

node.js

A quick tour

About



- Felix Geisendörfer
- 23 years
- Berlin, Germany

@felixge

nodeJS

Core Contributor

&

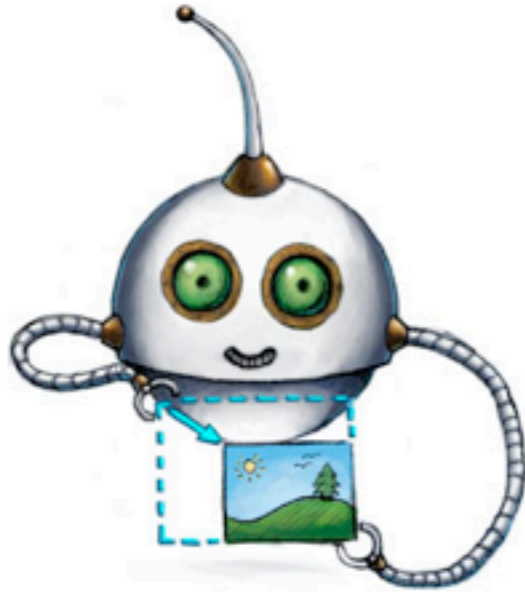
Module Author



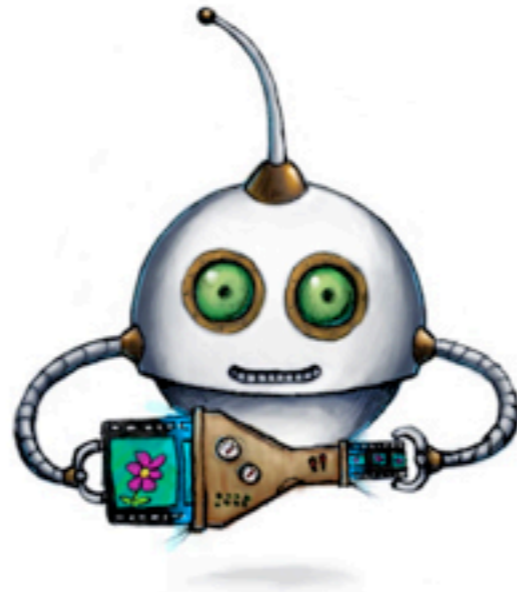
node-mysql



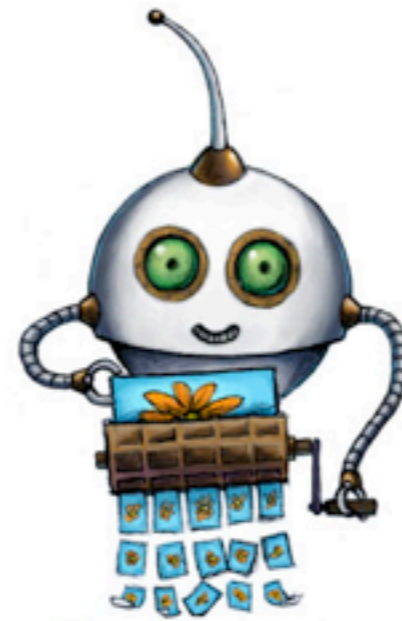
node-formidable



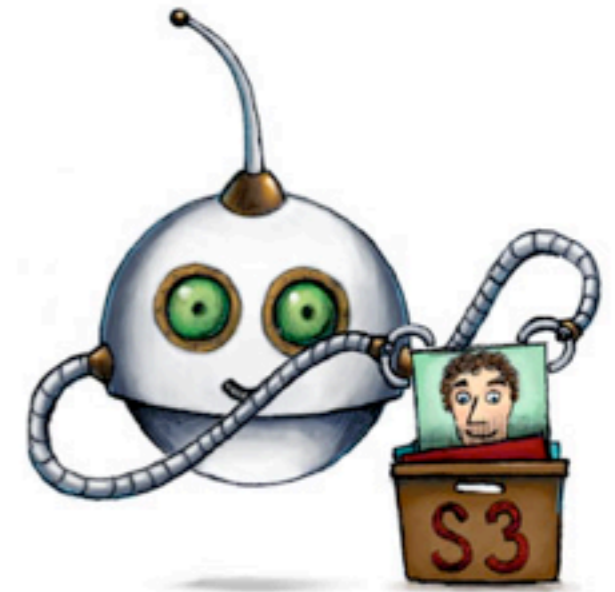
Resize images



Encode videos



Extract thumbnails



Store in S3

File uploading & processing as an infrastructure service for web & mobile applications.

Audience?



Node's goal is to provide an easy way to build scalable network programs.

Node.js

- Created by Ryan Dahl
- Google's V8 JavaScript engine (no DOM)
- Module system, I/O bindings, Common Protocols

Installing

```
$ git clone \
git://github.com/ry/node.git
$ cd node
$ ./configure
$ make install
```


Hello World

```
$ cat test.js
```

```
console.log( 'Hello World' );
```

```
$ node test.js
```

```
Hello World
```

Ingredients

libeio

libev



c-ares

V8

http_parser

Philosophy

- Just enough core-library to do I/O
- Non-blocking
- Close to the underlying system calls

Non-blocking I/O

Blocking I/O

```
var a = db.query( 'SELECT A' );  
console.log( 'result a:', a );
```

```
var b = db.query( 'SELECT B' );  
console.log( 'result b:', b );
```

Time = SUM(A, B)

Non-Blocking I/O

```
db.query('SELECT A', function(result) {  
  console.log('result a:', result);  
});
```

```
db.query('SELECT B', function(result) {  
  console.log('result b:', result);  
});
```

Time = MAX(A, B)

Non-Blocking I/O

- Offload most things to the kernel and poll for changes (select, epoll, kqueue, etc.)
- Thread pool for anything else

Single Threaded

```
var a = [];  
function f() {  
    a.push(1);  
    a.push(2);  
}  
  
setTimeout(f, 10);  
setTimeout(f, 10);
```


Single Threaded

```
var a = [];  
function f() {  
    a.push(1);  
    a.push(2);  
}  
  
setTimeout(f, 10);  
setTimeout(f, 10);
```

- [1, 1, 2, 2] ?
- [1, 2, 1, 2] ?
- BAD ACCESS ?

Single Threaded

```
var a = [];  
function f() {  
    a.push(1);  
    a.push(2);  
}  
  
setTimeout(f, 10);  
setTimeout(f, 10);
```

- [1, 1, 2, 2]

- **[1, 2, 1, 2]**

- BAD ACCESS

API Overview

CommonJS Modules

```
$ cat hello.js
```

```
exports.world = function() {  
  return 'Hello World';  
};
```

```
$ cat main.js
```

```
var hello = require('./hello');  
console.log(hello.world());
```

```
$ node main.js
```

```
Hello World
```

Child processes

```
$ cat child.js
```

```
var spawn = require('child_process').spawn;  
var cmd = 'echo hello; sleep 1; echo world;';  
  
var child = spawn('sh', ['-c', cmd]);  
child.stdout.on('data', function(chunk) {  
    console.log(chunk.toString());  
});
```

```
$ node child.js
```

```
"hello\n"
```

```
# 1 sec delay
```

```
"world\n\n"
```

Http Server

```
$ cat http.js
```

```
var http = require('http');  
http.createServer(function(req, res) {  
  setTimeout(function() {  
    res.writeHead(200);  
    res.end('Thanks for waiting!');  
  }, 1000);  
}).listen(4000);
```

```
$ curl localhost:4000
```

```
# 1 sec delay
```

```
Thanks for waiting!
```

Tcp Server

```
$ cat tcp.js
```

```
var tcp = require('tcp');  
tcp.createServer(function(socket) {  
  socket  
    .on('connect', function() {  
      socket.write("Hi, How Are You?\n> ");  
    })  
    .on('data', function(data) {  
      socket.write(data);  
    });  
}).listen(4000);
```

```
$ nc localhost 4000
```

```
Hi, How Are You?
```

```
> Great!
```

```
Great!
```

DNS

```
$ cat dns.js
```

```
var dns = require('dns');  
dns.resolve('nodejs.org', function(err, addresses) {  
  console.log(addresses);  
});
```

```
$ node dns.js
```

```
[ '8.12.44.238' ]
```


Watch File

```
$ cat watch.js
```

```
var fs = require('fs');  
fs.watchFile(__filename, function() {  
  console.log('You changed me!');  
  process.exit(0);  
});
```

```
$ node watch.js
```

```
# edit watch.js
```

```
You changed me!
```

And much more

- UDP
- Crypto
- Assert
- Buffer
- Script
- EcmaScript5

...

Suitable Applications

Suitable Applications

- Single-page apps
- Real time
- Crawlers

More Applications

- Async chaining of unix tools
- Streaming
- File uploading

Interesting projects

Package management



Web Frameworks

- Express.js (Sinatra clone)
- Fab.js (Mind-bending & awesome)

DOM

- `node-htmlparser`
- `jquery`
- `node.io`

WebSockets

Socket.IO

1171 modules in npm

Ready for production?

Ready for production?

- 0.4 was released last week
- API has settled down for most parts
- Very few bugs, but YMMV

Things that suck

Things that suck

- Stack traces are lost at the event loop boundary
- Utilizing multiple cores requires multiple processes
- V8: 1.9 GB heap limit / GC can be problematic

Community

Benevolent Dictator For Life



Ryan Dahl

Community

- Mailing list (nodejs, nodejs-dev)
- IRC (#node.js)

Hosting



Questions?



@felixge / felix@debuggable.com

Bonus Slides

Speed

Speed

```
var http = require('http')
var b = new Buffer(1024*1024);

http.createServer(function (req, res) {
  res.writeHead(200);
  res.end(b);
}).listen(8000);
```

Speed

100 concurrent clients
1 megabyte response

node 822 req/sec

nginx 708

thin 85

mongrel 4

(bigger is better)

Speed

- NGINX peaked at 4mb of memory
- Node peaked at 60mb
- Very special circumstances

Questions?



@felixge / felix@debuggable.com