

POKE**API**

**MAIN GOAL**

Consumir la PokeApi y obtener: un Pokemon por su nombre o por su ID, un Pokemon aleatorio y una lista de Pokemones desde un id hasta la cantidad que yo decida inclusivo (por ejemplo, quiero desde el Pokemon 3 y quiero ver 10, entonces debería de poder ver del Pokemon 3 al 13).

¿Qué es lo que voy a necesitar de este proyecto?

+ Cada petición debo de poder ver: nombre del Pokemon, ID, sprite de frente y detrás (shiny o normal, macho o hembra) y un movimiento.

+ Que aprendas a usar Node Js y Express en el backend para hacer el consumo de la Api

+ Aprendas a usar las promesas (new Promise con .then/.catch, NO try/catch ni fetch aunque si se te dificulta, puedes usarlas)

+ Aprendas a usar Postman para probar tu backend

Puntos extra si usas Git y GitHub y haces tus ramas para cada función. Lo ideal es un frontend para visualizar las cosas de mejor manera, pero me conformo con un muy buen backend. Importa más el backend que el frontend, pero el frontend da puntos extra.

**SETTING UP THE ENVIROMENT & INSTALLATION**

* At this point we assume that we’ve already installed Git, npm (with NodeJS) and, at least, the new Windows Terminal, otherwise the command prompt for Windows
* Create a new project with npm init –-y, install Bootstrap, jquery and popper.js with the npm install bootstrap jquery popper.js also install the SASS package as global, with npm install sass -g
* Optional: In order to watch changes in real time in browser, install the gulp-cli as a global package, also the gulp-sass and browser sync packages as dev dependencies. Install them with npm install gulp-cli -g and npm install gulp gulp-sass browser-sync --save-dev. They’ll be useful for live coding.
* Create a file called “gulpfile.js”, paste the following script and now by typing “gulp watch” we’ll be able to see the changes in real time (front-end)

const gulp = require('gulp');

const sass = require('gulp-sass');

const browserSync = require('browser-sync').create();

//compile scss into css

function style() {

return gulp.src('src/scss/\*\*/\*.scss')

.pipe(sass().on('error',sass.logError))

.pipe(gulp.dest('src/css'))

.pipe(browserSync.stream());

}

function watch() {

browserSync.init({

port: 5000,

server: {

baseDir: ".",

index: "index.html"

}

});

gulp.watch('src/scss/\*\*/\*.scss', style)

gulp.watch('./\*.html').on('change',browserSync.reload);

gulp.watch('src/js/\*\*/\*.js').on('change', browserSync.reload);

}

exports.style = style;

exports.watch = watch;

**DEVELOPMENT**

* First, go to the POKE**API** page, available at: <https://github.com/PokeAPI/pokedex-promise-v2>
* Get involved with it, read a little bit & learn whatever you can about Pokemon
* We’re going to use NodeJS to call the API and receive a Pokemon by ID, name or by
* Also browserify is required. Browserify allow us to run code blah blah
* Install it running the following command: npm install browserify
* Now you can “compile” the requires in a single file with the following command:

browserify input.js > output.js