

Ejercicio 1

En el siguiente [enlace](#) se explica paso a paso cómo acceder a una de las API de OpenWeather para obtener los datos meteorológicos de una ciudad. Sigue el tutorial. No hagas copy-paste y estudia el código mientras tecleas.

Una vez seguido el tutorial obtenemos el siguiente código, en mi caso he realizado algunos cambios para mejorar el resultado.

```
const request = require('request');
const argv = require('yargs').argv;

let apiKey = 'd260f0f685af9aa06d210df1bcd6322f';
let city = argv.c || 'portland';

let url = `http://api.openweathermap.org/data/2.5/weather?q=${city}&units=imperial&appid=${apiKey}`;
console.log(city);

request(url, function(err, response, body){
  if(err){
    console.log('error:', error);
  }else{
    let weather = JSON.parse(body);

    let temp = Math.floor(weather.main.temp - 32);
    let resultado = Math.floor(temp * 5)/9;
    let temperatura = parseInt(resultado);

    let message = `En ${city} hay ${weather.main.temp} en farenheit, que en grados son ${temperatura} grados`;
    console.log(message);
  }
});
```

Probamos con la temperatura que hay en Madrid

```
root@debian-ivan:/home/ivan/node/node-weather# node index.js -c Madrid
Madrid
En Madrid hay 57.2 en farenheit, que en grados son 13 grados
root@debian-ivan:/home/ivan/node/node-weather#
```

Comprobamos que es cierto.



Ejercicio 2

Busca una API pública. En el siguiente [listado](#) tienes varias, aunque puedes utilizar cualquier otra. Ten en cuenta que algunas de estas API necesitarán que te registres o que esperes una aprobación para poder usar el API Key. Otras, sin embargo, no tienen ni siquiera API Key.

Una vez escogida una API consulta su documentación y crea un programa en Node que te permita visualizar los datos que proporciona esa API.

La API que yo he seleccionado ha sido la de twitter, con ella he creado un Script que obtiene todos los Tweets según un determinado #hashtag. A partir de un objeto que nos devuelve llamado "statuses" he seleccionado algunas de sus propiedades. A partir de esas propiedades he obtenido valores como el texto del tweet, el autor, el dispositivo desde el que se escribió, la fecha, etc..

"Para complicarlo aun mas" he creado un servidor http en el que muestro los tweets de una manera mas "bonita".

Primero voy a explicar un poco el código.

Primera Parte

```
const argv = require('yargs').argv;
var http = require('http');
var Twitter = require('twitter');
const fs = require('fs');
var client = new Twitter({
  consumer_key: '...',
  consumer_secret: '...',
  access_token_key: '...',
  access_token_secret: '...'
});
let hashtag = argv.c;
//console.log(hashtag);
// EL OBJETO (statuses) devuelve .created_at .text ,etc..
```

Creamos una instancia

He añadido el modulo **yargs** de la práctica anterior para recibir por consola el parámetro. También los módulos **http**, **twitter** y **fs**

Agregamos las Keys que nos proporcionan

Almacenamos en una variable el parámetro

Buscamos
el hashtag

Creamos
nuestro
servidor web

Creamos la
estructura del
documento

Leemos un archivo css
donde tengo los estilos

Con un **foreach**
recorremos el
objeto **statuses**

Llamamos algunas
de sus
propiedades

```
// Le damos un nombre a la variable que vamos a usar para el hashtag
client.get('search/tweets', {q: `#${hashtag}`}, function(error, tweets, response) {
  http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/html'} , {'Content-Type': 'text/css'})
    var css = fs.readFileSync('styles.css', {encoding: 'utf8'});

    res.write('<html>');
    res.write('<head>');
    res.write('<meta charset="utf-8">');
    res.write('<style type="text/css">');
    res.write(css);
    res.write('</style>');
    res.write('</head>');
    res.write('<body>');
    res.write('<h2>Últimos Tweets del hashtag #${hashtag}</h2>')
    tweets.statuses.forEach(function(tweet){

      res.write("&<blockquote class='twitter-tweet'>");

      res.write(`<pre>Hora:${tweet.created_at} Escrito con ${tweet.source} </pre>`);
      res.write(`Por el Usuario: ${tweet.user.name}</pre>`);
      res.write("<p>");
      res.write(`Escribio: ${tweet.text}</pre>`);
      res.write("</p>");

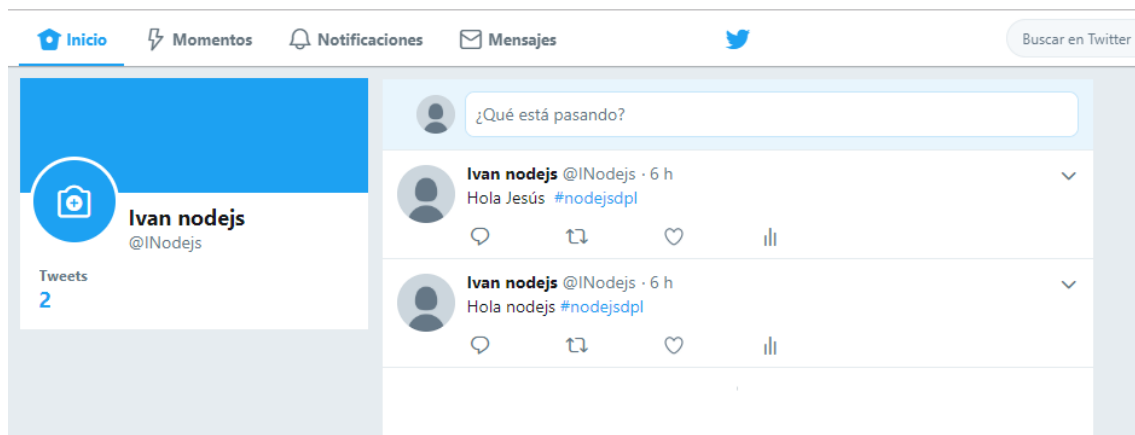
      res.write("</blockquote>");
    });
  });
});
```

```
res.write('</body>');
res.write('</html>');
res.end();

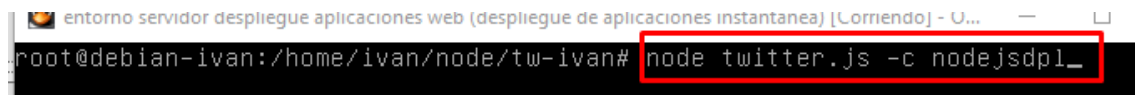
}).listen(8080);

}); //final client
```

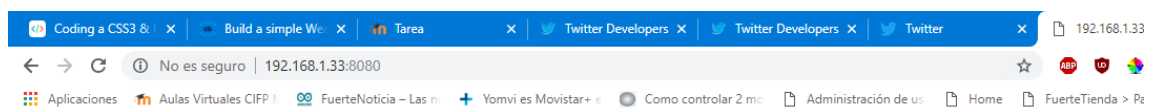
Vamos a ver un ejemplo, en este caso yo he creado dos Tweets con el hashtag #nodejsdpl



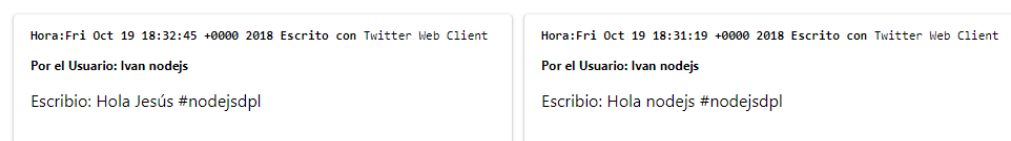
Introducimos en consola el hashtag



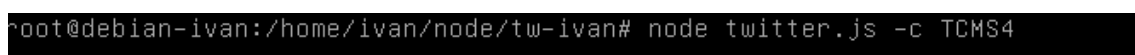
Entramos en el navegador y vemos el resultado correcto



Últimos Tweets del hashtag #nodejsdpl



Por último, probaremos con el hashtag actualmente mas comentado.



Últimos Tweets del hashtag #TCMS4

