

JSON- Folio Giratorio

1. Introducción

- Acrónimo de JavaScript Object Notation, es un formato de texto ligero utilizado para el intercambio de datos.
- Debido a su amplia adopción como alternativa de XML, se considera un lenguaje independiente.
- Ventajas:
 - Es más sencillo escribir un analizador sintáctico (parser)
 - Se puede analizar fácilmente utilizando la función eval()
- Es frecuente también el uso de XML y JSON, como por ejemplo Google Maps.

2. Nombre y Pronunciación

- En inglés, JSON se pronuncia de forma acronímica, como el nombre de la letra J seguido de la sílaba «son».
- («yéison»), por adaptación fonética del acrónimo en inglés;

3. Sintaxis

Los tipos de datos disponibles con JSON son:

- Números: Se permiten negativos , naturales, y decimales separados por punto.
- Cadenas: Son secuencias de caracteres. Se ponen entre comillas(“ ”)
- Booleanos: Representan sólo dos valores, o verdadero o falso
- NULL: Representa un valor nulo.
- Array: Representa una lista de valores que pueden ser de cualquier tipo.
- Objetos: Son colecciones no ordenadas, con la sintaxis <nombre>:<valor>, separados por comas y entre llaves

4. Modelos de procesamiento

Al ser un formato muy extendido, se han desarrollado APIs para muchos lenguajes distintos, que permiten analizar, generar procesar..etc todos estos datos.

Modelo de objeto: el Json completo se guarda en memoria, en formato árbol, que es analizado y modificado con las APIs apropiadas. Este modelo consume muchos recursos pero es muy flexible para manipular el contenido..

Modelo de flujo: Los datos son leídos o escritos en bloques. Cada vez que se lee un bloque, se genera un evento. El cliente puede procesar el contenido escuchando este tipo de eventos.

5. Uso de Json

En teoría, es trivial analizar JSON en JavaScript usando la función `JSON.parse()` incorporada en el lenguaje. En la práctica, las consideraciones de seguridad por lo general recomiendan no usar *eval* sobre datos crudos y debería usarse un analizador JavaScript distinto para garantizar la seguridad. El analizador proporcionado por JSON.org usa `eval()` en su función de análisis, protegiéndola con una expresión regular de forma que la función sólo ve expresiones seguras.

```
var http_request = new XMLHttpRequest();
var url = "http://example.net/jsondata.php"; // Esta URL debería
devolver datos JSON
```

```
// Descarga los datos JSON del servidor.
http_request.onreadystatechange = handle_json;
http_request.open("GET", url, true);
http_request.send(null);
```

```
function handle_json() {
  if (http_request.readyState == 4) {
    if (http_request.status == 200) {
      var json_data = http_request.responseText;
      var the_object = eval("(" + json_data + ")");
    } else {
      alert("Ocurrió un problema con la URL.");
    }
    http_request = null;
  }
}
```

6. Comparación

XML goza de mucho menor soporte y ofrece menos herramientas de desarrollo. Los entornos en el servidor normalmente requieren que se incorpore una función u objeto analizador de JSON.

Algunos programadores, especialmente los familiarizados con el lenguaje C, encuentran JSON más natural que XML, pero otros desarrolladores encuentran su escueta notación algo confusa, especialmente cuando se trata de datos fuertemente jerarquizados o anidados muy profundamente.