

Handlebars in Node.js (tutorial) - Programming sage

 medium.com/programming-sage/handlebars-in-node-js-tutorial-a30a41fc6206

November 2,
2018

Handlebars and Node.js (tutorial)



When I decided to build my portfolio site the only backend language I knew was Node.js. So I started looking how can I generate dynamic HTML pages from my Node.js server. So, I stumbled upon HTML templating,

"what is HTML templating" you ask?

well, you know, how we type static HTML code to make websites? Now, imagine we want to change a few things on our page e.g. User's photo, Name, etc. We can do this using javascript(client-side) or we can generate an HTML file dynamically(server-side) upon parsing the request from user. The latter is faster and is known as HTML templating.

Popular contenders

Node.js w/ express supports a lot of HTML templating engines. Some popular choices are:

1. Pug
2. Mustache
3. Handlebars
4. Marko
5. EJS

Handlebars

After a lot of consideration, I chose Handlebars as the templating engine for my portfolio. It's minimal, powerful and blends well with express.

Configuration.

Let's start by installing a few dependencies. I hope you have node installed. If not, do so, [here](#). Nowadays, npm is shipped with Node.js. So, you should have both node and npm installed just by installing Node.js. Remember you can always check Node and NPM by using the following commands:

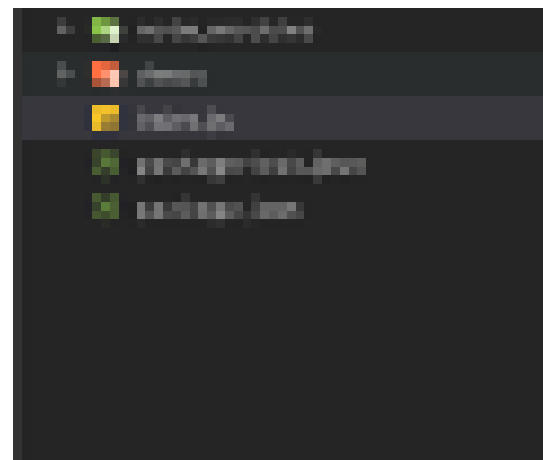
```
node -v
```

```
npm -v
```

let's create a project using `npm init`

or you can download a boilerplate [here](#). It should look something like this.

Boilerplate setup.



'Nuff talk, show me some code

1. Install **express** and save it as a dependency using `npm i --save express`
2. Install **hbs** and save it as a dependency using `npm i --save hbs`

3. In the **index.js** file import express and path dependencies as, `const app = require('express')() const path = require('path')`
4. Set views directory and views engine as Handlebars using, `app.set('views',path.join(__dirname,"views")) app.set("view engine","hbs")`
5. Setup a **GET** route at '/' using, `app.get('/',(req,res)=>{ res.render('index'); })`
6. Finally, setup port for our application using, `app.listen(5000);`
7. Our code should look something like,



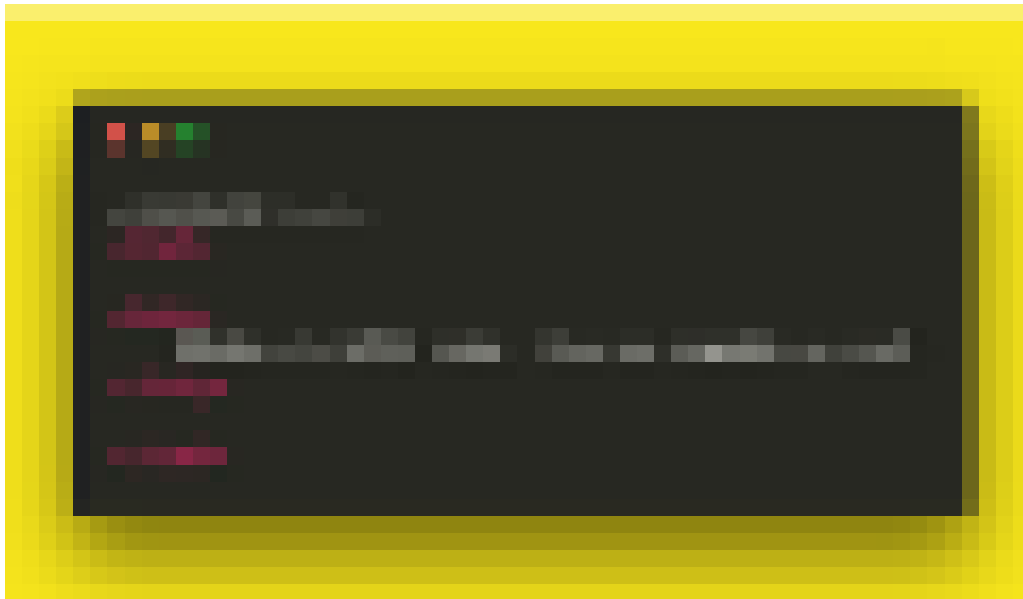
Index.js file

Did you notice we are rendering an '**index**' file at '/' route?

Well, it's time for us to create the index.hbs file inside our views folder.

HTML templating in action

We can start with something very simple. Create a file called index.hbs inside the views folder. Write this basic code inside our index.hbs and save the file.



Index.hbs file

“Well, I could’ve done better using static HTML code, all by myself”

You might say, But trust me, hang on it’s gonna be a lot fun, real soon.

Let’s take this code for a ride

1. Open terminal and go your project directory.
2. Type `node index.js`
3. Now open browser and type `http://localhost:5000` in the address bar.
4. You should see something like,



Node app running.

Unleashing the power of HTML templates.

We don’t use HTML templating for what we did above but we can do something much more powerful after setting things up. Now let’s assume, you want to render a different list every

time a user visits your site. You can't do that with static HTML right? But, with Templating you can. Make the following changes to index.js file.



Index.js file

and some more changes to our **index.hbs** file.



index.hbs file

Save everything, and run the app using `node index.js`

Now go to `localhost:5000` and hit refresh several times, you should see a new list of people's names every time you hit refresh. Something like this.



Dynamic list generated from server using HTML templating.

Bear in mind this is pure html, no client-side javascript involved.

That's all folks

This was a very basic introduction to HTML templating with Node.js, in the next article we'll dive deep into the world of **Handlebars**. The final code is available [here](#).

Happy Coding.
