Node.js

Express

API

Postgres - hosted on a linux server (ubuntu).

Where should we be storing the API on the server?

Where should it be so it can talk to the web app?

Application server or database server?

How to format a json file?

We are having trouble connecting our API to the database. Ask about permissions linux server permissions specifically. -Said we had to move files into the right place

How do we connect to the application?

Ask about rec.body.\_\_\_\_?

Read Write and Execute.

----------------------------------------------------------------------------------------------------------------------------

bruce@Victory:~/Documents$ npm start HerbyAPI.js

npm ERR! Error: ENOENT, open '/home/bruce/Documents/node\_modules/HerbyAPI.js/package.json'

npm ERR! If you need help, you may report this log at:

npm ERR! <http://github.com/isaacs/npm/issues>

npm ERR! or email it to:

npm ERR! <npm-@googlegroups.com>

npm ERR! System Linux 3.13.0-24-generic

npm ERR! command "/usr/bin/nodejs" "/usr/bin/npm" "start" "HerbyAPI.js"

npm ERR! cwd /home/bruce/Documents

npm ERR! node -v v0.10.25

npm ERR! npm -v 1.3.10

npm ERR! path /home/bruce/Documents/node\_modules/HerbyAPI.js/package.json

npm ERR! code ENOENT

npm ERR! errno 34

npm ERR!

npm ERR! Additional logging details can be found in:

npm ERR! /home/bruce/Documents/npm-debug.log

npm ERR! not ok code 0

----------------------------------------------------------------------------------------------------------------------------

BELOW IS EVERYTHING HE DID

^[[A^[[ALast login: Wed Nov 4 20:50:44 on ttys000

^[[B148-100-142-225:~ CharlieRopes$ ssh bruce@10.10.7.162

bruce@10.10.7.162's password:

Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-24-generic x86\_64)

\* Documentation: https://help.ubuntu.com/

12 packages can be updated.

10 updates are security updates.

\*\*\* System restart required \*\*\*

Last login: Wed Nov 4 21:32:02 2015 from 148.100.142.225

bruce@Victory:~$ cd Documents

bruce@Victory:~/Documents$ ls

Capping.sql HerbyAPI.js npm-debug.log package.json

bruce@Victory:~/Documents$ npm start HerbyAPI.js

npm ERR! Error: ENOENT, open '/home/bruce/Documents/node\_modules/HerbyAPI.js/package.json'

npm ERR! If you need help, you may report this log at:

npm ERR! <http://github.com/isaacs/npm/issues>

npm ERR! or email it to:

npm ERR! <npm-@googlegroups.com>

npm ERR! System Linux 3.13.0-24-generic

npm ERR! command "/usr/bin/nodejs" "/usr/bin/npm" "start" "HerbyAPI.js"

npm ERR! cwd /home/bruce/Documents

npm ERR! node -v v0.10.25

npm ERR! npm -v 1.3.10

npm ERR! path /home/bruce/Documents/node\_modules/HerbyAPI.js/package.json

npm ERR! code ENOENT

npm ERR! errno 34

npm ERR!

npm ERR! Additional logging details can be found in:

npm ERR! /home/bruce/Documents/npm-debug.log

npm ERR! not ok code 0

bruce@Victory:~/Documents$ more /home/bruce/Documents/npm-debug.log

0 info it worked if it ends with ok

1 verbose cli [ '/usr/bin/nodejs', '/usr/bin/npm', 'start', 'HerbyAPI.js' ]

2 info using npm@1.3.10

3 info using node@v0.10.25

4 error Error: ENOENT, open '/home/bruce/Documents/node\_modules/HerbyAPI.js/pack

age.json'

5 error If you need help, you may report this log at:

5 error <http://github.com/isaacs/npm/issues>

5 error or email it to:

5 error <npm-@googlegroups.com>

6 error System Linux 3.13.0-24-generic

7 error command "/usr/bin/nodejs" "/usr/bin/npm" "start" "HerbyAPI.js"

8 error cwd /home/bruce/Documents

9 error node -v v0.10.25

10 error npm -v 1.3.10

11 error path /home/bruce/Documents/node\_modules/HerbyAPI.js/package.json

12 error code ENOENT

13 error errno 34

14 verbose exit [ 34, true ]

bruce@Victory:~/Documents$ pwd

/home/bruce/Documents

bruce@Victory:~/Documents$ pwd

/home/bruce/Documents

bruce@Victory:~/Documents$ ls

Capping.sql HerbyAPI.js npm-debug.log package.json

bruce@Victory:~/Documents$ man ln

bruce@Victory:~/Documents$ ln node\_modules /home/bruce/Documents/

ln: failed to access ‘node\_modules’: No such file or directory

bruce@Victory:~/Documents$ ln -s /home/bruce/Documents node\_modules

bruce@Victory:~/Documents$ ls

Capping.sql HerbyAPI.js **node\_modules** npm-debug.log package.json

bruce@Victory:~/Documents$ npm start HerbyAPI.js

npm ERR! Error: ENOTDIR, open '/home/bruce/Documents/node\_modules/HerbyAPI.js/package.json'

npm ERR! If you need help, you may report this log at:

npm ERR! <http://github.com/isaacs/npm/issues>

npm ERR! or email it to:

npm ERR! <npm-@googlegroups.com>

npm ERR! System Linux 3.13.0-24-generic

npm ERR! command "/usr/bin/nodejs" "/usr/bin/npm" "start" "HerbyAPI.js"

npm ERR! cwd /home/bruce/Documents

npm ERR! node -v v0.10.25

npm ERR! npm -v 1.3.10

npm ERR! path /home/bruce/Documents/node\_modules/HerbyAPI.js/package.json

npm ERR! code ENOTDIR

npm ERR! errno 27

npm ERR!

npm ERR! Additional logging details can be found in:

npm ERR! /home/bruce/Documents/npm-debug.log

npm ERR! not ok code 0

bruce@Victory:~/Documents$ rm node\_modules

bruce@Victory:~/Documents$ ls

Capping.sql HerbyAPI.js npm-debug.log package.json

bruce@Victory:~/Documents$ mkdir node\_modules

bruce@Victory:~/Documents$ mv package.js^C

bruce@Victory:~/Documents$ cp package.json node\_modules/

bruce@Victory:~/Documents$ npm start HerbyAPI.js

npm ERR! Error: ENOENT, open '/home/bruce/Documents/node\_modules/HerbyAPI.js/package.json'

npm ERR! If you need help, you may report this log at:

npm ERR! <http://github.com/isaacs/npm/issues>

npm ERR! or email it to:

npm ERR! <npm-@googlegroups.com>

npm ERR! System Linux 3.13.0-24-generic

npm ERR! command "/usr/bin/nodejs" "/usr/bin/npm" "start" "HerbyAPI.js"

npm ERR! cwd /home/bruce/Documents

npm ERR! node -v v0.10.25

npm ERR! npm -v 1.3.10

npm ERR! path /home/bruce/Documents/node\_modules/HerbyAPI.js/package.json

npm ERR! code ENOENT

npm ERR! errno 34

npm ERR!

npm ERR! Additional logging details can be found in:

npm ERR! /home/bruce/Documents/npm-debug.log

npm ERR! not ok code 0

bruce@Victory:~/Documents$ ls

Capping.sql HerbyAPI.js **node\_modules** npm-debug.log package.json

bruce@Victory:~/Documents$ ls^C

bruce@Victory:~/Documents$ npm start HerbyAPI.jsc^C

bruce@Victory:~/Documents$ which npm

/usr/bin/npm

bruce@Victory:~/Documents$ more /usr/bin/npm

#!/usr/bin/nodejs

;(function () { // wrapper in case we're in module\_context mode

// windows: running "npm blah" in this folder will invoke WSH, not node.

if (typeof WScript !== "undefined") {

WScript.echo("npm does not work when run\n"

+"with the Windows Scripting Host\n\n"

+"'cd' to a different directory,\n"

+"or type 'npm.cmd <args>',\n"

+"or type 'node npm <args>'.")

WScript.quit(1)

return

}

process.title = "npm"

var log = require("npmlog")

log.pause() // will be unpaused when config is loaded.

log.info("it worked if it ends with", "ok")

var fs = require("graceful-fs")

, path = require("path")

, npm = require("../lib/npm.js")

, npmconf = require("npmconf")

, errorHandler = require("../lib/utils/error-handler.js")

, configDefs = npmconf.defs

, shorthands = configDefs.shorthands

, types = configDefs.types

, nopt = require("nopt")

// if npm is called as "npmg" or "npm\_g", then

// run in global mode.

if (path.basename(process.argv[1]).slice(-1) === "g") {

process.argv.splice(1, 1, "npm", "-g")

}

log.verbose("cli", process.argv)

var conf = nopt(types, shorthands)

npm.argv = conf.argv.remain

if (npm.deref(npm.argv[0])) npm.command = npm.argv.shift()

else conf.usage = true

if (conf.version) {

console.log(npm.version)

return

}

if (conf.versions) {

npm.command = "version"

conf.usage = false

npm.argv = []

}

log.info("using", "npm@%s", npm.version)

log.info("using", "node@%s", process.version)

// make sure that this version of node works with this version of npm.

var semver = require("semver")

, nodeVer = process.version

, reqVer = npm.nodeVersionRequired

if (reqVer && !semver.satisfies(nodeVer, reqVer)) {

return errorHandler(new Error(

"npm doesn't work with node " + nodeVer

+ "\nRequired: node@" + reqVer), true)

}

process.on("uncaughtException", errorHandler)

if (conf.usage && npm.command !== "help") {

npm.argv.unshift(npm.command)

npm.command = "help"

}

// now actually fire up npm and run the command.

// this is how to use npm programmatically:

conf.\_exit = true

npm.load(conf, function (er) {

if (er) return errorHandler(er)

npm.commands[npm.command](npm.argv, errorHandler)

})

})()

bruce@Victory:~/Documents$ more /usr/bin/npm

#!/usr/bin/nodejs

;(function () { // wrapper in case we're in module\_context mode

// windows: running "npm blah" in this folder will invoke WSH, not node.

if (typeof WScript !== "undefined") {

WScript.echo("npm does not work when run\n"

+"with the Windows Scripting Host\n\n"

+"'cd' to a different directory,\n"

+"or type 'npm.cmd <args>',\n"

+"or type 'node npm <args>'.")

WScript.quit(1)

return

}

process.title = "npm"

var log = require("npmlog")

log.pause() // will be unpaused when config is loaded.

log.info("it worked if it ends with", "ok")

var fs = require("graceful-fs")

, path = require("path")

, npm = require("../lib/npm.js")

, npmconf = require("npmconf")

, errorHandler = require("../lib/utils/error-handler.js")

, configDefs = npmconf.defs

, shorthands = configDefs.shorthands

, types = configDefs.types

, nopt = require("nopt")

// if npm is called as "npmg" or "npm\_g", then

// run in global mode.

if (path.basename(process.argv[1]).slice(-1) === "g") {

process.argv.splice(1, 1, "npm", "-g")

}

log.verbose("cli", process.argv)

var conf = nopt(types, shorthands)

npm.argv = conf.argv.remain

if (npm.deref(npm.argv[0])) npm.command = npm.argv.shift()

else conf.usage = true

if (conf.version) {

console.log(npm.version)

return

}

if (conf.versions) {

npm.command = "version"

conf.usage = false

npm.argv = []

}

log.info("using", "npm@%s", npm.version)

log.info("using", "node@%s", process.version)

// make sure that this version of node works with this version of npm.

var semver = require("semver")

, nodeVer = process.version

, reqVer = npm.nodeVersionRequired

if (reqVer && !semver.satisfies(nodeVer, reqVer)) {

return errorHandler(new Error(

"npm doesn't work with node " + nodeVer

+ "\nRequired: node@" + reqVer), true)

}

process.on("uncaughtException", errorHandler)

if (conf.usage && npm.command !== "help") {

npm.argv.unshift(npm.command)

npm.command = "help"

}

// now actually fire up npm and run the command.

// this is how to use npm programmatically:

conf.\_exit = true

npm.load(conf, function (er) {

if (er) return errorHandler(er)

npm.commands[npm.command](npm.argv, errorHandler)

})

})()

bruce@Victory:~/Documents$

bruce@Victory:~/Documents$ more /usr/bin/npm

#!/usr/bin/nodejs

;(function () { // wrapper in case we're in module\_context mode

// windows: running "npm blah" in this folder will invoke WSH, not node.

if (typeof WScript !== "undefined") {

WScript.echo("npm does not work when run\n"

+"with the Windows Scripting Host\n\n"

+"'cd' to a different directory,\n"

+"or type 'npm.cmd <args>',\n"

+"or type 'node npm <args>'.")

WScript.quit(1)

return

}

process.title = "npm"

var log = require("npmlog")

log.pause() // will be unpaused when config is loaded.

log.info("it worked if it ends with", "ok")

var fs = require("graceful-fs")

, path = require("path")

, npm = require("../lib/npm.js")

, npmconf = require("npmconf")

, errorHandler = require("../lib/utils/error-handler.js")

, configDefs = npmconf.defs

, shorthands = configDefs.shorthands

, types = configDefs.types

, nopt = require("nopt")

// if npm is called as "npmg" or "npm\_g", then

// run in global mode.

if (path.basename(process.argv[1]).slice(-1) === "g") {

process.argv.splice(1, 1, "npm", "-g")

}

log.verbose("cli", process.argv)

var conf = nopt(types, shorthands)

npm.argv = conf.argv.remain

if (npm.deref(npm.argv[0])) npm.command = npm.argv.shift()

else conf.usage = true

if (conf.version) {

console.log(npm.version)

return

}

if (conf.versions) {

npm.command = "version"

conf.usage = false

npm.argv = []

}

log.info("using", "npm@%s", npm.version)

log.info("using", "node@%s", process.version)

// make sure that this version of node works with this version of npm.

var semver = require("semver")

, nodeVer = process.version

, reqVer = npm.nodeVersionRequired

bruce@Victory:~/Documents$ locate npmconf

/usr/share/npm/node\_modules/npmconf

/usr/share/npm/node\_modules/npmconf/.npmignore

/usr/share/npm/node\_modules/npmconf/README.md

/usr/share/npm/node\_modules/npmconf/config-defs.js

/usr/share/npm/node\_modules/npmconf/node\_modules

/usr/share/npm/node\_modules/npmconf/npmconf.js

/usr/share/npm/node\_modules/npmconf/package.json

/usr/share/npm/node\_modules/npmconf/test

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/.npmignore

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/index.js

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/node\_modules

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/package.json

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/readme.markdown

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/test

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/node\_modules/proto-list

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/node\_modules/proto-list/README.md

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/node\_modules/proto-list/package.json

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/node\_modules/proto-list/proto-list.js

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/node\_modules/proto-list/test

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/node\_modules/proto-list/test/basic.js

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/test/broken.js

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/test/broken.json

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/test/chain-class.js

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/test/env.js

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/test/find-file.js

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/test/ignore-unfound-file.js

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/test/ini.js

/usr/share/npm/node\_modules/npmconf/node\_modules/config-chain/test/save.js

/usr/share/npm/node\_modules/npmconf/test/00-setup.js

/usr/share/npm/node\_modules/npmconf/test/basic.js

/usr/share/npm/node\_modules/npmconf/test/builtin.js

/usr/share/npm/node\_modules/npmconf/test/fixtures

/usr/share/npm/node\_modules/npmconf/test/save.js

/usr/share/npm/node\_modules/npmconf/test/fixtures/builtin

/usr/share/npm/node\_modules/npmconf/test/fixtures/globalconfig

/usr/share/npm/node\_modules/npmconf/test/fixtures/userconfig

bruce@Victory:~/Documents$ ls /usr/share/npm/node\_modules/

**chmodr** **chownr** **editor** **fstream-npm** **init-package-json** **npmconf** **npm-registry-client** **npm-user-validate** **opener** **read-installed** **uid-number**

bruce@Victory:~/Documents$ ls /usr/share/npm/node\_modules/fstream-npm/

**example** fstream-npm.js **node\_modules** package.json README.md

bruce@Victory:~/Documents$ cd /usr/share/npm/

bruce@Victory:/usr/share/npm$ ls

**bin** **doc** **lib** **man** **node\_modules** npmrc package.json

bruce@Victory:/usr/share/npm$ cd node\_modules/

bruce@Victory:/usr/share/npm/node\_modules$ ls

**chmodr** **chownr** **editor** **fstream-npm** **init-package-json** **npmconf** **npm-registry-client** **npm-user-validate** **opener** **read-installed** **uid-number**

bruce@Victory:/usr/share/npm/node\_modules$ ls init-package-json/

default-input.js example.js init-package-json.js **node\_modules** package.json README.md **test**

bruce@Victory:/usr/share/npm/node\_modules$ ls init-package-json/node\_modules/

**promzard**

bruce@Victory:/usr/share/npm/node\_modules$ ls init-package-json/node\_modules/promzard/

example/ .npmignore package.json promzard.js README.md test/

bruce@Victory:/usr/share/npm/node\_modules$ ls init-package-json/node\_modules/promzard/

**example** package.json promzard.js README.md **test**

bruce@Victory:/usr/share/npm/node\_modules$ ls init-package-json/node\_modules/promzard/

example/ .npmignore package.json promzard.js README.md test/

bruce@Victory:/usr/share/npm/node\_modules$ ls npmconf/

config-defs.js **node\_modules** npmconf.js package.json README.md **test**

bruce@Victory:/usr/share/npm/node\_modules$ ls npmconf/C^C

bruce@Victory:/usr/share/npm/node\_modules$ more npm

npmconf/ npm-registry-client/ npm-user-validate/

bruce@Victory:/usr/share/npm/node\_modules$ more npm

npmconf/ npm-registry-client/ npm-user-validate/

bruce@Victory:/usr/share/npm/node\_modules$ more npmconf/README.md

# npmconf

The config thing npm uses

If you are interested in interacting with the config settings that npm

uses, then use this module.

However, if you are writing a new Node.js program, and want

configuration functionality similar to what npm has, but for your

own thing, then I'd recommend using [rc](https://github.com/dominictarr/rc),

which is probably what you want.

If I were to do it all over again, that's what I'd do for npm. But,

alas, there are many systems depending on many of the particulars of

npm's configuration setup, so it's not worth the cost of changing.

## USAGE

```javascript

var npmconf = require('npmconf')

// pass in the cli options that you read from the cli

// or whatever top-level configs you want npm to use for now.

npmconf.load({some:'configs'}, function (er, conf) {

// do stuff with conf

conf.get('some', 'cli') // 'configs'

conf.get('username') // 'joebobwhatevers'

conf.set('foo', 'bar', 'user')

conf.save('user', function (er) {

// foo = bar is now saved to ~/.npmrc or wherever

})

})

```

bruce@Victory:/usr/share/npm/node\_modules$