

15/04/2023

Version 3

JSON

JÉRÉMY PERROUULT



JSON

SON HISTOIRE

L'HISTOIRE DE JSON

- **JSON** (Java**S**cript **O**bject **N**otation) existe depuis le début des années 2000
- Sa première spécification officielle date de 2013
- Yahoo et Google ont commencé à l'utiliser en 2005 et 2006
 - C'est à partir de là qu'il se démocratise

L'HISTOIRE DE JSON

- Tout comme le **XML**, **JSON** permet stocker et d'échanger des informations
- Contrairement au **XML**, il n'est pas structuré
 - C'est d'ailleurs dans ce format que sont manipulées les bases de données documentaires
- L'énorme avantage face au **XML**
 - Pas de description
 - Moins verbeux et donc, plus léger
- Sa syntaxe est très proche des syntaxes *Object de JavaScript*

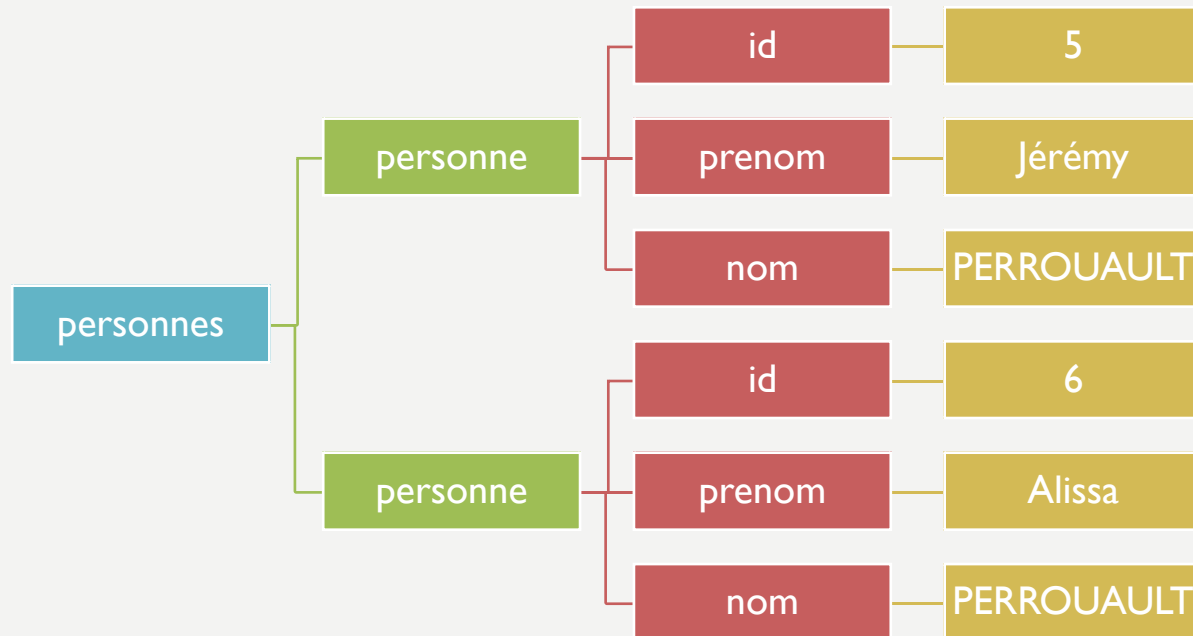


JSON

SON FORMALISE

FORMALISME

- L'information est hiérarchique mais non structurée
 - On peut la représenter sous la forme d'un arbre
 - Avec **une** racine (élément principal) et des branches (sous-éléments) sur plusieurs niveaux



FORMALISME

- **JSON** accepte plusieurs types
 - Objet
 - Array (tableaux)
 - Nombre
 - Chaîne de caractères
 - Booléens (true, false)
 - Null

FORMALISME

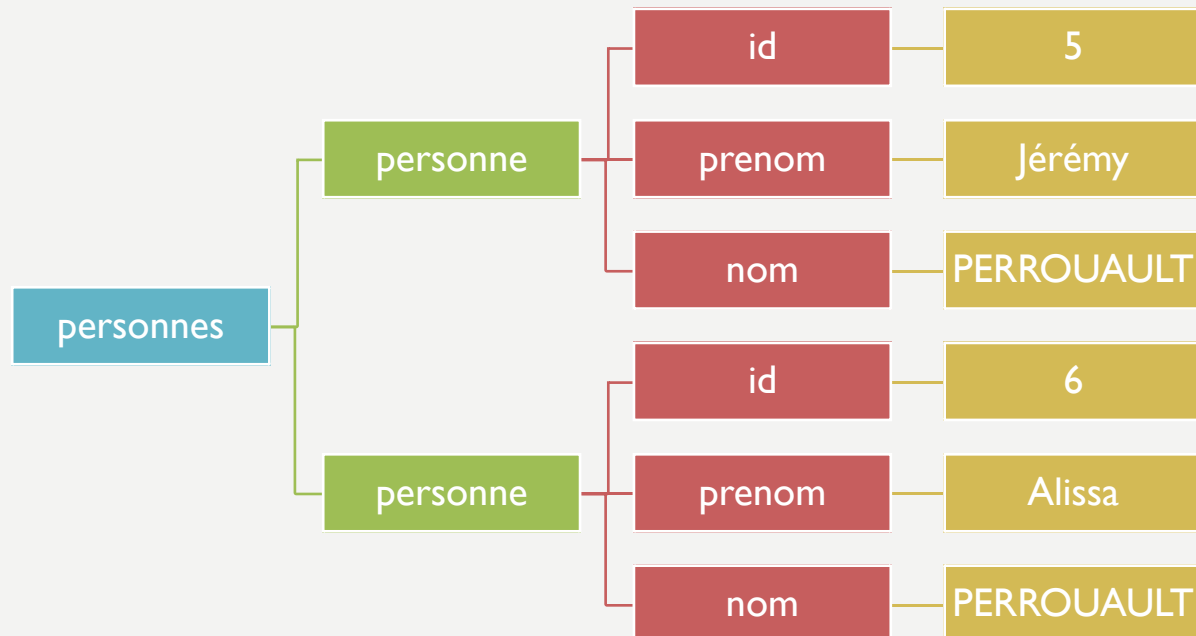
- Les objets se formalisent avec des accolades { }
- Les tableaux se formalisent avec des crochets []
- Le nom de l'élément est suivi de 2 points (:) et de sa valeur
- Les éléments sont séparés par des virgules (sauf pour le dernier)

```
{  
  "nom": "PERROUAULT",  
  "prenom": "Jérémy"  
}
```

```
{  
  "nom": "PERROUAULT",  
  "prenom": "Jérémy",  
  "emails": [  
    { "adresse": "jeremy.perrouault@ascadis.fr" },  
    { "adresse": "jeremy.perrouault@gmail.com" }  
  ]  
}
```


FORMALISME – EXERCICE

- Reprenons le schéma précédent
 - Le traduire en **JSON**
 - **personnes** est un tableau de **personne**



FORMALISME – EXERCICE

- Traduire en JSON
 - 2 personnes (la deuxième est inactive et n'a pas de numéro de téléphone)
 - id
 - active (O/N)
 - nom
 - prénom
 - telephone
 - adresses (une liste, même s'il n'y a qu'une seule adresse)
 - rue
 - code postal
 - ville



API

MANIPULATION GOOGLE API

API

- **JSON** est largement utilisé dans les **API** (**A**pplication **P**rogramming **I**nterface) Web
 - Un service web rendu par une société / une application
 - Consommable par une autre application

API – EXERCICE

- Utiliser JSON Place Holder
 - <https://jsonplaceholder.typicode.com/posts>
 - <https://jsonplaceholder.typicode.com/albums>
 - <https://jsonplaceholder.typicode.com/photos>
- Retrouver le titre de l'album de la photo #15

API – EXERCICE

- Rechercher la liste des pizzerias à NANTES (avec un navigateur)
 - En utilisant *Google Places API* (Il vous faudra une clé d'API)
 - Clé d'API `AlzaSyBiAmyKDEL6lhU7af_oCPzDWFDYzp0y9wg`
 - URL de recherche <https://maps.googleapis.com/maps/api/place/textsearch/json>
 - Requête pour restaurants à Sydney `?query=restaurants+in+Sydney&key=cle`
 - Lire et interpréter le flux JSON de *Google Places API*
 - Retrouver le nom d'une pizzeria et son adresse
 - Afficher ses photos (si disponible)