

Formato Json

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos.

Está constituido por dos estructuras:

- ✓ Una colección de pares de nombre/valor. (*Objeto*, registro, estructura, diccionario, etc..)
- ✓ Una lista ordenada de valores. (Arreglos, vectores, listas o secuencias).

```
{  
  "Nombre" : "Juan",  
  "Edad": 28,  
  "Aficiones": ["Música", "Cine", "Tenis"],  
  "Residencia": "Madrid"  
}
```

Formato Json

USO.

Comúnmente es utilizado para transmitir datos en aplicaciones web.

Ejemplo: enviar algunos datos desde el servidor al cliente, así estos datos pueden ser mostrados en páginas web, o vice versa.

Notas:

- JSON es sólo un formato de datos
- Una coma o dos puntos mal ubicados pueden producir que un archivo JSON no funcione.

Ajax

- Es una técnica de JavaScript que nos permite consultar información o enviar información desde y hacia un servidor externo.
- La información que descargamos puede ser recibida usualmente como texto o como objeto JSON. Lo más usual es JSON, para así poder manipularlo como un objeto de JavaScript.

AJAX

OBTENIENDO UN JSON

<https://mdn.github.io/learning-area/javascript/oojs/json/superheroes.json>

Para obtener el JSon, se utilizara un API llamado [XMLHttpRequest](#) .

- Para empezar, se debe almacenar la URL del JSON que se quiere recuperar en una variable. Agregue lo siguiente al final del código JavaScript:

```
var requestURL = 'https://mdn.github.io/learning-area/javascript/oojs/json/superheroes.json';
```

AJAX

- Para crear una solicitud, se necesita crear una nueva instancia de objeto de solicitud desde el Constructor `XMLHttpRequest`, utilizando la palabra clave `new`.

```
var request = new XMLHttpRequest();
```

- Ahora es necesario abrir una nueva solicitud utilizando el método `open()`.

```
request.open('GET', requestURL);
```

- Se envía la solicitud con el método `send()`:

```
request.responseText = 'json'; request.send();
```

AJAX

- Para poder esperar por la respuesta a retornar desde el servidor y luego, manejarla.

```
request.onload = function() {  
  var superHeroes = request.response;  
  console.log(superHeroes);  
}
```

AJAX CON JQUERY

Con jQuery se ha simplificado los pedidos AJAX que llevaban mucho código. La más simple función (\$.get) que permite leer un JSON de Internet y ejecutar una función cuando los datos han llegado.

```
$.get("https://cotizaciones.com/dolar", function(respuesta) {  
    var dolar = respuesta.monedas.dolar;  
    $("#cotizacionDolar").html(dolar)  
});
```

AJAX CON FETCH

Los Navegadores modernos incluyen una función llamada fetch (pedir), que se parece bastante a \$.get de jQuery y permite hacer pedidos AJAX sin necesidad de tener la librería jQuery de forma fácil y rápida. El mismo código pero usando fetch sería:

```
fetch("https://cotizaciones.com/dolar")
  .then( function(respuesta) {
    return respuesta.json(); // le decimos que queremos JSON
  })
  .then( function (respuesta) {
    var dolar = respuesta.monedas.dolar;
    document.getElementById("cotizacionDolar").innerHTML = dólar;
  });
```


AJAX CON FETCH

Para navegadores que no soportan fetch se puede agregar una pequeña librería que la haga compatible.

```
<script src="https://unpkg.com/unfetch/polyfill"></script>
```

La lista de navegadores compatibles con fetch está actualizada y disponible en

<https://caniuse.com/#search=fetch>