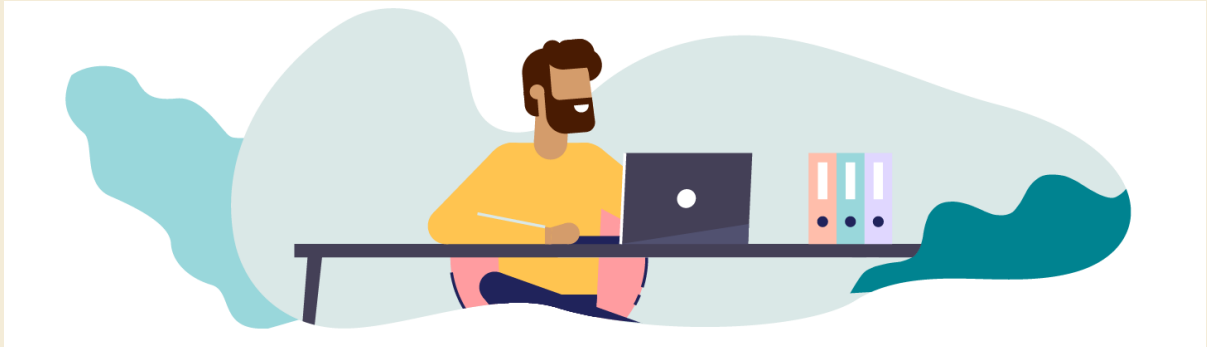


# Exploitez des données au format JSON

6–8 minutes



Vous avez maintenant un plan d'action : utiliser des données au format JSON issues du site actuel des Bonnes Pièces. Vous souhaitez cependant mettre à jour ces données avec trois nouvelles informations sur les produits :

- un lien vers l'image du produit ;
- la description et la disponibilité du produit.

Pour y parvenir, vous devez être capable de comprendre le format JSON afin de lire et modifier les données et leur structure.

## Découvrez la syntaxe JSON

Comme son nom l'indique, le JSON (JavaScript Object Notation) est issu du langage JavaScript. C'est un format plus limité car il est utilisé par des programmes écrits dans d'autres langages, comme Python, le C ou le PHP. Il a donc vocation à s'adapter à cette diversité d'environnements.

Il dispose de :

- 4 types primitifs : les nombres, les textes, les booléens, la valeur null ;
- 2 types structurants : les objets et les tableaux.

## Les nombres

Les nombres s'écrivent avec les chiffres de zéro (0) à neuf (9). Ils peuvent inclure une partie décimale, séparée par un point, à droite de la partie entière. Enfin, le signe moins (-) placé à gauche du premier chiffre vous permet de rendre un nombre négatif.

Par exemple :

```
1805  
6.28  
-9.81
```

## Les textes

Les textes sont constitués d'une série de caractères. Dit simplement, un caractère correspond à une touche sur votre clavier. Pour savoir où un texte commence et où il finit, vous devez le délimiter à l'aide de guillemets "".

Par exemple :

```
"Un texte en JSON"  
"Le double guillemet \" délimite un texte"
```

(Par exemple :

Sans backslash :

```
"Le symbole guillemet " délimite une chaîne de caractères"
```

Avec backslash :

```
"Le symbole guillemet \" délimite une chaîne de caractères"
```

## Les booléens et la valeur null

Les booléens comportent deux valeurs : true et false. Elles permettent de représenter une information de vérité : vrai ou faux.

Par exemple, est-ce qu’une pièce automobile est en stock ? Si oui, alors nous utiliserons la valeur `true`. Sinon, alors ce sera la valeur `false`.

```
true  
false
```

La valeur **`null`** permet d’indiquer l’absence de valeur.

Par exemple, une pièce automobile vient d’être ajoutée à notre page, mais l’équipe Marketing n’a pas encore eu le temps de la prendre en photo. En l’absence d’image du produit, on utilisera la valeur `null` :

```
null
```

## Les tableaux

Un tableau est représenté avec une paire de crochets ouvrant “[” et fermant “]”. Tous les éléments s’écrivent à l’intérieur de ces crochets. Lorsqu’il y a plusieurs éléments, vous devez les séparer à l’aide de virgules. Si un tableau ne possède pas encore d’éléments, il sera vide.

```
[]  
["un"]  
[0, 1, 2]
```

## Les objets

Les objets sont délimités par des accolades ouvrante “{” et fermante “}”. Les informations qu’elles contiennent sont identifiées par des propriétés. Ces dernières doivent être délimitées par des doubles guillemets, suivis du symbole deux points “:” avant de laisser place à la valeur concrète.

Par exemple, pour déclarer un objet qui représente un utilisateur du

site web des Bonnes Pièces, qui s'appelle Louis et qui a 19 ans, vous devez écrire :

```
{ }  
  
{ "prenom": "Louis" }  
  
{ "prenom": "Louis", "age": 19 }
```

## Modifiez vos données au format JSON

Nous avons fait connaissance avec le langage JSON, utilisons-le maintenant pour modifier les données de nos pièces automobiles.



Une pièce automobile est composée de plusieurs informations, que nous représenterons ainsi :

- son nom, avec le type texte ;
- son prix, avec le type nombre ;
- et sa catégorie, également avec le type texte.

Ces informations forment un tout cohérent. Nous allons donc les structurer dans un **objet**.

Par exemple, pour structurer l'objet "Ampoule LED" avec un prix de 60 euros dans la catégorie Optiques, nous écrirons :

```
{  
  "nom": "Ampoule LED",  
  "prix": 60,  
  "categorie": "Optiques"  
}
```

Nous souhaitons maintenant ajouter une information : un lien vers une image de la pièce, de type texte.

Pour effectuer cette modification, nous devons d'abord identifier à quel endroit du fichier rajouter cette information. Étant donné que chaque pièce aura un lien vers une image, il faudra modifier

chaque objet de chaque pièce. Ici, nous ajouterons donc une propriété intitulée *image* à la suite de la propriété *categorie*.

Pour renseigner la valeur de la propriété, nous devons utiliser les images qui se situent dans le dossier 'images' de notre projet. Nous procédons ainsi de la manière suivante :

```
{  
  "nom": "Ampoule LED",  
  "prix": 60,  
  "categorie": "Optiques",  
  "image": "images/ampoule-led.png"  
}
```

Dans le fichier pieces-auto.json, nous avons plusieurs pièces et donc plusieurs objets. Ces derniers sont **regroupés** pour former un **tableau** à l'aide des crochets ouvrant [ et fermant ].

```
[  
  {  
    "nom": "Ampoule LED",  
    "prix": 60,  
    "categorie": "Optiques"  
  },  
  {  
    "nom": "Plaquettes de frein (x4)",  
    "prix": 40,  
    "categorie": "Freinage"  
  }  
]
```

**Récapitulons en vidéo**

Vous pouvez revoir les différentes étapes de cette démonstration dans la vidéo ci-dessous :

## À vous de jouer

À votre tour maintenant !

Depuis la branche GitHub [P1C3-Exercice](#), éditez le fichier `pieces-autos.json` et ajoutez deux informations pour chaque pièce automobile :

- Une propriété '*description*' :
- Ampoule LED : "Distance d'éclairage : 100 mètres !".
- Plaquettes de frein (x4) : "Une qualité de freinage optimale, par tous les temps."
- Ampoule boîte à gants : "Pour y voir clair dans l'habitacle."
- Liquide de frein : *Aucune description, ne pas rajouter de propriété.*
- Balai d'essuie-glace : "Performances d'essuyage au top ! Longueur : 550 mm."
- Une propriété '*disponibilite*' pour la disponibilité :
- Ampoule LED : oui.
- Plaquettes de frein (x4) : oui.
- Ampoule boîte à gants : non.
- Liquide de frein : oui.
- Balai d'essuie-glace : oui.

## Corrigé

Vous pouvez vérifier votre travail en consultant la branche [P1C3-Solution](#) et en regardant la vidéo ci-dessous :

## En résumé

- Le format JSON permet de décrire vos données sur l'aspect :

- de leur structure même ;
- des valeurs concrètes de vos informations.
- Il supporte des types :
  - primitifs (nombres, textes, booléens et la valeur null) ;
  - structurants (des objets pour structurer plusieurs informations, et des tableaux pour regrouper plusieurs informations).
- Grâce au format JSON, vous pouvez facilement créer et/ou modifier des données.

*Et voilà, vous savez désormais lire, écrire et modifier des données au format JSON ! Prochaine étape : générer une page web à partir de ces données !*