



Martin Fillafer

NodeJS

- Javascript (ECMAScript) Webserver
- Ryan Dahl, 2009
- Google Chrome V8 (Virtual Machine)
- data- VS computation-centric
- Javascript Concepts
 - event - based
 - asynchronous / non - blocking

NodeJS Fundamentals (1)

- non-blocking I/O => event loop
 - no multi-thread management overhead
 - many concurrent connections
- REPL
- node package manager (NPM)
- **process**
- Timers
 - `setTimeout`, `setInterval`

NodeJS Fundamentals (2)

- Buffer
 - fixed length, outside of V8
- EventEmitter
 - `obj.on('evt', cb)`, `obj.once`, `obj.emit`
- Streams
 - events: `data`, `end`, `error`
 - `pipe`
- Modules
 - `exports.fun`, `module.exports`

NodeJS Modules

- Core
 - `fs`, `http`, `net`, `path`, `url`, ...
- lodash (Underscore.js extension)
- Express - web framework
 - `express-generator`
- Async.js
 - `map`, `filter`, `series`, `parallel`
- Socket.IO – WebSocket
- Node-ORM



DEMO
Time

Pros and Cons

- scalability
- functional → expressiveness,
parallelism = no multithreading → deadlocks
- propper error handling
 - `callback(err, ...)`
- almost everything comes per module
- asynchronous code = far more complex
- production system more complicated than
Apache2 + PHP (unhandled exceptions)

Do's and Don'ts

- NodeJS + Backbone.js / Angular.js
 - JSON handling, APIs
- Realtime Communication (WebSockets), Proxies
- Heavy computation
 - Solution: WebWorkers pattern
- Serving files in big scale
 - `app.use(express.static(__dirname + '/public'))`
 - use Nginx as Proxy
- Write Console Tools

Sources

- What is NodeJS
 - <http://stackoverflow.com/a/6782438>
- Core Modules Docu
 - <http://www.nodejs.org/api/index.html>
- Introduction to Node.js with Ryan Dahl
 - http://www.youtube.com/watch?v=jo_B4LTHi3I
- nodeschool.io - open source lesson
 - <http://nodeschool.io/#workshoppers>
- Mixu's Node book
 - <http://book.mixu.net/node/index.html>