

Guión API IP-Stack

Para obtener los datos de una IP pública se puede utilizar esta URL:

`http://api.ipstack.com/`

Para poder hacerles consultas mediante Ajax hay que obtener una clave de acceso. En nuestro caso es: 43aae9d7454a6f5a9eaf7d0b180fc6d2

Teniendo la clave de acceso y la IP de la que queremos obtener los datos se hace la consulta por Ajax mediante esta URL:

```
var access_key = '43aae9d7454a6f5a9eaf7d0b180fc6d2';

var ip="13.87.56.3"

var url = 'http://api.ipstack.com/' + ip + '?access_key=' + access_key;
```

Los datos que devuelve están en formato JSON y tiene las siguientes propiedades:

```
{
  "ip": "2.139.164.235",
  "type": "ipv4",
  "continent_code": "EU",
  "continent_name": "Europe",
  "country_code": "ES",
  "country_name": "Spain",
  "region_code": "MD",
  "region_name": "Madrid",
  "city": "Madrid",
  "zip": "28001",
  "latitude": 40.42618179321289,
  "longitude": -3.6851398944854736
}
```

Guión Geolocalización

Es necesario utilizar una API, como puede ser la Google Maps:

```
<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBDaeWicvigtp9xPv919E-
RN0xfvC-Hqik&libraries=geometry"></script>
```

Esta API permite crear mapas de Google, marcar unas coordenadas en un mapa, calcular distancia entre coordenadas.

Vamos a utilizar los siguientes métodos:

Método para obtener las coordenadas del ordenador que utiliza el usuario

```
navigator.geolocation.getCurrentPosition(f) ;
```

Este método lo tienen integrado todos los navegadores, y devuelve, a través de la función manejadora f, las coordenadas del ordenador que está utilizando el usuario. Estas coordenadas se utilizan con la siguiente sintaxis:

```
function f(pos){  
  
    //pos.coords.latitude -> guarda la latitud  
  
    //pos.coords.longitude -> guarda la coordenada longitud  
  
}
```

Objeto para crear un mapa

```
<div id="map"></div>  
  
var map = new google.maps.Map(document.getElementById("map"), {  
  
    zoom: 3, //para que salgan todos los continentes, 10 una región  
  
    center: coord  
  
});
```

Al objeto google.maps.Map se le pasan 2 argumentos (hay otros opcionales):

- **elem_HTML**: es el elemento HTML en el que se va a crear el mapa. Por ejemplo, si tenemos el elemento: `<div id="map"></div>`, crearíamos el mapa así:
google.maps.Map(document.getElementById("map"), {...})
- El segundo argumento es un objeto literal de 2 propiedades:
 - **zoom**: para indicar con qué zoom queremos que nos muestre el mapa.
 - **center**: para indicar las coordenadas que van a ser el centro del mapa. Estas coordenadas deben ser expresadas mediante la siguiente sintaxis:

```
{  
    lat: valor,  
    lng: valor  
};
```

Objeto para marcar unas coordenadas en un mapa ya existente

```
var marker = new google.maps.Marker({  
    position: coord, // position a marcar en el mapa  
    map: map        // map es el mapa en el que vamos a hacer la marca  
});
```

position tiene esta sintaxis:

```
{  
    lat: valor,  
    lng: valor  
};
```

Método para calcular la distancia entre 2 coordenadas

```
var dist=google.maps.geometry.spherical.computeDistanceBetween(coord1, coord2)
```

coord1 y coord2 tienen esta sintaxis:

```
{
    lat: valor,
    lng: valor
};
```