

Transferencia datos con JSON

<http://www.json.org/>



Objetivos



- Necesitamos una estructura para crear mensajes complejos usando separadores
- Necesitamos un sintaxis parecida a los elementos de HTML: `<element attribute="value">content</element>`
- Queremos una sintaxis general para definir jerarquías (datos anidados).

XML vs JSON

JSON y XML son lenguajes para la transmision de los datos:

- XML hay muchas herramientas para gestionar este tipo de datos
- JSON tiene una sintaxis mas natural y mas fácil de usar

Estructura de almacenaje datos



- Queremos transmitir un mensaje compuesto por muchos elementos

My note:

BEGIN

FROM: Alice Smith (alice@example.com)

TO: Robert Jones (roberto@example.com)

SUBJECT: Tomorrow's "Birthday Bash" event!

MESSAGE (english):

Hey Bob,

Don't forget to call me this weekend!

PRIVATE: true

END

- Alternativas XML o JSON



Ejemplo con XML

```
<?xml version="1.0" encoding="UTF-8"?>
<note private="true">
  <from>Alice Smith (alice@example.com)</from>
  <to>Robert Jones (roberto@example.com)</to>
  <to>Charles Dodd (cdodd@example.com)</to>
  <subject>Tomorrow's "Birthday Bash"
  event!</subject>
  <message language="english">
    Hey guys, don't forget to call me this weekend!
  </message>
</note>
```



Ejemplo con JSON

```
{  
  "private": "true",  
  "from": "Alice Smith (alice@example.com)",  
  "to": [  
    "Robert Jones (roberto@example.com)",  
    "Charles Dodd (cdodd@example.com)"  
  ],  
  "subject": "Tomorrow's \"Birthday Bash\" event!",  
  "message": {  
    "language": "english",  
    "text": "Hey guys, don't forget to call me this weekend!"  
  }  
}
```

JSON vs XML



- **Como el XML**

- El JSON es una información almacenata como texto (no codificada)
- JSON es "self-describing" (facil a leer para el usuario)
- JSON tiene jerarquias (valores anidados)
- JSON puede ser analizado con un parsed de JavaScript
- Los datos de JSON pueden ser transportados usando AJAX

- **Al contrario del XML**

- No hay un tag de fin
- Es mas compacto
- Mas facil al escribir
- Se puede analizar usando un parser JavaScript "eval()" pero es peligroso!!!
- Permite el uso de arrays
- No tiene palabras reservadas



Estructura JSON

- En JSON se pueden utilizar dos tipos de estructuras:
 - Una coleccion de nombres/valores. En distintos lenguajes eso corresponde a un objeto, un registro, una estructura, un diccionario, una tabla de hash, o un array asociativo.
 - Una lista de valores ordenados. En muchos lenguajes corresponde a un *array*, vector, lista, o sequence.



Estructura JSON

- Un objeto
 - Empieza con “{” y acaba con “}”. Cada nombre es seguido por “:” y los valores/parejas son separados por “,”
- Un array
 - Empieza con “[” y acaba con “]” . Los valores son separados por “,”.
- Un valor
 - Puede ser un *string*, un *numero*, un booleano o “null”, un *object* or un *array*. Esas estructuras pueden ser anidadas.

Ejemplo:

```
{  
  "employees": [  
    { "firstName":"John" , "lastName":"Doe" },  
    { "firstName":"Anna" , "lastName":"Smith" },  
    { "firstName":"Peter" , "lastName":"Jones" }  
  ]  
}
```


Ejemplo



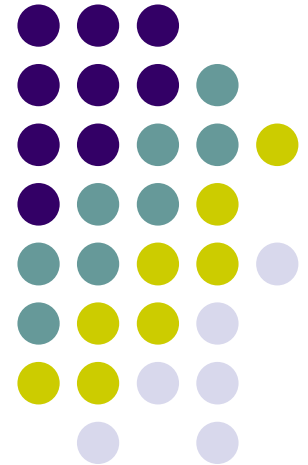
```
var data =
{
  "window": {
    "title": "Sample Widget",
    "width": 500,
    "height": 500
  },
  "image": {
    "src": "images/logo.png",
    "coords": [250, 150, 350, 400],
    "alignment": "center"
  },
  "messages": [
    {"text": "Save", "offset": [10, 30]},
    {"text": "Help", "offset": [0, 50]},
    {"text": "Quit", "offset": [30, 10]},
  ],
  "debug": "true"
}
```

Como podemos acceder a sus valores?

- El titulo de la ventana?
`var title = data.window.title;`
- La tercera coordenada de la imagen?
`var coord = data.image.coords[2];`
- El numero de mensajes?
`var len = data.messages.length;`
- El y-offset del ultimo mensaje?

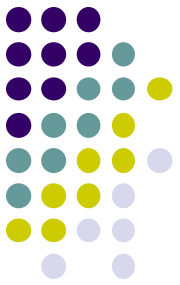
`var y = data.messages[len - 1].offset[1];`

JSON y DOM



JSON y DOM

http://www.w3schools.com/json/json_intro.asp



```
<!DOCTYPE html><html><body>
<h2>JSON Object Creation in JavaScript</h2>
```

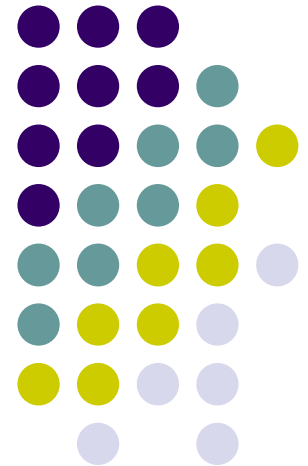
```
<p>
Name: <span id="jname"></span><br>
Age: <span id="jage"></span><br>
Address: <span id="jstreet"></span><br>
Phone: <span id="jphone"></span><br>
</p>
```

```
<script>
var JSONObject = {
  "name": "John Johnson",
  "street": "Oslo West 16",
  "age": 33,
  "phone": "555 1234567"};
document.getElementById("jname").innerHTML=JS
ONObject.name;
document.getElementById("jage").innerHTML=JS
ONObject.age;
document.getElementById("jstreet").innerHTML=JS
ONObject.street;
document.getElementById("jphone").innerHTML=JS
ONObject.phone;
</script>
</body></html>
```

JSON Object Creation in JavaScript

Name: John Johnson
Age: 33
Address: Oslo West 16
Phone: 555 1234567

JSON y JavaScript





Métodos de JSON

- `JSON.parse(string)`
 - convierte el string de datos JSON en un objeto de JavaScript
- `JSON.stringify(object)`
 - convierte un objeto definido en un string de datos de JSON (el contrario de `JSON.parse`)
- Se puede usar Ajax para recuperar datos en un formato JSON
- Una vez transformado en objeto, se puede interactuar como se hace con cualquier objeto JavaScript



Función eval()

- Para convertir un JSON en un objeto se puede usar la función eval()
`var myObject = eval('(' + myJSONtext + ')');`
- La función eval() es muy rapida en ejecutarse.
- ATENCION!!! Permite ejecutar **cualquier programa JavaScript** , por lo tanto genera problemas de seguridad. Se debería usar eval() solo si la fuente del JSON es fiable. Por ejemplo si la origen del JSON es la misma pagina web.
- En todos los otros casos, se prefiere usar un parser JSON.



Ejemplo JSON.parse()

```
<ul id="groups"></ul>
var jsonString = '[
  {"name":"Manchester GTUG",
   "meetup":"First Monday of every month",
   "tags":["gtug","google","manchester","madlab"]},

  {"name":"Manchester jQuery Group",
   "meetup":"First Tuesday of every month",
   "tags":["jquery","javascript","jresig","madlab"]},

  {"name":"Hybrid!",
   "meetup":"First Monday of every month",
   "tags":["jquery","javascript","jresig","madlab"]}]';
```

```
var myData = JSON.parse(jsonString);
```

```
$(document).ready(function() {
  var $grouplist = $('#groups');
  $.each(myData, function() {
    $('<li>' + this.name +
    '</li>').appendTo($grouplist);
  });
});
```

Resultado:

- Manchester GTUG
- Manchester jQuery Group
- Hybrid!

Fuente : <http://jsfiddle.net/qmacro/NJMyD/>

Ejemplo JSON.parse()



Fichero JSON

```
{
  "name": "mkyong",
  "age": 30,
  "address": {
    "streetAddress": "88 8nd Street",
    "city": "New York"
  },
  "phoneNumber": [
    {
      "type": "home",
      "number": "111 111-1111"
    },
    {
      "type": "fax",
      "number": "222 222-2222"
    }
  ]
}
```

<script>

```
var data = '{"name": "mkyong","age": 30,"address": {"streetAddress": "88 8nd Street","city": "New York"},"phoneNumber": [{"type": "home","number": "111 111-1111"}, {"type": "fax","number": "222 222-2222"}]'};
```

```
var json = JSON.parse(data);
```

```
alert(json["name"]); //mkyong
alert(json.name); //mkyong
```

```
alert(json.address.streetAddress); //88 8nd Street
alert(json["address"].city); //New York
```

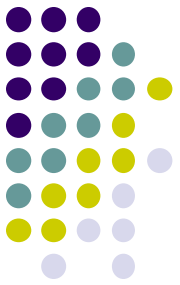
```
alert(json.phoneNumber[0].number); //111 111-1111
alert(json.phoneNumber[1].type); //fax
```

```
alert(json.phoneNumber.number); //undefined
```

</script>

Ejercicios colaborativos

<https://codeshare.io/COW06>

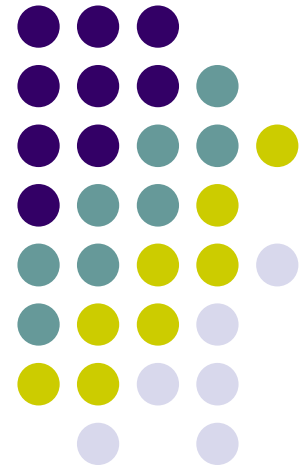


```
var data = JSON.parse(this.responseText);
```

```
{
  "window": {
    "title": "Sample Widget",
    "width": 500,
    "height": 500
  },
  "image": {
    "src": "images/logo.png",
    "coords": [250, 150, 350, 400],
    "alignment": "center"
  },
  "messages": [
    {"text": "Save", "offset": [10, 20]},
    {"text": "Help", "offset": [0, 50]},
    {"text": "Quit", "offset": [30, 15]}
  ],
  "debug": "true"
}
```

- Considerando el JSON y el código
 - Como recupero el titulo del window?
 - La tercera coordenada de la imagen?
 - El numero de mensajes?
 - El y-offset del ultimo mensaje?

JSON y Ajax





JSON y Ajax

- Cualquier dato que se carga con AJAX puede ser almacenado usando el formato JSON en un servidor.
- Ajax permite recuperar esos ficheros cuando necesario, y procesarlos con un parser

- Ejemplo:

Cricketer Details

Name	Country
Sachin	India

Update Details

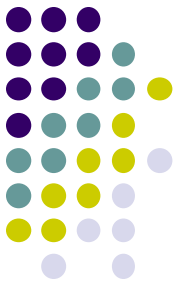
- Clicando

Cricketer Details

Name	Country
brett	Australia

JSON y Ajax

http://www.tutorialspoint.com/json/json_ajax_example.htm



```
<html><head> <script>
function loadJSON()
{
    var data_file = "IP_host//data.json";
    var http_request = new XMLHttpRequest();
}

http_request.onreadystatechange = function(){
    // Javascript function JSON.parse to parse
    // JSON data
    var jsonObj =
        JSON.parse(http_request.responseText);

    // jsonObj variable now contains the data
    // structure and can be accessed as
    // jsonObj.name and jsonObj.country.

    document.getElementById("Name").innerHTML = jsonObj.name;

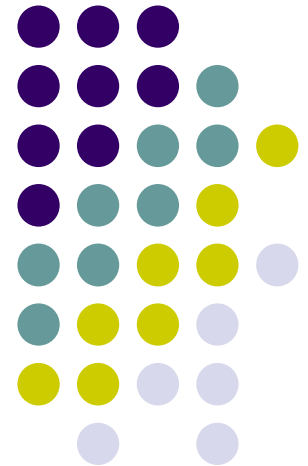
    document.getElementById("Country").innerHTML = jsonObj.country;
}
http_request.open("GET", data_file, true);
http_request.send();
}</script>
```

```
</head>
<body>
<h1>Cricketer Details</h1>
<table class="src">
<tr><th>Name</th><th>Country</th></tr>
<tr><td><div id="Name">Sachin</div></td>
<td><div id="Country">India</div></td></tr>
</table>

<div class="central">
<button type="button" onclick="loadJSON()">Update
    Details </button>
</div>
</body>
</html>
```

```
IP_host//data.json
{"name": "brett", "country": "Australia"}
```

JSON con PHP





funciones JSON

- PHP incluye las siguientes funciones globales par interactuar con JSON:
 - [json_decode\(string\)](#) es un parser que devuelve un string JSON o devuelve un objeto *associative array* (como JSON.parse en JavaScript)
 - [json_encode\(object\)](#) devuelve un JSON considerando un objeto dado (como JSON.stringify en JavaScript)



ENCODING

Ejemplo json_encode():

```
<?php
```

```
$arr = array('a' => 1, 'b' => 2, 'c' => 3, 'd' =>  
4, 'e' => 5);
```

```
echo json_encode($arr);
```

```
?>
```

Produce

```
{"a":1,"b":2,"c":3,"d":4,"e":5}
```

Ejemplo JSON



```
<?php
$data = array(
    "library" => "Odegaard",
    "category" => "fantasy",
    "year" => 2012,
    "books" => array(
        array("title" => "Harry Potter", "author" => "J.K. Rowling"),
        array("title" => "The Hobbit", "author" => "J.R.R. Tolkien"),
        array("title" => "Game of Thrones", "author" => "George R. R. Martin"),
        array("title" => "Dragons of Krynn", "author" => "Margaret Weis"),
    )
);

print json_encode($data);
?>
```


Resultado JSON



```
{  
  "library": "Odeggaard",  
  "category": "fantasy",  
  "year": 2012,  
  "books": [  
    {"title": "Harry Potter", "author": "J.K. Rowling"},  
    {"title": "The Hobbit", "author": "J.R.R. Tolkien"},  
    {"title": "Game of Thrones", "author": "George R. R. Martin"},  
    {"title": "Dragons of Krynn", "author": "Margaret Weis"},  
  ]  
}
```



ENCODING Otro ejemplo

- El código

```
<?php
class Emp {
    public $name = "";
    public $hobbies = "";
    public $birthdate = "";
}
$e = new Emp();
$e->name = "sachin";
$e->hobbies = "sports";
$e->birthdate = date('m/d/Y h:i:s a', "8/5/1974 12:20:03 p");
$e->birthdate = date('m/d/Y h:i:s a', strtotime("8/5/1974 12:20:03"));

echo json_encode($e);
?>
```

- Produce

```
{"name":"sachin","hobbies":"sports","birthdate":"08/05/1974 12:20:03 pm"}
```



DECODING

- PHP `json_decode()`
- la function permite convertir un objeto JSON en código PHP.
- La función `var_dump()` permite visualizar el contenido de una estructura (tipo y valor)
- Ejemplo, el código:

```
<?php
$json = '{"a":1,"b":2,"c":3,"d":4,"e":5}';

var_dump(json_decode($json));
var_dump(json_decode($json, true));
?>
```

Produce

```
array(5) {
    ["a"] => int(1)
    ["b"] => int(2)
    ["c"] => int(3)
    ["d"] => int(4)
    ["e"] => int(5)
}
```