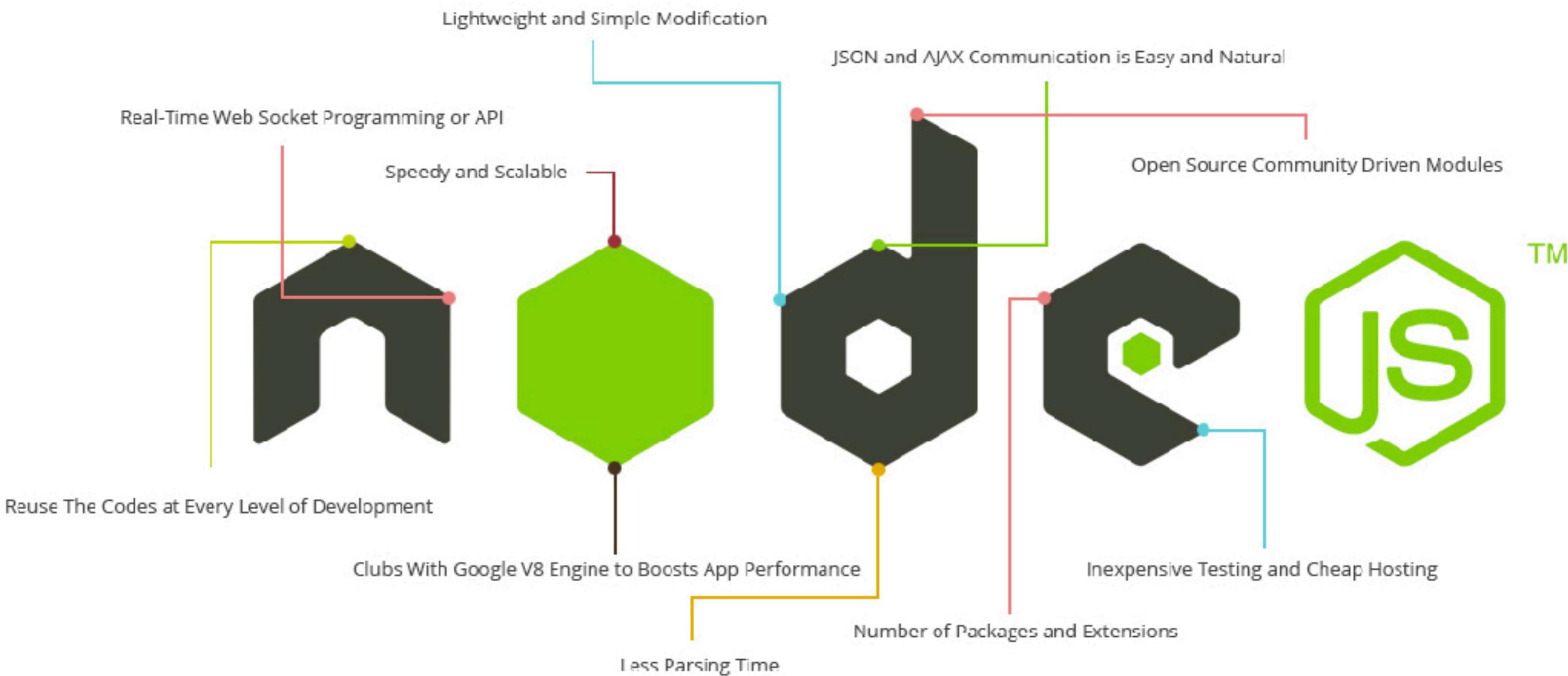


in action 5 days

@somkiat.cc



บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่



Popular ?

[nodejs / node](#)

Watch 2,093 Star 31,000 Fork 5,701

Code Issues 680 Pull requests 339 Projects 5 Wiki Pulse Graphs

Node.js JavaScript runtime 🎉🐢🚀🌟 <https://nodejs.org>

16,182 commits 131 branches 406 releases 1,236 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

Commit	Message	Time Ago
bnoordhuis	src: move trace_event.h include to internal header	Latest commit b1d3c7e 3 days ago
.github	meta: modify pull request template for prepending	28 days ago
benchmark	queryString: improve unescapeBuffer performance	a day ago
deps	deps: upgrade libuv to 1.10.2	14 days ago
doc	tools,doc: add Google Analytics tracking.	14 hours ago
lib	zlib: be strict about what strategies are accepted	13 hours ago

<https://github.com/nodejs/node>



Who use Node.js ?

Manufacturing



General Motors

Johnson Controls

SIEMENS

Financial



citigroup

Goldman
Sachs

PayPal



eCommerce

amazon.com



ebay

TARGET

Zappos[®]

Media



CONDÉ NAST

DOWJONES

The New York Times

SONY.

Technology

salesforce.com



box



YAHOO!

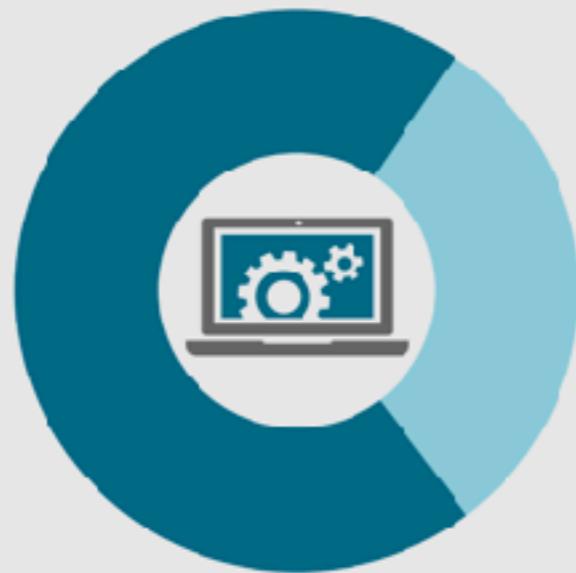
<https://strongloop.com/node-js/why-node/>



Trend of Node.js

60%

of developers were using
Node.js for web apps



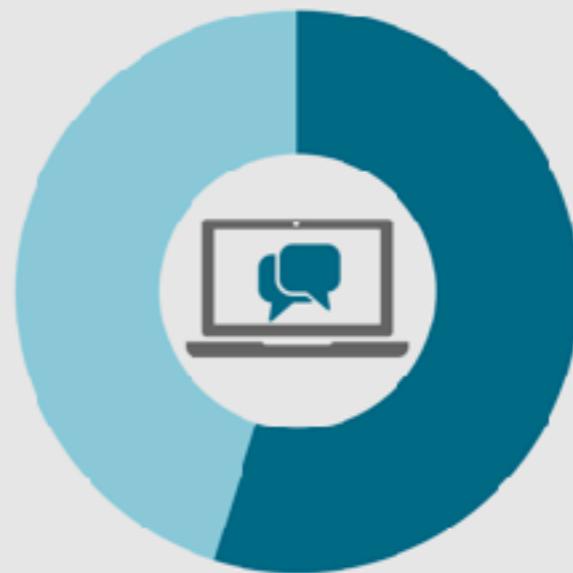
40%

using it for
mobile apps



40%

using it for
realtime web apps



26%

planning on using
in the next year

27%

planning on using
in the next year

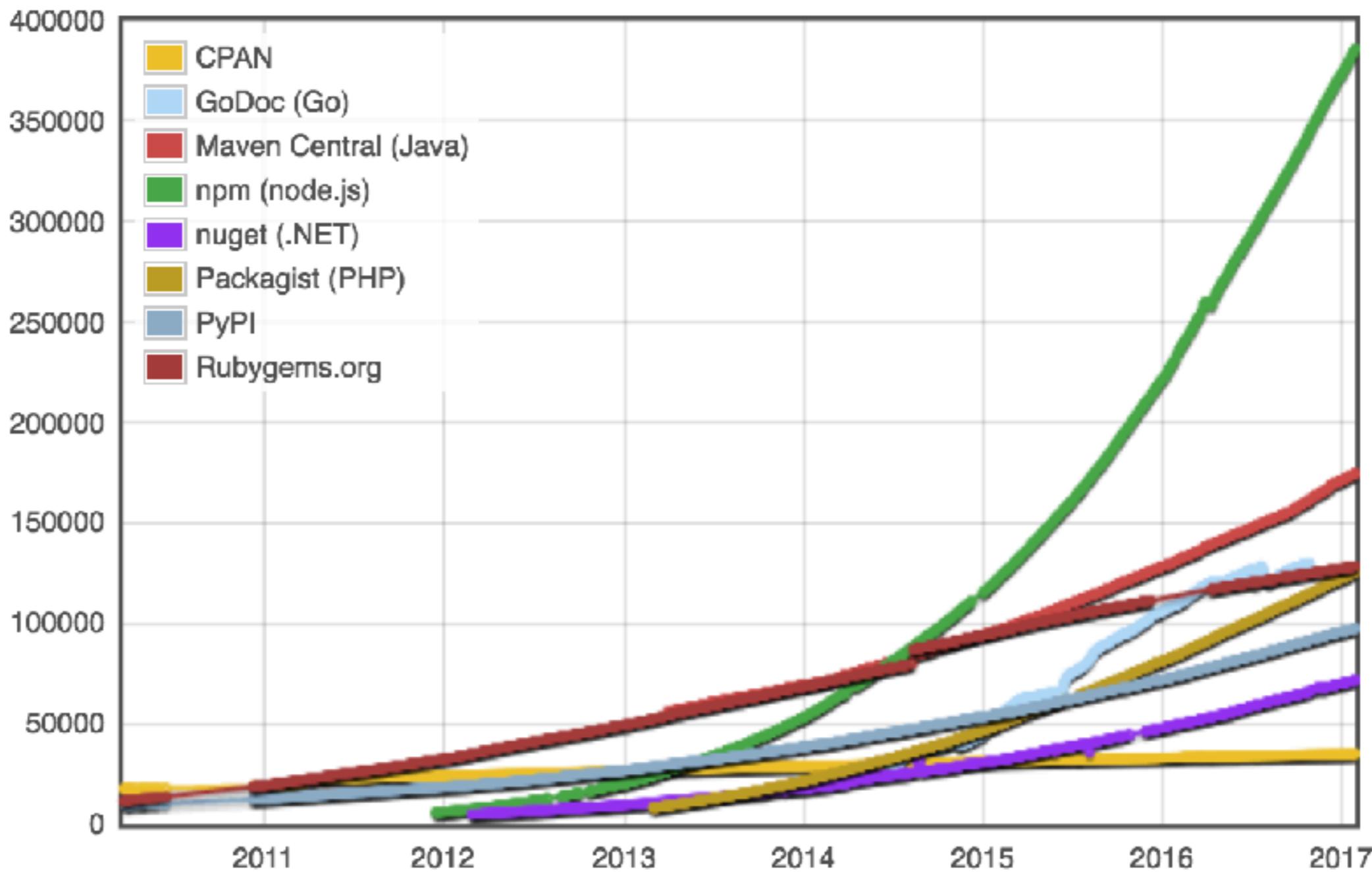
33%

planning on using
in the next year

<https://strongloop.com/node-js/why-node/>



Module count



<http://www.modulecounts.com/>



Day 1



Welcome to Node.js

What is Node.js ?

JavaScript on server

Asynchronous and Event loop

Type of application with Node.js

Sample app



What is Node.js ?

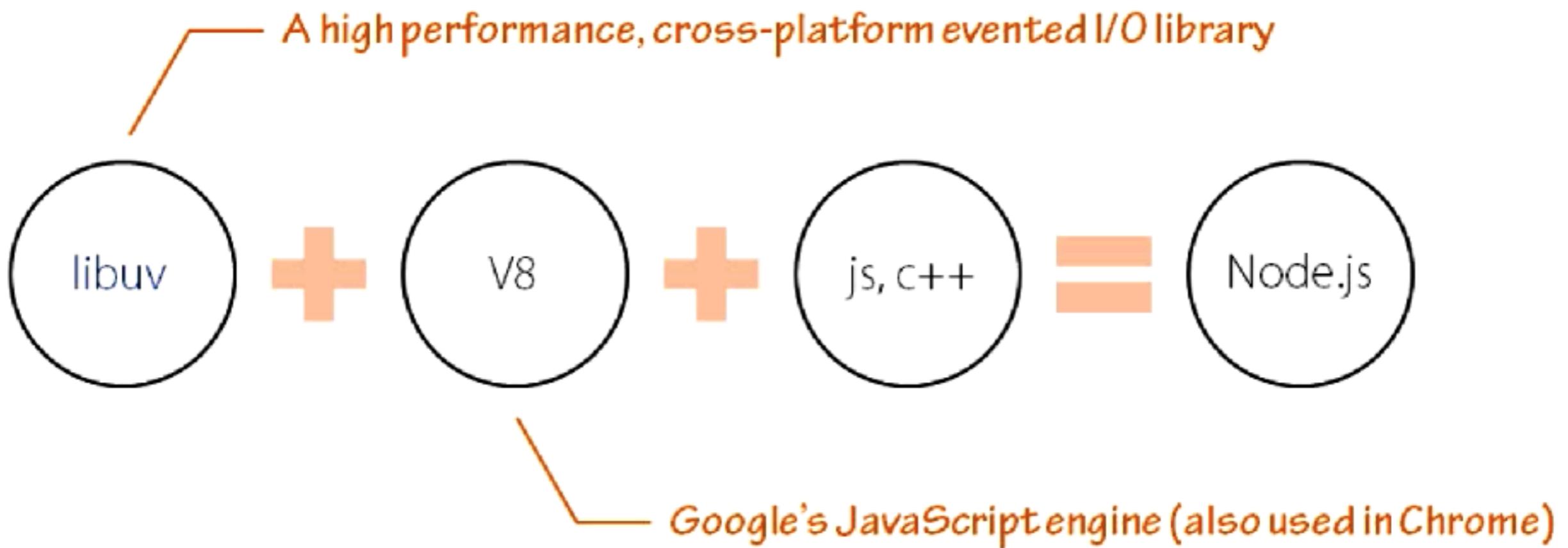
Platform

build on Chrome's **JavaScript** runtime

Fast and **Scalable** application



Building block



What is Node.js ?

Event-driven ?

Non-blocking I/O ?



What is Node.js ?

Event-driven ?

Non-blocking I/O ?



Installation

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

Download for macOS (x64)

v6.9.4 LTS
Recommended For Most Users

v7.4.0 Current
Latest Features

Other Downloads | Changelog | API Docs Other Downloads | Changelog | API Docs

Or have a look at the [LTS schedule](#).



Hello Node.js

```
$node -v
```



Develop with IDE

Atom

Sublime

Cloud9 IDE



C9.io

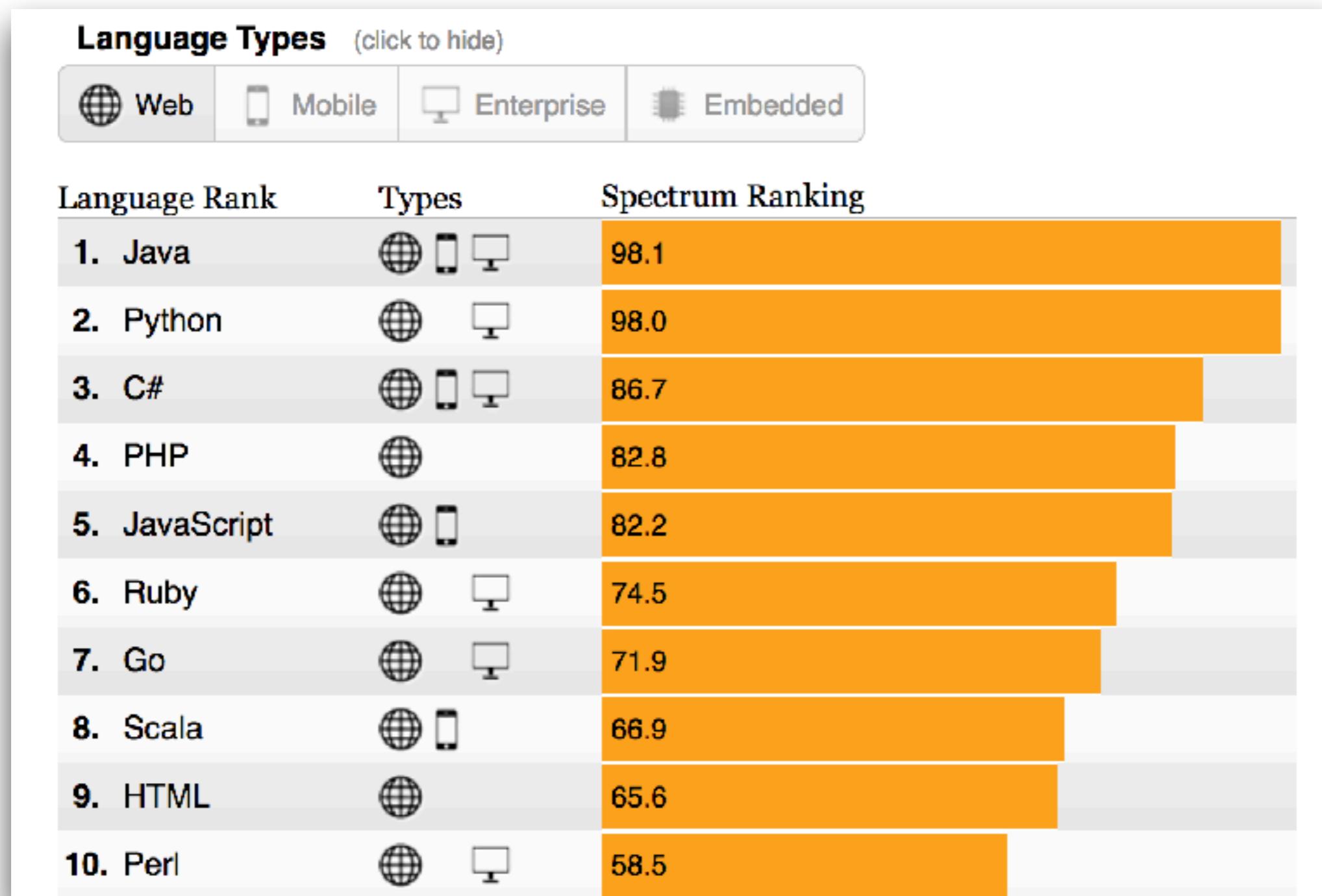
The screenshot shows the Cloud9 IDE interface with several windows open:

- edit_session.js**: A file containing JavaScript code related to session editing. It includes comments explaining methods like `getValue` and `getSelection`.
- hello.js**: A file containing a Node.js script that creates a simple HTTP server. It responds with "Hello World222\n" and listens on port 8080.
- README.md**: A file containing the project's README content.
- package.json**: A file defining the project's dependencies, including "ace" as the main module and "asyncjs", "jsdom", "cmd-loader", and "device" as devDependencies.

The bottom of the interface features a toolbar with various icons for file operations, search, and help.



Build on JavaScript



Build on JavaScript

Write once, Run anywhere

Frontend and Backend



Node use V8

Virtual machine by google chrome
Boost performance



<https://developers.google.com/v8/>



Benefits

Write web app in **one language**

Working with **JSON** (native)

Working with **NoSQL** database

Follow **ECMAScript** standard



Asynchronous & Event driven

Non-blocking I/O
Event loop

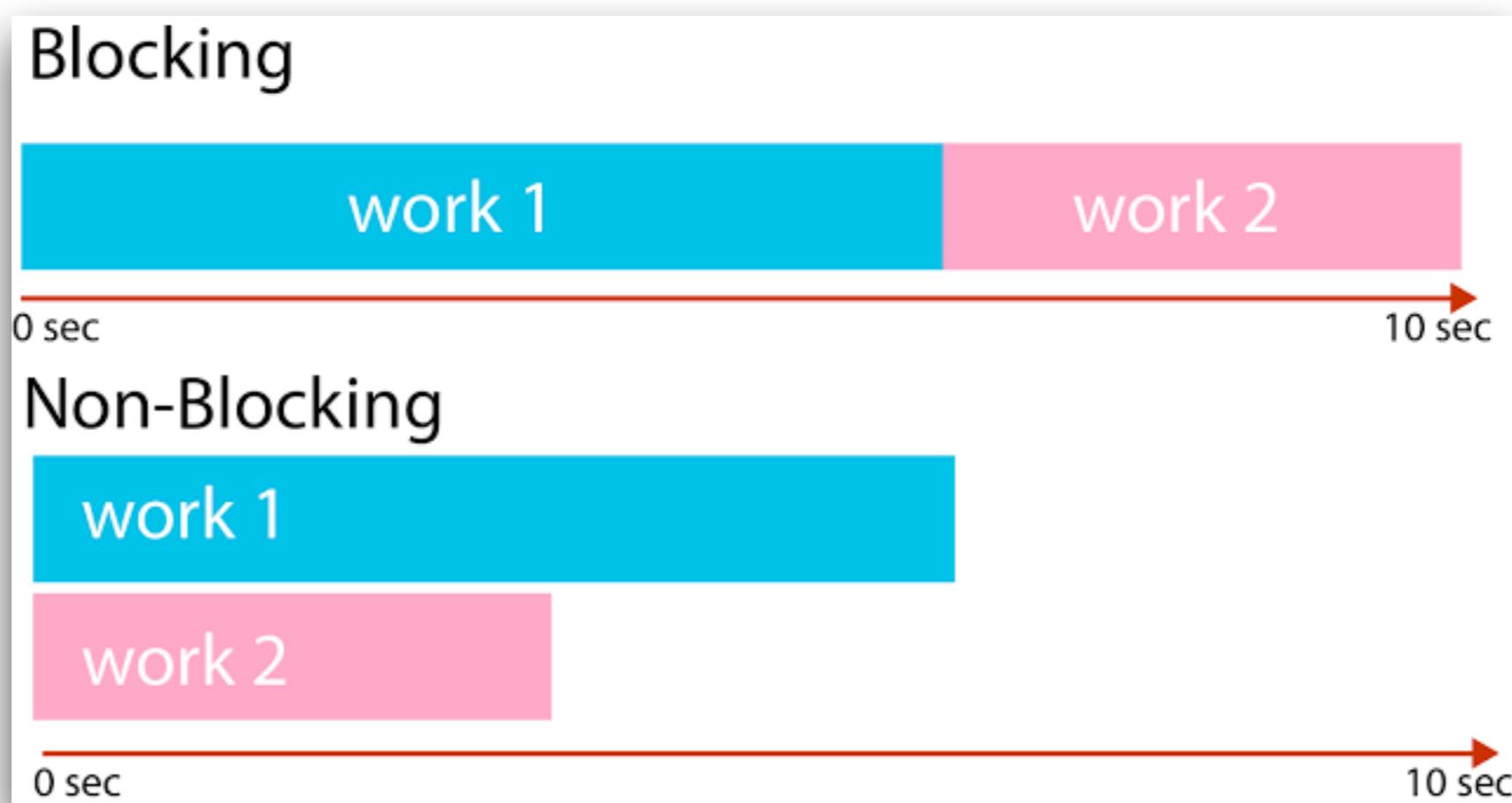


Blocking vs Non-blocking I/O

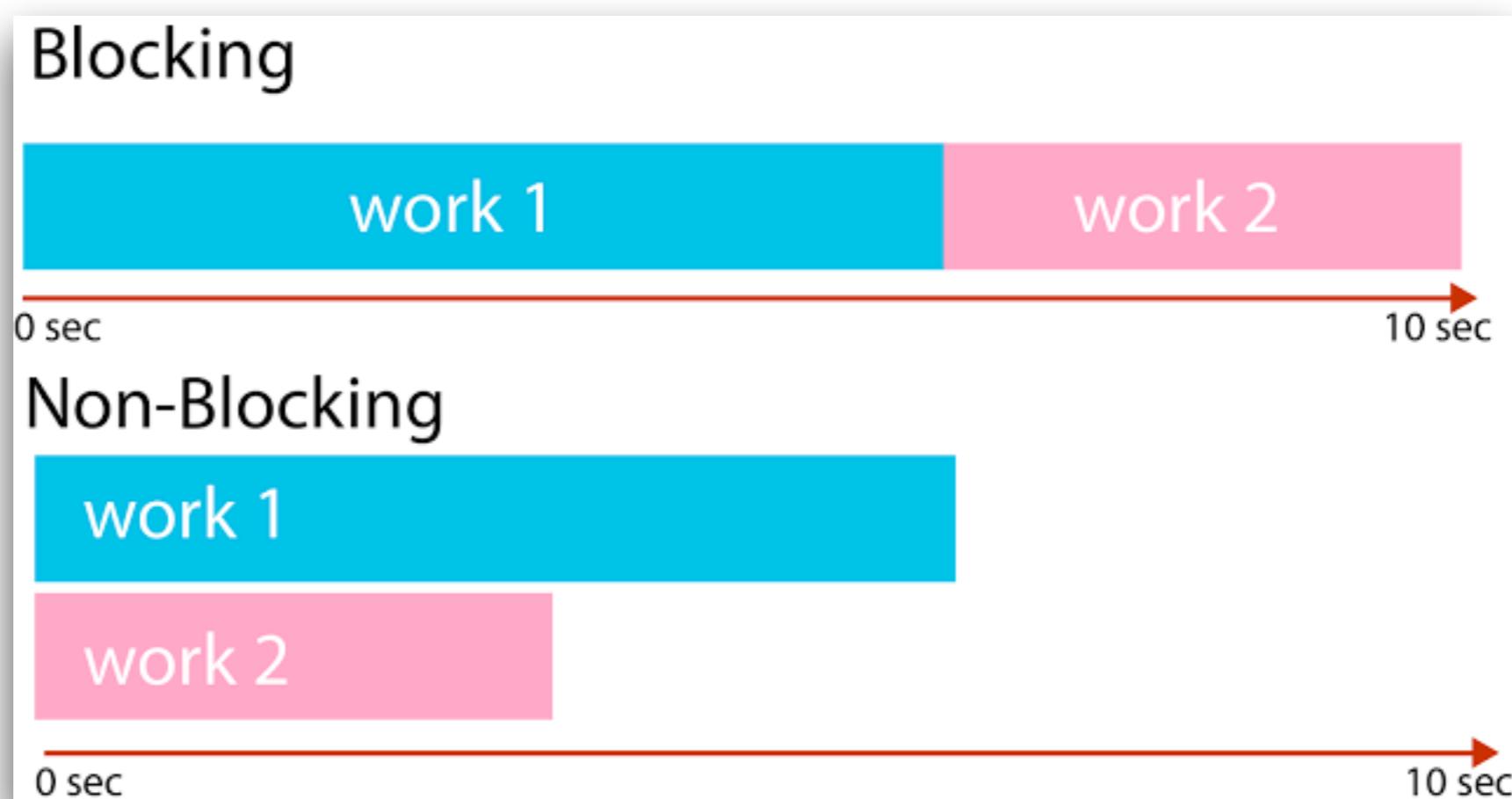
"I/O" refers primarily to interaction
with
the system's **disk** and **network** supported



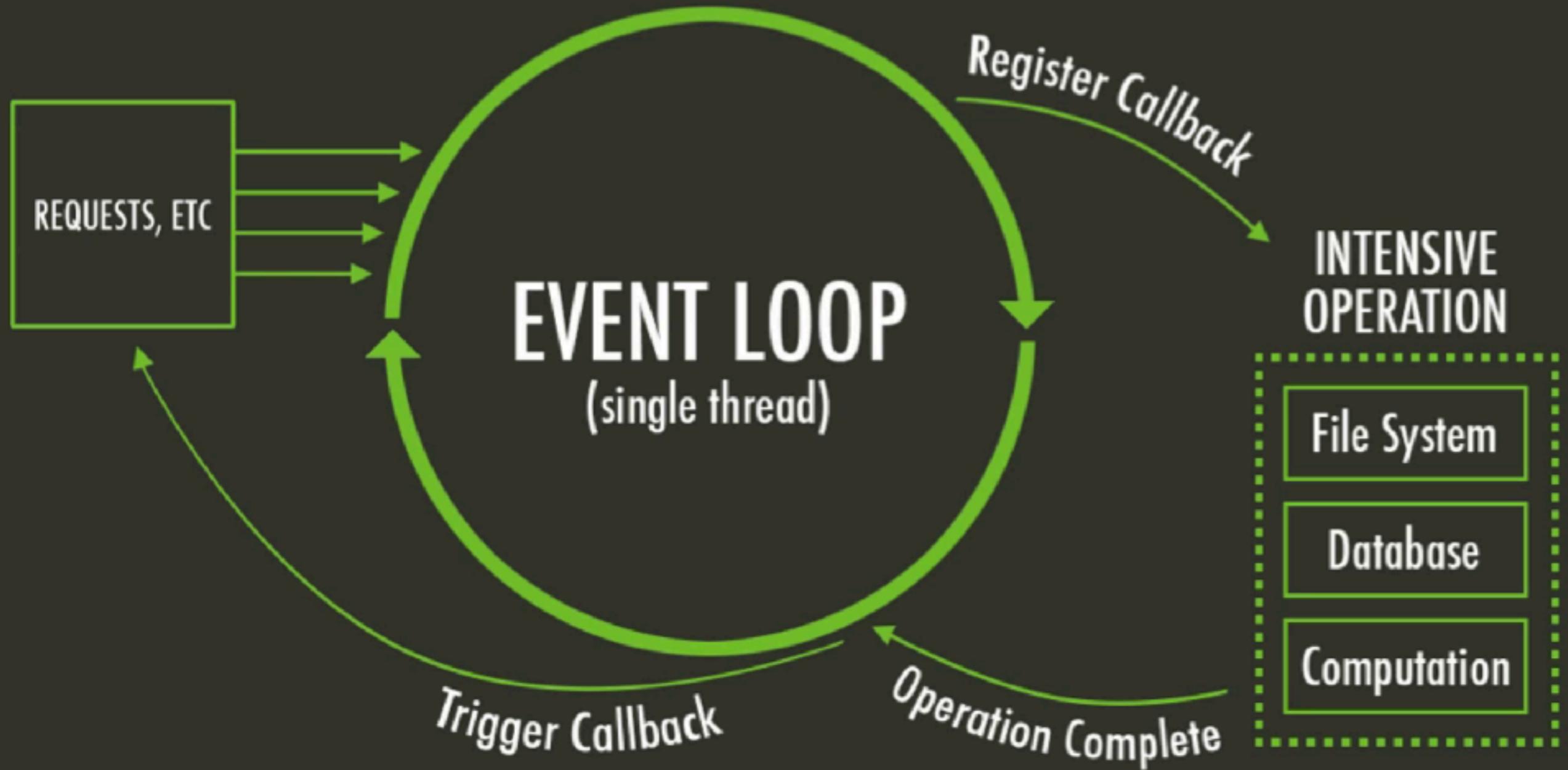
Blocking vs Non-blocking I/O



Synchronous vs Asynchronous



Event loop



Non-blocking by default

All of the I/O method in standard library
provide
asynchronous (non-blocking)



Blocking I/O

Method name is end with **Sync**

eg.

readFileSync()



Let's coding



บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

Basic of JavaScript



Basic of JavaScript

Variables and Assignment

Operators

Conditionals

Loops

Functions



Variables and Assignment

```
var counter = 0;  
var message = "Hello world";  
var isLogin = false;  
  
var users = ["User 1", "User 2"];  
var mixData = [1, 2, false, "User 1", "User 2"];
```



Variables and Assignment

```
var dataDictionary = {"id": 1, "name": "Name 1";  
  
console.log(dataDictionary['id']);  
console.log(dataDictionary.id);  
console.log(dataDictionary.name);
```



Variables and Assignment

```
var userList = [{"id": 1, "name": "Name 1"},  
                {"id": 2, "name": "Name 2"}];  
  
console.log(userList[0].id);  
console.log(userList[0].name);
```



Variables and Assignment

```
var add = function(firstNumber, secondNumber){  
    return firstNumber + secondNumber;  
}  
  
console.log(add(1, 2));
```



Loop

```
var users = ["User 1", "User 2"];  
  
for (var i = 0; i < users.length; i++) {  
    console.log(users[i]);  
}
```



Loop

```
var users = ["User 1", "User 2"];  
  
for (var index in users) {  
    console.log(index + "="> + users[index]);  
}
```



Loop

```
var users = ["User 1", "User 2"];  
  
for (user of users) {  
    console.log(user);  
}
```



Loop

```
var users = ["User 1", "User 2"];  
users.forEach(function(user){  
    console.log(user);  
})
```



Loop

```
var users = ["User 1", "User 2"];  
  
while(user = users.reverse().pop()){  
    console.log(user);  
}
```



Functions

```
function hello(name) {  
    console.log("Hello " + name);  
}  
  
hello("somkiat");
```



Functions

```
var hello = function(name) {  
    console.log("Hello " + name);  
}  
  
hello("somkiat");
```



Functions

```
var hello = (name) => {  
    console.log("Hello " + name)  
}  
  
hello("somkiat");
```



Start with Node.js



Send parameter from command line



Let's start

\$node command.js **somkiat**

1st parameter



Try by yourself

```
process.argv.forEach((val, index) => {  
    console.log(`#${index}: ${val}`);  
});
```

https://nodejs.org/docs/latest/api/process.html#process_process_argv



Blocking I/O

Read data from file system

```
const fs = require('fs');
const data = fs.readFileSync('./data.json');
```



Blocking I/O

\$node sync_read_file.js

```
Start read file ...
{
  "result": "Hello world!!"
}

Finish read file
```



Non-blocking I/O

Read data from file system

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

fs.readFile('./data.json', (error, data) => {
    if(error) throw error
    console.log(data.toString());
});
```



Non-blocking I/O

Callback function

```
const fs = require('fs');
fs.readFile('./data.json', (error, data) => {
  if(error) throw error
  console.log(data.toString());
});
```



Non-blocking I/O

Callback function

```
const fs = require('fs');
fs.readFile('./data.json', function(error, data){
    if(error) throw error
    console.log(data.toString());
});
```



Non-blocking I/O

Callback function

```
const fs = require('fs');
var callback = function(error, data) {
  if(error) throw error
  console.log(data.toString());
};
fs.readFile('./data.json', callback);
```



Non-blocking I/O

\$node async_read_file.js

```
Start read file ...
Finish read file
{
  "result": "Hello world!!"
}
```



Try by yourself

```
fs.readFile('./data.json', callback);  
fs.readFile('./data2.json', callback);
```



Try by yourself

```
setTimeout( function(error, content{  
    console.log("Call ...");  
}, 2000);  
  
console.log("Finish");
```



Mixing blocking and non-blocking



Bad pattern

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

fs.readFile('data.json', (err, data) => {
  if (err) throw err;
  console.log(data);
});

fs.unlinkSync('data.json');
```



Bad pattern

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

fs.readFile('data.json', (err, data) => {
  if (err) throw err;
  console.log(data);
});

fs.unlinkSync('data.json');
```

Non-blocking

Blocking



Good pattern

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

fs.readFile('data.json', (err, data) => {
    if (err) throw err;
    console.log(data);

    fs.unlink('data.json', (err) => {
        if (err) throw err;
    });
});
```

Non-blocking

Non-blocking

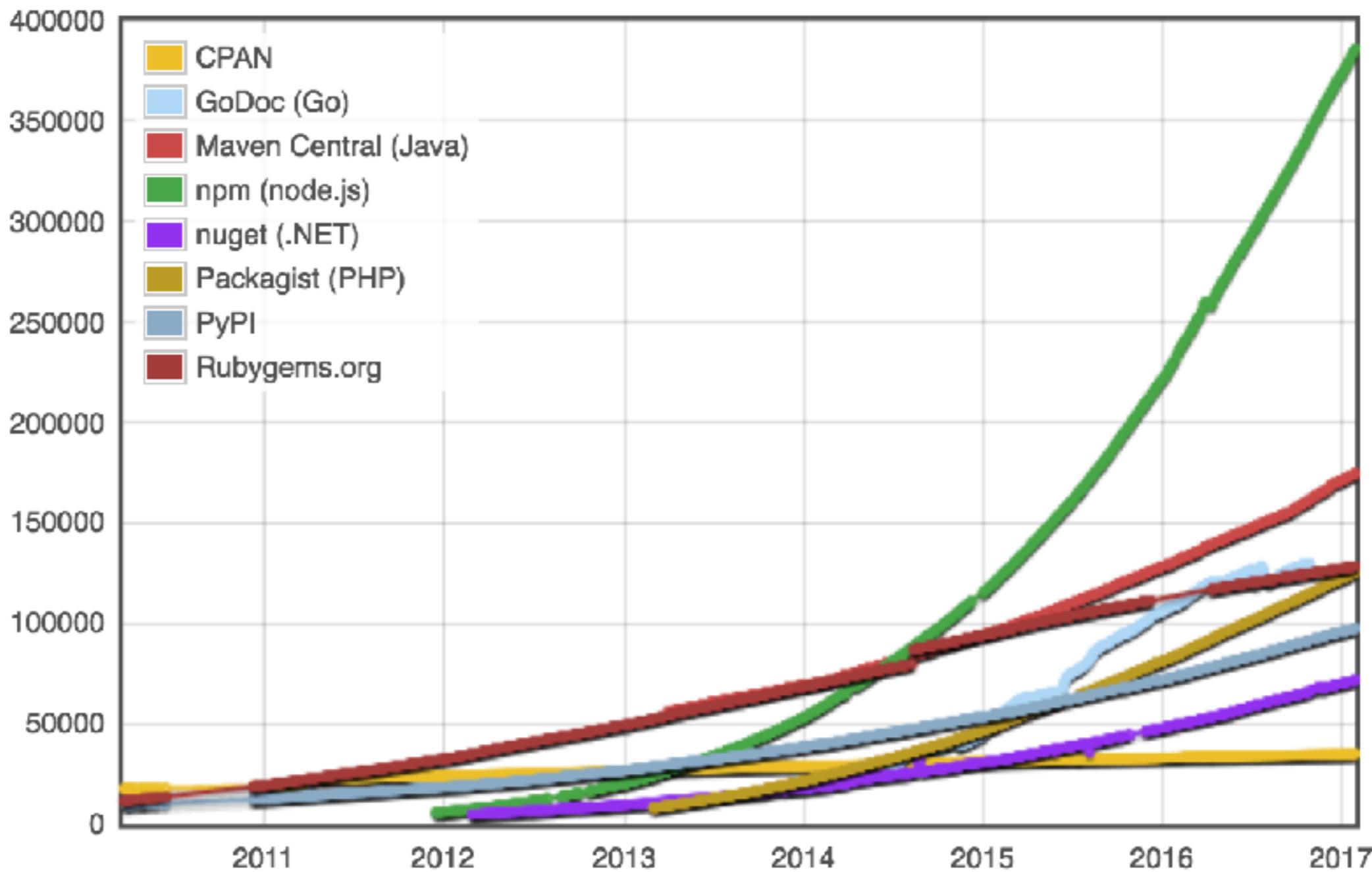


Modular



บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

Module count



<http://www.modulecounts.com/>



Include external module

```
$npm install <module name>
```



Include external module

```
var express = require('express');
var io = require('socket.io');
```



NPM (Node Package Manager)

<https://www.npmjs.com/>

The screenshot shows the homepage of npmjs.com. At the top left is the red 'npm' logo. To its right is a search bar with the placeholder 'find packages'. Further right is a search icon (a magnifying glass) and a 'sign up or log in' link with a small user icon. The main headline 'Build amazing things' is displayed prominently in white text against a dark blue background. Below the headline, a paragraph explains what npm is: 'npm is the package manager for JavaScript. Find, share, and reuse packages of code from hundreds of thousands of developers — and assemble them in powerful new ways.' A large orange 'Get started' button is located at the bottom left of the main content area. The background features a stylized illustration of a city skyline with various buildings, a bridge, a crane, and a bear carrying boxes. The bottom of the page has a solid orange footer bar.

npm

find packages

Q sign up or log in

Build amazing things

npm is the package manager for JavaScript. Find, share, and reuse packages of code from hundreds of thousands of developers — and assemble them in powerful new ways.

Get started

บริษัท สยามซัมนาณภิจ จำกัด และเพื่อนพ้องน้องพี่

Find package

npm | sign up or login 

SEARCH BY Best Overall Quality Popularity Maintenance

12814 PACKAGES FOUND
for "json"

json trentm
a 'json' command for massaging and processing JSON on the command line
v9.0.4

JSON coolaj86
Douglas Crockford's json2.js
v1.0.0

jsonfile jprichardson
Easily read/write JSON files.
v2.4.0

Yellow Pages Canada,
[microapps](#), [QuikOrder](#) and
lots of other companies are
hiring Javascript developers.
See all 32 hiring companies.

Thanks to [npms.io](#)
For improving our search. Want
to learn more about these
results?



Create your module/package



Calculator.js

```
function add(firstNumber, secondNumber) {  
    return firstNumber + secondNumber;  
}  
  
function minus(firstNumber, secondNumber) {  
    return firstNumber - secondNumber;  
}
```



Export calculator module

```
module.exports.add = add;  
module.exports_MINUS = minus;
```



How to use ?

```
var calculator = require('./calculator.js');

console.log(calculator.add(1, 2));
console.log(calculator.minus(1, 2));
```



Publish your module to NPM



บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

Configuration NPM

\$npm set init.author.name "<name>"

\$npm set init.author.email "<email>"

\$npm set init.author.url "<url>"



Add npm's user

\$npm adduser



Create node module

```
$npm init
```



Package.json

```
{  
  "name": "somkiat_calculator_node",  
  "version": "1.0.1",  
  "description": "Try to create my module",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "author": "",  
  "license": "ISC"  
}
```

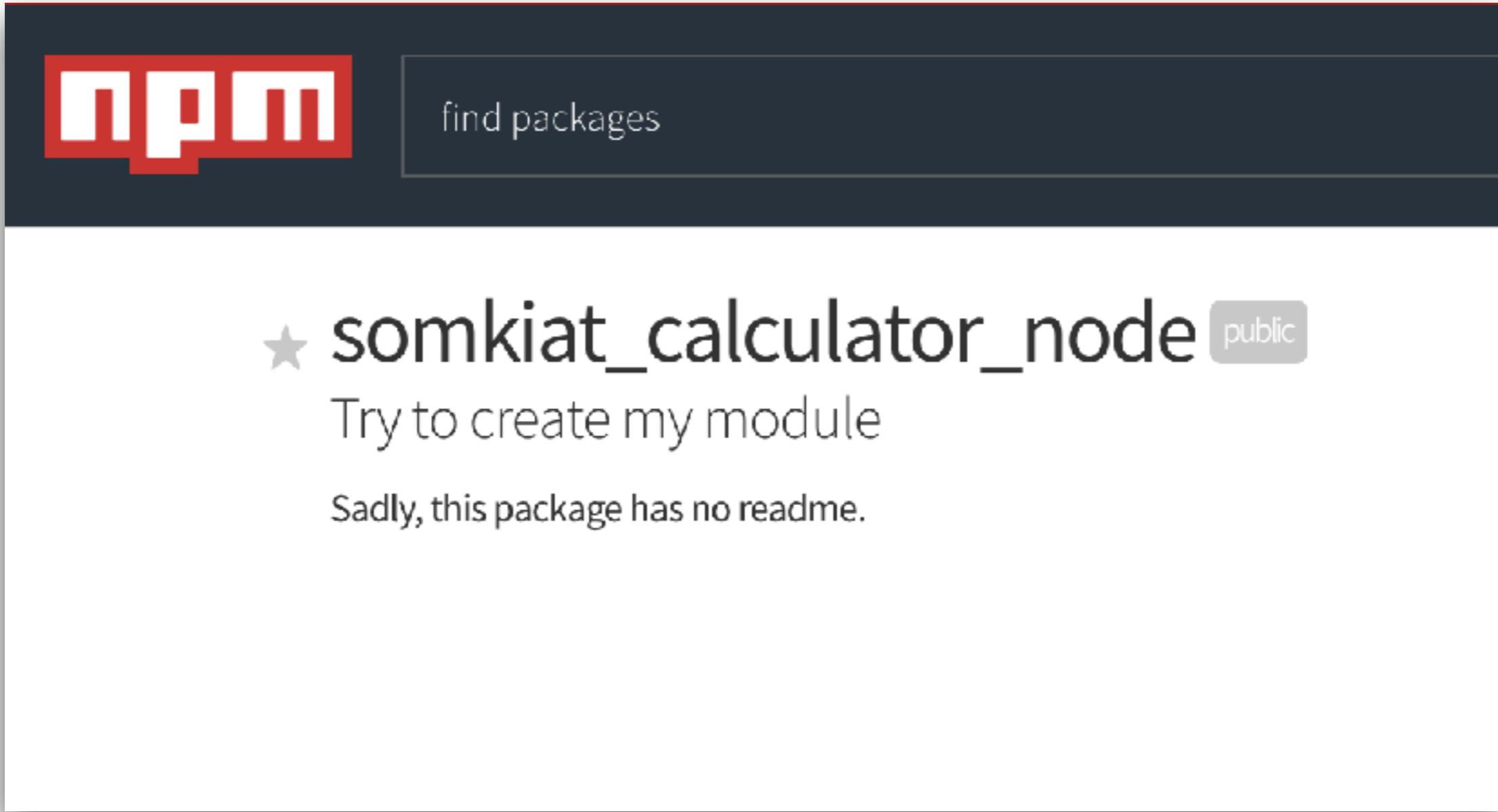


Publish to npm

```
$npm publish
```



Find your package



The screenshot shows the npm search interface. At the top left is the npm logo. To its right is a search bar with the placeholder text "find packages". Below the search bar, the results for the package "somkiat_calculator_node" are displayed. The package name is shown in large, bold, dark text. To its right is a small "public" badge. Below the package name is the text "Try to create my module". Underneath that, it says "Sadly, this package has no readme.".

★ somkiat_calculator_node public

Try to create my module

Sadly, this package has no readme.



Callback hell



Callback Hell !!

The pyramid of doom ...

```
node95.js *  
1 var floppy = require('floppy');  
2  
3 floppy.load('disk1', function (data1) {  
4     floppy.prompt('Please insert disk 2', function () {  
5         floppy.load('disk2', function (data2) {  
6             floppy.prompt('Please insert disk 3', function () {  
7                 floppy.load('disk3', function (data3) {  
8                     floppy.prompt('Please insert disk 4', function () {  
9                         floppy.load('disk4', function (data4) {  
10                            floppy.prompt('Please insert disk 5', function () {  
11                                floppy.load('disk5', function (data5) {  
12                                    // if node.js would have existed in 1995  
13                                });  
14                            });  
15                        });  
16                    });  
17                });  
18            });  
19        });  
20    });  
21});  
22
```



How to solve this problem ?



How to solve this problem ?

Nested approach

Modular approach

Async/Await approach

Promise approach

Generator approach



Workshop

Largest file in folder



Hello HTTP Server



บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

Hello HTTP Server

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(3000);
console.log('Server running at http://localhost:3000/');
```

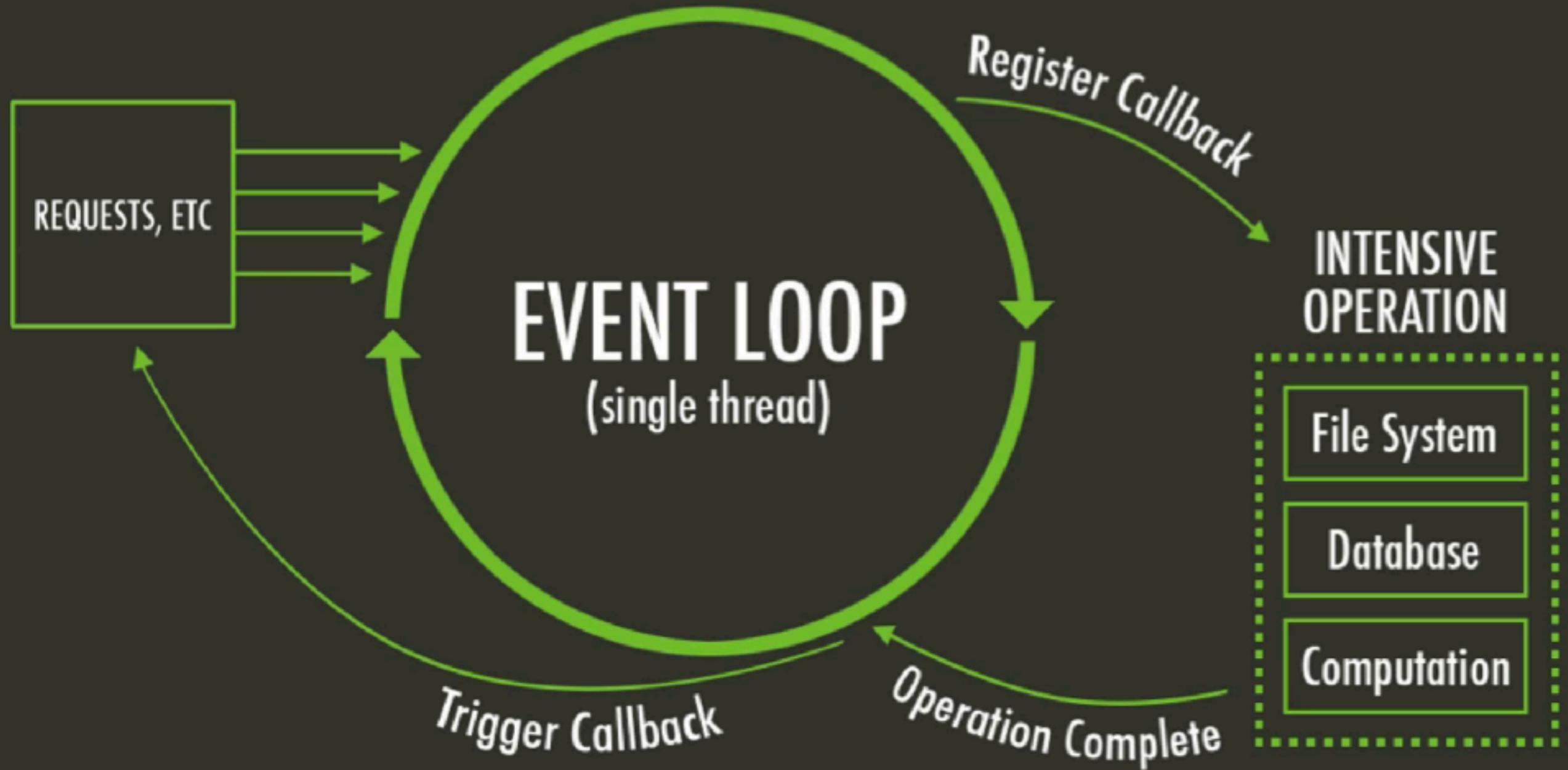


Hello HTTP Server

```
$node hello_server.js
```



Event loop



Improve structure

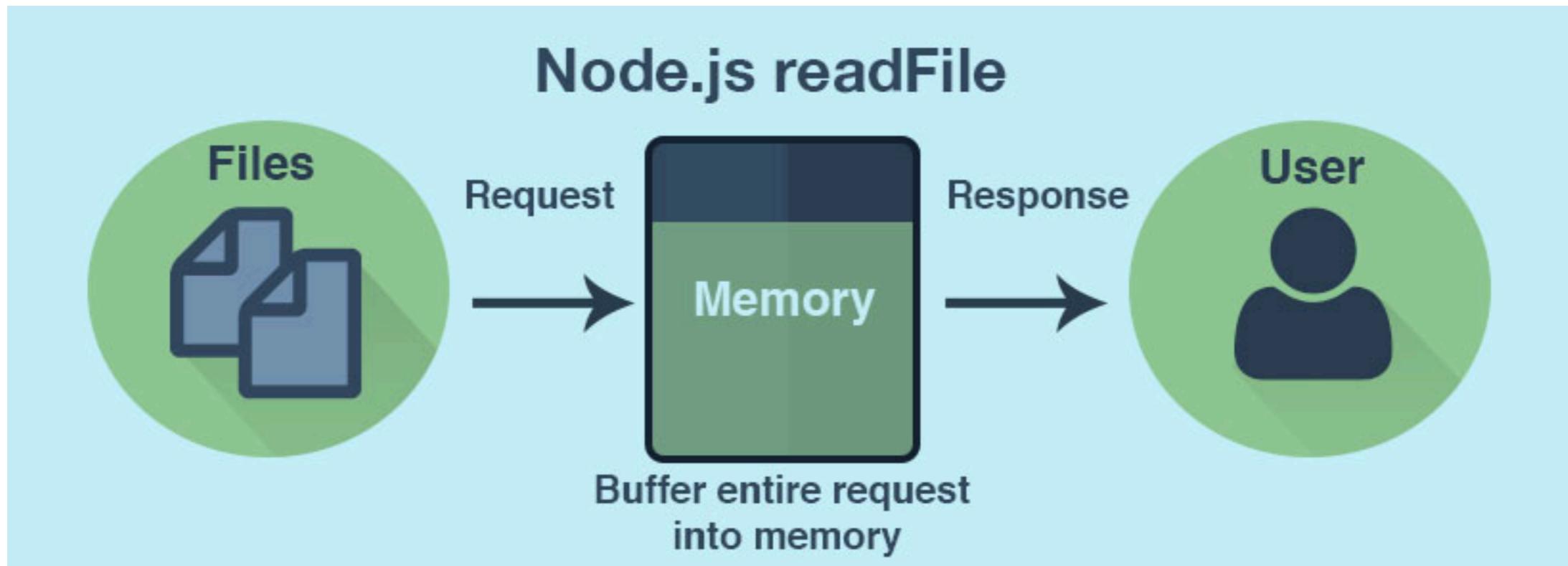
```
var http = require('http');  
  
var server = http.createServer();  
server.on('request', function(request, response){  
  response.writeHead(200, {'Content-Type': 'text/plain'});  
  response.end('Hello World\n');  
});  
server.listen(3000);  
  
console.log('Server running at http://localhost:3000/');
```



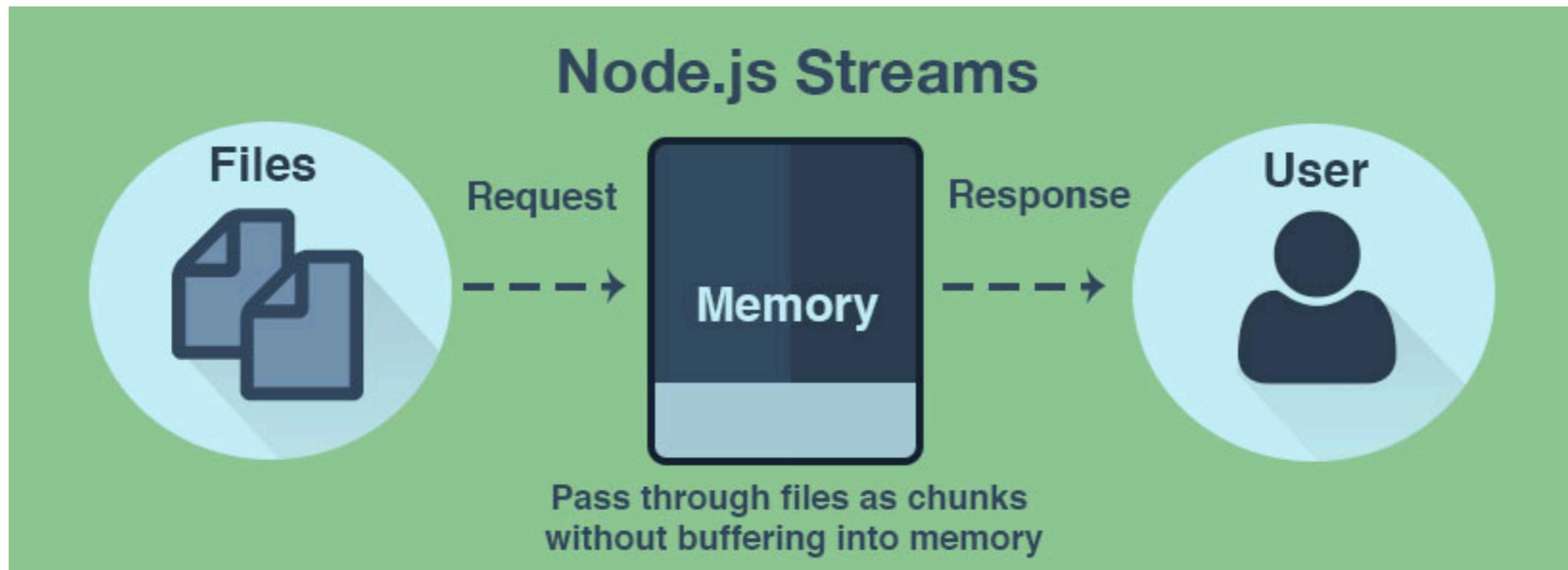
Streaming data server



Read file with Blocking



Read file with stream



Read file with stream

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

var stream = fs.createReadStream('data.txt', 'UTF-8');

var data = ";
```



Read file with stream

```
stream.once('data', function(){
    console.log('Started Reading File');
});

stream.on('data', function(chunk){
    process.stdout.write(`chunk: ${chunk.length} \n`);
    data += chunk;
});

stream.on('end', function(){
    console.log(`Finished Reading File ${data.length}`);
});
```



Develop streaming server

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

http.createServer(function(request, response){
    response.writeHead(200, {'Content-Type': 'application/pdf'});
    var stream = fs.createReadStream('data.txt', 'UTF-8');
    stream.pipe(response);
}).listen(3000);

console.log('Server running at http://localhost:3000/');
```



Workshop with Upload file ?



Upload file

```
http.createServer(function(request, response){  
  var stream = fs.createWriteStream('data2.txt', 'UTF-8');  
  request.pipe(stream);  
  request.on('end', function() {  
    response.end('uploaded!');  
  });  
}).listen(3000);
```

Create file
on server



Upload file

```
http.createServer(function(request, response){  
  var stream = fs.createWriteStream('data2.txt', 'UTF-8');  
  request.pipe(stream);  
  
  request.on('end', function() {  
      response.end('uploaded!');  
  });  
  
}).listen(3000);
```



Handle end event



Summary

Build on JavaScript

Event and Asynchronous

Designed for data-intensive app

Designed for real-time app



Day 2



Building web with Node.js



Building web with Node.js

Handling HTTP requests

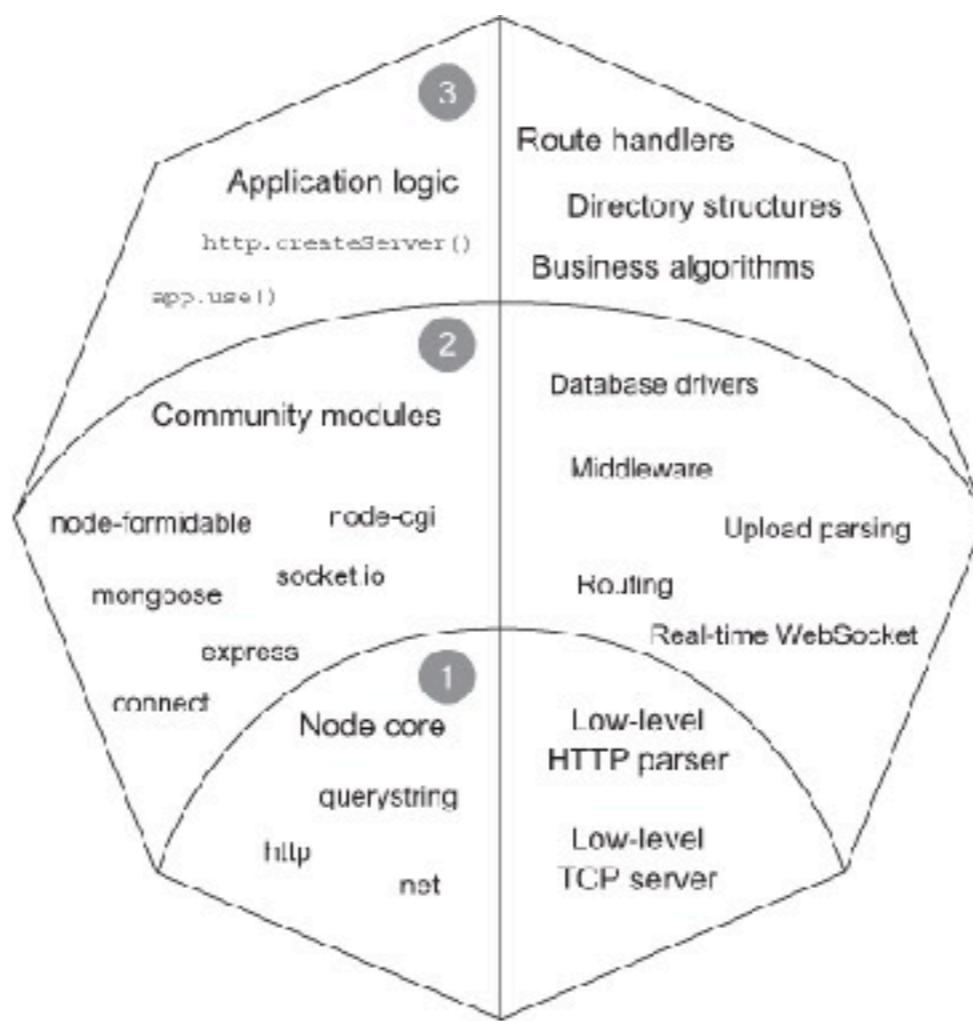
Building RESTful APIs

Serving static files

Accept user input from forms



Layer of Node web app



1 Node's core APIs are always lightweight and low-level. This leaves opinions, syntactic sugar, and specific details up to the community modules.

2 Community modules are where Node thrives. Community members take the low-level core APIs and create fun and easy-to-use modules that allow you to get tasks done easily.

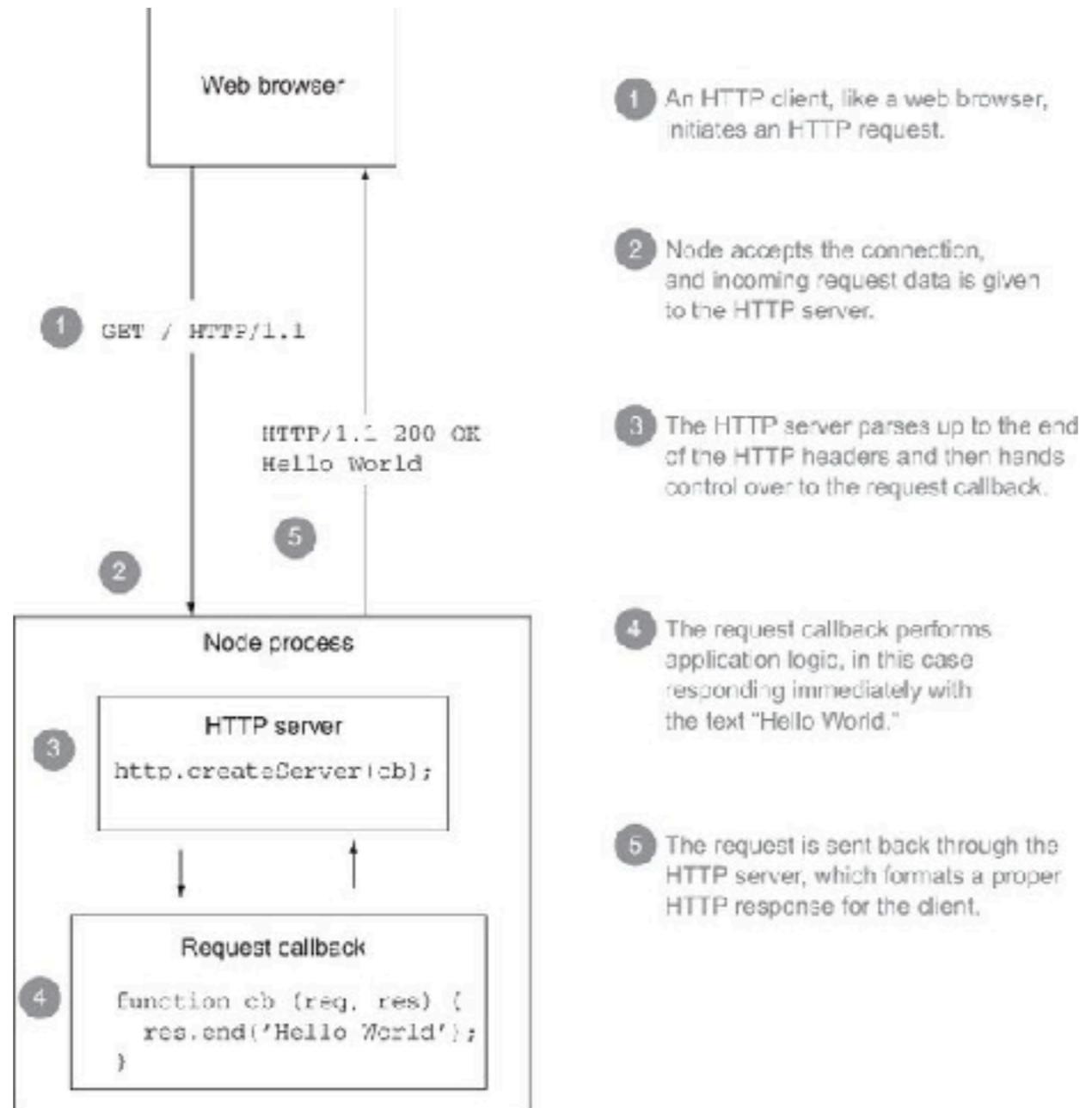
3 The application logic layer is where your app is implemented. The size of this layer depends on the number of community modules used and the complexity of the application.



Basic of HTTP server



HTTP Server



Handling request

```
var http = require('http');
var server = http.createServer(function(req, res){
  // handle request
});
```



Send response

```
var http = require('http');
var server = http.createServer(function(req, res){
  res.write('Hello World');
  res.end();
});
```



Send response

```
var http = require('http');
var server = http.createServer(function(req, res){
  res.end('Hello World');
});
```



Listen for incoming request

```
var http = require('http');
var server = http.createServer(function(req, res){
    res.end('Hello World');
});
server.listen(3000);
```



Setting response headers

```
var http = require('http');
var server = http.createServer(function(req, res){
    var body = 'Hello World';
    res.setHeader('Content-Length', body.length);
    res.setHeader('Content-Type', 'text/plain');
    res.statusCode = 200;
    res.end(body);
});
server.listen(3000);
```



Building



RESTFul ?



- GET
- POST
- PUT
- DELETE

RESTFul

POST : Add data

GET : Retrieve data

DELETE : Delete data

PUT : Update data



Let's coding



บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

express

<http://expressjs.com/>



บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

Express

Fast

Easy and Simple

Flexible

Minimal



Installation

```
$npm install express
```



Hello express

```
var express = require('express');
var app = express();
app.get('/', function(request, response){
    response.setHeader('Content-Type', 'text/plain');
    response.end('Hello, world!');
});
app.listen(3000);
```

Routing



Routing

```
app.get('/', function(request, response){  
    response.end('Hello world with GET');  
});
```

```
app.post('/', function(request, response){  
    response.end('Hello world with POST');  
});
```



Routing with parameters

```
app.get('/users/:userId/books/:bookId',  
    function (req, res) {  
  
        res.send(req.params)  
  
    })
```



App.route()

```
app.route('/book')
  .get(function (req, res) {
    res.send('Get a random book')
  })

  .post(function (req, res) {
    res.send('Add a book')
  })

  .put(function (req, res) {
    res.send('Update the book')
  });
}
```



Route modular

```
var express = require('express')
var router = express.Router()

router.get('/', function (req, res) {
  res.send('Birds home page')
})

router.get('/about', function (req, res) {
  res.send('About birds')
})

module.exports = router
```



Route modular

```
var birds = require('./birds');  
app.use('/birds', birds)
```



Working with JSON



Working with JSON

```
var photos = [];  
  
photos.push({  
    name: "First",  
    path: "image01.jpg"  
});  
  
app.get('/photo', function(request, response){  
    response.json(photos)  
});
```



Let's workshop with REST



Let's workshop with REST

HTTP Method	URL	Action
GET	/photo	Get all photo
GET	/photo/:id	Get photo by id
POST	/photo	Add new photo
PUT	/photo	Update photo
DELETE	/photo/:id	Delete photo by id



Easy to start with generator



Express generator

```
$npm install express-generator -g
```



Express generator

\$express -h

```
Usage: express [options] [dir]
```

Options:

-h, --help	output usage information
--version	output the version number
-e, --ejs	add ejs engine support
--pug	add pug engine support
--hbs	add handlebars engine support
-H, --hogan	add hogan.js engine support
-v, --view <engine>	add view <engine> support (ejs hbs hjs jade pug twig vash) (defaults to jade)
-c, --css <engine>	add stylesheet <engine> support (less stylus compass sass) (defaults to plain css)
--git	add .gitignore
-f, --force	force on non-empty directory

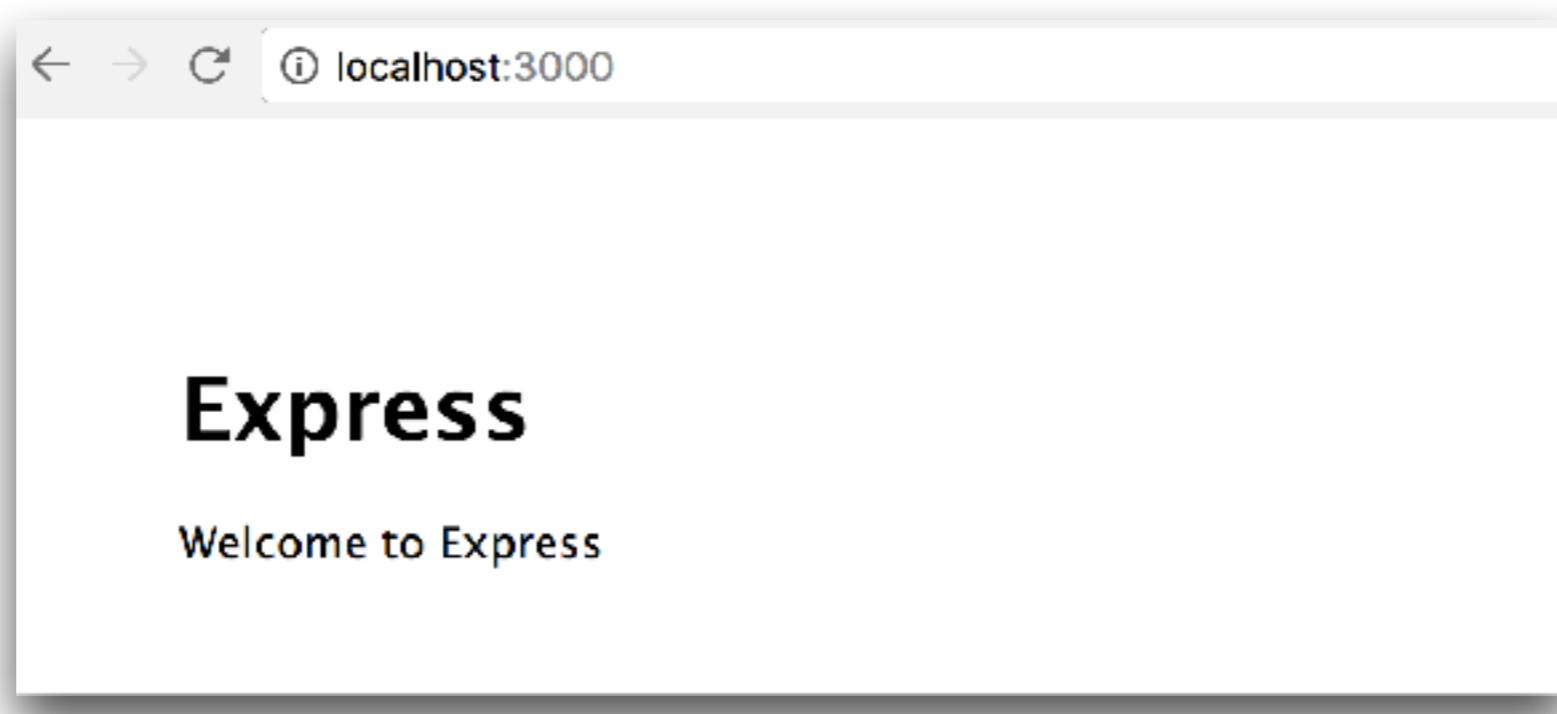


Create application

```
$express --view=ejs myapp
```

```
$npm install
```

```
$npm start
```



Server static files



Static files

Cascade Style Sheet (CSS)

JavaScript

Images

Audio/VDO



Static with express

```
app.use(express.static('public'))
```

```
app.use('/static', express.static('public'))
```

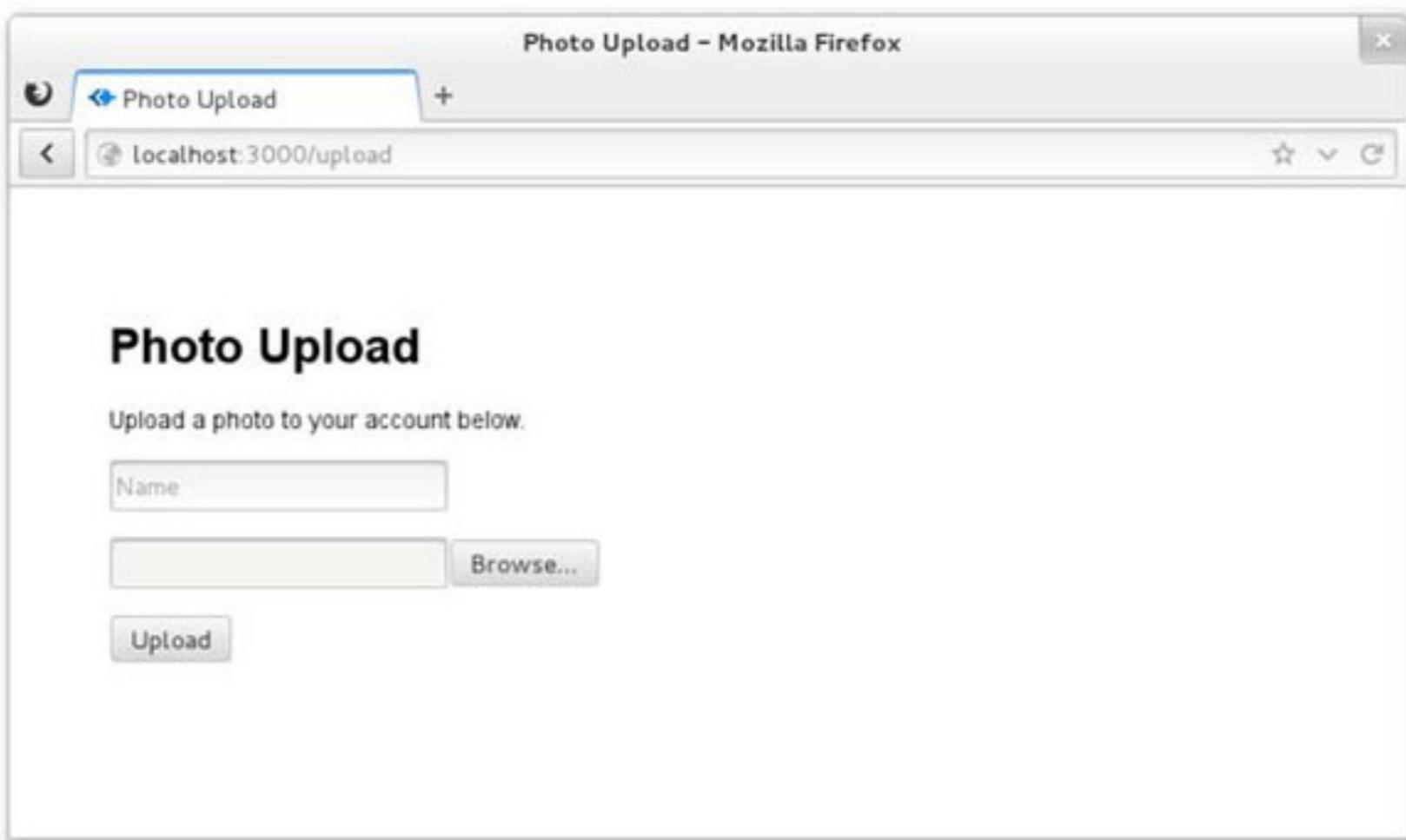
```
app.use('/static', express.static(path.join(__dirname, 'public')))
```



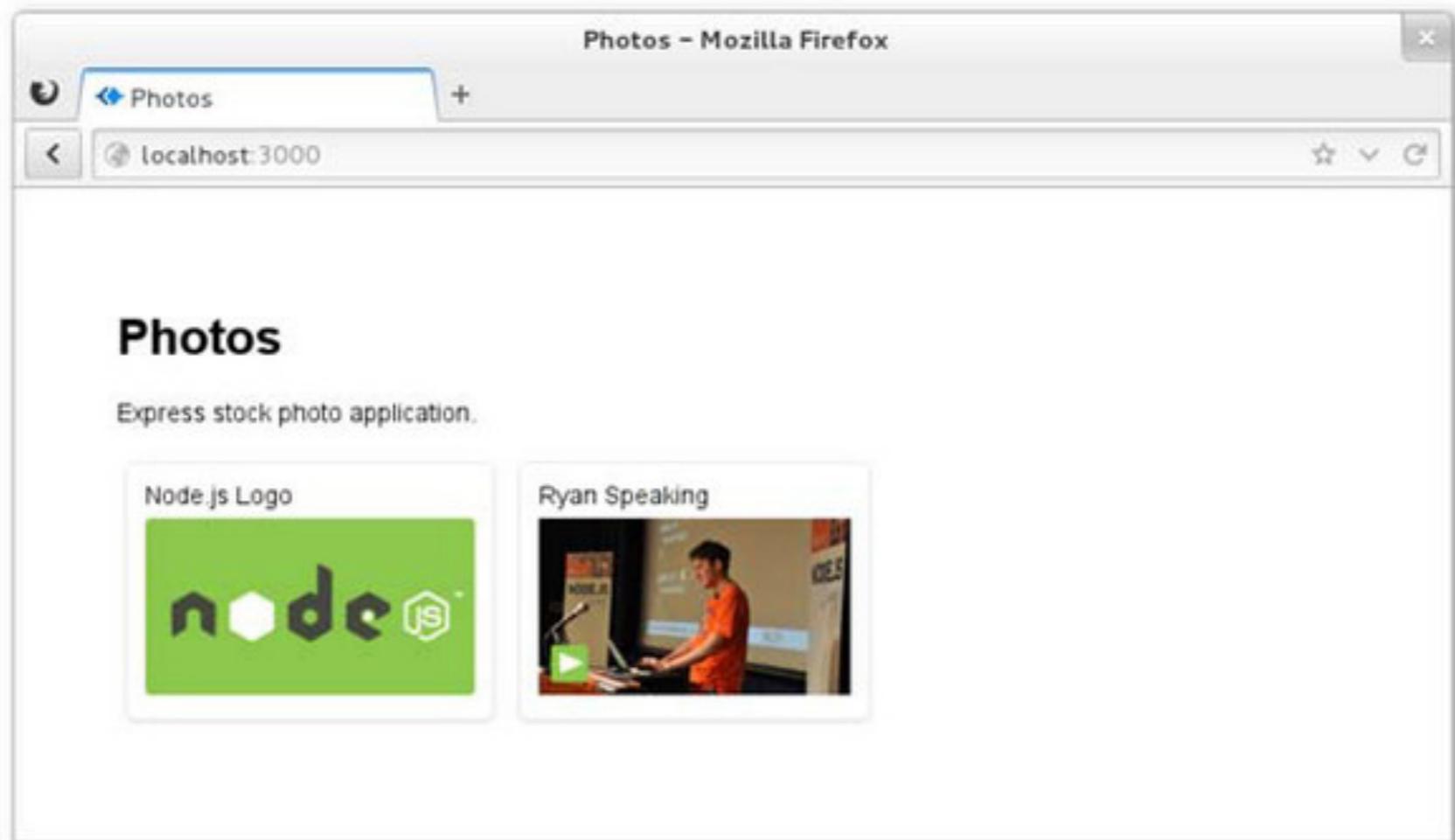
Let's workshop



Let's workshop



Let's workshop



Day 3



Manage data



elasticsearch



kibana



Elasticsearch

The screenshot shows the official Elasticsearch website homepage. At the top, there is a navigation bar with the elastic logo, links for Products, Cloud, Services, Customers, Learn, downloads, contact, a search icon, and a language switcher set to EN. Below the navigation is a large blue banner with the text "Get started with...". It features three circular icons: one for Elasticsearch (magnifying glass over a wavy line), one for Logstash (vertical bars), and one for Kibana (bar chart). Each icon has a corresponding text label and a "Overview Video" or "Introduction Video" button below it.

Get started with...

Elasticsearch

Logstash

Kibana

Overview Video

Introduction Video

Kibana 101 Video

<https://www.elastic.co/>



**SELECT
FROM TABLE
WHERE TEXT **LIKE** '%SHIT%'**



High-Availability

Plug-ins

Lucene

Scalability

Distributed

RESTFul

API

JSON



elasticsearch

open-source

realtime, search
and analytics

documentation

document store

engine

JAVA



Use cases

Full-text search

Data store

Analytics

Alert

Ads



Who USE ?

Path



github

foursquare™



SONY



XING

salesforce desk™

IGN

mozilla



KLOUT

infochimps



SCOUT 24



Fog Creek SOFTWARE

Revinate

picturesafe



Client libraries

- Java
- PHP
- Ruby
- Python
- JavaScript
- NodeJS
- Go
- Scala
- .Net
- Clojure
- Erlang
- R



Let's start with Elasticsearch



Download and Install

<https://www.elastic.co/downloads/elasticsearch>

installation

1



**Download and unzip the latest
Elasticsearch distribution**

2



**Run `bin/elasticsearch` on Unix,
or `bin/elasticsearch.bat` on Windows**

3



Run `curl-X GET http://localhost:9200/`



Start elasticsearch

./bin/elasticsearch

```
[2017-01-27T11:36:29,612] [INFO ] [o.e.n.Node          ] [] initializing ...
[2017-01-27T11:36:29,746] [INFO ] [o.e.e.NodeEnvironment ] [jFPio5Y] using [1] data paths, mounts [[/ (/dev/disk1), net usable_space [32.7gb], net total_space [232.6gb], spins? [unknown], types [hfs]
[2017-01-27T11:36:29,747] [INFO ] [o.e.e.NodeEnvironment ] [jFPio5Y] heap size [1.9gb], compressed ordinary object pointers [true]
[2017-01-27T11:36:29,749] [INFO ] [o.e.n.Node          ] [jFPio5Y] node name [jFPio5Y] derived from node ID [jFPio5Y6RkaiWRhKnq]
iWRhKnq]; set [node.name] to override
[2017-01-27T11:36:29,754] [INFO ] [o.e.n.Node          ] [jFPio5Y] version[5.1.2], pid[36610], build[cBc4c16/2017-01-11
18:39.146Z], OS[Mac OS X/10.12.1/x86_64], JVM[Oracle Corporation/Java HotSpot(TM) 64-Bit Server VM/1.8.0_40/25.45]
[2017-01-27T11:36:31,825] [INFO ] [o.e.p.PluginsService ] [jFPio5Y] loaded module [aggs-matrix-stats]
[2017-01-27T11:36:31,825] [INFO ] [o.e.p.PluginsService ] [jFPio5Y] loaded module [ingest-common]
[2017-01-27T11:36:31,825] [INFO ] [o.e.p.PluginsService ] [jFPio5Y] loaded module [lang-expression]
[2017-01-27T11:36:31,826] [INFO ] [o.e.p.PluginsService ] [jFPio5Y] loaded module [lang-groovy]
[2017-01-27T11:36:31,826] [INFO ] [o.e.p.PluginsService ] [jFPio5Y] loaded module [lang-mustache]
[2017-01-27T11:36:31,826] [INFO ] [o.e.p.PluginsService ] [jFPio5Y] loaded module [lang-painless]
[2017-01-27T11:36:31,826] [INFO ] [o.e.p.PluginsService ] [jFPio5Y] loaded module [percolator]
[2017-01-27T11:36:31,826] [INFO ] [o.e.p.PluginsService ] [jFPio5Y] loaded module [reindex]
[2017-01-27T11:36:31,827] [INFO ] [o.e.p.PluginsService ] [jFPio5Y] loaded module [transport-netty3]
[2017-01-27T11:36:31,827] [INFO ] [o.e.p.PluginsService ] [jFPio5Y] loaded module [transport-netty4]
[2017-01-27T11:36:31,828] [INFO ] [o.e.p.PluginsService ] [jFPio5Y] no plugins loaded
[2017-01-27T11:36:36,231] [INFO ] [o.e.n.Node          ] [jFPio5Y] initialized
[2017-01-27T11:36:36,231] [INFO ] [o.e.n.Node          ] [jFPio5Y] starting ...
[2017-01-27T11:36:36,604] [INFO ] [o.e.t.TransportService] [jFPio5Y] publish_address {127.0.0.1:9300}, bound_addresses {[fe80::1]:9300}, {[::1]:9300}, {127.0.0.1:9300}
[2017-01-27T11:36:39,723] [INFO ] [o.e.c.s.ClusterService] [jFPio5Y] new_master {jFPio5Y}{jFPio5Y6RkaCuEiWRhKnq
iWRhKnq}{[127.0.0.1:9300]}, reason: zen-disco-elected-as-master ([0] nodes joined)
```



Hello elasticseach

```
← → ⌂ ⓘ localhost:9200

{
  name: "jFPio5Y",
  cluster_name: "elasticsearch",
  cluster_uuid: "35D5Rs5AQn2IDjpQdPYJmA",
  - version: {
      number: "5.1.2",
      build_hash: "c8c4c16",
      build_date: "2017-01-11T20:18:39.146Z",
      build_snapshot: false,
      lucene_version: "6.3.0"
    },
  tagline: "You Know, for Search"
}
```



Manage data with



kibana



Download and Install

<https://www.elastic.co/downloads/kibana>



Start kibana

./bin/kibana

```
log [04:44:47.787] [info][status][plugin:kibana@5.1.2] Status changed from uninitialized to
log [04:44:47.896] [info][status][plugin:elasticsearch@5.1.2] Status changed from uninitialized
ing for Elasticsearch
log [04:44:47.959] [info][status][plugin:console@5.1.2] Status changed from uninitialized to
log [04:44:48.345] [info][status][plugin:timelion@5.1.2] Status changed from uninitialized to
log [04:44:48.350] [info][listening] Server running at http://localhost:5601
log [04:44:48.352] [info][status][ui settings] Status changed from uninitialized to yellow -
n is yellow
log [04:44:53.522] [info][status][plugin:elasticsearch@5.1.2] Status changed from yellow to
Kibana index found
log [04:44:54.268] [info][status][plugin:elasticsearch@5.1.2] Status changed from yellow to
ready
log [04:44:54.269] [info][status][ui settings] Status changed from yellow to green - Ready
```



Hello kibana

<https://www.elastic.co/downloads/kibana>

The screenshot shows the Kibana Management interface at the URL `localhost:5601/app/kibana#/management/kibana/index/?_g=()`. The left sidebar has a pink header with the Kibana logo and blue buttons for Discover, Visualize, Dashboard, Timelion, Dev Tools, and Management. The Management button is highlighted. The main area has a grey header with tabs for Index Patterns, Saved Objects, and Advanced Settings. A warning message says: "Warning No default index pattern. You must select or create one to continue." Below it, the title "Configure an index pattern" is displayed in large font. A sub-instruction reads: "In order to use Kibana you must configure at least one index pattern. Index patterns are used to Elasticsearch index to run search and analytics against. They are also used to configure fields." There are two checkboxes: one checked for "Index contains time-based events" and one unchecked for "Use event times to create index names [DEPRECATED]". A text input field contains the value "logstash-*". At the bottom, there is an unchecked checkbox for "Do not expand index pattern when searching (Not recommended)".



Dev Tools



Manage data

Create

Read

Update

Delete



Compare with RDBMS

DATABASE

TABLE

ROW

COLUMN

INDEX

TYPE

DOCUMENT

FIELD



Create data

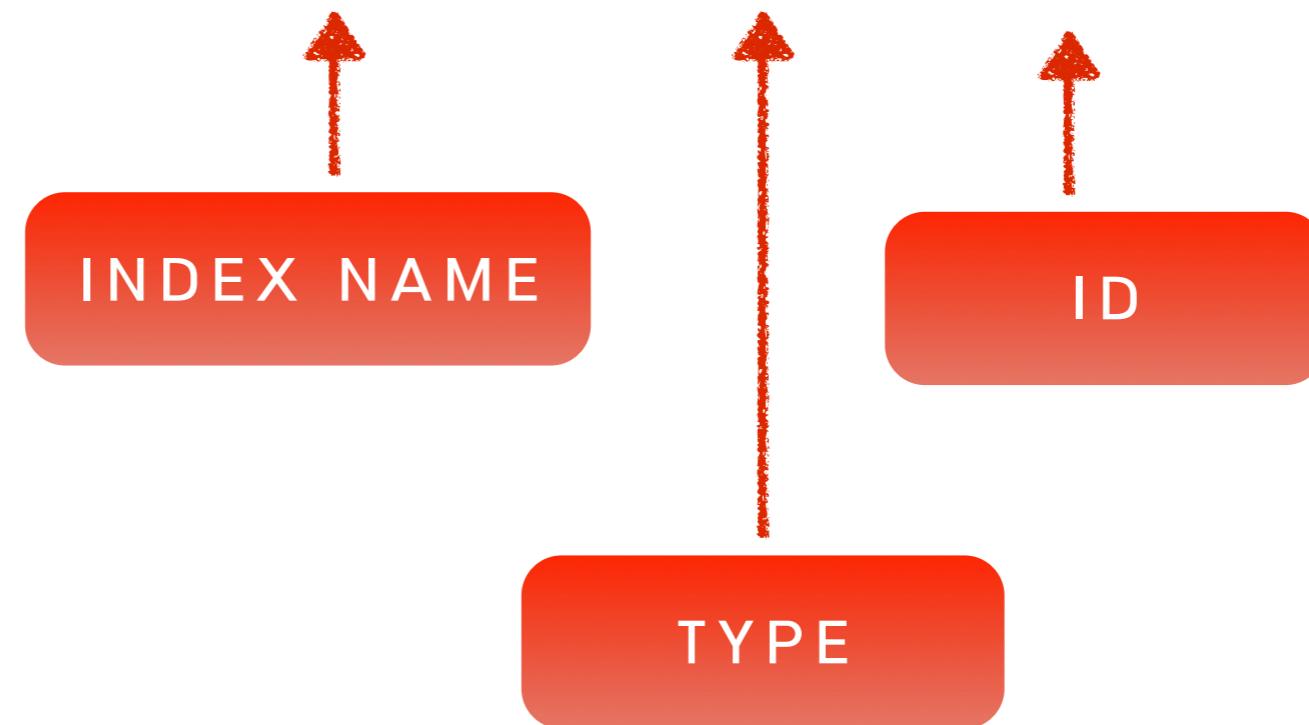
```
1 | PUT /media/photo/1
2 | {
3 |   "name": "photo 01",
4 |   "path": "/path/image01"
5 | }
6 |
7 |
8 |
9 |
10|
11|
12|
```

```
1 | {
2 |   "_index": "media",
3 |   "_type": "photo",
4 |   "_id": "1",
5 |   "_version": 1,
6 |   "result": "created",
7 |   "_shards": {
8 |     "total": 2,
9 |     "successful": 1,
10|     "failed": 0
11|   },
12|   "created": true
| }
```



Create data

PUT /media/photo/1



Read data

GET /media/photo/1

```
1 | GET /media/photo/1
2 |
3 |
4 |
5 |
6 |
7 |
8 |
9 |
10|
11|
```

```
1 | {
2 |   "_index": "media",
3 |   "_type": "photo",
4 |   "_id": "1",
5 |   "_version": 1,
6 |   "found": true,
7 |   "_source": {
8 |     "name": "photo 01",
9 |     "path": "/path/image01"
10|   }
11| }
```



Update data

PUT /media/photo/1

```
1 PUT /media/photo/1
2 {
3     "name": "Update photo 01",
4     "path": "/path/image01"
5 }
6
7
8 GET /media/photo/1
9
10
11
12
13
14
```

```
1 {
2     "_index": "media",
3     "_type": "photo",
4     "_id": "1",
5     "_version": 4,
6     "result": "updated",
7     "_shards": {
8         "total": 2,
9         "successful": 1,
10        "failed": 0
11    },
12    "created": false
13}
```



Update partial data

POST /media/photo/1/_update

```
1 | POST /media/photo/1/_update
2 | {
3 |   "doc": {
4 |     "name": "Update partial photo 01"
5 |   }
6 |
7 | }
8 |
9 |
10 | GET /media/photo/1
11 |
12 |
```

```
1 | {
2 |   "_index": "media",
3 |   "_type": "photo",
4 |   "_id": "1",
5 |   "_version": 9,
6 |   "result": "noop",
7 |   "_shards": {
8 |     "total": 0,
9 |     "successful": 0,
10 |     "failed": 0
11 |   }
12 | }
```



Delete data

DELETE /media/photo/1

```
1 DELETE /media/photo/1
2
3
4 GET /media/photo/1
5
6
7
8
9
10
11
12
13
```

```
1 { "found": true,
2   "_index": "media",
3   "_type": "photo",
4   "_id": "1",
5   "_version": 10,
6   "result": "deleted",
7   "_shards": {
8     "total": 2,
9     "successful": 1,
10    "failed": 0
11  }
12 }
13 }
```



Workshop Query data



Working with Node.js



Install package

\$npm install elastic search



Connect to elasticsearch

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

var client = new elasticsearch.Client( {
  hosts: [ 'http://localhost:9200/' ]
});

client.cluster.health({},function(err,resp,status) {
  console.log("-- Client Health --",resp);
});
```



Connect to elasticsearch

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

var client = new elasticsearch.Client( {
  hosts: [ 'http://localhost:9200/']
});

client.cluster.health({},function(err,resp,status) {
  console.log("-- Client Health --",resp);
});
```



Cluster health

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

var client = new elasticsearch.Client( {
  hosts: [ 'http://localhost:9200/' ]
});

client.cluster.health({}, function(err,resp,status) {
  console.log("-- Client Health --",resp);
});
```



Cluster health

```
-- Client Health -- { cluster_name: 'elasticsearch',
status: 'yellow',
timed_out: false,
number_of_nodes: 1,
number_of_data_nodes: 1,
active_primary_shards: 6,
active_shards: 6,
relocating_shards: 0,
initializing_shards: 0,
unassigned_shards: 6,
delayed_unassigned_shards: 0,
number_of_pending_tasks: 0,
number_of_in_flight_fetch: 0,
task_max_waiting_in_queue_millis: 0,
active_shards_percent_as_number: 50 }
```



Create index

```
client.indices.create({ index: 'media',  
  function(err,resp,status) {  
    if(err) {  
      console.log(err);  
    }else {  
      console.log("create",resp);  
    }  
  } );
```



Delete index

```
client.indices.delete({ index: 'media',
  function(err,resp,status) {
    if(err) {
      console.log(err);
    }else {
      console.log("create",resp);
    }
  });
});
```



Create data

```
client.index({  
    index: 'media',  
    id: '1',  
    type: 'photo',  
    body: {  
        "name": "image 01",  
        "path": "image_01.jpg"  
    }  
},function(err,resp,status) {  
    console.log(resp);  
});
```



Create data

```
client.index({  
    index: 'media',  
    id: '1',  
    type: 'photo',  
    body: {  
        "name": "image 01",  
        "path": "image_01.jpg"  
    }  
},function(err,resp,status) {  
    console.log(resp);  
});
```



Count data

```
client.count({  
    index: 'media',  
    type: 'photo'  
},  
function(err,resp,status) {  
    console.log("Photo",resp);  
});
```



Delete data

```
client.delete({  
    index: 'media',  
    id: '1',  
    type: 'photo'  
},function(err,resp,status) {  
    console.log(resp);  
});
```



Search data

```
client.search({  
    index: 'media',  
    type: 'photo',  
    body: {  
        query: {  
            match: { "_all": "image" }  
        },  
    }  
})
```



Search data

```
client.search({  
    index: 'media',  
    type: 'photo',  
    body: {  
        query: {  
            match: { "_all": "image" }  
        },  
    }  
})
```



Same result

```
1 | GET /media/photo/_search
2 | {
3 |   "query": {
4 |     "match": {
5 |       "_all": "image"
6 |     }
7 |   }
8 | }
9 |
10|
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25| }
```

```
1 | {
2 |   "took": 43,
3 |   "timed_out": false,
4 |   "_shards": {
5 |     "total": 5,
6 |     "successful": 5,
7 |     "failed": 0
8 |   },
9 |   "hits": {
10|     "total": 1,
11|     "max_score": 0.2876821,
12|     "hits": [
13|       {
14|         "_index": "media",
15|         "_type": "photo",
16|         "_id": "1",
17|         "_score": 0.2876821,
18|         "_source": {
19|           "name": "image 01",
20|           "path": "image_01.jpg"
21|         }
22|       }
23|     ]
24|   }
25| }
```



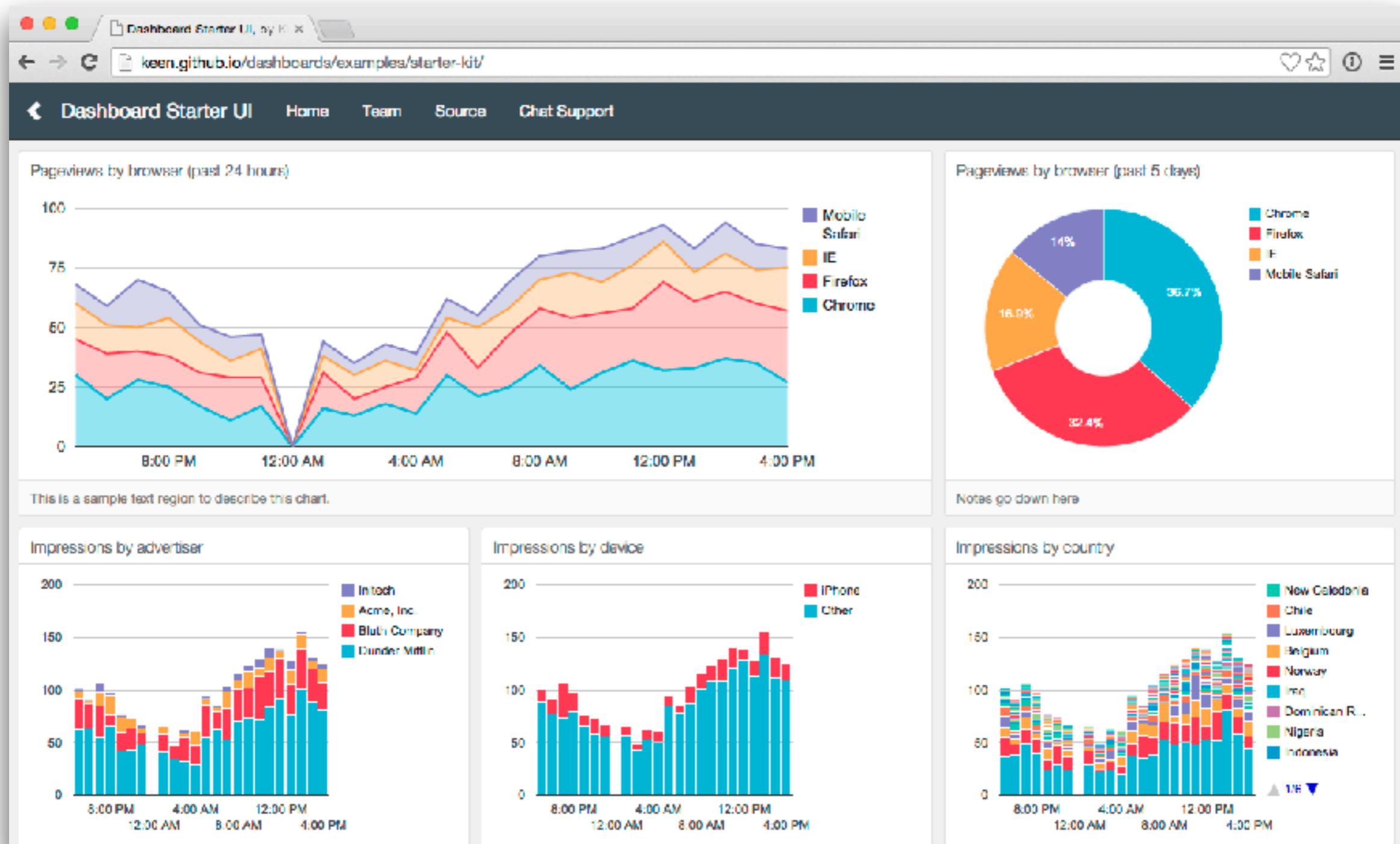
Workshop keep data in Elasticsearch



Day 4 & 5



Realtime analytic



Structure



express



socket.io



elasticsearch

Workshop

