Open in app

Bi Yoo

254 Followers About Follow

You can now subscribe to get stories delivered directly to your inbox.

Got it

How to set up a server with node.js & express and deploy to heroku

Bi Yoo Sep 8, 2017 · 3 min read

Hello world! 😊

This article will show you how to set up node.js server using express. It's also available in Korean, my native language.

It assumes that you are somewhat familiar with javascript and you have latest version of node and npm installed.

Install express and create a simple "hello world" server

1. Create a directory. I'd recommend something like "server"
2. In the directory run `npm init`
3. Install express `npm install — save express`
4. Create `index.js` file in the same directory, this will be the entry point of your server.
5. In that file, import express module `const express = require("express")`
   **Node does not support ES6 import yet thus we are using CommonJS*
6. Declare your app and hook it up with express `const app = express()`
7. Let's make it do something. Basic syntax for that goes like this,

```
app.get("/", (req, res) => {

   res.send({ hello: "world" });

});
```

8. Make your app listen to requests made to port 3000 by `app.listen(3000)`

9. Turn your newly created server by running `node index.js` in your terminal.

10. Go to `localhost:3000` to check it out! (pretty easy huh?) At this point, your `index.js` file should look like this,

```
const express = require("express");

const app = express();

app.get("/", (req, res) => {

    res.send({ hello: "world" });

});

app.listen(3000);
```

Deploy your server to Heroku pt.1

Now we have a server, let's put it out on the internet. We are going to use Heroku as our remote server, so sign up for an account.

1. Heroku assigns port for you. So instead pointing the server to localhost 3000, we will assign heroku port for our app to listen to. Let's do, `const PORT = process.env.PORT` You can name that const whatever you'd like.

2. Let's pass that port information into our app by switching to `app.listen(PORT)`. Now our app will listen to `process.env.PORT`

3. Before we push the app to Heroku, we have few more things to set up. We need to tell Heroku which version of node and npm we are using for our app. Go to `package.json` and insert,

```
"engines":{
    "node": //your node version ,
    "npm": //your npm version
```

**you can check the versions of those by running* `-v` *. eg)* `npm -v`

4. We also need to tell Heroku what command to run to start the server. Go back to `package.json` , in the `“scripts”` key, assign `“start”: “node index.js”`

5. Last thing! We need to make sure that we don’t upload all the modules we have. Create `.gitignore` file, and inside type `node_modules` .

Deploy your server to Heroku pt.2

We are now ready to deploy our server to Heroku. We will make use of heroku CLI, so if you haven’t installed it, install it globally. We also need git, so be sure to install that as well!

1. After making sure you have everything installed, initialize a git repository in your server directory. Run `git init`
2. Login to Heroku in your terminal. Run `heroku login`
3. Initiate a heroku server by running `heroku create`. This will generate remote git repo as well.
4. Make sure your repo and heroku repo are connected. Run `git remote add heroku YOURHEROKUGITADDRESS` *(make sure you put your heroku git address)*
5. Add & Commit your repo. Not sure how to? Bookmark this page and look for “git add” and “git commit” It should be straight-forward.
6. Finally push it! Run, `git push heroku master`
7. Hooray! You now have a express server on the web check it out by running `heroku open`

Hope this helps. Please shoot me any questions if you are stuck — thanks!

Repo link is here!

JavaScript Heroku Expressjs