-map** An example of a [map](#) utility function that can be used to structure your routes differently. This demonstrates you can do whatever you want, if you like it, do it.
- **route-seperation** An example of spreading your route handlers over multiple files.
- **search** An example of doing a database query over XHR with a web server backed by redis
- **session** An example of storing information in a session. The session is either backed by redis or memory.
- **static-files** An example of serving static files for your web server
- **vhost** An example of handling multiple sub domains as seperate applications in a singlue web server
- **web-service** An example JSON REST api web server. Contains API key authentication and error handling

Credit for the applications goes to

- @tj & express maintainers

hapi inspired examples

These examples are clones of [hapi](#) examples demonstrating how to author web apps without frameworks.

- **tail** An example of handling async errors in your applications after you have ended the HTTP response
- **validate-api** An example of how to add validation logic to your HTTP route handlers. Both validating query params & the HTTP body

Credit for the application's goes to

- @hapijs & hapi maintainers

connect inspired examples

These examples are clones of [connect](#) examples demonstrating how to author web apps without frameworks.

- **basic-auth** An example of handling basic authorization style authentication of a web server.
- **csrf** An example of adding csrf security to a form in a web application

Credit for the applications goes to

- @tj & connect maintainers

Todo

- ☒ Finish porting express examples
- ☐ continue porting hapi examples
- ☐ continue connect examples
- ☐ Port koajs examples
- ☐ Port restify examples
- ☐ Port partial.js examples
- ☐ Finish documentation
- ☐ Get community feedback
- ☐ Author new fresh examples

Installation

```
npm install http-framework
```

Contributors

- Raynos
- Matt-Esch
- maxogden

MIT Licenced