Server-side application

This document shows steps to install back-end software.

Preliminaries

Download and install:

1. Node.js (https://nodejs.org/en/), a server-side JavaScript network application framework (Install LTS, not latest)
2. Git (http://git-scm.com/download/), a version control system (If you haven’t done so)

   Windows Stick with the default options and click `Next` until finished.

   Mac OS X Got an error? Try:

   - Apple menu → System Preferences… → Security & Privacy → Open Anyway.
   - Otherwise, try: Spotlight Search (press `F4` or Command-Space) → Terminal. Then, type `git` and follow instructions to install git.

   Linux Use `sudo` with your package manager.

Initial project setup

1. Open the command-line.

   Windows Windows Key → Type "Git Bash"

   Mac & Linux Open "Terminal"

2. From the command line, install `express-generator` globally. Express (http://expressjs.com/) is the web framework for Node.js. Do NOT run this command with `sudo`.

```
npm install express-generator -g
```

NOTE

> Did you get an `EACCES` error? If so, do this:
>
> ```
> sudo chown -R $(whoami) $(npm config get prefix)/{lib/node_modules,bin,share}
> ```
>
> Want to clean what you did and start over? If so, do this:
>
> ```
> npm cache clean --force
> ```

3. Use `express-generator` to generate boilerplate code (minimal code to get things working). You can rename `Node0-Basics` to whatever you want. Note that we will create more projects later. So name it properly (i.e., COMP335/Node0-Basics).

   ```
   express --no-view Node0-Basics
   ```

4. Go into the project folder you created. After this step, assume you are in this folder.

   ```
   cd Node0-Basics
   ```

5. Examine the files placed in this folder:

   ```
   .
   ├── app.js
   ├── bin
   │   └── www
   ├── package.json
   ├── public
   │   ├── index.html
   │   ├── images
   │   ├── javascripts
   │   └── stylesheets
   │       └── style.css
   ├── routes
   │   ├── index.js
   │   └── users.js
   ```

`app.js`	The entry point into this project.
`bin/www`	A wrapper script for `app.js`
`package.json`	Configures project name, scripts, and dependencies.
`public`	Where to place client-side code.
`routes`	Where to place server-side code.

6. Examine the `package.json` file. Notice the following dependencies:

`cookie-parser`	Parses HTTP Cookies (https://www.npmjs.com/package/cookie-parser)
`debug`	Debugging utility (https://www.npmjs.com/package/debug)

`express`	Node web framework (https://www.npmjs.com/package/express) (documentation (http://expressjs.com/en/guide/routing.html))
`morgan`	HTTP request logger (https://www.npmjs.com/package/morgan)

Run the project locally

1. From the command line, install dependencies listed in `package.json`. This command may take a while to run. Once it installs all dependencies successfully, you will see a new folder `node_modules`.

```
npm install
```

> **NOTE**
>
> Did you get an `high severity vulnerabilities` error? If so, do this:
>
> ```
> npm audit fix --force
> ```

2. Run the server to see the example project.

```
npm start
```

3. Open http://localhost:3000/ in your browser. You should see **Welcome to Express**.
4. In the command line, `Ctrl-C` stops the server.

Last updated 2024-02-14 08:05:17 -0500