

Extra Notes - Node.js & Express Starter App

- Dr Nick Hayward

Starter App

A collection of notes on creating a starter app for Node.js and Express

Contents

- Intro
- getting started
- initial structure
 - server.js
- define Git Ignore
- initial Heroku config
- initial GitHub push
- setup Heroku to host project
 - use Heroku CLI with multiple accounts
 - rename heroku app from cli

intro

This starter template creates a basic Express and Node.js based app, which may be pushed to GitHub and hosted on Heroku.

getting started

Start by creating a new directory for the local project on the host machine, e.g.

```
mkdir node-express-starter
```

Then, we can simply change to this directory and start development

```
cd node-express-starter
```

As we're building a Node.js based app, we can start by initialising our local project. e.g.

```
npm init
```

As this command is run, we can then answer some simple questions to add metadata for the current project. A `package.json` file will be created in the root directory of the current project.

We also need to add version control to this project, which is necessary for both GitHub and Heroku usage. e.g.

```
git init
```

We can then install Express using NPM,

```
npm i express --save
```

initial structure

For a Node.js and Express based app we can start by adding a `server.js` file to a `server` directory at the root of the project.

```
|-- node-express-starter
  |-- .git
  |-- node_modules
  |-- package.json
  |-- server
  |-- server.js
```

We might also add an initial directory to serve static files, e.g. `public`

```
|-- node-express-starter
  |-- .git
  |-- node_modules
  |-- package.json
  |-- public
  |   |-- index.html
  |-- server
  |-- server.js
```

server.js

Add some initial setup and config to the `server.js` file to get a basic server up and running.

```
// require node module 'path' - built-in module
const path = require('path');
// require express module
const express = require('express');

// define path to static dir public
const publicDir = path.join(__dirname, '../public');
// define variable to call express methods
var app = express();

// configure express static middleware
app.use(express.static(publicDir));

// start server on port 3030 - add callback function
app.listen(3030, () => {
  console.log('server running on port 3030');
});
```

This basic server will allow us to server static files from the `public` directory, e.g. a starter `index.html` file

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>starter template</title>
  </head>
  <body>
    <h3>Express starter template</h3>
  </body>
</html>
```

define Git Ignore

To push a Git based project to GitHub, we usually start by defining a `.gitignore` file. e.g.

```
node_modules/
```

We use this file to define project files and directories Git will omit for tracking &c., such as the Node.js directory for installed modules.

initial Heroku config

We need to update `server.js` to define a port for the process environment, e.g.

```
const port = process.env.PORT || 3030;
```

This allows us to use the app on a configured Heroku hosting, and for local testing with the port 3030.

We'll update our server to use this path,

```
app.listen(port, () => {
  console.log(`server running on port ${port}`);
});
```

Then, we need to update the `package.json` file to tell Heroku how to start the app, and the required minimum version of Node.js.

```
...
"scripts": {
  "start": "node server/server.js",
  "test": "echo \"Error: no test specified\" && exit 1"
},
"engines": {
  "node": "9.2.1"
},
...
```

initial GitHub push

This assumes an existing GitHub account, and requirement to create a new repo for this project.

On GitHub, we may simply create a new repository. This may then be used with the local Git project repository.

To push our local project to this new GitHub repo, we need to commit any changes in the local repository before adding the remote repo.

```
git status
git add .
git commit -m "initial commit"
```

Then, we can run the commands GitHub provides for a newly created repo, e.g.

```
git remote add origin git@github.com:your_username/repo_name.git
git push -u origin master
```

This will set the remote repo for synchronisation with the current local Git repo, and then push the recent commit to this remote repo.

setup Heroku to host project

In the root directory of the local project, we can use the Heroku Toolbelt CLI tool to create a new Heroku project, e.g.

```
heroku create
```

and then push the local Git repo to Heroku, e.g.

```
git push heroku master
```

Heroku will create the new app and push it to a hosted instance online. It will also return the URL for this new app, which we can then view with the following command,

```
heroku open
```

or by simply entering the project's hosted URL in a browser.

use Heroku CLI with multiple accounts

By default, Heroku's Toolbelt CLI tool does not support multiple accounts for a local machine. However, there is a supported plugin, `heroku-accounts`, which adds this functionality.

We can install it using the following command,

```
heroku plugins:install heroku-accounts
```

This will add the accounts plugin to the currently install Heroku Toolbelt CLI tool.

Then, we can add some accounts to allow us to switch account as needed for a local project. e.g.

```
heroku accounts:add personal
```

The Heroku tool will then request account credentials for the Heroku account to link, e.g. username and password.

To switch to this account, we may use the following command

```
heroku accounts:set personal
```

To check accounts for the current machine, we may use the following command

```
heroku accounts
```

To check for the currently active account,

```
heroku accounts:current
```

rename heroku app from cli

We can rename and update a heroku app using the CLI toolbelt, again in the root directory of the enclosing git repository

```
heroku apps:rename new_name --app current_name
```

This will update the name locally, and on the Heroku servers. It will also update the git repository name for heroku push commands.