

JSON

JavaScript Object Notation

Table des matières

Présentation.....	1
Les objets JSON.....	3
Les tableaux JSON.....	4
PHP et JSON.....	6
Ouvrir et récupérer le contenu d'un fichier json.....	7
Convertir les objets JSON en variables PHP.....	7

Présentation

JSON pour « JavaScript Object Notation » est un format léger l'échange de données entre applications. Par exemple, le contenu d'un fichier JSON peut être alimenté par un extract d'une base de données et utilisé d'un autre côté par une application web.

L'avantage de cette méthode est qu'il n'est pas nécessaire d'établir une connexion à un serveur de base de données pour accéder à une donnée. Les données sont accessibles par le biais du fichier JSON.

Le fichier JSON est basé sur un format texte complètement indépendant d'un langage de programmation, mais la syntaxe est très proche de ce que connaissent les développeurs. En effet, le format JSON est basé sur le principe du couple ou de paire clé/valeur.

C'est typiquement ce que l'on retrouve avec les tableaux associatifs en PHP.

Le format JSON est décrit dans le RFC 8259 et disponible ici : <https://www.rfc-editor.org/rfc/rfc8259.html>

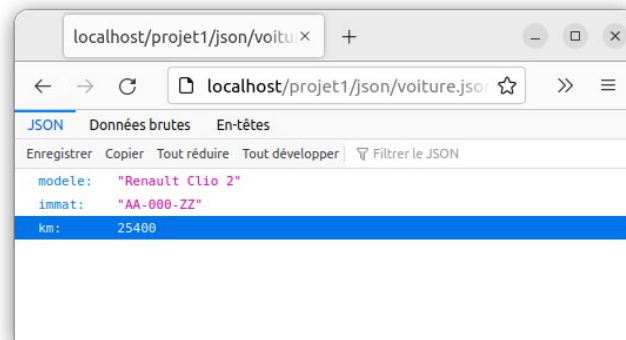
Le principe est très simple:

- Un objet JSON un ensemble de clé/valeur
- Un objet JSON commence par { et se termine par }
- Chaque nom/valeur est séparée par une virgule

Voici un exemple :

```
{ "modele" : "Renault Clio 2",  
  "immat" : "AA-000-ZZ",  
  "km" : 25400  
}
```

Recopiez le contenu JSON et enregistrez-le sous le nom « vehicules.json » Ouvrez votre fichier « vehicules.json » avec un navigateur web.



A retenir

Un objet, qui est un ensemble de couples **nom/valeur non ordonnés**.

- Un objet **commence par une accolade gauche {** et **se termine par accolade droite }**.
- Un **nom** est séparé d'une **valeur** par deux-points (:)
- Les couples nom/valeur sont séparés par une virgule.

Une valeur JSON peut être soit un objet, un tableau, un nombre ou une chaîne de caractères. JSON autorise les valeurs booléennes et null à la condition qu'ils soient notés en minuscules.

Voici le récapitulatif des types de valeur autorisé avec le format JSON :

- un objet
- un tableau
- une chaîne de caractères
- une valeur
- "true"
- "false"
- "null"

Les objets JSON

Un objet est représenté par une paire de crochets encadrant zéro ou plusieurs paires nom/valeur que l'on appelle des membres. Le nom d'un objet doit être une chaîne de caractère unique. Les deux-points qui viennent après chaque nom, séparant le nom de la valeur. Une seule virgule sépare une valeur d'un nom suivant. Les noms dans un objet DOIVENT être uniques.

```
{ "Renault": {  
    "modele" : "Renault Clio 2",  
    "immat" : "AA-000-ZZ",  
    "km" : 25400  
  }  
}
```

- 1) Pour commencer, créez un répertoire json à la racine de votre serveur apache.
- 2) Recopiez l'exemple ci-dessus et enregistrez-le dans un fichier vehicules.json dans le répertoire json précédemment créé.
- 3) Ouvrez votre navigateur et saisissez l'url pour accéder au fichier véhicules.json depuis votre navigateur web.

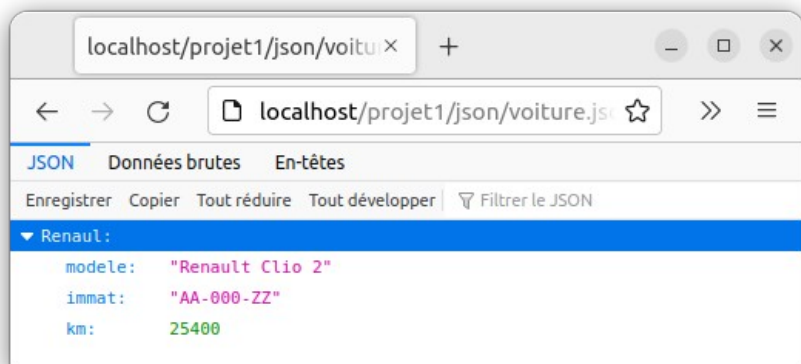
Un objet dont tous les « noms » sont uniques est interopérable, car tout logiciel recevant cet objet sera capable de faire correspondre les associations entre les noms et les valeurs.

En revanche, lorsque les noms d'un objet ne sont pas uniques, le comportement du logiciel qui reçoit un tel objet peut devenir imprévisible.

Ceci peut se manifester par le signalement des doublons, une erreur ou l'échec dans l'analyse des nom/valeur. Tout dépendra de comment a été implémenté JSON par l'application en question.

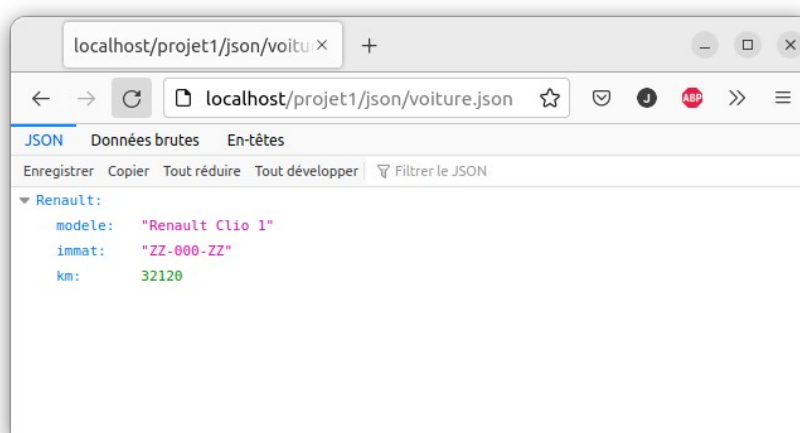
Faisons un essai avec le navigateur Firefox. Ajoutez un autre objet mais avec le même nom :

```
{  
  "Renault": {
```

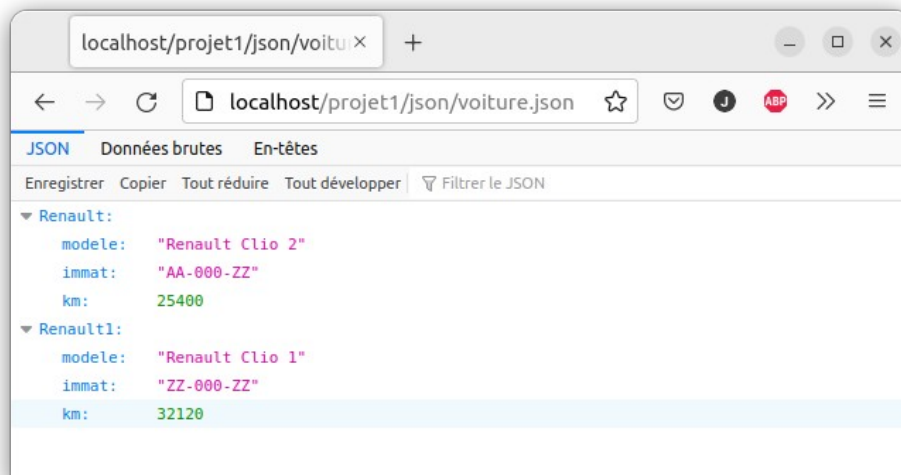


```
{
  "modele" : "Renault Clio 2",
  "immat" : "AA-000-ZZ",
  "km" : 25400
},
"Renault": {
  "modele" : "Renault Clio 1",
  "immat" : "ZZ-000-ZZ",
  "km" : 32120
}
}
```

Pour mon cas, Firefox affiche uniquement le dernier objet portant le nom « Renault ».



Pour corriger cela, renommez le dernier objet en « renault1 » par exemple.



Cette fois-ci, les deux objets ont bien été mappés par Firefox.

Testons autre chose. Au sein d'un même objet, utilisez cette fois-ci deux fois le même nom. Par exemple remplacer « immat » par « modele ».

Résultat, cette valeur n'a pas été mappée.

La règle à respecter est donc 1 objet = 1 nom unique

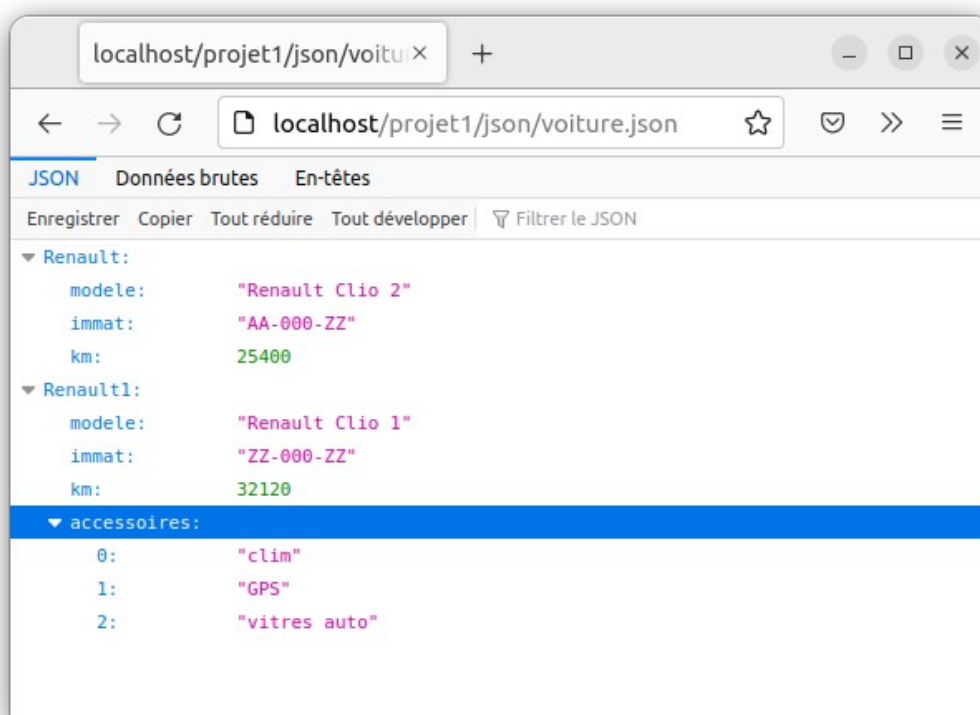
Les tableaux JSON

Un tableau est une collection de valeurs ordonnées. Un tableau commence par une ouverture de crochet [et se termine par une fermeture de crochet]. Les valeurs à l'intérieur du tableau sont séparées par une virgule.

Ajoutez un tableau à l'un des objets pour recenser les accessoires disponible d'un véhicule, par exemple

```
"accessoires": ["clim", "GPS", "vitres auto"]
```

Remarquez que Firefox génère tout seul l'index du tableau.



Revenons un instant sur le problème de mappage que l'on vu précédemment avec les noms d'objet. Pour conserver le 2ème objet « Renault » nous avons été obligés de le renommer. Pas très pratique ...

Imaginons que l'on souhaite malgré tout conserver un seul nom d'objet « Renault » (il peut y avoir d'autre marque mais on souhaite avoir un seul objet portant ce nom. Est-ce que les tableaux pourraient nous aider à résoudre ce point ?

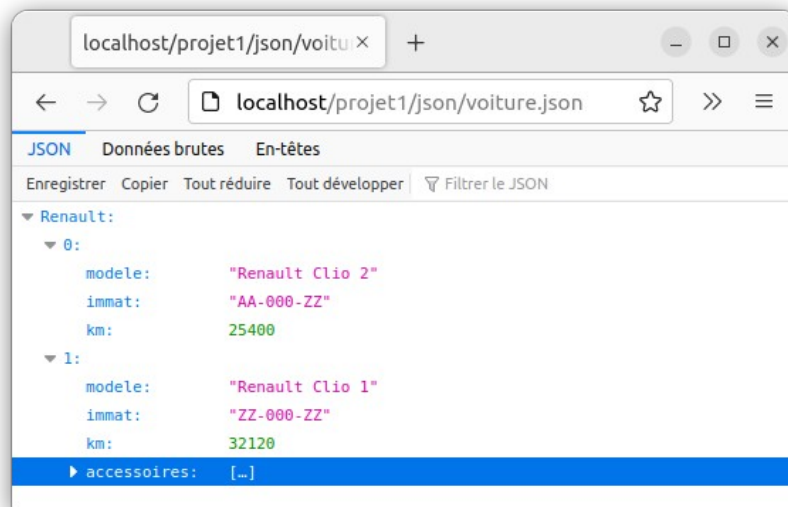


Figure 1: Modification à réaliser

Et si l'on stockait nos objets représentant les véhicules dans un tableau, qu'est-ce que cela donnerait ?

A vous de jouer.

Travail à rendre

1. Réalisez les modifications sur le fichier `vehicules.json` pour qu'il corresponde l'imprimé écran « Figure 1 ».
2. Créer un fichier json qui stocke les coordonnées de Pierre, Paul et Jacques. Vous devez y renseigner le prénom, l'email et le téléphone

PHP et JSON

Comme on l'a vu, JSON a été conçu pour l'échange de données. Ici nous allons utiliser un fichier JSON comme source de données pour une application PHP.

Notez qu'il n'est pas nécessaire d'installer quoi que ce soit pour manipuler un fichier JSON avec PHP. Ces fonctions font partie intégrante du cœur de PHP.

Voici quelques fonctions intéressantes pour manipuler un fichier JSON:

- **json_decode** — Récupère une valeur JSON et de la convertie en une variable PHP.
- **json_encode** — Renvoie la représentation JSON d'une valeur ou d'une variable

- **file_get_contents** : Permet d'ouvrir un fichier (JSON au hasard) et de stocker son contenu dans une variable
- **json_last_error_msg** — Retourne le message de la dernière erreur survenue lors de l'appel à la fonction `json_encode()` ou `json_decode()`

Commençons par le début.

Ouvrir et récupérer le contenu d'un fichier json

Pour cela PHP met à votre disposition la fonction `file_get_contents`. Il suffit de fournir en argument l'emplacement du fichier à lire à cette fonction.

```
<?PHP
$fichier = file_get_contents('vehicules.json');
var_dump($fichier);
?>
```

Notez qu'ici le fichier « véhicule.json » se trouve dans le même répertoire que le fichier php.

La réponse de `var_dump` nous indique que le contenu de `$fichier` est une chaîne de 222 caractères.

```
string(222) "{ \"Renault\": [{ \"modele\" : \"Renault Clio 2\",
\"immat\" : \"AA-000-ZZ\", \"km\" : 25400 }, { \"modele\" : \"Renault Clio
1\", \"immat\" : \"ZZ-000-ZZ\", \"km\" : 32120, \"accessoires\":
[\"clim\", \"GPS\", \"vitres auto\"] } ] } "
```

Comprenez bien que cette chaîne de caractères n'est pas reconnu à ce stade comme étant une donnée structurée au format JSON. C'est une simple suite de caractères stockée dans une variable. La fonction `file_get_contents` nous a permis uniquement d'accéder au contenu d'un fichier. Rien de plus.

Doc PHP

Pour plus d'information sur l'utilisation `file_get_contents()` :
<https://www.php.net/manual/fr/function.file-get-contents.php>

Convertir les objets JSON en variables PHP

Maintenant que nous avons accès au contenu du fichier JSON, nous devons convertir cette chaîne de caractères en objet JSON. C'est-à-dire, demander à PHP d'identifier les objets JSON à partir de cette chaîne de caractères, puis de les convertir en tableaux associatifs. Cette fonctionnalité est réalisée par la fonction PHP `json_decode()`.

Doc PHP

Pour information `json_decode()` permet également de convertir les objets JSON en objet PHP. Nous n'aborderons pas ce point dans ce TD. Pour plus d'informations :

<https://www.php.net/manual/fr/function.json-decode.php>

Pour convertir automatiquement les objets JSON de notre fichier en tableaux PHP nous allons :

1. Créer une nouvelle variable pour stocker ces tableaux
2. Utilisez la fonction `json_decode()`
3. Indiquer à `json_decode` que l'on souhaite convertir tous les objets JSON en tableaux associatifs

Variable utilisée pour récupérer les tableaux associatifs

Données JSON brut à convertir

Indique à `json_decode` que l'on souhaite tous convertir en tableaux

`$tableau = json_decode($fichier), true;`

Faisons l'essai

```
<?php
$fichier = file_get_contents('voiture.json');
$tableau = json_decode($fichier,true);
print_r($tableau);
?>
```

Plusieurs choses sont à noter :

1. On a utilisé `print_r` au lieu de `var_dump`. Il est souvent plus lisible que `var_dump`.
2. On peut lire que nous avons un tableau associatif dont la clé est « Renault », ce tableau contient deux tableaux indexés. Chaque tableau indexé contient des clé/valeur dont une des valeurs est un tableau.

```
Array (
    [Renault] => Array (
        [0] => Array ( [modele] => Renault Clio 2 [immat] => AA-000-ZZ [km] => 25400 )
        [1] => Array ( [modele] => Renault Clio 1 [immat] => ZZ-000-ZZ [km] => 32120 [accessoires] => Array ( [0] => clim [1] => GPS [2] => vitres auto ) )
    )
)
```

3. La conversion de notre chaîne de 222 caractères a correctement fonctionné. Nous avons maintenant des tableaux qui ont été générés à partir l'analyse du fichier au format JSON.

A partir de ce point nous pouvons utiliser le mécanisme de PHP pour accéder aux donnée d'un tableau associatif, comme par exemple appeler la clé « Renault », puis l'index 1, puis le nom « immat » pour accéder à cette donnée.

```
<?php
$fichier = file_get_contents('voiture.json');
$tableau = json_decode($fichier,true);
echo"<BR>";echo $tableau["Renault"][1]["immat"];
?>
```

Travail à rendre

3. Faites une page PHP qui affiche la liste des immatriculations à partir du fichier concessionnaire.json (disponible sur moodle).
4. Faites une page PHP qui affiche le modèle, l'immatriculation et le km d'un véhicule à partir du concessionnaire.json.
5. Faites une page PHP qui affiche toutes les données disponibles pour le modèle de véhicule de votre choix.
6. Réalisez un formulaire :
 - a) Votre formulaire devra être en mesure d'afficher une liste déroulante qui contient l'ensemble des modèles de véhicule présent dans le fichier concessionnaire.json
 - b) puis dans un second temps après la validation du formulaire, d'afficher correctement l'ensemble des données que l'on possède sur celui-ci

Petit indice : Pour votre liste de véhicule à sélectionner ; Rien ne vous interdit de réaliser une boucle afin de générer les balises <option value="volvo">Volvo</option> de l'élément <SELECT ...> ... Après, moi je dis ça ...