



# FUTURE DECODED

6-7 OTT '16 / MILANO

IN PARTNERSHIP WITH:



CommunityDays.it

[www.futuredecoded.it](http://www.futuredecoded.it)



#FutureDecoded



# Node.js per lo sviluppatore .NET


Gianluca Carucci  
Software Engineer & Agile Coach

@rucka

<http://gianluca.carucci.org>



[www.futuredecoded.it](http://www.futuredecoded.it)

 #FutureDecoded





# communities map {me}





Node.js

WHY NAME?



# cosa succede ad un webserver quando eseguiamo questa richiesta?

```
var orders = from o in data.Orders
              where o.Amount > 10000
              select o;
ShowOrders(orders);
```







# quanto aspettiamo?

- >> L1-cache **3** cicli
- >> L2-cache **14** cicli
- >> RAM **250** cicli
- >> Disco **41.000.000** cicli
- >> Rete **240.000.000** cicli

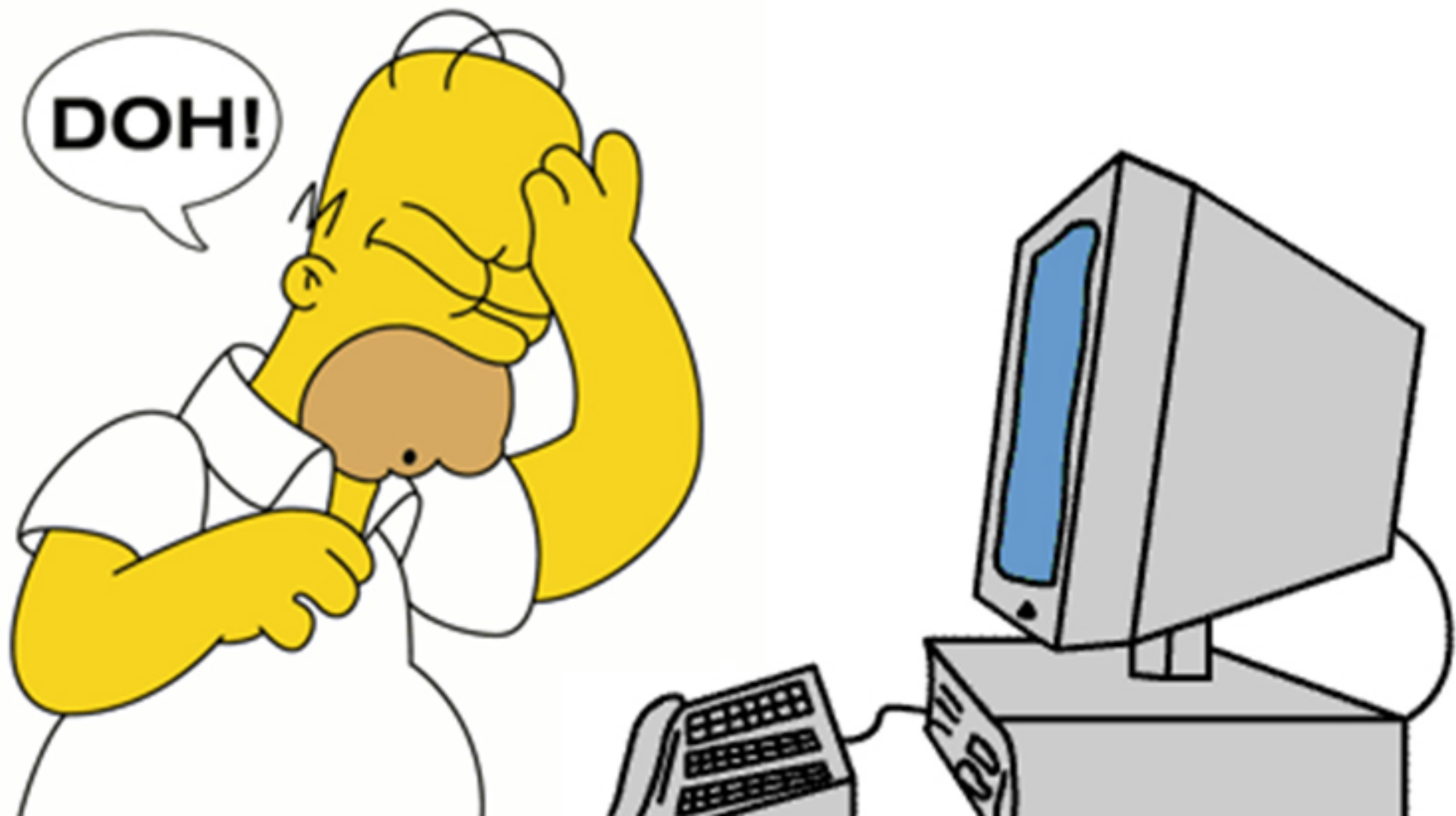


soluzione (?)

web server multithreading

Apache







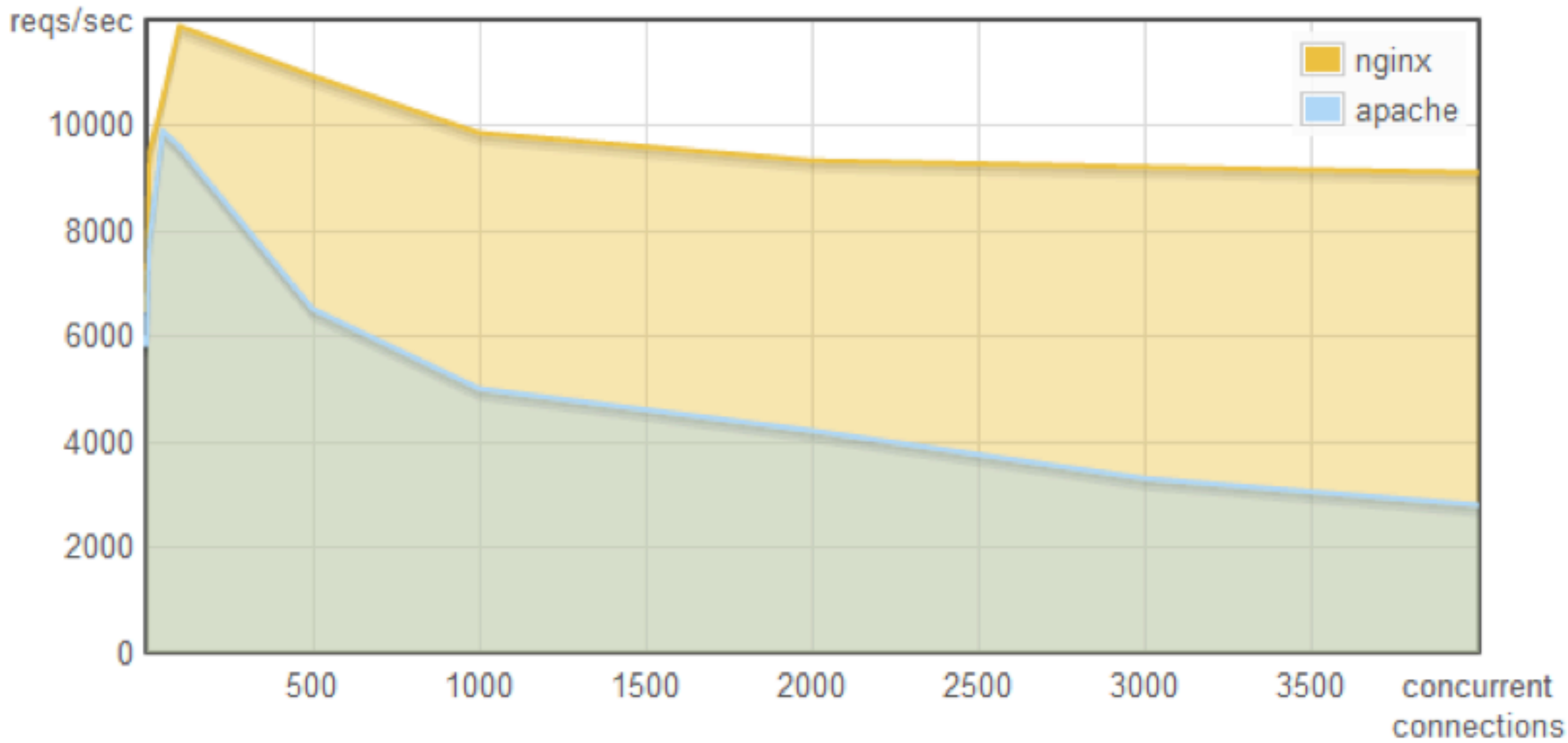
A man wearing a straw hat and a woman wearing boxing gloves are shown in a boxing stance. The man is on the left, wearing a light blue button-down shirt and a straw hat. The woman is on the right, wearing a black polka-dot shirt and black boxing gloves with 'EVERLAST' and 'EverPro' branding. She is punching the man's face. The background is a solid light blue.

# Apache vs NGINX <sup>1</sup>

---

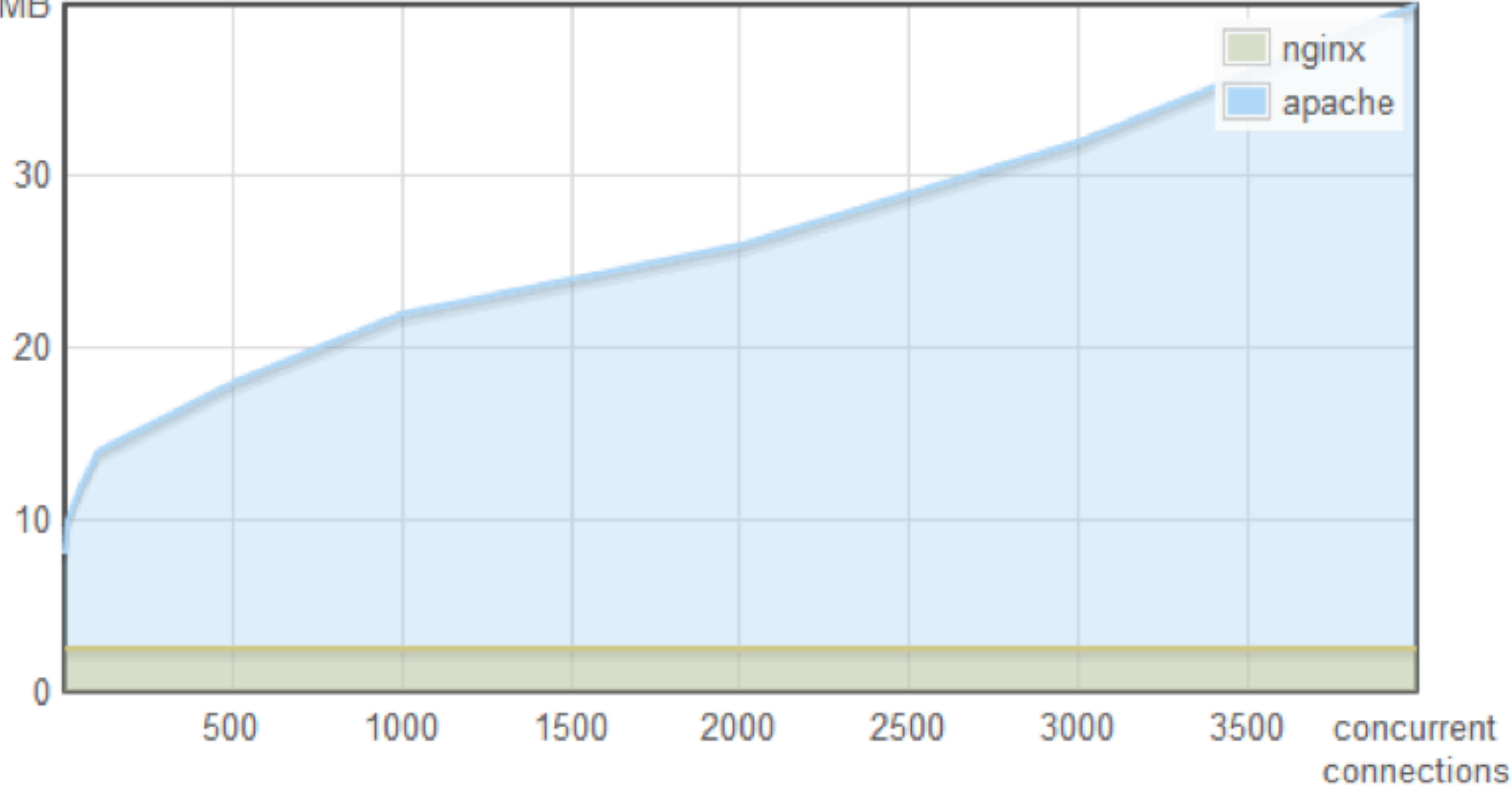
<sup>1</sup> <http://blog.webfaction.com/2008/12/a-little-holiday-present-10000-reqssec-with-nginx-2/>







memory in MB





A person wearing a blue Adidas hoodie is holding a playing card (the Ace of Spades) with both hands. The card has 'CASINO' written on it. The background is a colorful, abstract geometric pattern. The text 'dov'è la magia?' is overlaid in yellow.

dov'è la magia?



web server **single threaded**,  
bastato su **event loop**



A top-down view of a wooden cutting board with the word "Ricette" written in white cursive script. The board is surrounded by fresh ingredients: tomatoes, cucumbers, green onions, a yellow pepper, a red bell pepper, a green pepper, a red pepper, bay leaves, garlic, and fresh parsley.

Ricette



prendiamo un modello  
monothread,  
asincrono



# scegliamo un linguaggio

```
$.post("my/rest/api", function (data, status) {  
    alert("Data: " + data + "\nStatus: " + status);  
});
```





A bartender is shown in a close-up, focused on pouring a drink. He is holding a metal shaker in his left hand and a fine-mesh strainer in his right hand, filtering the liquid into a glass. The background is blurred, showing a bar setting with various bottles and equipment. The lighting is warm and slightly dim, creating a professional and intimate atmosphere.

shakeriamo bene...





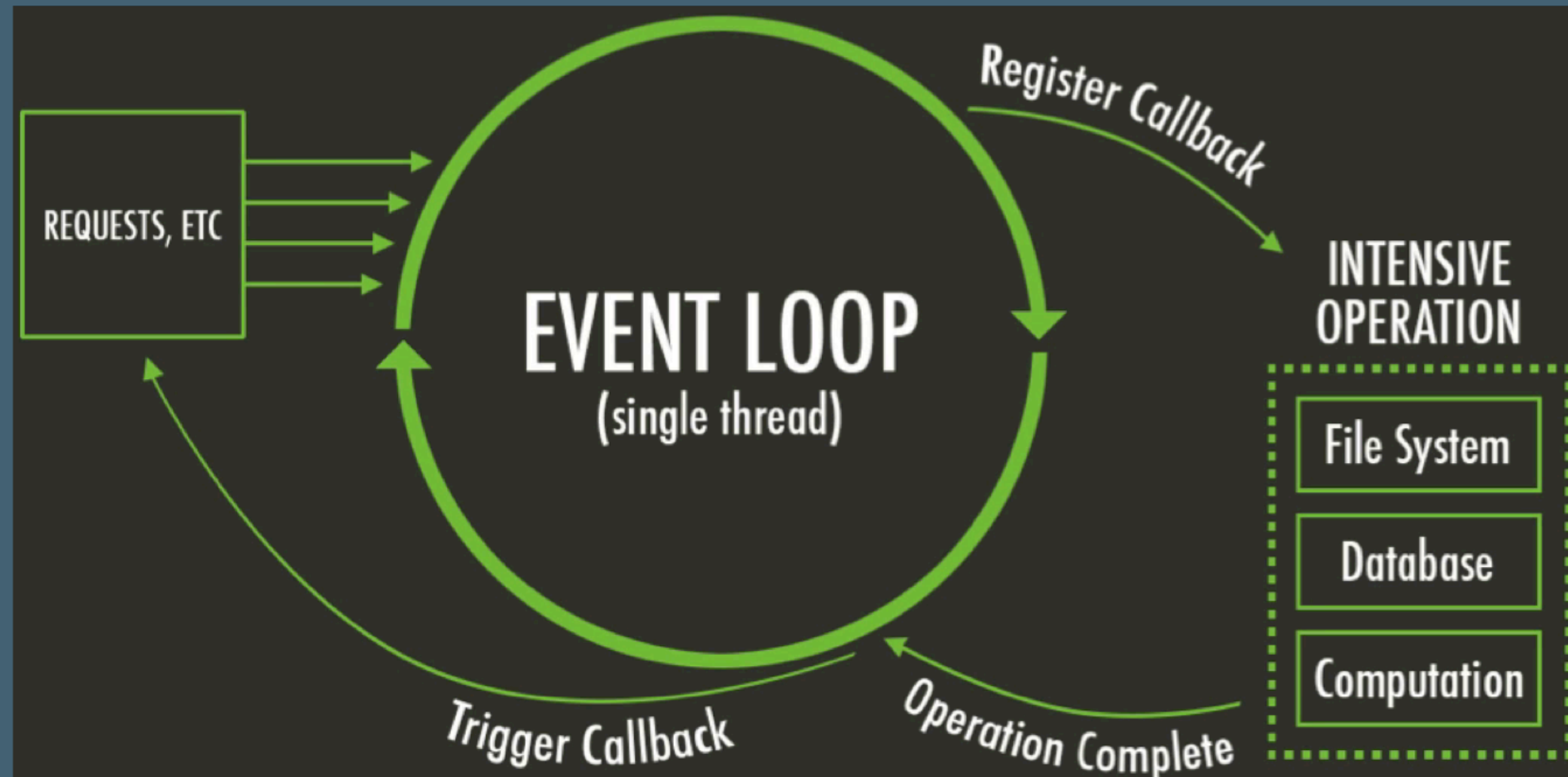
# Node.js viene presentato<sup>2</sup> alla European JsConf 1'8 novembre 2009 da Ryan Dahl

---

<sup>2</sup> <https://www.youtube.com/watch?v=ztspvPYybIY>



# architettura







**Node.js**   
@nodejs



 **Segui**

Node.js has an estimated 5 million users and is growing at 100% per year. [#NodeInteractive](#)

 Visualizza traduzione

RETWEET  
**72**

MI PIACE  
**82**



09:06 - 15 set 2016





find packages



npm is the package manager for **javascript.**



**306 113**  
total packages



**74 494 287**  
downloads in the last day



**1 192 831 758**  
downloads in the last week



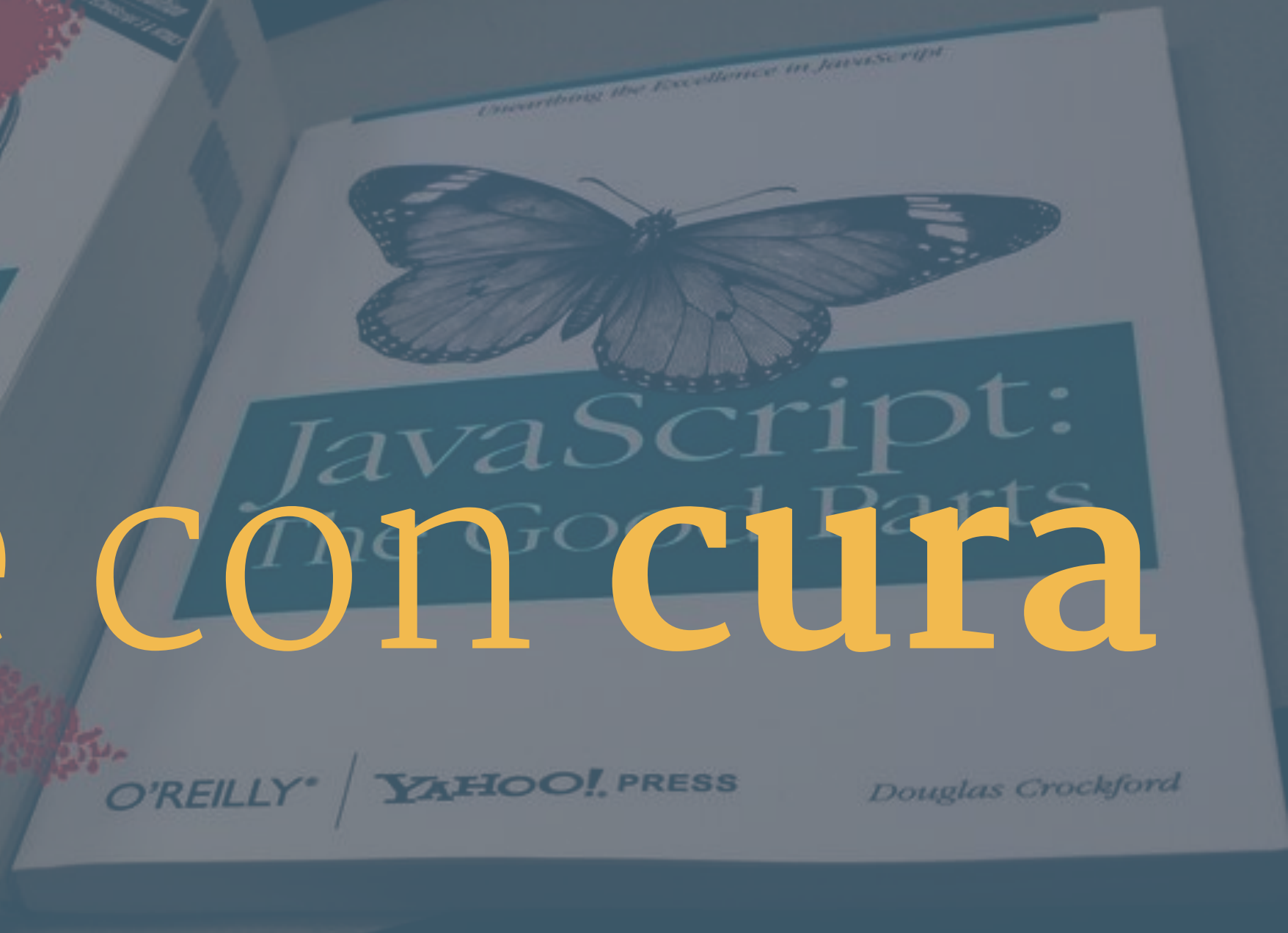
**4 751 059 780**  
downloads in the last month

*prima di iniziare...*

javascript non è un  
optional



scegliete con cura





# web application in 5 passi



# 1

## API<sup>3</sup>

- Assertion Testing
- Buffer
- C/C++ Addons
- Child Processes
- Cluster
- Command Line Options
- Console
- Crypto
- Debugger
- DNS
- Domain

- Path
- Process
- Punycode
- Query Strings
- Readline
- REPL
- Stream
- String Decoder
- Timers
- TLS/SSL
- TTY

---

<sup>3</sup> <https://nodejs.org/en/docs/>



find packages



npm is the package manager for javascript.



306 113  
total packages



74 494 287  
downloads in the last day



1 192 831 758  
downloads in the last week



4 751 059 780  
downloads in the last month

packages people 'npm install' a lot



browserify

browser-side require() the node way  
13.0.1 published 2 months ago by jnm



grunt-cli

The grunt command line interface  
1.2.0 published 3 months ago by vladikoff



bower

The browser package manager  
1.7.9 published 3 months ago by sheerun

express

express

Fast, unopinionated, minimalist web framework...  
4.14.0 published 2 weeks ago by dougwilson



npm

a package manager for JavaScript  
3.9.1 published 1 month ago by zkat



cordova

Cordova command line interface tool  
6.2.0 published a month ago by stevegill



pm2

Production process manager for Node.js ap...  
1.1.3 published 2 months ago by tknew



karma

Spectacular Test Runner for JavaScript.  
0.13.22 published 4 months ago by dignifiedquire



coffee-script

Unfancy JavaScript  
1.10.0 published 10 months ago by jashkenas



# amici in 5 minuti!

```
$ mkdir todo
```

```
$ cd todo
```

```
$ npm init
```

```
$ echo "console.log('hello cdays16\!');" > index.js
```

```
$ node index.js
```

```
>> hello cdays16!
```

```
const http = require('http');

const hostname = '127.0.0.1';
const port = process.env.PORT || 8000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  if (req.url === '/ping') return res.end('pong!\n');
  return res.end('Hello World!\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```





3

web framework  
(express)

# npm install express --save

```
var express = require('express');
var app = express();

app.get('/ping', (req, res) {
  res.send('pong!');
});

app.patch('/api/:id', (req, res) => {
  db.update(req.params.id, req.body, todo => {
    res.send(createTodo(req, todo));
  });
});

app.listen(process.env.PORT || 8000);
```



# template engine

`npm install jade --save`

```
//app.js
var express = require('express');
var app = express();
app.set('view engine', 'jade');

app.get('/', (req, res) {
  res.render('index', {
    title: 'cdays 2016',
    message: 'hello!'
  });
});
```

```
//index.jade
html
  head
    title!= title
  body
    h1!= message
```

template engine *Razor* like

npm install vash --save

```
<ul class="@model.active ? 'highlight' : ''">
  @model.forEach(function(m) {
    <li>@m.name</li>
  })
</ul>
```



template engine *Razor* like

```
npm install bliss --save
```

```
@!(customer,orders)
```

```
<h1>Welcome @customer.name!</h1>
```

```
<ul>
```

```
@orders.forEach(function(order){
```

```
    <li>@order.title</li>
```

```
})
```

```
</ul>
```

SQL, as you would write it

4

accesso al db (knex  
[+bookshelf])

```
knex.select('*')  
  .from('accounts')  
  .where('activated', 1)  
  .then(function(accounts) {  
    // ...  
  })  
  .catch(function(err) {  
    // handle error  
  })  
}
```

```
// select * from "accounts" where "activated" = 1
```



# npm install knex mssql pg --save

```
const knex = require('knex');

const connectionString = process.env.DATABASE_URL || 'postgres://postgres:pwd@localhost:5432/todo';
bug: false});
const db = knex({client: 'pg', connection: connectionString, searchPath:'knex,public'}, {debug: false});
//const db = knex({client: 'mssql',connection: connectionString,searchPath:'knex,public'}, {de
db.schema.createTableIfNotExists('items', table => {
  table.increments();
  table.string('title', 40).nullable();
  table.boolean('completed').defaultTo(false);
  table.integer('order').defaultTo(99999);
}).asCallback((e, r) => (if (e) return console.error(e); else console.log('database ready')));

export function get (id, cb) {
  db('items').select().where('id', '=', id)
    .asCallback((e, r) => (if (e) return console.error(e); else cb(r[0])));
}

export function create (title, order, cb) {
  db.into('items')
    .insert([{'title': title, 'order': order}]).returning('*')
    .asCallback((e, r) => (if (e) return console.error(e); else cb(r[0])));
}
```



by Jared Hanson

# Passport

## Simple, unobtrusive authentication for Node.js

Passport is authentication middleware for Node.js. Extremely flexible and modular, Passport can be unobtrusively dropped in to any Express-based web application. A comprehensive set of strategies support authentication using a username and password, Facebook, Twitter, and more.

Home

Documentation

Features

Strategies

# 5 autenticazione (passport)

```
passport.authenticate('facebook');
```

+300 Strategies Now!

[VIEW ALL STRATEGIES](#)



# npm install passport passport-local --save

```
var passport = require('passport')
    , LocalStrategy = require('passport-local').Strategy;

passport.use(new LocalStrategy(
  function (username, password, done) {
    User.findOne({ username: username }, function (err, user) {
      if (err) { return done(err); }
      if (!user) return done(null, false, { message: 'Incorrect username.' });
      if (!user.validPassword(password)) return done(null, false, { message: 'Incorrect password.' });

      return done(null, user);
    });
  }
));

app.post('/login', passport.authenticate('local', {
  successRedirect: '/',
  failureRedirect: '/login',
  failureFlash: true
}));
```

# npm install passport passport-twitter --save

```
var passport = require('passport')
    , TwitterStrategy = require('passport-twitter').Strategy;

passport.use(new TwitterStrategy({
  consumerKey: TWITTER_CONSUMER_KEY,
  consumerSecret: TWITTER_CONSUMER_SECRET,
  callbackURL: "http://www.example.com/auth/twitter/callback"
},
  function(token, tokenSecret, profile, done) {
    User.findOrCreate(profile, function(err, user) {
      if (err) { return done(err); }
      done(null, user);
    });
  }
));

// Redirect the user to Twitter for authentication.  When complete, Twitter will redirect the user back to the application at /auth/twitter/callback
app.get('/auth/twitter', passport.authenticate('twitter'));

// Twitter will redirect the user to this URL after approval.  Finish the authentication process by attempting to obtain an access token.
//If access was granted, the user will be logged in.  Otherwise, authentication has failed.
app.get('/auth/twitter/callback',
  passport.authenticate('twitter', { successRedirect: '/', failureRedirect: '/login' }));
```

# bonus track

unit testing



# npm install mocha superhost should --save-dev

```
var request = require('supertest');
var should = require("should");

describe("GET /ping", function () {
  it("respond 'pong!'", function (done) {
    var app = require('express')();
    app.get('/ping', function (req, res) {
      res.send('pong!');
    });
    api.configure(app);
    request(app).
      get('/ping').
      expect(200, 'pong!', done);
  });
});
```

FROM STEPHENIE MEYER THE AUTHOR OF THE TWILIGHT SAGA

CHOOSE

CHOOSE

CHOOSE

TO LOVE  
Node.js cosa ci facciamo?

BELIEVE

FIGHT

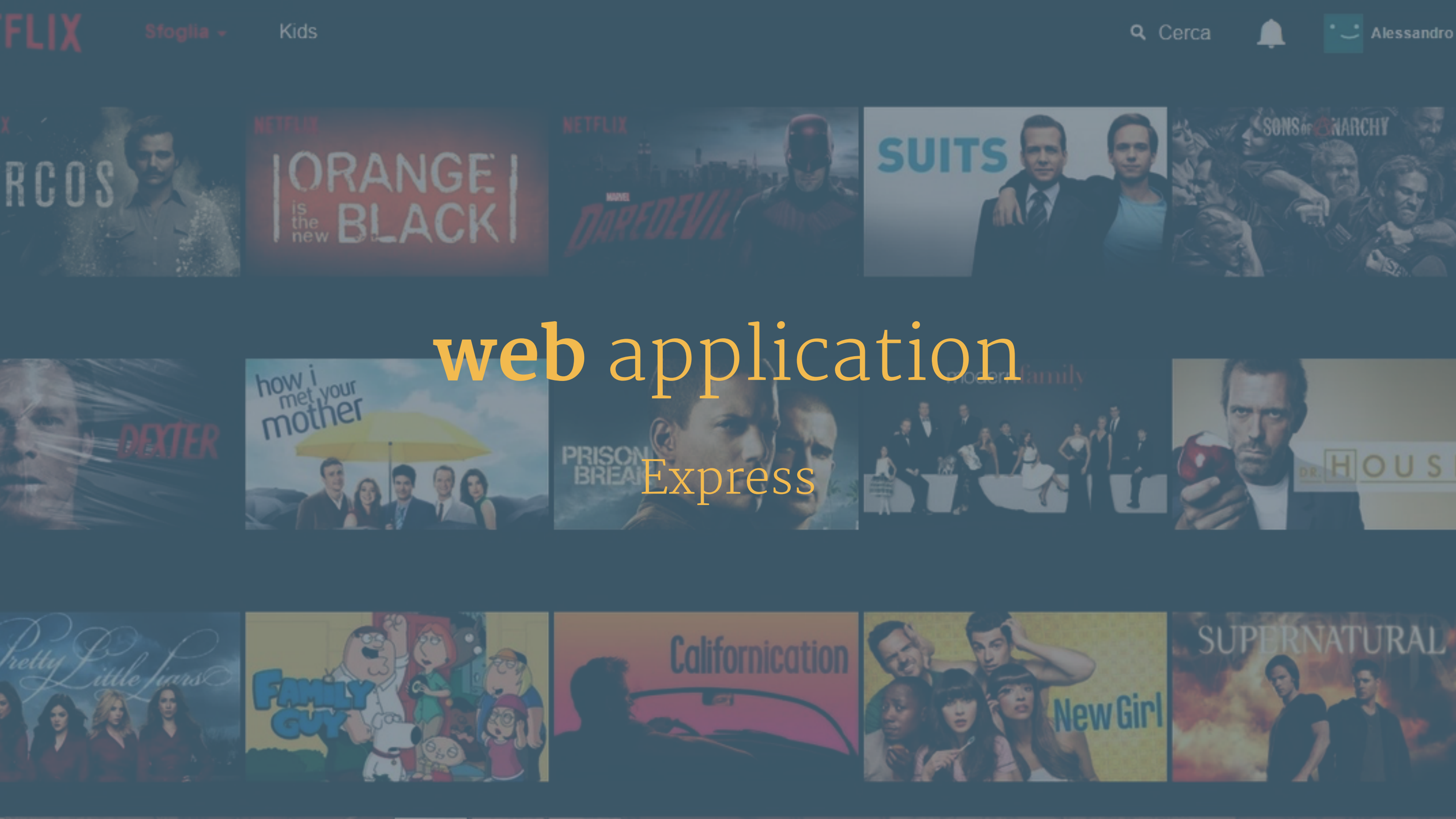
THE HOST

SCREEN AND DIRECTED BY ANDREW NICCOL

YAHOO! MOVIES

IN THEATERS 03.20





FLIX

Sfoglia ▾

Kids

🔍 Cerca



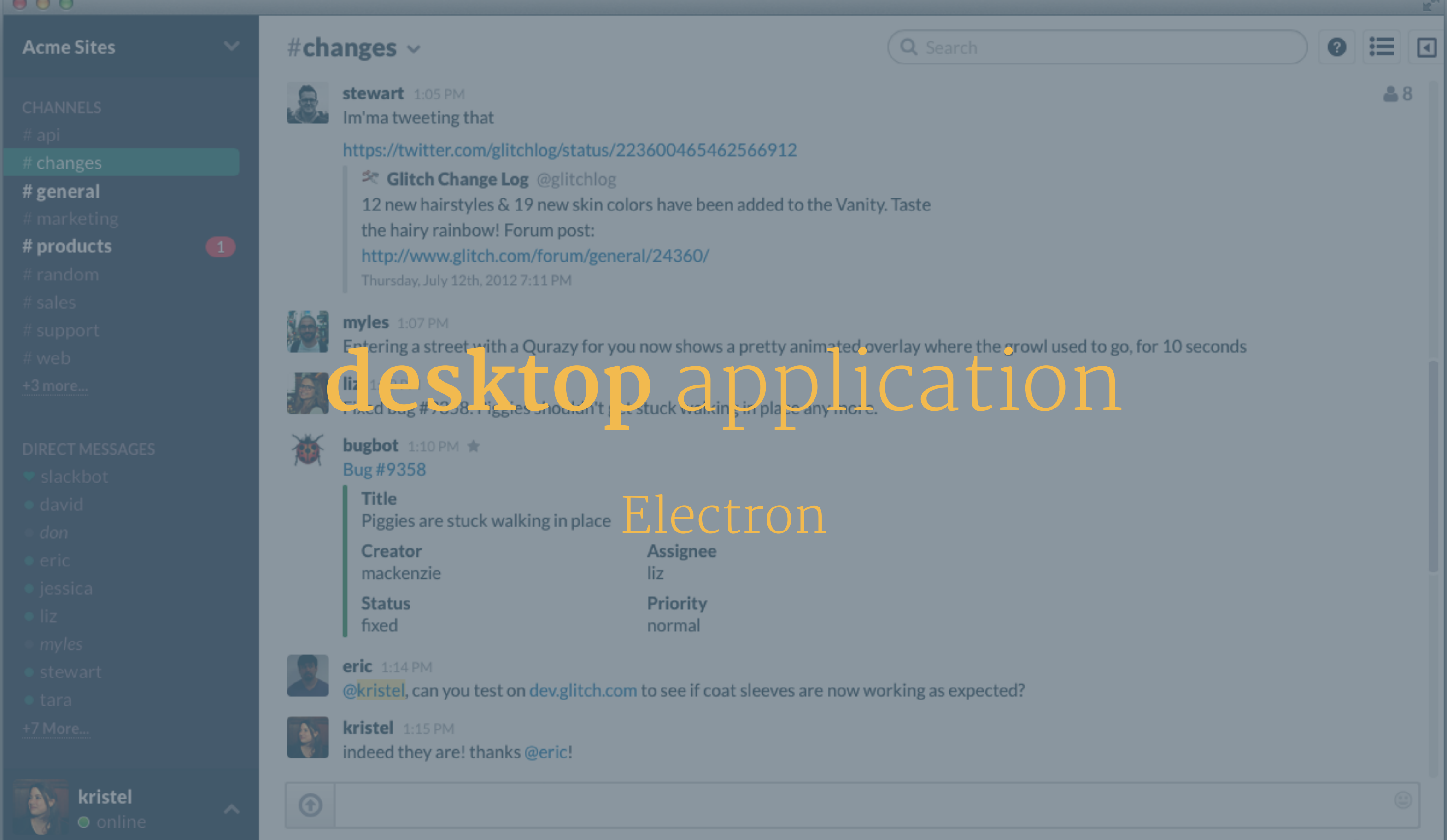
Alessandro



web application

Express





Acme Sites

CHANNELS

- # api
- # changes
- # general
- # marketing
- # products 1
- # random
- # sales
- # support
- # web
- +3 more...

DIRECT MESSAGES

- slackbot
- david
- don
- eric
- jessica
- liz
- myles
- stewart
- tara
- +7 More...

 **kristel**  
online

#changes

Search

Help, List, and Close icons

-  **stewart** 1:05 PM  
Im'ma tweeting that  
<https://twitter.com/glitchlog/status/223600465462566912>  
 **Glitch Change Log** @glitchlog  
12 new hairstyles & 19 new skin colors have been added to the Vanity. Taste the hairy rainbow! Forum post:  
<http://www.glitch.com/forum/general/24360/>  
Thursday, July 12th, 2012 7:11 PM
-  **myles** 1:07 PM  
Entering a street with a Qurazy for you now shows a pretty animated overlay where the cowl used to go, for 10 seconds
-  **liz** 1:08 PM  
Fixed bug #9358. Piggies shouldn't get stuck walking in place any more.
-  **bugbot** 1:10 PM ★  
Bug #9358  

<b>Title</b>	Piggies are stuck walking in place
<b>Creator</b>	mackenzie
<b>Assignee</b>	liz
<b>Status</b>	fixed
<b>Priority</b>	normal
-  **eric** 1:14 PM  
[@kristel](#), can you test on [dev.glitch.com](http://dev.glitch.com) to see if coat sleeves are now working as expected?
-  **kristel** 1:15 PM  
indeed they are! thanks [@eric](#)!

Message input area with a send button (upward arrow) and an emoji picker button (smiley face).



# tool e plugin

(es. grunt, gulp, mocha, forever...)

## Function Url:

https://nirsclassicfunc.azurewebsites.net/api/httpTriggerNode51?code=11be4x7huJoc821a7919-b04rwr1pclq5ayhlu0j15bqwa5rk9ylx00rsr0hd18c1lvterve29

## Code

Save

```
1 module.exports = function(context, req) {  
2   context.log('Node.js HTTP trigger function processed a request. RequestUrl:' + req.originalUrl);  
3  
4   if (req.query.name || (req.body && req.body.name)) {  
5     context.res = {  
6       // status: 200, /* Default */  
7       body: "Hello " + (req.query.name || req.body.name),  
8     };  
9   }  
10  else {  
11    context.res = {  
12      status: 400,  
13      body: "Please pass a name in the query string or the request body.",  
14    };  
15  }  
16  context.done();  
17 };
```

## Develop

1 of 4

The fastest way to edit code is with the code editor, but you can also use Git. This example uses Node.js, but many other languages are also supported. This page also includes a log stream and a test console for helping you debug your function.

Next

Skip the tour and start coding

## Logs

Pause Clear

2016-09-29T17:31:39 Welcome, you are now connected to log-streaming service.





Come lo lancio?

# console application

## foreverjs

```
[sudo] npm install -g forever
$ forever
usage: forever [start | stop | stopall | list] [options] SCRIPT [script options]
```

### options:

start	start SCRIPT as a daemon
stop	stop the daemon SCRIPT
stopall	stop all running forever scripts
list	list all running forever scripts

```
$ forever start simple-server.js
$ forever list
[0] simple-server.js [ 24597, 24596 ]
$ forever list
[0] simple-server.js [ 24611, 24596 ]
```



## Connections



HEXPOWER7 (HEXPOWER7\Scot

Application Pools

Sites

Default Web Site

aspnet\_client

iPhoneTest

LiveStreaming

Mvc1

node



## Modules

Use this feature to configure the native and managed code modules that process requests made to the Web server.


Group by: No Grouping


Name	Code	N
FormsAuthen...	System.Web.Security.FormsA...	N
HttpCacheM...	%windir%\System32\inetsrv\...	N
Http...	%...	N


web server

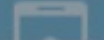
iis node module



 ALL ITEMS


 WEB SITES  
0

 VIRTUAL MACHINES  
0

 MOBILE SERVICES

web sites preview

You have no web sites. Create one to get started!

CREATE A WEB SITE 

NEW




 COMPUTE


 DATA SERVICES


 APP SERVICES

 NETWORKS

 STORE

 WEB SITE  
preview

 VIRTUAL MACHINE  
preview

 MOBILE SERVICE  
preview

 CLOUD SERVICE

 QUICK CREATE

 CREATE WITH DATABASE

 FROM GALLERY

cloud

node azure

URL

nodejsexample



.azurewebsites.net

REGION

East US



# container

## docker

```
FROM node:6.2.2
```

```
RUN mkdir /nodejs
```

```
COPY index.js /nodejs/
```

```
WORKDIR /nodejs
```

```
ENTRYPOINT ["node", "index.js"]
```

```
EXPOSE 80
```





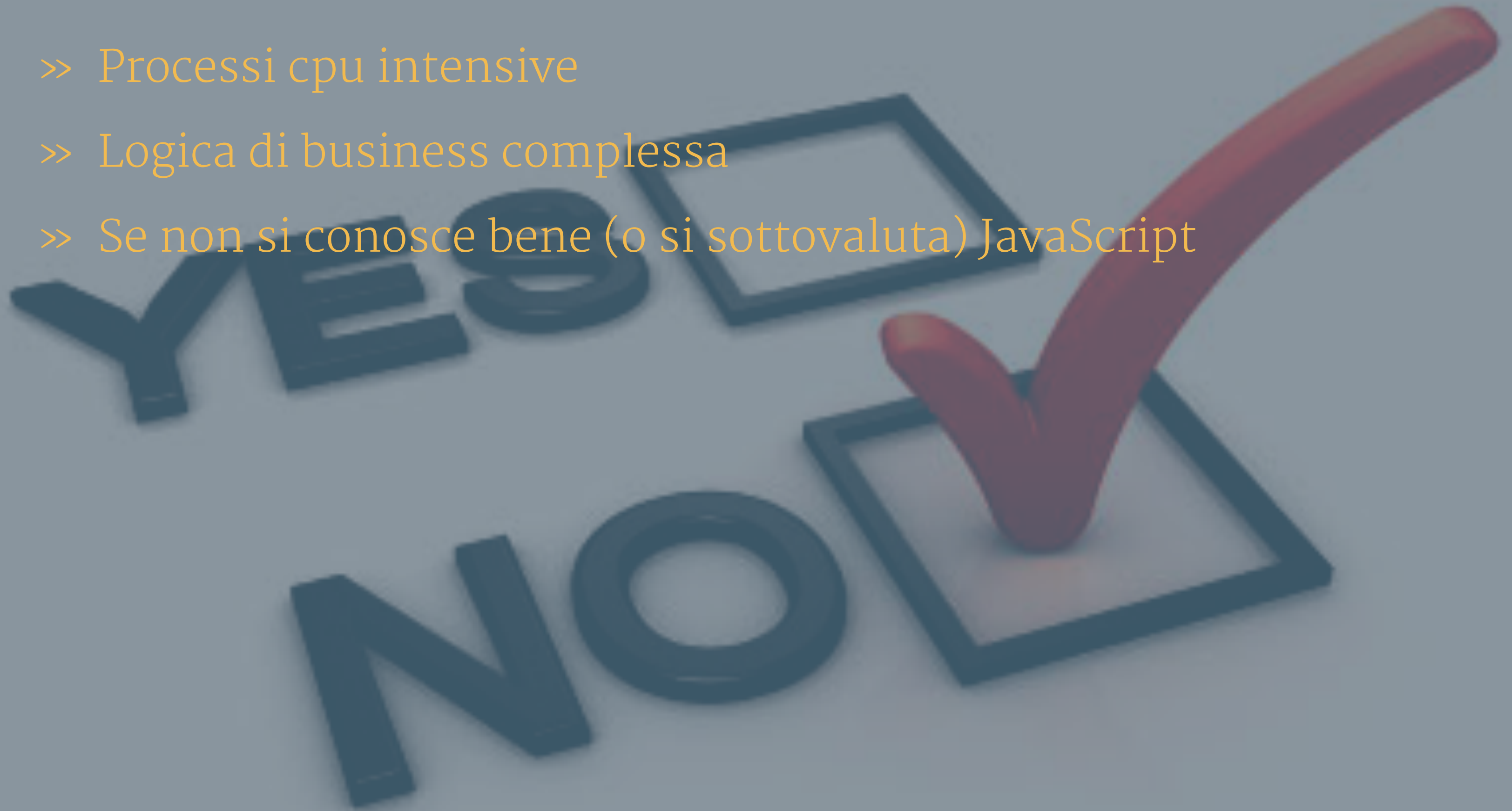
A hand-drawn decision tree diagram on a light blue background. The diagram features a central node with a question mark '?' inside a diamond shape. Three arrows branch out from this central node: one pointing up and to the left towards the word 'Yes', one pointing up and to the right towards the word 'No', and one pointing down and to the right towards the word 'maybe'. A black pen is visible in the bottom right corner, having just finished drawing the diagram.

# Node.Js quando?



- 
- >> Tanti I/O (network, files, db...)
  - >> Realtime application (online games, collaboration tools)
  - >> Come «frontoffice» di un batch server (scritto in Java, .NET, Scala....)
  - >> Vogliamo un unico stack tecnologico tra client e server
  - >> Vogliamo costruire velocemente tools per lo sviluppo

- » Processi cpu intensive
- » Logica di business complessa
- » Se non si conosce bene (o si sottovaluta) JavaScript





THANK YOU








# Domande?

Materiale su [http://  
www.communitydays.it/](http://www.communitydays.it/)

[www.futuredecoded.it](http://www.futuredecoded.it)

 #FutureDecoded