

Consumo de API de Google Maps usando javascript

A+ extra de Acme-Associations

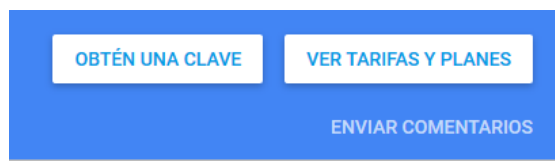
ACMESOME42

5 de junio de 2017

Autor: Alberto Bujalance Muñoz

API Key

En primer lugar tendremos que pedir una clave para usar la API de Google maps en [este enlace](#).



Contenido

Autenticación para la
API estándar: claves de
API

La clave estándar es gratuita y permite hasta 25000 cargas de mapa y 2500 peticiones de geocoding (el servicio que usaremos) diarias. En nuestro sistema esto es más que suficiente.

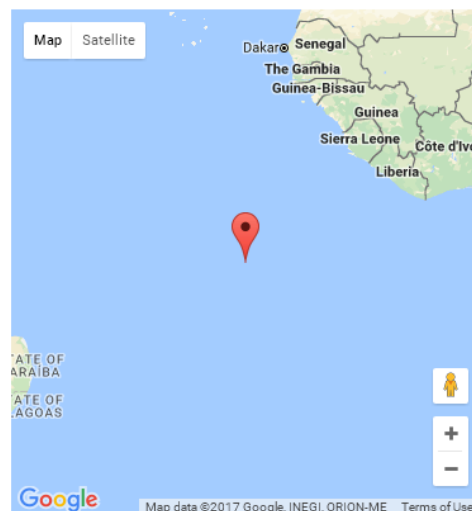
Mostrar un mapa

Consumimos la API de dos formas distintas. La primera y más simple es mostrar un mapa con un marcador. Esto lo podemos hacer en la vista de “mostrar actividad” siempre que dicha actividad tenga un lugar creado.

```
function initMap() {  
  
    var lati = '${latitude}';  
  
    var longi = '${longitude}';  
  
    var uluru = {lat: parseFloat(lati), lng: parseFloat(longi)};  
  
    var map = new google.maps.Map(document.getElementById('map'), {  
  
        zoom: 4,  
  
        center: uluru  
  
    });  
  
    var marker = new google.maps.Marker({  
  
        position: uluru,  
  
        map: map  
  
    });  
  
}
```

Place direction

Address: Calle chorizo

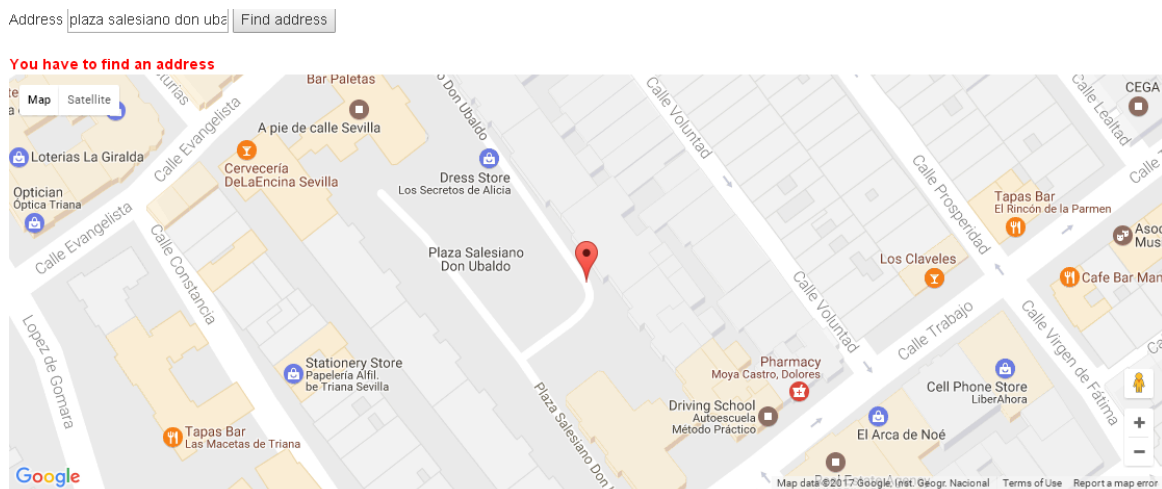


El Código arriba mostrado se encarga de recoger unas variables de longitud y latitud pasadas a la vista y asignarlas al mapa para que se centre en las coordenadas además de crear un marcador en

la misma posición. Con los controles del mapa se puede ampliar y reducir a voluntad para localizar mejor el sitio en cuestión.

Añadir coordenadas a un mapa

Si seleccionamos una actividad sin lugar (Actividad 1 al poblar la base de datos) podremos añadir un lugar nuevo. En este caso habrá una barra de texto y se nos permitirá buscar un lugar para establecer sus coordenadas. Una vez de acuerdo el usuario, se puede guardar el lugar y se pondrá como el lugar establecido para la actividad. Es obligatorio poder encontrar un lugar para enviar el formulario.



El código viene dividido en varios métodos. El primero inicializa el mapa de la misma manera que cuando se mostraba la actividad pero sin marcador. Para el resto de métodos se definen 2 variables que pasaran entre ellos para guardar los valores del mapa y el marcador.

```
var map;  
var marker;
```

El segundo método se encarga de recibir una dirección textual, devuelve el primer par de coordenadas encontrado y centra el mapa en esas coordenadas. Además usa un tercer método para crear un marcador nuevo y situarlo en esas coordenadas también.

```

function searchAddress() {

    var addressInput = document.getElementById('address-input').value;

    var geocoder = new google.maps.Geocoder();

    geocoder.geocode({address: addressInput}, function(results, status) {

        if (status == google.maps.GeocoderStatus.OK) {

            var myResult = results[0].geometry.location; // reference LatLng value

            var lati = myResult.lat();
            var longi = myResult.lng();

            createMarker(myResult); // call the function that adds the marker

            map.setCenter(myResult);

            map.setZoom(9);

            document.getElementById('latitude').value = lati;
            document.getElementById('longitude').value = longi;
            document.getElementById('address').value = document.getElementById('address-input').value;

        } else { // if status value is not equal to "google.maps.GeocoderStatus.OK"

            // warning message

            alert("The Geocode was not successful for the following reason: " + status);

        }

    });

}

```

Por

```
function createMarker(latlng) {  
  
    // If the user makes another search you must clear the marker variable  
    if(marker != undefined && marker != ""){  
        marker.setMap(null);  
        marker = "";  
    }  
  
    marker = new google.maps.Marker({  
        map: map,  
        position: latlng  
    });  
  
}
```

último, hay que asegurarse de que usamos nuestra clave de API en algún lugar de nuestra vista. En nuestro caso se incluye en el layout.

```
<script async defer  
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCOHVq8CTYmEOaRdZl6y15RQBmo9u3A3EA&callback=initMap">  
</script>
```