Tutorials

Languages: en-US

Getting Started

This tutorial is compatible with hapi v17 and newer

Overview

This tutorial will show how to set up a basic hapi server that displays "Hello World!" in your browser.

Installing hapi

Create a new directory `myproject`, and from there:

- Run: `cd myproject`, this goes into the created project folder.
- Run: `npm init` and follow the prompts. This will generate a package.json file for you.
- Run: `npm install @hapi/hapi`, this will install the latest version of hapi as a dependency in your package.json.

Creating a Server

A very basic hapi server looks like the following:

```
'use strict';

const Hapi = require('@hapi/hapi');

const init = async () => {

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

    await server.start();
    console.log('Server running on %s', server.info.uri);
};

process.on('unhandledRejection', (err) => {

    console.log(err);
    process.exit(1);
});

init();
```

First, you require hapi. Then you initialize a new `Hapi.server()` with connection details containing a port number to listen on and the host information. After that you start the server and log that it's running.

When creating a server, you can provide a hostname, IP address, a Unix socket file, or Windows named pipe to bind the server to. For more details, see the API reference.

The `host` property set to `localhost` is likely the safest choice. In a docker container, however, the localhost may not be accessible outside of the container and using `host: '0.0.0.0'` may be needed.

Adding Routes

After you get the server up and running, its time to add a route that will display "Hello World!" in your browser.

```
'use strict';

const Hapi = require('@hapi/hapi');

const init = async () => {

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

    server.route({
        method: 'GET',
        path: '/',
        handler: (request, h) => {

            return 'Hello World!';
        }
    });

    await server.start();
    console.log('Server running on %s', server.info.uri);
};

process.on('unhandledRejection', (err) => {

    console.log(err);
    process.exit(1);
});

init();
```

Save the above as `index.js` and start the server with the command `node index.js`. Now you'll find that if you visit `http://localhost:3000` in your browser, you'll see the text 'Hello, World!'.

The `method` property can be any valid HTTP method, array of HTTP methods, or an asterisk to allow any method.

The `path` property defines the path including parameters. It can contain optional parameters, numbered parameters, and even wildcards. For more details, see the routing tutorial.

The `handler` function performs the main business logic of the route and sets the response. The `handler` must return a value, a promise, or throw an error.

Everything Else

hapi has many, many other capabilities and only a select few are documented in tutorials here. This tutorial was intentionally minimal, we highly recommend you to check out the plugins tutorial. It will give some more knowledge to better organize your hapi project. We also have other tutorials of interest, please use the list to your left to check them out. Everything else is documented in the API reference and, as always, feel free to ask questions on github or just visit us on slack.