

Server-side Web Development

UD 2.1 JSON i YAML



IES Jaume II El Just
Tavernes de la Valldigna
Departament d'Informàtica
Curs 2024-25

Índex

1	JSON	2
1.1	Sintaxi JSON	2
1.2	Tipus de valors	2
1.2.1	Array	3
1.2.2	Objecte	3
1.2.3	Strings	3
1.2.4	Número	4
1.2.5	Booleà	4
1.2.6	Null	4
1.3	Dades JSON emmagatzemades	4
2	YAML	5
2.1	Sintaxis de YAML	6
2.2	Tipus de dades	6
2.2.1	Valors escalars	7
2.2.2	Objectes	7
2.2.3	Seqüències	8
2.2.4	Diccionaris	12
3	Referències	13

1 JSON

JSON és un format que emmagatzema informació estructurada i s'utilitza principalment per transferir dades entre un servidor i un client.

El fitxer és bàsicament una alternativa més simple i lleugera a l'XML (llenguatge de marcatge estés, per les seves sigles en anglès) que compta amb funcions similars.

Els desenvolupadors usen JSON per treballar amb AJAX (JavaScript asíncron i XML, per les sigles en anglès). Aquests formats funcionen bé junts per aconseguir la càrrega asincrònica de les dades emmagatzemades, el que significa que un lloc web pot actualitzar el vostre informació sense actualitzar la pàgina.

Aquest procés és més fàcil de fer amb JSON que amb XML/RSS. I avui, com molts llocs web estan adoptant AJAX, el fitxer .json s'ha tornat molt popular.

1.1 Sintaxi JSON

Ja saps què és un fitxer JSON, ara has de saber que per crear correctament un fitxer **.json**, has de seguir la sintaxi correcta.

Hi ha dos elements centrals en un objecte JSON: **claus** (*Keys*) i **valors** (*Values*).

Les **Keys** han de ser cadenes de caràcters (*strings*). Com el seu nom indica, aquestes contenen una seqüència de caràcters envoltats de cometes.

Els **Values** són un tipus de dades JSON vàlides. Podeu tenir la forma d'un array, objecte, cadena (string), booleà, número o nul.

Un **objecte** JSON comença i acaba amb **claus** `{ }`. Podeu tenir dos o més parells de claus/valor dins, amb una **coma** per separar-los. Així mateix, cada key és seguida per dos punts per distingir-la del valor.

Exemple:

```
{"ciutat": "Nova York", "país": "Estats Units"}
```

Aquí tenim dos parells de clau/valor: ciutat i país són les claus; Nova York i els Estats Units són els valors.

1.2 Tipus de valors

Com s'esmentava abans, els valors contenen un tipus de dades JSON vàlid, com:

1.2.1 Array

Un **array** (conegut com a arranjament o vector) és una **col·lecció ordenada de valors**. Està envoltat de **claudàtors** `[]` i cada valor dins està separat per una coma.

Un valor d'un array pot contenir objectes JSON, el que significa que utilitza el mateix concepte de parell clau/valor. Per exemple:

```
{"estudiants": [  
  {"Nom": "Tom", "Cognom": "Jackson"},  
  {"Nom": "Linda", "Cognom": "Garner"},  
  {"Nom": "Adam", "Cognom": "Cooper"}  
]}
```

En aquest cas, la informació entre claudàtors és un array, que té tres objectes.

1.2.2 Objecte

Un **objecte** conté una **clau** i un **valor**. Hi ha dos punts després de cada clau i una coma després de cada valor, que també distingeix cada objecte. Tots dos estan entre cometes.

L'objecte, com a valor, ha de seguir la mateixa regla que un objecte comú. Exemple:

```
{"empleats": {"nom": "Tom", "cognom": "Jackson"}}
```

Ací, empleats és la clau, mentre que tot el que és dins de les claus és l'objecte.

1.2.3 Strings

Un **string** (cadena de caràcters) és una seqüència establerta de zero o més caràcters Unicode. Està tancat entre **dos cometes dobles**.

Aquest exemple mostra que Tom és un string ja que és un conjunt de caràcters dins una cometa doble:

```
{"Nom": "Tom"}
```

1.2.4 Número

El número a JSON ha de ser un nombre enter o un punt flotant, com

```
{"Edat": 30}
```

1.2.5 Booleà

Pots utilitzar true o false com a valor, de la següent manera:

```
{"Casat": false }
```

1.2.6 Null

És per mostrar que no hi ha informació, valor nul.

```
{"Tipus de sang": null }
```

1.3 Dades JSON emmagatzemades

Hi ha dues formes de emmagatzemar dades JSON: **objecte** i **vector**. El primer es veu així:

```
{  
  "nom": "Tom",  
  "cognom": "Jackson",  
  "gènere": "masculí"  
}
```

Les claus indiquen que és un objecte JSON. Té tres parelles clau/valor que estan separades per comes. A cada parella, tens les claus (nom, cognom i gènere) seguides de dos punts per distingir-los dels valors (Tom, Jackson, masculí). Els valors en aquest exemple són strings. Per això també estan entre cometes, com les claus.

Un altre mètode per emmagatzemar dades és un **vector** (array). Fes-li una ullada a aquest exemple:

```
{  
  "nom": "Tom",  
  "cognom": "Jackson",  
  "gènere": "masculí",  
  "hobby": ["futbol", "lectura", "natació"]  
}
```

El que diferencia això del mètode anterior és el quart parell clau/valor. Hobby és la clau i hi ha diversos valors (futbol, lectura, natació) entre claudàtors, que representen un vector.

Per a representar un document json que tinga correspondència amb XML es necessita un element arrel, el qual és un objecte el valor del qual és un array d'objectes. Per exemple:

```
{"empleats": [  
  {"nom": "John", "cognom": "Doe"},  
  {"nom": "Anna", "cognom": "Smith"},  
  {"nom": "Peter", "cognom": "Jones"}  
]}
```

empleats és l'arrel del document json i els 3 valors que conté estan dins d'un array. Aquest document es podria representar en forma de llista així:

```
Empleats:  
- Nom: John  
- Cognom: Doe  
  
- Nom: Anna  
- Cognom: Smith  
  
- Nom: Peter  
- Cognom: Jones
```

2 YAML

YAML significa *Ain't Markup Language*, és un format de dades per a serialització, intercanvi de dades i arxius de configuració.

Aquests fitxers contenen dades de text organitzades en una estructura jeràrquica.

Admet tipus escalars (enter, cadenes, flotant, booleà, temps) i tipus de col·lecció (array, llista).

Punts importants:

- És **sensible a majúscules i minúscules**
- L'extensió de fitxer és **.yaml**
- **No es permeten les tabulacions**

2.1 Sintaxis de YAML

Un document yaml conté diverses línies de text amb les regles següents.

En un alt nivell, els documents yaml s'interpreten com a mapes o diccionaris:

- La primera línia comença amb ---
- Cada línia de text conté parelles de claus i valors
- La clau i els valors estan separats per dos punts (:) i espai
- S'usa el **sagnat a l'estil Python** per a indicar que un element està dins d'un altre.
- Utilitzeu **espais en compte de tabulacions** per al sagnat
- Es poden escriure comentaris d'una línia començant per #

Un mapa senzill de clau-valor:

```
clau: valor
```

2.2 Tipus de dades

YAML admet els tipus de dades següents:

Tipus	Descripció
String	Les cadenes es tanquen amb o sense cometes simples o dobles
Nombre	Pot ser nombres enters o valors flotants
Booleà	Accepta true o false
Seqüència	Admet array o llista
Set	Set emmagatzema els elements no ordenats sense duplicats

Tipus	Descripció
Map	Els mapes emmagatzemen parelles clau-valor com ara un hash map
Binari	Dades binàries de cadenes octals, imatges, fitxers
Timestamp	Marca de temps

2.2.1 Valors escalars

Els escalars són **nombres enters** o **valors flotants**, **cadenes** i **valors booleans**. Les cadenes es poden tancar amb cometes simples, dobles o sense cometes:

```
valor: 11
pi: 3.14
isEnabled: false
strvalue1: 'valor de cadena1'
strvalue2: "valor de cadena2"
strvalue3: valor de cadena3
```

2.2.2 Objectes

Recordem com és un objecte en JSON:

```
{
  "id": 1
  "nom": "Miquel Sanchis"
}
```

La seua representació en YAML seria:

```
---
id: 1
nom: Miquel Sanchis
```


2.2.3 Seqüències

Un **array** és un grup de valors similars amb un sol nom. En YAML, un array representa una única clau assignada a diversos valors. Cada valor comença amb un guionet – seguit d'espai:

```
clau1:  
  - valor1  
  - valor2  
  - valor3
```

Es pot representar en una línia tancant els valors entre claudàtors []:

```
clau1: [valor1,valor2,valor3]
```

El que també es pot representar en vàries línies:

```
clau1: [valor1,  
        valor2,valor3]
```

La representació en JSON seria:

```
{  
  "clau1": [  
    "valor1",  
    "valor2",  
    "valor3"  
  ]  
}
```

Els arrays també es poden representar sense clau:

```
[un, dos, tres, quatre]
```

```
- un  
- dos  
- tres  
- quatre
```

En JSON seria:

```
[  
  "un",  
  "dos",  
  "tres",  
  "quatre"  
]
```

Més exemples:

```
## Array de nombres enters  
numbers: [  
  1,  
  2,  
  3,  
  4  
]
```

```
## Array de nombres enters i decimals  
numbers: [  
  1.5,  
  2.9,  
  3,  
  4  
]
```

Per a representar **arrays d'arrays**, o **arrays multidimensionals**, fem la notació indentada:

```
estudiants:  
-  
  id: 213  
  nom: Franc  
  dades:  
    - estudis: English  
      estudi_id: 1  
    - edat: 23  
    - nacionalitat: USA  
      codi: 97845
```

L'equivalent en JSON seria:

```
{
  "estudiants": [
    {
      "id": 213,
      "nom": "Franc",
      "dades": [
        {
          "estudis": "English",
          "estudi_id": 1
        },
        {
          "edat": 23
        },
        {
          "nacionalitat": "USA",
          "codi": 97845
        }
      ]
    }
  ]
}
```

Un **array d'objectes** conté una **llista d'objectes** (un objecte conté parells clau-valor):

```
---
alumnes:
  - id: 1
    nom: Franc
  - id: 2
    nom: Mara
```

El que en JSON seria:

```
{
  "alumnes": [
    {
      "id": 1,
      "nom": "Franc"
    },
  ],
}
```

```
{
  "id": 2,
  "nom": "Mara"
}
]
```

Es poden niuar arrays dins d'objectes així com objectes dins d'arrays:

```
---
alumnes:
  - id: 1
    nom: Franc
    assignatures:
      - Programació
      - Matemàtiques
  - id: 2
    nom: Mara
    assignatures:
      - Estructura de computadors
      - Programació
      - Física
```

EL mateix exemple en JSON:

```
{
  "alumnes": [
    {
      "id": 1,
      "nom": "Franc",
      "assignatures": [
        "Programació",
        "Matemàtiques"
      ]
    },
    {
      "id": 2,
      "nom": "Mara",
      "assignatures": [
        "Estructura de computadors",
        "Programació",

```

```
    "Física"  
  ]  
}  
]
```

2.2.4 Diccionaris

Un diccionari conté parelles clau-valor.

Un diccionari en Yaml es pot representar en dues sintaxis:

- Mapatge fluid
- Mapatge de blocs

En el **mapatge fluid**, les parelles de clau i valor estan separades per una coma i totes les parelles s'inclouen entre claus {}:

```
# diccionari  
mysqlatabase: {hostname: localhost, port: 3012, username: root, password:  
  ↪ root}
```

En el **mapatge de blocs**, cada parella clau-valor es representa en una línia amb indentació:

```
# diccionari  
mysqlatabase:  
  hostname: localhost  
  port: 3012  
  username: root  
  password: root
```

L'equivalent en JSON:

```
{  
  "mysqlatabase": {  
    "hostname": "localhost",  
    "port": 3012,  
    "username": "root",  
    "password": "root"  
  }  
}
```

3 Referències

- [Json.org](#)
- [JSON W3Schools](#)
- [YAML Tutorial](#)