



Modern Web Development 2018

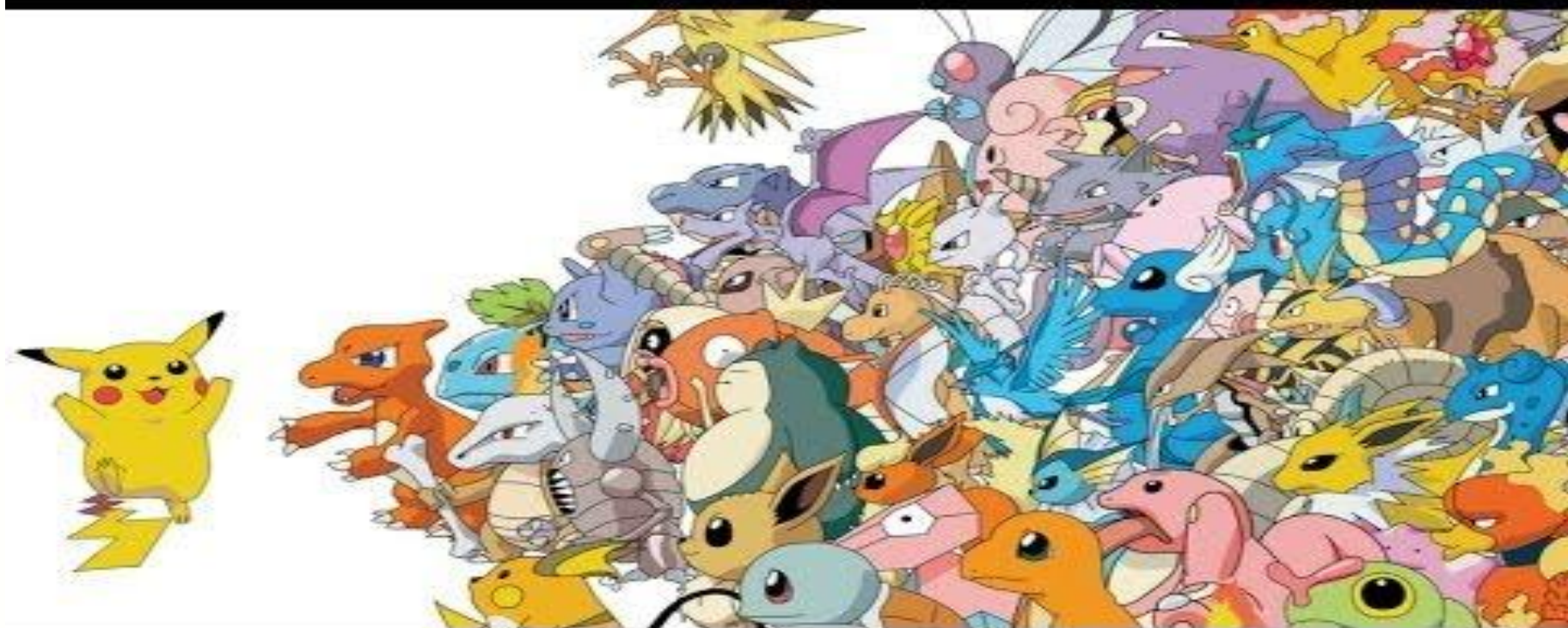
Course summary

Backend

- Node and HTTP APIs with Express
- MongoDB, NoSQL and mongoose

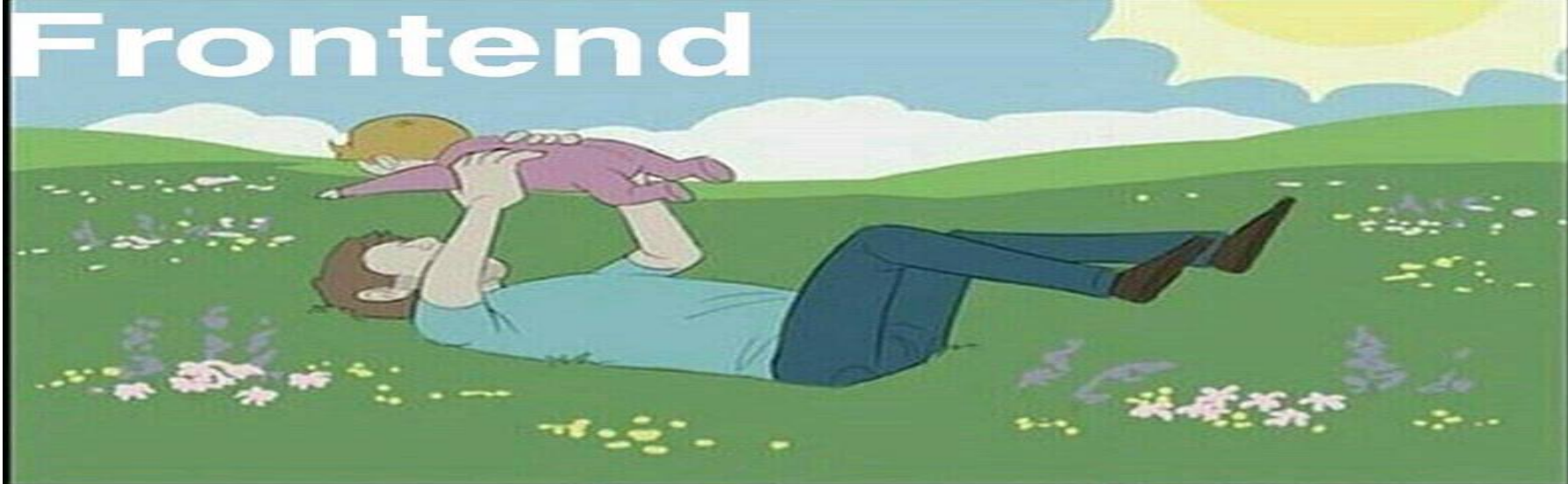
Frontend

- HTML, CSS, responsive websites (using Bootstrap)
- SPA's with Vue.js



frontend + backend = fullstack

Frontend



Backend



Node and HTTP APIs with Express

Summary

- What is a REST API?
- What is HTTP?
- What is node.js?
- How to use callbacks/clojures

REST API

An API is a set of functions that allows applications to access features of your service (os, web application etc.).

REST is a (robust) architectural style of building API's. Usually REST API's are built on top of HTTP (REST and HTTP were developed in parallel by Roy Fielding).

All REST API's use as a basic unit of organization resources (mostly json and xml documents).

The REST API enables manipulation of documents.

HTTP

Protocol over TCP that is the basis of data communication on the internet.

It uses URLs for identification of resources.

Ex: <https://www.advancedinstaller.com/index.html>

It exposes HTTP verbs for actions.

Most common verbs: **GET, POST, DELETE**

GET

With the HTTP GET request the server will respond back with some resource: an html file, a json etc.

A GET request can have query params:

<https://www.example.com/resource?param1=someValue>

The server can respond accordingly.

POST

Used for creating resources or sending data (pictures, comments, username and passwords etc.)

The body length has a bigger limit (it depends on the server).

If you use HTTPS the request is encrypted.

PUT

Used for updating a resource. If the resource does not exist then the server must create it.

The PUT verb is idempotent (calling multiple times does not change the results).

PUT: <https://www.example.com/article/comment/1234567>

DELETE

Used for deleting a resource.

DELETE: <https://www.example.com/article/comment?author=John>

DELETE: <https://www.example.com/article/comment/1234567>

Query params

Query params are used for resource identification.

<https://www.example.com/news?country=Romania&page=2>

The query params string is limited(size depends on the server) and is visible to everyone that can see the request (ISP, proxies).

Body Contents

Typically POST and PUT requests have message body with the information that is needed to create/update the resource.

The format is `application/json` (json) or `application/www-form-urlencoded` (similar to query strings, just hidden in the message body).

There is no semantic reason for a DELETE and GET request to have a message body. If the request contains a message body it should be ignored by the server.

HTTP headers

HTTP headers allow the client and server to pass additional information with the request.

Format of a header is *name:value*.

Content-Type: application/json

Authorization: Basic YWxhZGRpbjpvcGVuc2VzYW1l



Node.js

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.

Source: nodejs.org

Read: Efficient servers written in JS

Node is *not* ready for production!

Used in production by:

- Netflix
- LinkedIn
- NASA
- Caphyon and others

Source:

<http://www.tothenew.com/blog/how-are-10-global-companies-using-node-js-in-production/>

Java and JavaScript

Java is to JavaScript as Ham is to HamSter



JS AA Java™

JavaScript key features

- JavaScript is a lightweight, dynamic, interpreted, prototype based programming language
- It is multi-paradigm (supports): object-oriented, imperative and declarative programming styles
- A language that runs on every browser(desktop and mobile) and on server (with Node).

Source: <https://developer.mozilla.org/bm/docs/Web/JavaScript>

JSON

JSON is a syntax for storing and exchanging data.

It is self-describing, hierarchical, can be parsed by JS very easy.

As opposed to XML is shorter, quicker to read and write and can use arrays.

```
{  
  "employee":{ "name":"John", "age":30, "city":"New York" }  
  "friends":[ "John", "Anna", "Peter" ]  
}
```

DEMO 1

\$ node

```
>console.log("Hello World Node!")
```

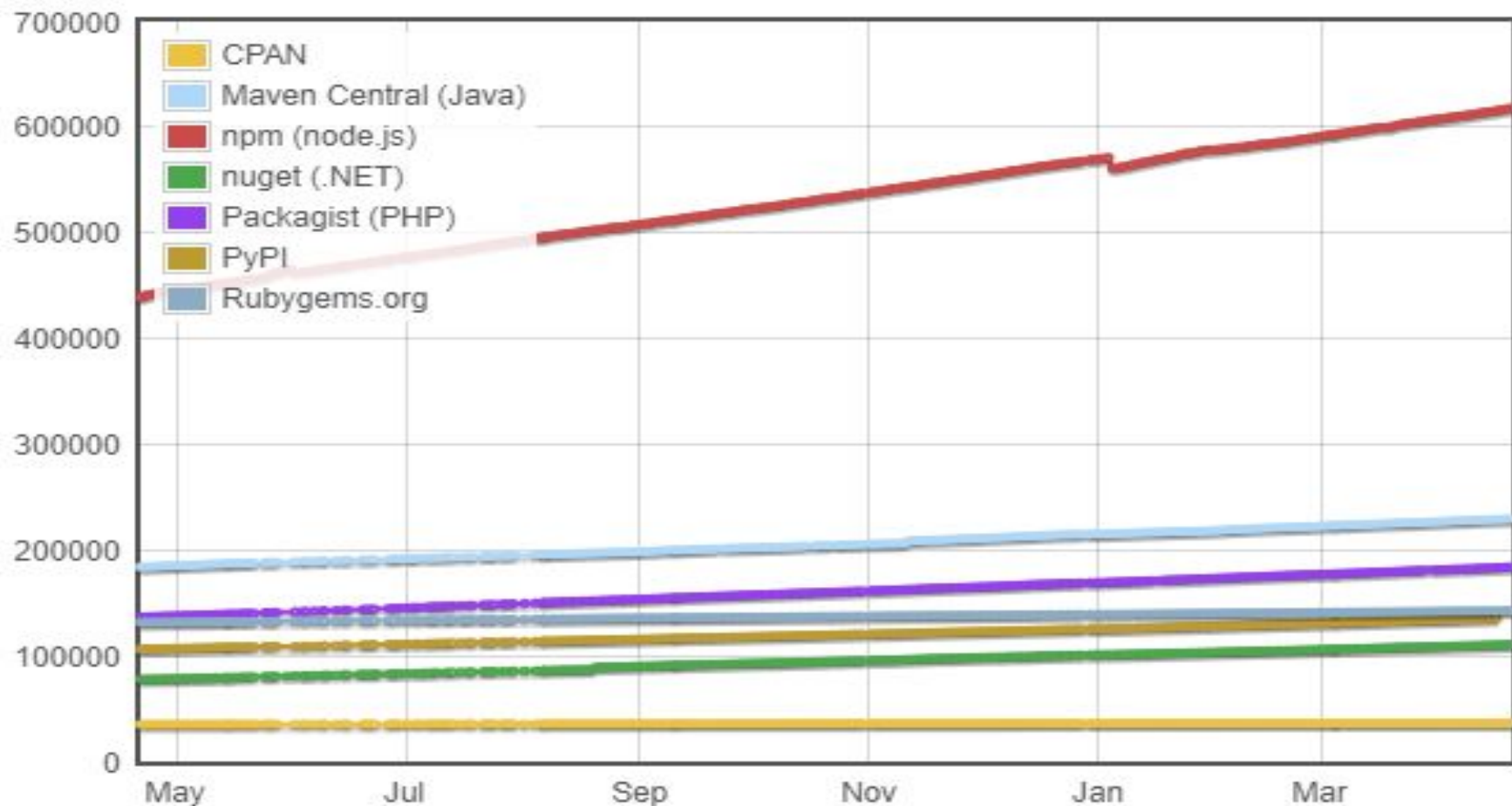

NPM

Npm is a tool to manage node packages.

Mostly written in JS (or compiled to JS from TS, Coffee Script or others).

More than 600k of libraries that can be reused so that you can focus on your product/idea.

Module Counts



Demo 2

Let's init a project in your work folder:

```
$ npm init
```

Let's install a package:

```
$ npm install --save express
```

Callbacks

A callback is a function that is to be executed after another function has finished

```
request('https://www.advancedinstaller.com/products.json', function (err, res,  
body) {
```

```
    // do something with the json
```

```
});
```

Additional resources

- HTTP: <https://developer.mozilla.org/en-US/docs/Web/HTTP>
- Javascript Guide:
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>
- Node documentation: <https://nodejs.org/en/docs/>
- Node tutorials: <https://www.tutorialspoint.com/nodejs/index.htm>
- REST: <http://www.restapitutorial.com>
- Roy Fielding doctoral dissertation:
<https://www.ics.uci.edu/~fielding/pubs/dissertation/>