PROGRAMACIÓN WEB EN ENTORNO SERVIDOR UD-15

JSON

ÍNDICE

[JSON 3](#_Toc155865849)

[CONTROL DE ERRORES CON JSON 3](#_Toc155865850)

[cURL 3](#_Toc155865851)

[cURL + JSON 4](#_Toc155865852)

# JSON

Un API es un sistema de comunicación entre lenguajes o equipos. Nos ayudará a recibir JSONs (JavaScript Object Notation) y a generar respuesta filtrando por verbos y enviando cabeceras especiales.

JSON (JavaScript Object Notation) es un formato de intercambio de datos ligero y fácilmente legible por humanos. PHP proporciona funciones incorporadas para trabajar con JSON, lo que facilita la manipulación de datos entre aplicaciones web. Aquí te presento una guía detallada sobre cómo usar JSON en PHP.

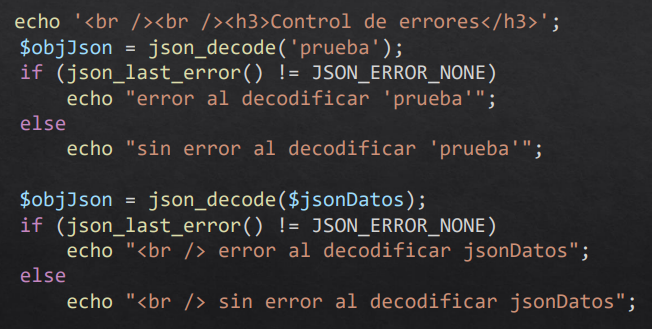
Básicamente tendremos dos funciones a la hora de usar JSONs:

* String: json\_encode(array): La función json\_encode() se utiliza para convertir datos de PHP a formato JSON. Es el que se utiliza en las llamadas a la web.
* Objeto: json\_decode(stringJson): La función json\_decode() se utiliza para convertir datos JSON en objetos PHP. Es el que se utiliza en las respuestas de la web.

# CONTROL DE ERRORES CON JSON

Cuando ejecutemos la función de codificar o decodificar JSON, podemos preguntar al método json\_last\_error()si se han producido errores. Si no se han producido errores nos devolverá la constante JSON\_ERROR\_NONE.

Se pueden ver el resto de respuestas que nos puede dar la función json\_last\_error() desde el siguiente link: <https://www.php.net/manual/es/json.constants.php>



# cURL

cURL (Client URL) es una biblioteca y una herramienta de línea de comandos para transferir datos con sintaxis de URL. En PHP, puedes utilizar la biblioteca libcurl para realizar comunicaciones con diferentes servidores y protocolos (HTTPS, HTTP, GET, POST, HTTP PUT, subidas FTP, proxies, cookies, y autenticación usuario+contraseña.). Antes de usar cURL, asegúrate de que la extensión esté instalada y habilitada en tu servidor. Hay que comprobar que cURL esté instalado.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

# cURL + JSON

Lo primero que hay que hacer es inicializar un objeto de tipo cURL mediante $objCurl = curl\_init();

No debemos olvidar cerrar siempre el ojeto cURL mediante el método curl\_close($objurl); cuando terminemos de usarlo.

Debemos seleccionar las opciones de la conexión, aunque como mínimo debemos definir la url mediante curl\_setopt($objCurl, CURLOPT\_URL, 'https://swapi.dev/api/people/10/');. El método curl\_setopt tiene muchas más opciones que se pueden consultar en el manual <https://www.php.net/manual/es/function.curl-setopt.php>

Por ejemplo podría poner que el método de envío fuera POST en lugar de GET con curl\_setopt($objCurl, CURLOPT\_POST, 1); o para enviar datos con post curl\_setopt($objCurl, CURLOPT\_POSTFIELDS, $arrDatos);

Cuando ya tenemos definidas las opciones para el objeto cURL (como mínimo la dirección), ejecutaremos para que nos devuelva el JSON con $objJSON = curl\_exec($objCurl);

Para controlar los errores que haya podido dar la consulta, puedo utilizar dos métodos curl\_errno($objCurl) me indica el número del error y curl\_error($objCurl) me indica el mensaje del error.

Una vez obtenida la respuesta, decodificaremos el mensaje JSON y pasaremos a utilizar los datos.

Normalmente la parte de cURL se implementa como un método estático de una clase auxiliar, a la que le indicamos la url y nos devuelve los datos que debemos procesar en nuestro programa.