

JSON

Antonio Espín Herranz

Introducción

- JavaScript Object Notation es un formato ligero de representación de datos.
- Idealmente pensado para el intercambio de datos, permite estructurar la información sin requerir XML.
- Normalmente, una representación JSON suele ser mas compacta que su equivalente XML.
- JSON puede ser analizado de forma más rápida que XML, por tanto es una buena opción allí donde el flujo de datos AJAX es muy grande.
- En **JavaScript**, JSON es analizado mediante la función **eval()**.

Introducción

- JSON está constituido por dos estructuras:
 - Una colección de pares de nombre/valor. Lo que como conocemos como un objeto.
 - Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arrays.
- Un **objeto** es un conjunto ordenado desordenado de pares nombre/valor. Comienza con una { y termina con }
- Cada nombre es seguido de : (dos puntos) y los pares **nombre / valor** van separados por , (comas).

Introducción

- Un **array** es una colección de valores. Un array comienza con [(corchete izquierdo) y termina con] (corchete derecho). Los valores se separan con , (coma).
- Un **valor**, puede ser una cadena de caracteres con comillas dobles o un número, o true o false o null, o un objeto, o un array.
- Las estructuras pueden anidarse. Podemos tener relaciones de composición entre objetos.

Introducción

- Una cadena de caracteres es una colección de 0 o más caracteres, encerrados entre comillas dobles, usando barras divisorias invertidas como escape.
- Secuencias de escape: `\n`, `\t`, `\r`, etc.

Notación

- Es simple transformar un documento XML a la notación JSON:
 - El documento JSON comenzará y terminará en una llave.
 - Cada elemento XML se transforma en una cadena de caracteres encerrada entre comillas dobles, a la que sigue el carácter de dos puntos.
 - En XML: <titulo>
 - En JSON: “titulo”:

Notación

- El contenido textual de los elementos aparecerá posteriormente, encerrado entre comillas y seguido de una coma, si hay mas contenido:
 - XML: <titulo>Ejemplo de JSON</titulo>
 - JSON: “titulo”:”Ejemplo de JSON”,

Notación

- Si el elemento de XML contiene elementos hijos, estos aparecerán encerrados entre corchetes.
- No necesario con el elemento raíz.
- Si un elemento contiene atributos, cada uno se trata como un elemento y su valor como el contenido textual del elemento. Si tiene varios, se separan por comas.

Notación

- Si un elemento contiene atributos, cada uno se trata como un elemento y su valor como el contenido textual del elemento. Si tiene varios se separa por comas.
- Los atributos de los elementos se encerrarán entre llaves.
- `<item valor="demo" nota="ejemplo" />`
- Se transforma en:
 - `"item" : [
 {"valor" : "demo", "nota" : "ejemplo" }
]`

Ejemplo en XML

```
<?xml version="1.0"?>
```

```
<menu id="fichero" valor="Fichero">
```

```
  <popup>
```

```
    <menuitem id="nuevo" value="Nuevo Fichero"/>
```

```
    <menuitem id="abrir" value="Abrir Fichero" />
```

```
    <menuitem id="salir" value="Salir del menú"/>
```

```
  </popup>
```

```
</menu>
```

Ejemplo en JSON

```
{ "menu" : {  
  "id": "fichero",  
  "valor": "Fichero",  
  "popup": {  
    "menuitem": [  
      {"id": "nuevo", "value": "Nuevo Fichero"},  
      {"id": "abrir", "value": "Abrir Fichero"},  
      {"id": "salir", "value": "Salir del menu"}  
    ]  
  }  
}
```

JSON y AJAX

- La forma de interactuar con XMLHttpRequest es muy similar, teniendo en cuenta que:
 - Se ha de manipular la respuesta del Servidor mediante la propiedad **responseText**.
 - Tras obtener la respuesta, se evalúa mediante **eval**.

Ejemplo

```
function funcionProcesamiento(){
```

```
if (http_request.readyState == 4){
```

```
// Carga completada:
```

```
if (http_request.status == 200){ // OK
```

```
var nuevoTitulo;
```

```
// Respuesta del Servidor:
```

```
var json_doc =  
    http_request.responseText;
```

```
alert(json_doc);
```

```
// Obtengo la información deseada:
```

```
var datos = eval("(" + json_doc + ")");
```

```
alert(".- id: " + datos.menu.id);
```

```
alert(".- valor: " + datos.menu.valor);
```

```
for (var i=0; i < datos.menu.popup.menuitem.length; i++)
```

```
    alert(datos.menu.popup.menuitem[i].id + " = " +  
        datos.menu.popup.menuitem[0].value);
```

```
} else
```

```
window.alert("respuesta del Servidor: " +  
    http_request.status);
```

```
}
```

```
}
```

PRACTICA:

JSON