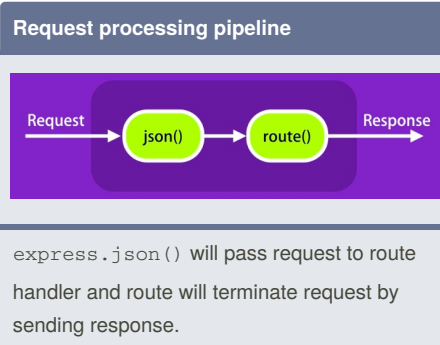## Module

Module	module has exports property e.g `module.exports = fn() =>{}` `module.exports = {hello, move}`
Require	returns module.exports
config	Storing configuration settings: **default, dev & prod** `` `config.get(confiProperty) ``

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.exports = hello()` to expose ur module

Express

install	`npm i express` `app = require('express')`
web server	`app.get('/', (req, res) => {` `res.send('hello')` `})` `app.listen(3000)`
route params	`req.params`
query params	`req.query`
middlewares	`app.use(body-parser.json())` parse body of req into json:- `app.use(express.json())` serves static files:- `app.use(express.static(path .join(__dirname,'public')))`

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

Request processing pipeline

`express.json()` will pass request to route handler and route will terminate request by sending response.

Built - in modules (mostly used)

path	to handle paths `path = require('path')` `path.join(__dirname, 'public')`
os	provide operating system info `os = require('os')` `os.freemem()` `os.uptime()` `os.userInfo()`
http	make Node to act as webserver `server = require('http')` `server.createServer(req, res)=>{})` `server.listen(3000)`
process	Node process info - global object `port = process.env.PORT || 3000`
events	to handle events `EventEmitter = require('events)` `emitter = new EventEmitter()` `emitter.on(eventName, listener)` `emitter.emit(eventName,[a rgs..])`
file system	to handle file system `fs = require('fs')` `fs.readfile(path, callback)` `fs.copyFile(source.txt,des t.txt, callback)`

set environment vars: `export PORT = 3000`

ewwee

require	rrnrnrnrnrnrnrnrnrnrn

www

w	w

Debugging

```
dbDebugger = require('debug')
('app:db')
startDebugger=
require('debug')('app:start')
`
```

✅ Two names are will create:- `app:db` & `app:start`. You can replace `console.log` with `dbDebugger` or `startDebugger`.

✅ Define console output:-
`export DEBUG=app:start` or
`export DEBUG=app:db` or
`export DEBUG=app:start,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

By **digotetso**
cheatography.com/digotetso/

Not published yet.
Last updated 11th April, 2018.
Page 1 of 1.