

JSON

Ph. Truillet

Octobre 2020– v. 1.0



JSON _ JavaScript Object Natation

JSON est un format d'échange de données, facile à lire par un humain et interpréter par une machine. Basé sur JavaScript, il est complètement indépendant des langages de programmation mais utilise des conventions qui sont communes à tous les langages de programmation (C, C++, Perl, Python, Java, C#, VB, JavaScript, ...).

Deux structures sont utilisées par JSON :

- Une collection de clefs/valeurs : **Object**. L'objet commence par un « { » et se termine par « } » et composé d'une liste non ordonnée de paires clefs/valeurs. Une clef est suivie de « : » et les paires clef/valeur sont séparés par « , »
- Une collection ordonnée d'objets : **Array**, liste ordonnée d'objets commençant par « [» et se terminant par «] », les objets sont séparés l'un de l'autre par « , »

JSON supporte plusieurs types de données :

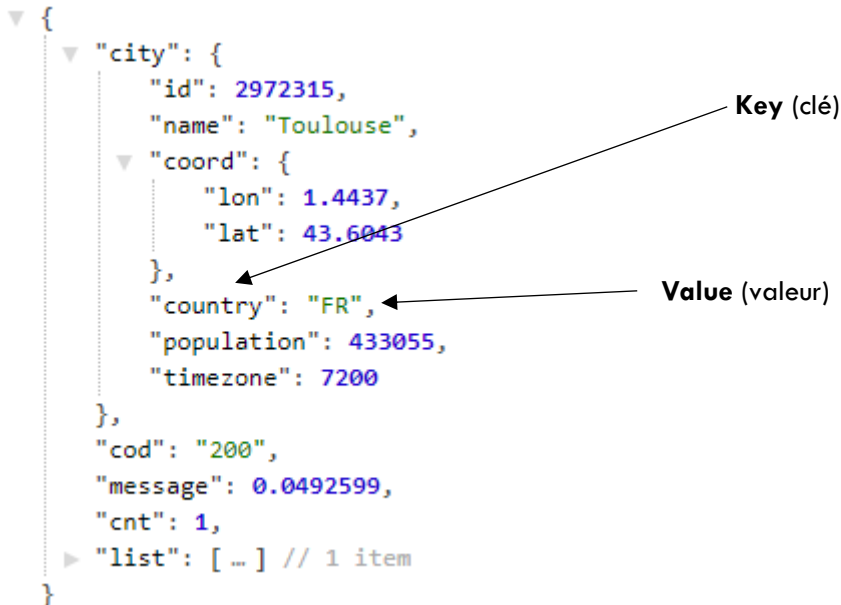
- **une Valeur**
 - **Numérique** : nombre entier ou flottant
 - **Chaîne de caractères** : ensemble de caractères Unicode (sauf une double quote " et un antislash \) entouré par des doubles guillemets.
 - **Booléen** : true ou false
 - La valeur **null**
- **un Tableau** : un ensemble ordonné de valeurs entouré par des crochets [et]. Chaque valeur est séparée par un caractère virgule. Les types des valeurs d'un tableau peuvent être différents
- **un Objet** : est composé de paires clé/valeur, chacune étant séparée par une virgule, entourées par des accolades { et }.
Une clé est obligatoirement une chaîne de caractères. Une valeur peut être littérale (chaîne de caractères, un nombre, un booléen, la valeur null), un objet ou un tableau. Une clé est séparée de sa valeur par le caractère « deux points : »

Les valeurs d'un objet ou d'un tableau peuvent être d'un de ces types.

Dans une chaîne de caractères, le caractère d'échappement est le caractère antislash qui permet notamment de représenter dans la chaîne de caractères :

- **\"** : une double quote
- **** : un antislash
- **\/** : un slash
- **\b** : un caractère backspace
- **\f** : un caractère formfeed
- **\n** : une nouvelle ligne
- **\r** : un retour chariot
- **\t** : une tabulation
- **\unnnn** : le caractère Unicode dont le numéro est nnnn

Un exemple de JSON



JSON et Processing.org

Afin de manipuler des données JSON avec Processing, plusieurs opérations sont nécessaires.

1. Charger les données JSON en mémoire

```
JSONObject json = loadJSONObject(chaine_JSON)
```

Ici, on charge dans la structure JSON le contenu d'un fichier JSON (souvent récupéré après une requête sur le web)

Il va falloir ensuite accéder à chaque champ en chargeant d'éventuels tableaux ou objets pour y arriver

2. Accéder à un tableau

Par exemple, nous pouvons extraire le tableau :

```
JSONArray values = json.getJSONArray("list");
```

3. Accéder à un objet

```
JSONObject list = values.getJSONObject(0);
```

4. Récupérer une valeur

```
float pressure = list.getFloat("pressure");
```



Liens

- Introducing JSON, <https://www.json.org/json-en.html>