

# node.js

A quick tour (V3)

# About

nodeJS

Contributor



node.js driver



Felix Geisendörfer



Co-founder

**formidable**

node.js file uploads

**Audience?**

**JavaScript?**

**What is node?**

# Server side JavaScript

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

# Server side JavaScript

```
$ cat test.js
```

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

```
$ node test.js
```

```
Hello World
```

# Server side JavaScript

```
$ node
```

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

```
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

# Benevolent Dictator For Life



Ryan Dahl

# Concurrency Model

# Non-blocking I/O

```
var fs = require('fs');

fs.readFile('test.txt', function(err, data) {
  if (err) throw err;

  console.log('Read file: ', data);
});

console.log('Reading file ...');
```

# Single Threaded

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

# 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 cmd = 'echo hello; sleep 1; echo world;',  
    spawn = require('child_process').spawn,  
    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> ");  
  });  
  socket.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

- Buffer

- Crypto

- Script

- Assert

- EcmaScript5

...

# Suitable Applications

- Web frameworks
- Real time
- Crawlers

# More Applications

- Process monitoring
- File uploading
- Streaming



**Interesting projects**

# Package management



# Web Frameworks

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

# DOM

- jsdom
- node-htmlparser
- apricot

# WebSockets

**Socket.IO**

# Protocol parsers

- node-formidable
- node-mysql



# Limitations

- Weak SSL support
- Weak Windows support
- 1GB Heap limit on x64 (V8)



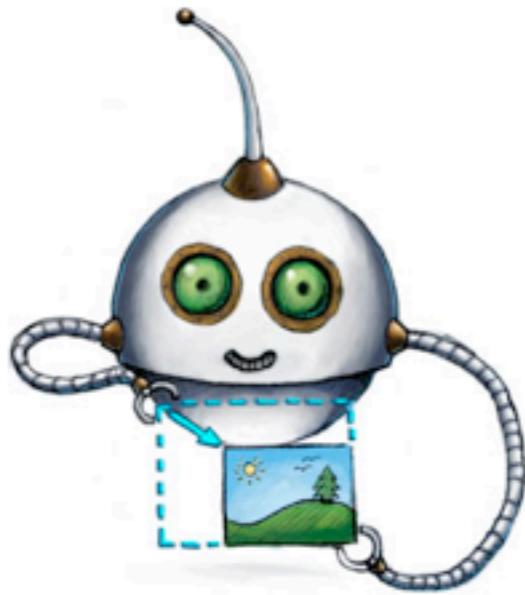
# Hosting



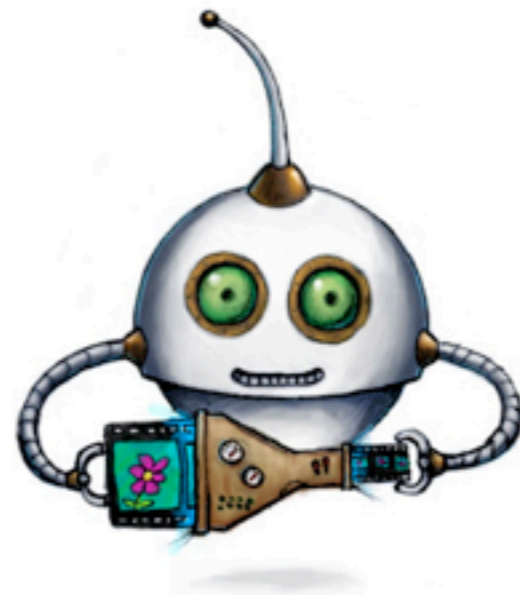
# Questions?



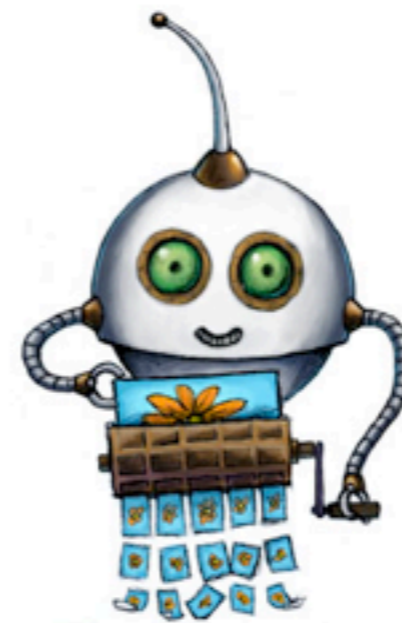
@felixge / [felix@debuggable.com](mailto:felix@debuggable.com)



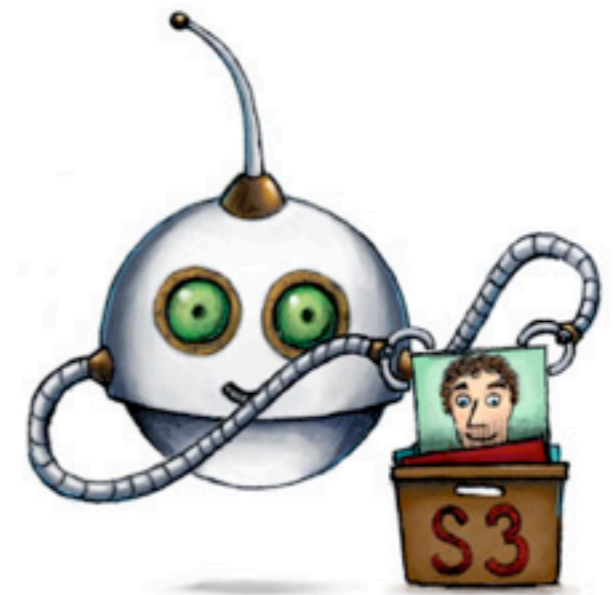
Resize images



Encode videos



Extract thumbnails



Store in S3

# Questions?



@felixge / [felix@debuggable.com](mailto:felix@debuggable.com)