Nodejs Cheat Sheet Cheat Sheet

by digotetso via cheatography.com/58474/cs/15450/

Module

Module	module has exports property e.g `modul e.ex ports = fn() =>{} module. ex ports = {hello, move}`
Require	returns module.exports
config	Storing configuration settings: **default, dev & prod** `confi g.g et( con fiP rop ert y)`

Node wraps you code with function expression & invokes it, this function expression has these argument: `exports, module, __dirname & __filename`.Now you have access to these argument within your code. You can use these argument e.g `module.ex ports = hell o()` to expose ur module

Express

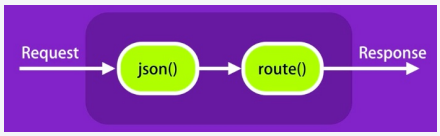
install	`npm i express app = requir e(' exp ress')`
web server	`app.ge t('/', (req, res) => { res.send('hello') }) app.li ste n(3000)`
route params	`req.params`
query params	`req.query`

Express (cont)

middlewares	`app.use(body-parser.json())` parse body of req into json:- `app.us e(e xpr ess.js on())` serves static files:- `app.use(express.static(path .join(__dirname,'public')))`

route handle is a middleware. request-processing-pipeline: Req -

Request processing pipeline



`expres s.j son()` will pass request to route handler and route will terminate request by sending response.

Built - in modules (mostly used)

path	to handle paths `path = requir e(' path') path.j oin (__ dir name, 'public')`
os	provide operating system info `os = requir e('os') os.fre emem() os.upt ime() os.use rInfo()`
http	make Node to act as webserver `server = requir e(' http') server.cr eat eSe rve r(req, res)=>{}) server.li ste n(3000)`
process	Node process info - global object `port = proces s.e nv.PORT ||` 3000

Built - in modules (mostly used) (cont)

events	to handle events `EventE mitter = requir e(' emitter = new EventE mit te emitter.on(eventName, liste emitte r.e mit (ev ent Nam`
file system	to handle file system `fs = requir e('fs') fs.rea dfi le( path, callb fs.cop yFi le( sou rce.tx lback)`

set environment vars: `export PORT = 3000`

ewwee

require	rrnrnrnrnrnrnrnrnrnrnrn

www

w	w

Debugging

`dbDebugger = requir e(' deb ug' ) ( startD ebu gger= requir e(' deb u :st art') `

Two names are will create:- `app:db & app:start`. You can replace `consol e.l og` with `dbDeb ugger` or `startD eb u gger`.

Define console output:-

`export DEBUG= app :start` or

`export DEBUG= app:db` or

`export DEBUG= app :st art ,app:db` or

`export DEBUG= app:*`

NPM

`npm init`	Create package.json file alt: `npm init --yes`
`npm i`	Install npm package alt: `npm i --save-dev` `npm i --save`
`npm list`	lookup dependencies & version alt: `npm list --depth=0`
`npm outdated`	check 4 outdated npm package
`npm update`	update npm package - only updates minor & patches
`npm un`	uninstall npm package
`npm login`	login in npm registry
`npm adduser`	create user on npm registry