

Desenvolupament d'aplicacions web

Desenvolupament web a l'entorn client

UT 4.2: JSON

Índex de continguts

JSON.....	3
Valors.....	4
Objectes.....	5
Arrays.....	6
Utilització de JSON.....	7
application/json.....	8



JSON

JavaScript Object Notation, llegit normalment com a “jaison” o “jotason”, **és un format d'intercanvi de dades** basat en una part de l'especificació d'ECMAScript, Javascript per els amics.

És un **format de text** completament independent del llenguatge, però que utilitza característiques i termes provinents dels llenguatges tipus C (C++, Java, Javascript, Python, ...)

Que sigui un format de text, com l'XML, implica que en utilitzar un JSON sempre es trobarà **dins d'una variable de tipus text o si és un literal, anirà tancat entre cometes**.

És una alternativa a XML. Com a característiques addicionals podem dir que és fàcil d'entendre i escriure per els humans i de generar i interpretar per les màquines. A més, la mateixa informació en format json ocupa molt menys espai que en format XML.

Un objecte amb JSON

```
{ "nom" : "Joan", "edat" : 4 }
```

L'equivalent amb XML:

```
<persona> <nom>Joan</nom> <edat>4</edat> </persona>
```

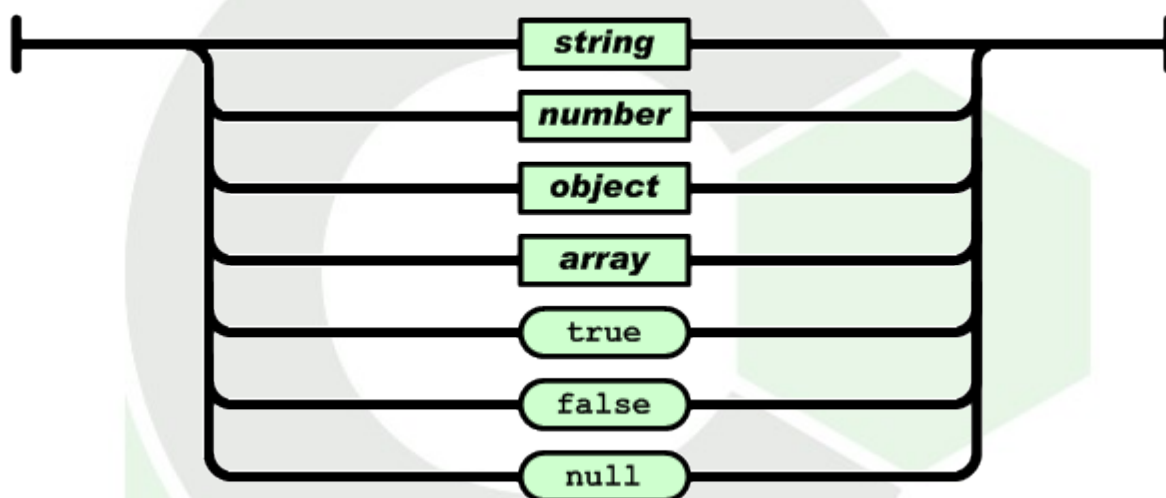
JSON es basa en dues estructures freqüents a molts llenguatges de programació, objectes i arrays.

Valors

Els arrays poden contenir valors i els objectes parells nom – valor, però què és un valor?

Un **valor** pot ser una cadena de caràcters entre cometes dobles, o un número, o true o false, o null, o un objecte o un array.

value



Exemples de valors vàlids:

```
"Jo"  
2345  
12.05  
true  
false  
null  
{ "nom" : "Joan" }  
[ "jo", 12.05, null ]
```

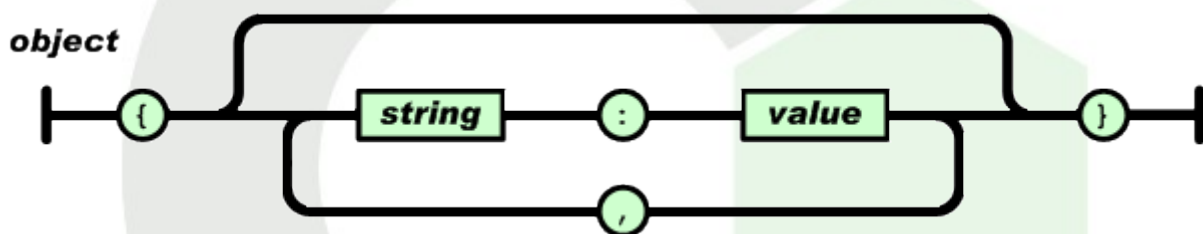
Els dos últims són, respectivament, un objecte i un array. Els veurem tot seguit.

Objectes

Objecte: Una col·lecció de parells nom - valor. És l'equivalent en altres llenguatges a objectes, registres, estructures, diccionaris, arrays associatius, ...

Un objecte comença amb { i acaba amb }. Cada nom d'atribut va entre **cometes dobles**, seguit per : i el valor. Els parells nom - valor estan separats per ,

```
{ "nom" : "Joan", "edat" : 4 }
```



L'anterior diagrama ens diu que un objecte pot tenir aquests tres formats, seguint els tres camins del diagrama:

1. Pot ser buit, si seguim el camí superior
2. Pot tenir un únic parell nom – valor si seguim el camí central
3. Pot tenir n parells nom – valor separats per comes si segueix el camí inferior després de definir cada parell.

Els següents exemples serien vàlids:

```
{ }
```

```
{ "nom" : "Joan" }
```

```
{ "nom" : "joan", "edat" : 18, "adreça" : { "carrer" : "Joan Miró", "numero" : 22, "cp" : 07300, "localitat" : "Inca" } }
```

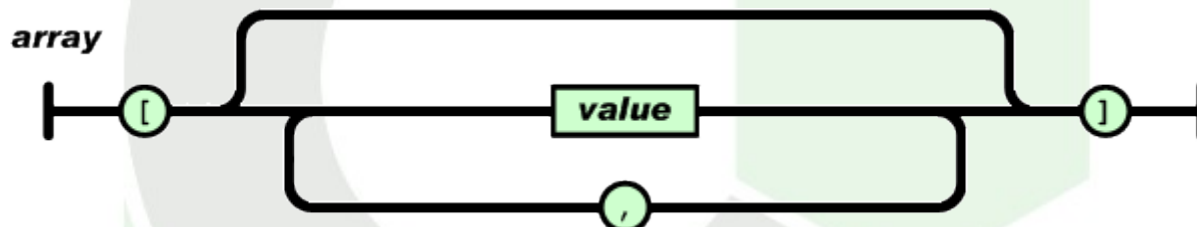
Com a valor podem tenir qualsevol dels valors descrits anteriorment, per tant un valor pot ser un objecte o un array també.

Arrays

Array: Una llista de valors. En la majoria dels llenguatges s'implementa com arrays, vectors, llistes o seqüències.

Un array comença amb `[` i acaba amb `]`. Els valors es separen amb `,`

```
'[ "joan" , 4 , "blau" , true ]'
```



L'anterior diagrama ens diu que un array pot tenir aquests tres formats, seguint els tres camins del diagrama:

1. Pot ser buit, si seguim el camí superior
2. Pot tenir un únic valor si seguim el camí central
3. Pot tenir n valors separats per comes si seguim el camí inferior després de cada valor.

Els següents exemples serien vàlids:

```
'[]'  
'[ 12 ]'  
'[ 12, "asd", 234.75 ]'  
'[ 12, { "nom" : "joan", "edat" : 18 } ]'  
'[ 12, [ "asd", 234.75 ], true ]'
```

Utilització de JSON

Una bona notícia: La majoria de llenguatges disposen de llibreries per convertir un objecte o un array a JSON i a l'inrevés, donat un JSON, convertir-lo a array o objectes.

Per exemple,

- amb php, convertir un array a json és tan senzill com passar-lo d'argument a la funció `json_encode`.
- amb JavaScript
 - convertir un JSON a objecte o array és tan fàcil com aplicar-li la instrucció `JSON.parse(cadenaJson)`. Avalua el contingut i el transforma sempre que tengui un format JSON correcte.

```
const enviatPerElServidor = '{ "nom" : "joan", "edat" : 18 }';  
console.log(enviatPerElServidor .nom); // error, una cadena de text no té la  
// propietat nom.  
  
const joan = JSON.parse( enviatPerElServidor );  
  
//dins joan tendrem l'objecte javascript { nom : "joan", edat : 18 }  
console.log(joan.nom); //Mostrarà joan
```

- Per convertir un objecte o array a JSON podem utilitzar `JSON.stringify`.

```
const joan = { nom : "joan", edat : 18 }  
const json = JSON.stringify(joan);  
  
console.log(json.nom); // error, una cadena de text no té la propietat nom.  
  
console.log(json); //Mostrarà { "nom" : "joan", "edat" : 18 }
```

- Per altres llenguatges podeu consultar a la pàgina <http://json.org/>

application/json

Un php, servlet, qualsevol cosa que faci que el servidor torni com a resultat un JSON hauria de:

- Posar la capçalera *content-type* a *application/json*
- Incloure només el JSON. Ni cap element HTML, ni cap altre cosa que el JSON.

La capçalera *content-type* indica al client el format de la informació que li està arribant. D'aquesta manera facilita que la tracti correctament.

Els valors d'aquesta capçalera estan definits per la [iana](https://www.iana.org/assignments/media-types/media-types.xhtml) i si teniu curiositat els podeu trobar a l'enllaç <https://www.iana.org/assignments/media-types/media-types.xhtml>

En programar serveis rest hauríeu de generar les respostes amb aquest format. Un exemple d'una resposta d'un servei REST que torna un JSON és el següent:

```
HTTP/1.1 200
Path=/; HttpOnly
Expires: 0
Content-Type: application/json
Transfer-Encoding: chunked
Date: Wed, 28 Sep 2022 16:45:23 GMT
Keep-Alive: timeout=60
Connection: keep-alive

{
  "id": 6,
  "title": "AGENT TRUMAN"
}
```