Programación

Uso de APIs

Vigésimo primera semana

Marzo 2022

Cruz García, Iago

Introducción

API

REST

Usando las APIs

- API sin clave
- API con clave
- POSTMAN

Introducción

El uso de APIs es bastante común a la hora de desarrollar páginas web. Estas ofrecen una interfaz, un puente, entre dos aplicaciones, generalmente una externa y la nuestra. Estas ofrecen datos o funciones externas a nuestro propio programa, que de otra manera serían imposibles.

Recomendación: A partir de ahora es necesario que a la hora de realizar ejercicios nos acostumbremos a buscar información en la documentación oficial. En caso de no poder resolver las dudas con la documentación ofrecida, el siguiente paso será preguntar las dudas en el foro de clase.

Documentación oficial de JavaScript

Foro de la asignatura

Lista de APIs públicas

APIs propietarias de Google

REST y sus funciones

API

API o *Application Programming Interface* o Interfaz de desarrollo de aplicaciones, es un puente entre dos estructuras. Permite conectarlas y utilizar las funcionalidades y datos en nuestro desarrollo para mejorar o ampliar nuestro programa. Estas no son nuevas, pues el uso de periféricos se debe gracias a una interfaz que proporciona el propio sistema operativo, por ejemplo, pero en este tema vamos a aprender a utilizar las APIs de desarrollo online.

Hay que tener en cuenta un detalle: muchas APIs utilizan un sistema de claves o *Keys* para conocer y limitar quien y desde donde se hace la llamada, para evitar sobrecargar los servidores que suministran las mismas. Por la necesidad de incluir la clave en la llamada a la API, algunas de estas no permiten ser llamadas desde JavaScript debido a que esa clave sería pública a cualquiera que usase nuestra página web, por lo que la seguridad sería un problema. Esto es porque están pensadas para que un servidor sea quien haga la llamada y utilice los datos para luego publicarlos en una página web.

REST

No vamos a ver REST en profundidad, pues tiene muchas capas, pero si necesitamos conocer los métodos que ofrece este estándar. Son 4:

- **GET:** Obtiene información de la base de datos de la API.
- **POST:** Crea nueva información en la base de datos de la API.
- **PUT:** Modifica información en la base de datos.
- **DELETE:** Elimina datos.

La mayoría de las APIs pública sólo permiten el uso de GET y en algunos casos, el uso de POST.

Veremos cómo utilizar REST utilizando AJAX, la librería de JQuery que permite hacer llamadas asíncronas a servidores.

Usando las APIs

La mayoría de bases de datos devolverán los valores en formato JSON, un formato muy ligero, alternativa a XML, para la transferencia de datos a través de web sobretodo. El formato de los datos es muy simple:

```
{
“Clave”:”valor”,
“Array”:[“valor1”,”valor2”,”valorN”]
}
```

Esto es un ejemplo muy reducido, la sintaxis de JSON puede ser más elaborada, pero para nuestro caso es suficiente de momento.

En JavaScript, utilizando AJAX, ya hará la conversión de datos de JSON a objetos. Si lo necesitáramos, podríamos usar la función ‘JSON.parse(json)’ para transformar los valores de JSON a un objeto.

Para acceder a los valores de JSON, lo haremos como si de un objeto se tratara, tal que:

```
var datos = “llamada a la API”
console.log (datos.clave1);
```

Una vez establecido lo que es una API, lo que es REST y el formato JSON de forma breve, podemos pasar a realizar llamadas a la API que designemos. En estos ejemplos veremos el uso de una API sin clave y otra con clave.

API sin clave

Utilizaremos la API "dog-api" que permite recoger imágenes de perros al azar con una simple llamada. En el mensaje, devolverá el valor de la url donde se encuentra la imagen de un perro y si la petición fue un éxito o hubo errores.

Veamos el código JavaScript, asumiendo que "imagen" es el id de un campo HTML del tipo img.

main.js

```
function __main__() {
        var settings = {
          "async": true,
          "type": "GET",
          "url": "https://dog.ceo/api/breeds/image/random",
    };

 $.ajax(settings).done(function(response) {
           console.log(response);
           if (response.status == "success") {
             document.getElementById("imagen").src = response.message;
           }
      });
}
__main__();
```

Desgranando el código…

- En **"var settings"** vamos a establecer una serie de parámetros, en formato JSON:

- **async**: establecemos a true, para indicar que la petición se hará de forma asíncrona, por lo que el navegador no se detendrá a esperar el mensaje, si no que continuará ejecutando el código del programa.
- **type**: tipo de petición, en este caso del tipo GET, es decir, que solo pedimos datos.
- **url**: la URL donde hacemos la petición, el objetivo de donde sacaremos los datos

- **$.ajax:** Esta la función de JQuery que permite realizar las peticiones a servidores dada una serie de parámetros, que establecimos en este caso en la variable settings anteriormente.
  - **(settings)**: indica los parámetros de la petición. Es buena idea hacerlos en una variable aparte para poder parametrizar los envios.
  - **.done**: Indica que es lo que debe hacerse cuando la petición se haga. En nuestro caso, realizará una función que recibe por parámetros la respuesta, *responsе* en este caso.
    - A continuación, mostramos por consola lo que contiene la respuesta, preguntamos si el estado de la respuesta fue un éxito y, si es así, después accedemos al valor que almacena la clave *message*, que en el caso de esta API es la imagen del perro, y lo asignamos a la imagen de nuestro HTML.

Como veis, la petición no es extremadamente compleja, pero sí que requiere conocer los valores que estamos utilizando, tanto en la petición como en la recepción.

API con clave

En este caso, voy a utilizar la librería de Riot Games para su juego, League of Legends. Es una API gratuita y abierta, que permite consultar múltiples factores, tanto de las partidas que juegan los usuarios como del estado del juego, pero hay que hacerse una cuenta para poder utilizarla, como la mayoría de APIs que requieren el uso de claves.

En este caso, hago una petición a la API para recuperar los datos del nombre del jugador que se elija:

main.js

```
function __main__() {
	var name = prompt("Nombre de usuario a buscar");
	var api_key = "claveAPIestablecidaPorElServidor";
	var settings = {
	"async": true,
	"type": "GET",
	"url":
"https://euw1.api.riotgames.com/lol/summoner/v4/summoners/by-name/"
+ name + "?api_key=" + api_key,
	"headers": {
	"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0)
Gecko/20100101 Firefox/98.0",
	"Accept-Language": "es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3",
	"Accept-Charset": "application/x-www-form-urlencoded;
charset=UTF-8"
```

```
        }
        };
        $.ajax(settings).done(function(response) {
        console.log(response);
        });

}

__main__()
```

En este ejemplo, en la **url** tenemos que introducir el valor que buscamos, **name**, y el valor de la clave API, **api_key**. La clave API en este caso la oculto, debido a que es de uso privado y no deben compartirse.

Cada API tiene su propia manera de construir las peticiones, por lo que para trabajar con diferentes interfaces, debéis leer la documentación que proporcionan y como os devuelven los datos.

POSTMAN

Como último apartado, os recomiendo el uso de la aplicación POSTMAN, que permite realizar llamadas a API in situ y comprobar los datos devueltos y los envíos. Es muy sencilla de usar y gratuita para uso personal.

Página web oficial de POSTMAN