JSON

- JSON: JavaScript Object Notation
- JSON es una sintaxis para guardar e intercambiar datos
- JSON es una alternativa al uso de XML

Ejemplo de JSON

- Un objeto Employees, con 3 registros de empleados:

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

Correspondencia con XML

```
<employees>
    <employee>
        <firstName>John</firstName> <lastName>Doe</lastName>
    </employee>
    <employee>
        <firstName>Anna</firstName> <lastName>Smith</lastName>
    </employee>
    <employee>
        <firstName>Peter</firstName> <lastName>Jones</lastName>
    </employee>
</employees>
```

Características

- Formato de intercambio de datos ligero
- Es independiente del lenguaje
- Es auto-descriptivo y fácil de comprender
- Utiliza sintaxis Javascript y es únicamente texto (como el XML), por tanto puede ser leído por cualquier lenguaje de programación
- Extensión .json y el MIME type es “application/json”

Javascript entiende JSON

- El formato de JSON es sintácticamente idéntico al código para crear objetos en Javascript
- Debido a esta similaridad, en lugar de utilizar un «parser», un programa en JavaScript puede utilizar funciones estándar para convertir datos en JSON en objetos nativos Javascript

```
<!DOCTYPE html><html><body>

<h2>JSON Object Creation in JavaScript</h2><p id="demo"></p>

<script>

var text = '{"name":"John Johnson","street":"Oslo West 16","phone":"555 1234567"}';

var obj = JSON.parse(text);

document.getElementById("demo").innerHTML =
obj.name + "<br>" +
obj.street + "<br>" +
obj.phone;

</script></body></html>
```

Parecido al XML

- Tanto el JSON com el XML son auto-descriptivos
- Tanto el JSON com el XML son jerárquicos (valores dentro de valores)
- Tanto el JSON com el XML puede ser «parseado» por muchos lenguajes de programación
- Tanto el JSON com el XML pueden ser recuperados a través del XHR

Y a diferencia del XML

- JSON no utiliza tags de finalización
- JSON es más corto
- JSON es más rápido de leer y escribir
- JSON puede utilizar arrays
- La principal diferencia es que el XML tiene que ser parseado con un XML parser mientras que el JSON puede ser parseado por cualquier función en Javascript

¿Por qué utilizar JSON?

- Se utilizará para aplicaciones en AJAX dado que es más rápido y fácil que XML
- Se recuperará un string JSON
- Se utilizará JSON.parse para parsear el valor

Sintaxis de JSON

- Es un subconjunto de la sintaxis de Javascript
- Los valores están en pares name/value
- Los valores están separados por coma
- Los objetos están delimitados por corchetes {}
- Los arrays quedan indexados usando los []
- Ahora bien, en JSON los nombres requiren las comillas dobles “letra”: “A”

Valores en un JSON

- Un número (integer or floating point)
- Una cadena de texto (in double quotes)
- Un Booleano (true o false)
- Un array (in [])
- Un objeto (in {})
- NULL

Objetos y Arrays en JSON

- Los objetos en JSON se escriben entre corchetes y pueden contener múltiples valores en parejas: { “firstName”: “John”, “lastName”: “Doe”}

- Y los arrays: "employees":[{"firstName":"John", "lastName":"Doe"}, {"firstName":"Anna", "lastName":"Smith"}, {"firstName":"Peter","lastName":"Jones"}]

```
<p id="demo"></p>

<script>
var employees = [
    {
    "firstName":"John",
    "lastName":"Doe"
    },
    {
    "firstName":"Anna",
    "lastName":"Smith"
    },
    {
    "firstName":"Peter",
    "lastName":"Jones"
    }
];
document.getElementById("demo").innerHTML =
employees[0].firstName + " " + employees[0].lastName;
</script>
```

```
<p id="demo"></p>

<script>
var employees = [
    {
    "firstName":"John",
    "lastName":"Doe"
    },
    {
    "firstName":"Anna",
    "lastName":"Smith"
    },
    {
    "firstName":"Peter",
    "lastName":"Jones"
    }
];
document.getElementById("demo").innerHTML =
employees[0]["firstName"] + " " + employees[0]["lastName"];
</script>
```

Usos

- Un uso típico de JSON es para leer datos de un servidor y mostrar esos datos en una página web
- Veamos un ejemplo en el que creamos un string en Javascript con notación/ sintaxis de JSON
- Usaremos la función JSON.parse(text) para convertir texto en JSON a un objeto en Javascript:

```
var text = '{ "employees" : [' +
'{ "firstName":"John" , "lastName":"Doe" },' +
'{ "firstName":"Anna" , "lastName":"Smith" },' +
'{ "firstName":"Peter" , "lastName":"Jones" } ]}';

var obj = JSON.parse(text);
```

```
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML =
obj.employees[1].firstName + " " + obj.employees[1].lastName;
</script>
```

- Los navegadores antiguos que no pueden utilizar la función JSON.parse() pueden utilizar la función eval() para convertir texto en JSOn a un objeto en JS:

```
var obj = eval ("(" + text + ")");
```

- Sin embargo, la función eval() puede ejecutar cualquier código JS así que esto presenta un problema potencial de seguridad. En entorno de producción es importante evitar su uso

Fuente: http://www.w3schools.com/js/js_json_intro.asp

EXTRA: Extensión Chrome para visualizar mejor un JSON: JSON Formater

EXTRA: API postman.com (aplicación web)

EXTRA: JSON-SERVER servir API estática:
https://github.com/typicode/json-server