

Sácale chispas a la API Rest de WordPress

Pero os aviso antes de empezar que
la API engancha y luego no puedes parar :)



**Para empezar un poco de
contexto y de historia**



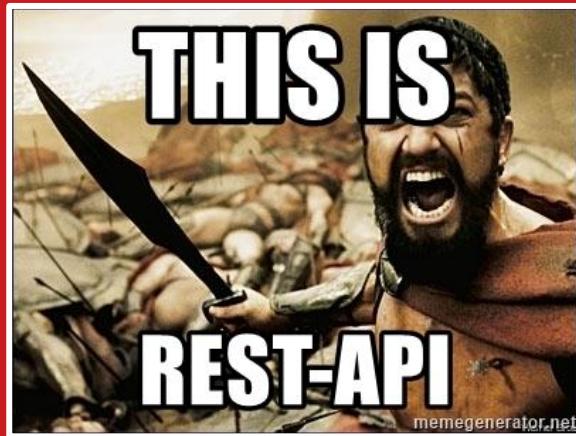
¿Qué es una API?

Una API (Application Programming Interfaces) es un conjunto de subrutinas, funciones y procedimientos que ofrece un software para ser utilizada por otro software.

¿Qué es una API REST?

Una API Rest es un tipo de API con una arquitectura específica:

- Arquitectura cliente-servidor.
- Funciona con peticiones HTTP y cada objeto tiene su propia URI.
- Estas peticiones son sin estado, deben contener toda la información necesaria para hacerse.
- Trabaja sobre 5 tipos de operaciones POST (crear), GET (leer y consultar), PUT (editar), DELETE (eliminar) y PATCH (editar un único dato).
- Devuelve los datos con un formato estándar, XML, JSON, ...



Historia de la API Rest de WordPress

- La API Rest salió a la luz en la versión 4.7 de WordPress, en **diciembre de 2016**, hace más de 5 años.
- Ha ido evolucionando y mejorando con cada nueva versión de WordPress.
- En **agosto de 2019** Woocommerce añade soporte API Rest a su plugin.
- La última versión ha sido con la versión 5.6 de WordPress, en **diciembre de 2020**.
- En **julio de 2021** lo hace YOAST SEO. Aunque por ahora solo acepte métodos GET.



Ventajas y desventajas de la API Rest de WordPress



Ventajas

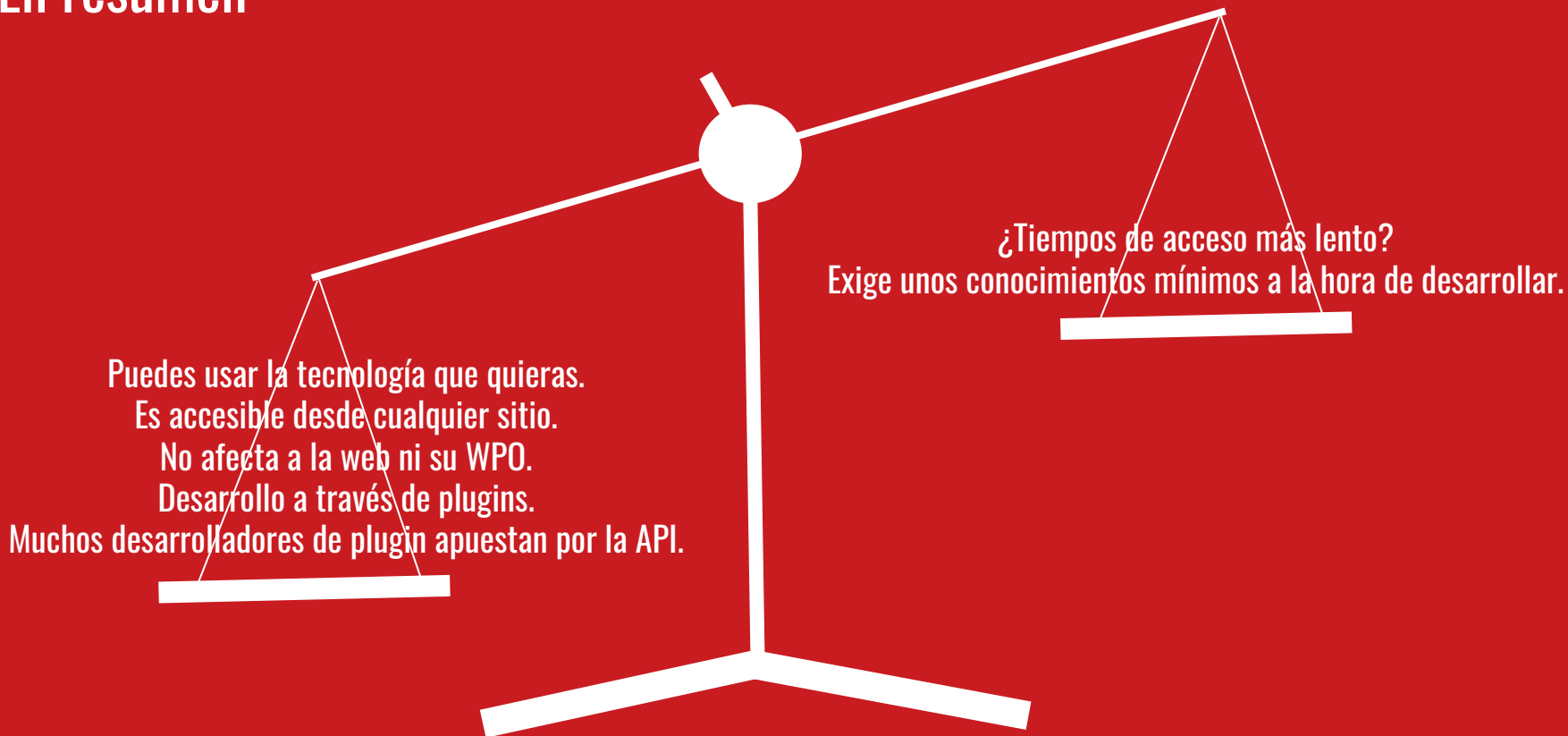
1. **La API ya está desarrollada y es accesible desde cualquier parte, solo tienes que llamarla.** Si la web es visible la API es accesible (por lo menos las llamadas POST y GET que son casi todas). Ayuda mucho cuando trabajas para clientes con alta seguridad, como la administración pública, donde modificar su código o acceder a sus bases de datos es un verdadero infierno.
2. La mayoría de las veces no vas a tener que desarrollar nuevas llamadas y en caso de tener que **desarrollar algún endpoint extra**, puede hacerse a través de plugins con un código sencillo. La API usa las mismas herramientas que la web (WP_Query, el loop, ...). Si lo sabes desarrollar para la web lo sabes desarrollar para la API.
3. En principio **no debería afectar al WPO** de la web. Nunca va a meter js, css, fuentes extra que ralentice la web y solo va a cargar el servidor cuando llamemos a la API.
4. **Libertad total de desarrollo.** Es una API, te da igual cómo funcione. Tú solo haces unas llamadas y recibes unos datos. No estás atado a ninguna tecnología, puedes acceder a la API con Java, con Python, con Javascript, ... Y puedes trabajar desde la propia web, desde una web en otro dominio, desde un ordenador en local, ... incluso desde otra API.
5. Muchos **plugin importantes** y conocidos como Yoast SEO o Woocommerce han **desarrollado sus propias funcionalidades de API**. Con lo cual es un trabajo que te quitas.

Desventajas

1. En teoría **es más lento**, ya que se suma a la gestión interna de la API (conexión a la BD, WP_Query, ...) el tiempo de las llamadas. Aunque las llamadas solo devuelven la información que se les ha asignado, no devuelven menús, ni barras laterales, ni pies, ...
2. Exige un **nivel de conocimiento superior**. Aunque ya hay mucha documentación y plugins para la API Rest, exige tener conocimientos avanzados, por ejemplo, para conectar la API con servicios de terceros.
3. **Y NO SE ME OCURREN MÁS.**



En resumen



Si todo esto no te ha convencido, aquí vamos a ver 4 ejemplos prácticos super-sencillos que usan la API de WordPress que van a convertirte en el héroe o la heroína de tu oficina.



1º ejemplo: Buscador multidominio



Buscador multidominio

Seguro que alguna vez nos hemos encontrado con un cliente que tiene una red de páginas en WordPress (una por cada parte del negocio) y en determinado momento decide juntarlo todo. Y siempre quieren un buscador que busque en todos los dominios. Esto supone un desarrollo grande, plugins de pago, múltiples conexiones a bases de datos, ...

Sin embargo todo se simplifica mucho si acudimos a API. Podemos crear un plugin que:

- Creamos un **código corto** que genera un formulario de búsqueda para poder meterlo donde queramos. Con el formulario va una llamada ajax que envía la búsqueda, recoge los datos de los resultados de búsqueda y los maqueta.
- Preparamos una llamada **WP AJAX** que reciba los datos del formulario, consulte a las APIs de las páginas que queremos y devuelva un json con los resultados.

Vaya Chapa :(

En poco más de 100 líneas de código tenemos un potente (y muy mejorable) buscador multidominio. Simplemente hacemos uso de un endpoint en todas las instalaciones de WordPress que queramos:

```
/wp-json/wp/v2/posts?search=[busqueda]
```

Solo tenemos que meter un código corto como este, con los dominios en un parámetro separado por comas para buscar en las webs de clubes deportivos de Irún.:

```
[buscador_multidominio  
dominios="https://realunionclub.com,http://erroibide.org,https://  
www.cdbidasoa.eus,https://cdsanmarcialirun.com" per_page="10"]
```

¿Y por qué es tan fácil y tan rápido desarrollar este plugin?

Porque la API ya ha desarrollado todo el tema de las búsquedas, nosotros solo debemos hacer las llamadas y juntar y ordenar los resultados.

[Ver código](#)

[Ver ejemplo](#)

2º ejemplo: Generador de boletines



El departamento de marketing te va a poner una estatua si les implementas este ejemplo.



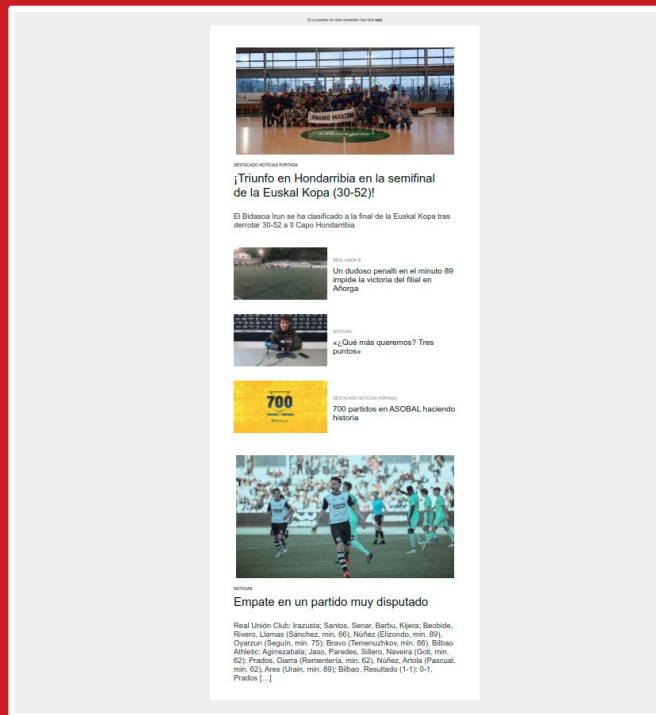
Generador de boletines

Puedes hacer aplicación web que con un simple formulario y sacando los datos de la API de uno o más WordPress genere el HTML que luego podrás mandar en formato boletín con MailChimp, MailRelay, HubSpot, Active Campaign, ...

Y si quieres mejorarlo, puedes ponerle diferentes formatos de noticia. En realidad no hay límites a las opciones que puedes meterle, pero la base es la misma, la API Rest de WordPress.

La tarea de maquetación, de corta y pega de textos y enlaces y de manejar imágenes se reduce ahora a segundos y encima con menos errores.

En el ejemplo que os muestro, podéis hacer un boletín deportivo de 4 clubes deportivos de Irún literalmente en segundos.



[Ver código](#)

[Ver ejemplo](#)

3º ejemplo: Sistema de logeo interdominios



¡¡¡¡OJO!!!! Nada de implementar este ejemplo sin securizarlo. Ahora mismo es un agujero de seguridad muy grande.



Sistema de logeo interdominios

Trabajamos con dos plugins. Uno que funciona como servidor y otro que funciona como cliente.

Con un plugin de servidor podemos crear un endpoint en la API que acepte llamadas GET con un usuario y una contraseña y no devuelve true/false si logea o no el usuario.

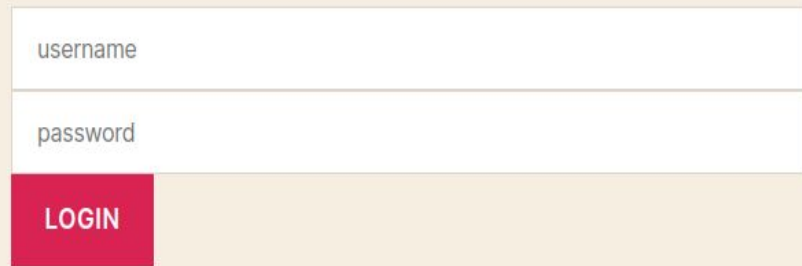
```
/wp-json/wp/login?username=XXXXX&password=XXXXX
```

El plugin de cliente crea un código corto

```
[api-login-client  
dominio="https://midominio.com"]<h1>Puedes ver esto  
por haber metido un usuario y contraseña  
correcto.</h1>[/api-login-client]
```

Este código corto muestra un formulario de logeo que al enviarlo conecta con el endpoint del servidor y prueba el usuario y la contraseña. Si es el servidor devuelve true muestra el contenido del código corto. Si no muestra un error y vuelve a sacar el formulario de logeo.

Imaginad, por ejemplo, que tenemos una tienda online y montamos una landing en otro dominio para una promoción. Podemos hacer que un código promocional de descuento solo sea visible en la landing para los que puedan loguearse con su cuenta de la tienda online.



A login form with two input fields labeled 'username' and 'password', and a red 'LOGIN' button below them.

[Ver código](#)

[Ver ejemplo](#)

4º ejemplo:

Conexión entre Google Sheets y Yoast SEO



La gente de SEO te van a hacer la ola.



Conexión entre Google Sheets y Yoast SEO

Con un plugin podemos crear un endpoint en la API que acepte llamadas POST con un token de seguridad para cambiar los campos personalizados que usa Yoast SEO para el title y la meta-descripción.

`/wp-json/wp/update-seo`

Para gestionarlo todo, simplemente programamos un AppScript en un Google Sheet que recorra todo las filas cuando detecta cambios.

Por cada fila hace un llamada al endpoint y le decimos que metas tiene que por cada por cada slug. En el ejemplo actual hasta podrías trabajar contra varios WordPress, uno en cada página cambiando el dominio y usando en todos el mismo token.

	A	B	C	D
1	Dominio	Slug	Title	Meta description
2	https://pruebas.enuttisworking.com	portada	Conexión entre Google Sheets y Yoast SEO	Home de la web
3	https://pruebas.enuttisworking.com	buscador-multidominio	Buscador multidominios	Buscador multidominios
4	https://pruebas.enuttisworking.com	logeo-interdominios	Sistema de logeo interdominios	Sistema de logeo interdominios
5				
6				
7				
8				
9				
10				
11				

```
Apps Script Ejemplo WordCamp Irun 2022

function updateYoastMetaDescription() {
  var data = SpreadsheetApp.getActiveSheet().getRange().getValues();
  for (row in data) {
    var row = data[row];
    if(row[0].match(/(http[s]?:\/\/.)*?(www.)*?[-a-zA-Z0-9@%~\+=]{2,256}\.[a-z]{2,6}\b(?:[-a-zA-Z0-9@%~\+=~?&\/=]*g)/)) {
      var options = {
        'method': 'post',
        'payload': {slug:row[1], title:row[2], desc:row[3]},
        'headers': {token: 'NkHf8htyeF3dVqx1C1Nbaci0m0vENY1C0J12boN'}
      };
      Logger.log(UrlFetchApp.fetch(row[0]+'-wp-json/wp/update-seo', options));
    }
  }
}
```

Ver código

Ver ejemplo

Otras ideas para usar la API Rest



Importador de contenidos.

Seguro que alguna vez algún cliente te ha pedido que el contenido de una web se publique automáticamente en otra web (vete tú a saber por qué, teniendo en cuenta lo que dice el SEO es que es contenido duplicado).

Podéis hacer que al guardar un post, su contenido se migre y se publique en otras web usando su API Rest.

Trabajar con ReactJS, AngularJS, ...

Puedes usar WordPress como tu CMS en desarrollos con ReactJS o AngularJS. Estos frameworks pueden tirar contra la API Rest de WordPress para sacar contenidos.

Por ejemplo, las noticias de la web del Athletic Club se gestionan desde un WordPress. La web, desarrollada en ReactJS, hace uso de la [API de ese WordPress](#) para mostrar las noticias.

Replicador de usuarios.

Como en el ejemplo anterior, puedes hacer que cuando un usuario se cree o se edite en un WordPress, se replique ese usuario en otros WordPress llamando a las API de esos WordPress.

Con eso conseguiremos que un mismo usuario funcione en varias instalaciones independientes.



Ahora estás pensando, igual que yo en su momento,
en todo el tiempo y dinero que has estado
malgastando estos 5 años en algunos proyectos por
no haber dedicado un rato a conocer y probar la API.



Así pues espero que cada vez que os presenten un proyecto nuevo, os preguntéis:

¿Puedo hacer uso de la API en este proyecto?

La respuesta no tiene que ser siempre **SÍ**, la API no es la panacea, pero es una opción muy interesante que muchos estábamos obviando hasta ahora.





That's all Folks!

Enlaces de interés

- Ejemplos en GitHub - <https://github.com/gwannon/WordCamp-Irun-2022>
- Documentación API Rest de WordPress - <https://developer.wordpress.org/rest-api/>
- Documentación API Rest de Woocommerce - <https://woocommerce.com/document/woocommerce-rest-api/>
- Documentación API Rest de Yoast SEO - <https://developer.yoast.com/customization/apis/rest-api/>

Sobre mí

Programador web con más de 15 años de experiencia, casi todos desarrollando con WordPress. Trabajando actualmente en Eñutt Comunicación S.L.

- GitHub - <https://github.com/gwannon>
- Codepen - <https://codepen.io/gwannon>
- Twitter - @gwannon
- LinkedIn - <https://www.linkedin.com/in/jorgemonclus/>
- Trabajo en Eñutt Comunicación S.L. - <https://enutt.net>

