

Usar la HTTP API para consumir recursos externos

Paulo Carvajal



WORDCAMP
PONTEVEDRA
2022

Soy Paulo Carvajal.

Más de 20 años construyendo sitios web y más de 15 años trabajando con WordPress.

Ahora mismo soy freelance, Toptal talent y Codeable WordPress expert.

@paulocarvajal_
paulocarvajal.com



Usar la HTTP API de WordPress

- Para qué sirve la HTTP API
- Funciones de la HTTP API de WordPress
- Ejemplo de uso de la HTTP API

Para qué sirve la HTTP API

Consumir una API (remota)

- Consumir una API: enviar, recibir o manipular datos.
- Usar los “verbos” HTTP:
 - GET
 - POST
 - PUT
 - DELETE
 - HEAD

Cosas que se pueden hacer

- Traer o enviar noticias, listados o información remota.
- Interactuar con un WordPress headless.
- Crear o interactuar con aplicaciones móviles.
- Enviar actualizaciones a otros sitios.
- Loguearse con credenciales de otro sitio.
- Crear usuario en otra web al registrarse.
- Traer o enviar archivos.

En general, todo lo que hace una API cualquiera.

Funciones de la HTTP API de WordPress

Funciones de WordPress

Funciones básicas

```
wp_remote_request( $url, $args );
```

Están en wp-includes/http.php

```
wp_remote_get( $url, $args );
```

```
wp_remote_post( $url, $args );
```

Son un wrapper de la clase:

```
wp_remote_head( $url, $args );
```

WP_Http

que está en

wp-includes/class-wp-http.php

Funciones de WordPress

Funciones básicas (y sus primas, seguras)

<code>wp_safe_remote_request(\$url, \$args);</code>	La URL se valida para evitar redireccionamientos y ataques de falsificación de solicitudes (request forgery attacks).
<code>wp_safe_remote_get(\$url, \$args);</code>	
<code>wp_safe_remote_post(\$url, \$args);</code>	
<code>wp_safe_remote_head(\$url, \$args);</code>	Pasan la url por:
	<code>wp_http_validate_url(\$url);</code>

Funciones de WordPress

Funciones auxiliares (I)

```
wp_remote_retrieve_response_code( $response );// array|WP_Error
```

```
wp_remote_retrieve_response_message( $response );
```

```
wp_remote_retrieve_headers( $response );
```

```
wp_remote_retrieve_header( $response, $header );
```

```
wp_remote_retrieve_body( $response );
```

Funciones de WordPress

Funciones auxiliares (II)

```
wp_remote_retrieve_cookies( $response );
```

```
wp_remote_retrieve_cookie( $response, $name );
```

```
wp_remote_retrieve_cookie_value( $response, $name );
```

Argumentos de un Request



```
$response = wp_remote_request( $url, $args );

if ( is_array( $response ) && ! is_wp_error( $response ) ) {
    $headers = wp_remote_retrieve_headers( $response );
    $body     = wp_remote_retrieve_body( $response );
    $api_response = json_decode( $body, true );
}
```



Argumentos de un Request



```
$args = array(  
    'body'      => $body,  
    'timeout'   => '5',  
    'redirection' => '5',  
    'httpversion' => '1.0',  
    'blocking'  => true,  
    'headers'   => array(),  
    'cookies'   => array(),  
    'method'    => 'DELETE', // GET, POST, HEAD, PUT  
);
```



Argumentos de un Request

```

$body = array(
    'name'      => 'Jane Smith',
    'email'     => 'someone@email.com',
    'subject'   => 'Checkout this HTTP API stuff',
    'message'   => 'Some text to be sent!',
);

```



Argumentos de un Request



```
$headers = array(  
    'referer'           => home_url()  
    'Authorization'    => 'Basic AUTH_KEY_HERE',  
    'content-type'     => 'application/json; charset=utf-8',  
    'vary'             => 'Accept-Encoding',  
    'Content-Type'     => 'application/json; charset=utf-8',  
    'content-encoding' => 'gzip',  
);
```



Ejemplo de uso de la HTTP API

Plugin para importar posts remotos

Plugin para importación

<https://api.chucknorris.io/>

<https://swapi.dev/>

#PonteWordCamp



[chucknorris.io](https://api.chucknorris.io/) is a free JSON API for hand curated Chuck Norris facts. [Read more](#)

SWAPI

The Star Wars API

(what happened to swapi.co?)

All the Star Wars data you've ever wanted:

Planets, Spaceships, Vehicles, People, Films and Species

From all SEVEN Star Wars films

Now with The Force Awakens data!

PONTEVEDRA
2022

Plugin para importación

Tareas principales

- Llamar al end-point (solicitar la info).
- Comprobar la comunicación y la recepción.
- Parsear la respuesta recibida.
- Importar la información en nuestro sistema.

Plugin para importación

Tareas por servicio

Request

!= Cambian los argumentos

Parse

!= Cambia el formato de los datos

Import

== Datos formateados

Plugin para importación

Iniciamos la petición pasando al “requester” la URL y el parser que queremos que utilice.

```
• • •  
$url = 'https://api.chucknorris.io/jokes/search?query=sleep';  
new chuck_request( $url, new chuck_parser() );
```



Clase requester

```
class chuck_request implements requester_interface {  
  
    private string $remote_url;  
    private array $args;  
    private parser_interface $parser;  
  
    public function __construct ( string $remote_url, parser_interface $parser ) {  
  
        $this->remote_url = $remote_url;  
        $this->parser = $parser;  
        $this->args = [  
            'timeout' => 10,  
            'headers' => [  
                'referer' => home_url(),  
            ],  
        ];  
    }  
  
    public function request():void {}  
  
}
```



Método request

```
$chuck_response = get_transient( 'retrieved_chuck_data' );

if ( false === $chuck_response ) {
    $response = wp_remote_get( $this->remote_url, $this->args );

    if ( ! is_wp_error($response)
        && ( 200 === wp_remote_retrieve_response_code( $response ) ) )
    {
        set_transient( 'retrieved_chuck_data', $response, HOUR_IN_SECONDS );
    }

    try {
        $body = json_decode( wp_remote_retrieve_body( $chuck_response ), false);
    } catch ( \JsonException $e ) {
        throw new \RuntimeException( 'Error Processing Request', 1 );
    }

    $posts = $this->parser->parse( $body->result );

    // Change behaviour here, or extract it to another Class:
    if( ! empty( $posts ) ) {
        foreach ( $posts as $post ) {
            wp_insert_post( $post );
        }
    }
}
```

- Chequeamos si hay transient.
- Chequeamos:
 - Si la respuesta es un error.
 - El código de respuesta es 200 (OK).
- Convertimos el json en array.
- Parseamos el array (\$body).
- Importamos los datos.

Método parse

```
public function parse( array $data ): array {
    return $this->format_data( $data );
}

private function format_data (array $data ): array {
    $posts = [];

    if( ! empty( $data ) ) {
        foreach ( $data as $datum ) {
            $post['post_title'] = $datum->value;
            $post['post_content'] = $datum->created_at . $datum->url;
            $post['post_status'] = 'publish';
            $posts[] = $post;
        }
    }

    return $posts;
}
```

- Encapsulamos el método público.
- Creamos el array “estandarizado”.

```
"created_at": "2020-01-05 13:42:18.823766",
"icon_url": "https://assets.chucknorris.host/img/avatar/chuck-norris.png",
"id": "qqthrspvtqyigfwvai2eq",
"updated_at": "2020-01-05 13:42:18.823766",
"url": "https://api.chucknorris.io/jokes/qqthrspvtqyigfwvai2eq",
"value": "Chuck Norris sleeps with a pillow under his gun."
```

Ejemplo de uso de la HTTP API

Patrón Strategy en OOP

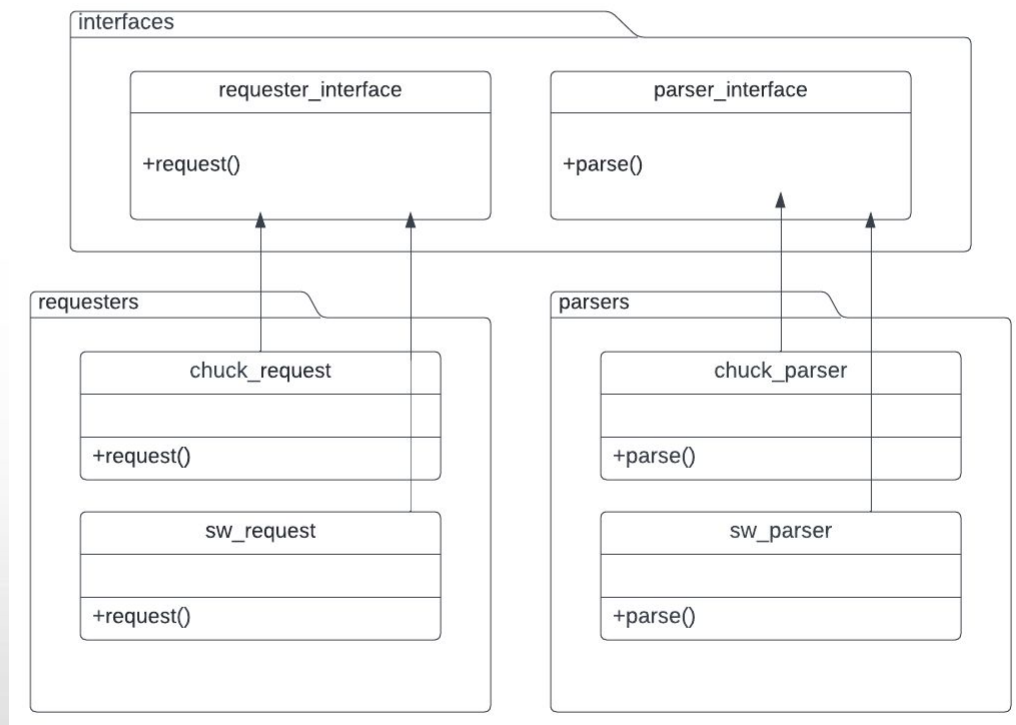


Patrón Strategy

Define y encapsula una familia de algoritmos, y los hace intercambiables.

Permite que los algoritmos varíen, independientemente del cliente que los use.

Patrón Strategy



¡Muchas gracias!



Referencias y documentación

<https://developer.wordpress.org/plugins/http-api/>

<https://torquemag.io/2017/09/guide-async-http-requests-wordpress/>

<https://requests.ryanmccue.info/docs/usage.html>

<https://torquemag.io/2017/09/guide-async-http-requests-wordpress/>

<https://gist.github.com/bueltege/6104254>

https://github.com/paulo-carvajal/wcpo2022_http_api