
node
JS
node essentials.js

About me

Bedis ElAcheche (@elacheche_bedis)

- Lazy web & mobile developer
- Official Ubuntu Member
- FOSS lover
- Javascript fanatic
- I'm writing bugs into apps since 2008

What to expect ahead..

- Data types
- Variables
- Conditional statements
- Loops
- Functions
- Comments

What to expect ahead..

What to expect ahead (for real)...

- Introduction
- Event loops
- Blocking and nonblocking I/O
- Modules
- NPM
- Demos
- Random memes

What is Node.js?

- Open source, cross platform development platform.

- A command line tool.

- Created by Ryan Dahl in 2009.

- Uses v8 JavaScript engine.

- Uses an event-driven, non-blocking I/O model.

Why Node.js?

- Free

- Open source

- Very lightweight and fast because it's mostly C/C++ code.

- Can handle thousands of concurrent connections with minimal overhead (CPU/Memory).

- There are lots of modules available for free.

When to use Node.js?

- HTTP servers.
- Chat applications.
- Online games.
- Collaboration tools.
- Desktop applications.
- If you are great at writing javascript code.

When not to use Node.js?

- Heavy and CPU intensive calculations on server side.

- Node.js is single thread

- You have to write logic by your own to utilize multi core processor and make it multi threaded.

Who uses Node.js?

- LinkedIn
- Microsoft
- Netflix
- PayPal
- SAP
- Walmart
- Uber
- ...

Event loops

- The core of event-driven programming.

- Almost all the UI programs use event loops to track the user event.

- Register callbacks for events.

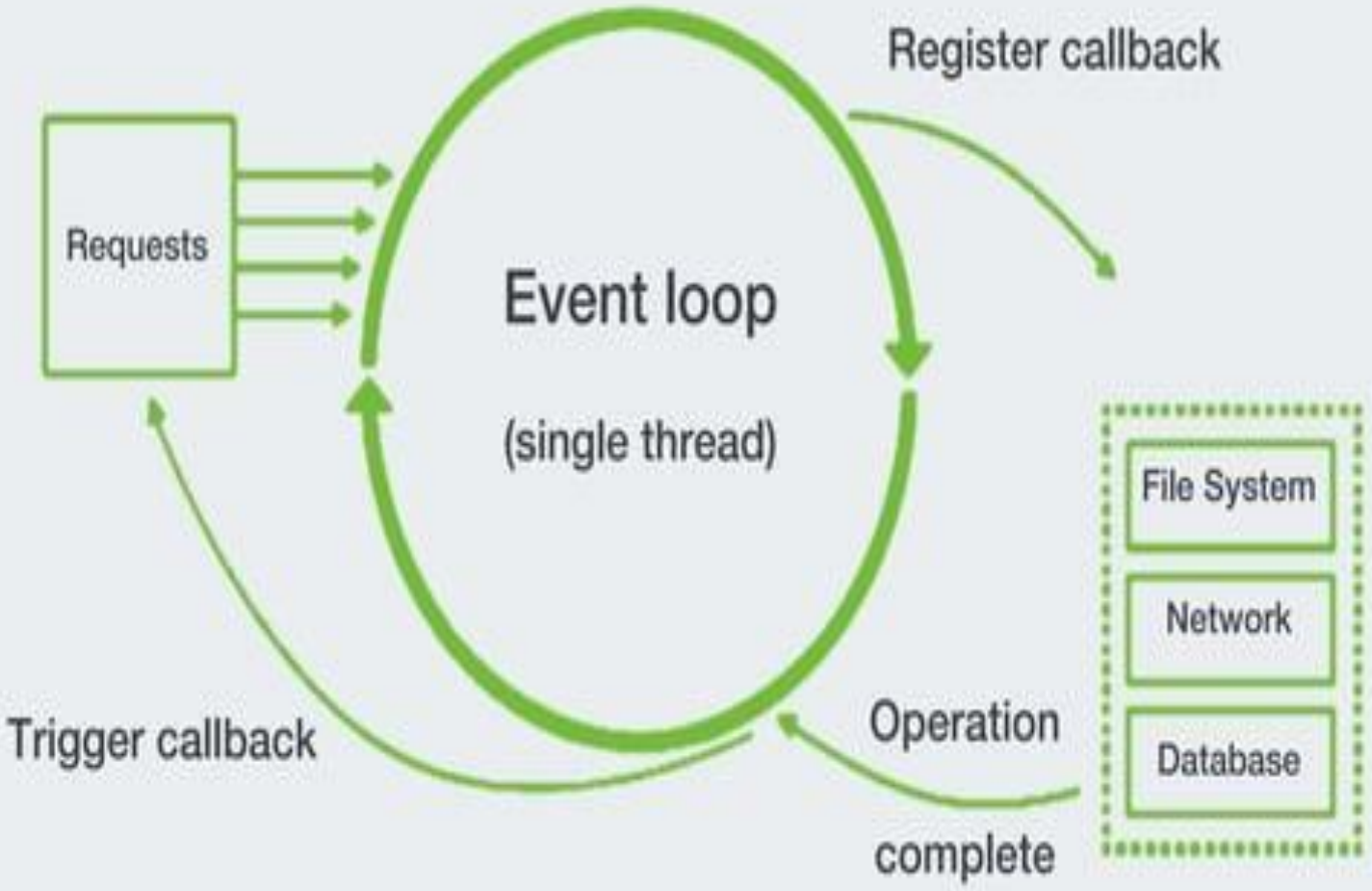
- Your callback is eventually fired.

Event loops

```
$('a').click(function() {
    console.log('clicked!');
});


$.get('slides.php', {
    from: 1,
    to: 5
}, function(data) {
    console.log('New slides!');
});
```

Event loops life cycle



Event loops life cycle

- Initialize empty event loop.
- Execute non-I/O code.
- Add every I/O call to the event loop.
- End of source code reached.
- Event loop starts iterating over a list of events and callbacks.
- Perform I/O using non-blocking kernel facilities.
- Event loop goes to sleep.
- Kernel notifies the event loop.
- Event loop executes and removes a callback.
- Program exits when event loop is empty.

Blocking and nonblocking I/O

```
// Blocking I/O

var attendees = db.query('SELECT * FROM attendees'); // Wait for result!

attendees.each(function(attendee) {

    var attendeeName = attendee.getName(); // Wait for result!

    sayHello(attendeeName); // Wait

});

startWorkshop(); // Execution is blocked!
```

Blocking and nonblocking I/O

```
// Nonblocking I/O

db.query('SELECT * FROM attendees', function(attendees) {

    attendees.each(function(attendee) {

        attendee.getName(function(attendeeName) {

            sayHello(attendeeName);

        });

    });

});

startWorkshop(); // Executes without any delay!
```

Blocking and nonblocking I/O

Events in Node.js

- An action detected by the program that may be handled by the program.

- Determine which function will be called next.

- Many objects in node emit events.

- You can create objects that emit events too.

Events in Node.js

```
// Custom event emitters
var EventEmitter = require('events');
var logger = new EventEmitter();
logger.on('new-attendee', function(message) {
    console.log('Welcome %s! Please have a seat.', message);
});
logger.on('attendee-left', function(message) {
    console.log('Goodbye %s! See you next time.', message);
});
logger.emit('new-attendee', 'John Doe');
// Welcome John Doe! Please have a seat.
logger.emit('new-attendee', 'Jane Doe');
// Welcome Jane Doe! Please have a seat.
logger.emit('attendee-left', 'John Smith');
// Goodbye John Smith! See you next time.
```

Events in Node.js

Node.js modules

- External libraries.

- One kind of package which can be published to NPM.

- Node.js heavily relies on modules.

- Creating a module is easy, just put your javascript code in a separate js file and include it in your code by using keyword require.

Node.js modules

Node.js modules

```
// greetings.js
var hello = function() {
    console.log('Hello %s!', name || '');
}
var bye = function() {
    console.log('Bye %s!', name || '');
}
module.exports = {
    hello: hello,
    bye: bye
};
```

```
// hola.js
module.exports = function(name) {
    console.log('Hola %s!', name || '');
};
```

```
// app.js
var greetings = require('./greetings');
var hola = require('./hola');

greetings.hello();
// Hello!
hola('John doe');
// Hola Jane doe!

// If we only need to call once

require('./greetings').bye('Jane');
// Goodbye Jane!
```

Node.js modules

Node.js modules

```
// look in same directory
var awesomeModule = require('./awesome-module')
// look in parent directory
var awesomeModule = require('../awesome-module')
// look in specific directory
var awesomeModule = require('/home/d4rk-5c0rp/lab/awesome-module');

// app.js located under /home/d4rk-5c0rp/lab/nodeWorkshop
var awesomeModule = require('awesome-module');
// search in node_modules directories
// /home/d4rk-5c0rp/lab/nodeWorkshop/node_modules
// /home/d4rk-5c0rp/lab/node_modules
// /home/d4rk-5c0rp/node_modules
// /home/node_modules
// /node_modules
```

NPM

- Package manager for node.
- Comes bundled with Node.js installation.
- Command line client that interacts with a remote registry.
- Allows users to consume and distribute JavaScript modules.
- Manage packages that are local dependencies of a particular project.
- Manage as well globally-installed JavaScript tools.

NPM registry

- Kind of an App store for developers.

- There are a whole lot of modules and tools available to make the process of building programs quicker and simpler.

- Look up the functionality you want, and hopefully found a module that does it for you.

Installing a NPM module

Install modules into local node_modules directory

```
npm install
npm install <tarball file>
npm install <tarball url>
npm install <name>
npm install <name>@<tag>
npm install <name>@<version>
npm install <name> [--save|--save-dev]
```

If a module requires another module to operate, NPM will download automatically!

Installing a NPM module

Install modules with executables globally

```
npm install gulp -g
```

Global NPM modules CAN'T be required

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

// Error: Cannot find module 'gulp'
```

Express.js

- Minimal and flexible Node.js framework.

- Free and open-source.

- Designed for building web applications and APIs.

Express.js

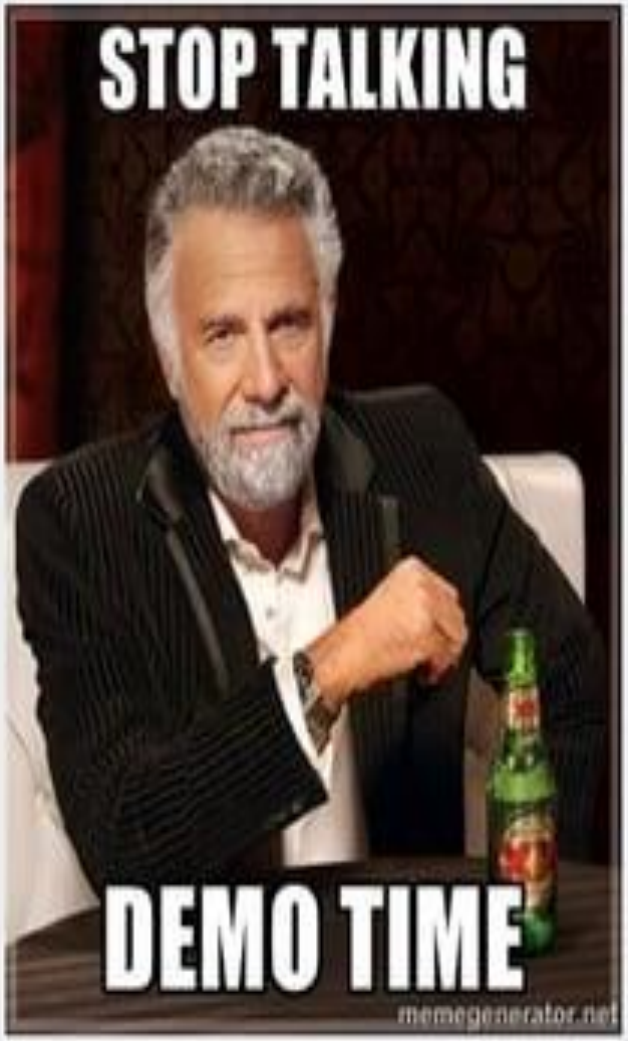
Socket.IO

- Realtime application framework.

- Enables bidirectional event-based communication.

- It has two parts: a client-side library that runs in the browser, and a server-side library for Node.js.

Socket.IO




```
slides.emit('Thank you!');
```