



JSON

FONDAMENTAUX



DÉBUTANT



SOMMAIRE

Introduction

Mise en place de l'environnement

Syntaxe & Conventions

Outils

JavaScript : Lecture

JavaScript : Écriture

API

JSON Server

Aller plus loin



INTRODUCTION



JSON

JAVASCRIPT OBJECT NOTATION



JavaScript Object Notation est un format de données textuelles dérivé de la notation des objets du langage JavaScript.

```
version: 1.0.0
description: ""
main: "index.js"
scripts: {
  start: "electron .",
  dev: "rollup -c -w",
  build: "rollup -c"
},
keywords: [],
author: "",
license: "ISC",
devDependencies: {
  electron: "8.2.1",
  electron-reload: "1.5.0",
  concurrently: "5.1.0",
  "@rollup/plugin-commonjs": "11.0.0",
  "@rollup/plugin-node-resolve": "7.0.0",
  rollup: "1.20.0",
  rollup-plugin-livereload: "1.0.0",
  rollup-plugin-svelte: "5.0.3",
  rollup-plugin-terser: "5.1.2",
  svelte: "3.21.0"
},
dependencies: {
  cron: "1.8.2",
  node-notifier: "7.0.0",
  rollup-plugin-scss: "2.4.0"
}
```



ENJEUX

POURQUOI JSON ?

Aujourd'hui, toutes les entreprises ont un système d'information et énormément de données sont stockées de manière hétérogène. Bien connaître et manipuler le format JSON constitue une corde en plus à l'arc de tout bon ingénieur informatique.

JSON

JavaScript Object Notation

- ✓ Très populaire
- ✓ Simple & Léger
- ✓ API REST

```
{  
  "user": {  
    "id": 1,  
    "firstname": "John",  
    "lastname": "Doe",  
    "elo_points": 1080  
  }  
}
```

EXEMPLE D'UN OBJET EN JSON





MISE EN PLACE DE L'ENVIRONNEMENT

CHOISIR UN NAVIGATEUR

OÙ NOTRE CODE VA S'EXÉCUTER ?



Firefox

Mozilla Corp.



Opera

Opera Software



Google Chrome

Google



Microsoft Edge

Microsoft



Veuillez éviter Microsoft Edge...

Les outils développeurs ainsi que le rendu de certains éléments ne sont pas à jour.



CHOISIR UN IDE / ÉDITEUR DE TEXTE

ENVIRONNEMENT DE DÉVELOPPEMENT



WebStorm
JetBrains

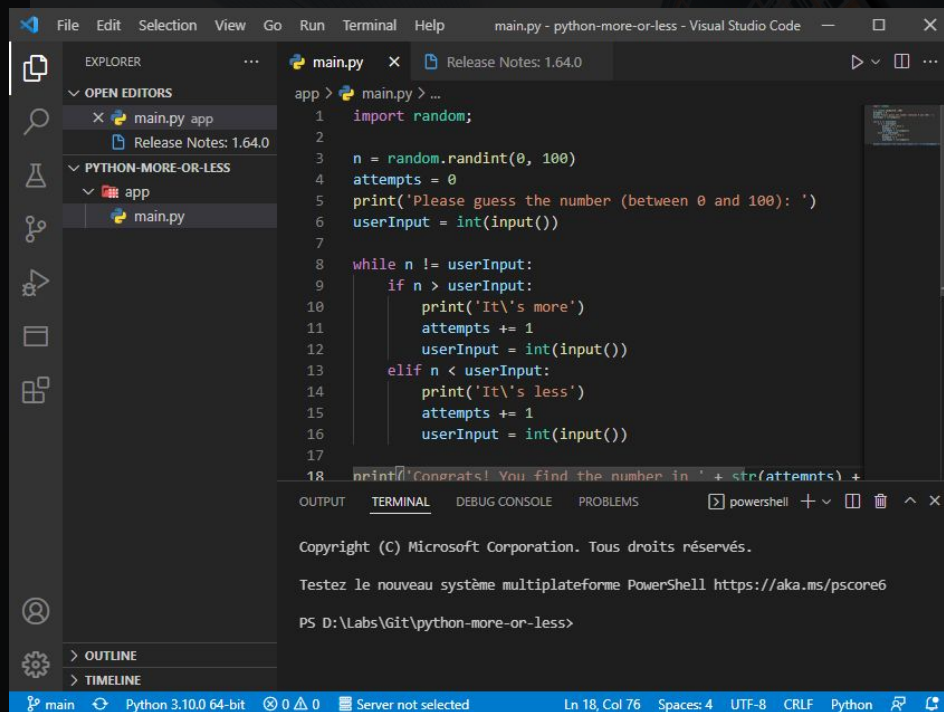


Visual Studio Code
Microsoft

<https://code.visualstudio.com/>

VISUAL STUDIO CODE

INTERFACE & RACCOURCIS



Gratuit

Système d'extensions

Connecté au système de versionning

Le plus populaire

Et bien plus...



NODE JS

RUNTIME JAVASCRIPT



Node JS
OpenJS Foundation

<https://nodejs.org/fr/>



Lors de l'installation de Node.js, veillez à bien ajouter le chemin parmi les variables d'environnement système.

```
node -v  
v19.01.0
```

--

TERMINAL / CMD

VÉRIFICATION DE L'INSTALLATION

NodeJS est un runtime JavaScript basé sur le moteur V8 de Google Chrome.
Il va nous être nécessaire pour avoir npm qui sera notre gestionnaire de packages / dépendances.

npm

EXTENSION CODE RUNNER

CÔTÉ PRATIQUE

.run

Code Runner

Jun Han



JavaScript

PHP

Python

TypeScript

Et bien plus...

Une fois l'extension installée, il suffira de sélectionner dans le fichier JavaScript le code à exécuter, et cliquer sur run code.



MDN

DOCUMENTATION & AUTONOMIE

Le travail d'un développeur est de résoudre des problèmes. Une des qualité nécessaire est de savoir comment trouver la bonne information sur la toile.

Pour les langages HTML, CSS et JavaScript, le site mdn fournit une documentation complète et détaillée.



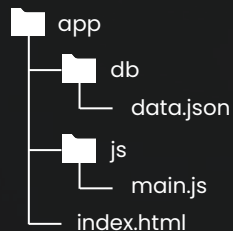
<https://developer.mozilla.org/fr/>



CRÉATION DU PROJET

KICK START JSON !

json-course



1. Créer et ouvrir le dossier json-course à l'aide de VS. Code.
2. Créer un dossier app.
3. Créer un fichier index.html dans le dossier app.
4. Créer dans le dossier db, le fichier data.json.
5. Créer dans le dossier js, le fichier main.js.



INDEX.HTML

UTILISER EMMET POUR LE MODÈLE DE DÉPART



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

index.html

L'extension Emmet est directement intégrée à Visual Studio Code.

Cette extension permet de créer très rapidement une structure HTML composée de plusieurs balises.

Raccourci Emmet : ! +



INTÉGRATION DU JAVASCRIPT

LA BALISE SCRIPT



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-...>
  <title>Document</title>
</head>
<body>

  <!-- Javascript -->
  <script src="js/main.js"></script>

</body>
</html>
```

index.html

La balise "<script>...</script>" permet de lier un fichier JavaScript à une page HTML.

Dans l'exemple ci-contre, le chemin relatif vers fichier JS est renseigné dans l'attribut "src"



SYNTAXE & CONVENTIONS

REPRÉSENTATION

LE JSON, À QUOI ÇA RESSEMBLE ?

Dans l'exemple ci-contre, nous remarquons que les objets se déclarent à l'aide d'accolades "{ ... }" et les tableaux à l'aide de crochets "[...]".

En JSON, les clés des objets sont toujours entourées par des guillemets, en anglais "double quotes" à la différence d'un objet en JavaScript.

```
{
  "cars": [
    {
      "brand": "Kia",
      "model": "Sportage",
      "year": 2016
    },
    {
      "brand": "Mazda",
      "model": null,
      "year": 2011
    }
  ]
}
```

COMMENTAIRE

C'EST SIMPLE, C'EST INTERDIT



Même si la coloration syntaxique colore en vert les commentaires sur VS. Code. Il est strictement interdit d'utiliser les commentaires dans un fichier JSON.

```
{
  "cars": [
    {
      "brand": "Kia",
      "model": "Sportage",
      "year": 2016
    },
    {
      "brand": "Mazda", // Marque
      "model": null,
      "year": 2011
    }
  ]
}
```


ÉLÉMENT RACINE UNIQUE

STRICTEMENT UN SEUL

Un document JSON est valide s'il contient strictement un seul élément racine. Cet élément racine peut être un tableau ou un objet.

Il est recommandé d'avoir un objet en élément racine. Avec la clef, on sait de quoi il s'agit.

[Cf. L'exemple correct ci-contre.]

```
{
  "cars": [
    {
      "brand": "Kia",
      "model": "Sportage",
      "year": 2016
    },
    {
      "brand": "Mazda",
      "model": null,
      "year": 2011
    }
  ]
}
```



```
{
  "brand": "Kia",
  "model": "Sportage",
  "year": 2016
}
{
  "brand": "Mazda",
  "model": null,
  "year": 2011
}
```



TYPES DE DONNÉES

NUMBER

```
let age = 20;  
typeof age;  
// returns number
```

NOMBRE ENTIER OU FLOTTANT

STRING

```
let firstName = 'John';  
typeof firstName;  
// returns string
```

CHAÎNE DE CARACTÈRES

BOOLEAN

```
let isOpen = false;  
typeof isOpen;  
// returns boolean
```

VRAI OU FAUX

NULL

```
let result = null;  
typeof result;  
// returns object
```

ABSENCE DE VALEUR / VIDE

A l'intérieur d'un fichier JSON, on retrouve les mêmes types de données que dans le langage JavaScript sauf pour le type "undefined".

TABLEAU

COLLECTION DE VALEURS



L'index d'un tableau est un nombre entier qui commence toujours à 0.

JSON

```
[  
  "banana",  
  "apple",  
  "orange",  
  "mandarin"  
]
```

fruits.json

"banana"	"apple"	"orange"	"mandarin"	Valeurs
0	1	2	3	Indices / Indexes



MANIPULER DES TABLEAUX

JAVASCRIPT

JS

#	EXEMPLES	RÉSULTATS
Ajouter un élément	<code>[3, 26].push(14);</code>	<code>[3, 26, 14]</code>
Modifier un élément	<code>let numbers = [5, 14, 7, 6];</code> <code>numbers[2] = 21;</code> <code>console.log(numbers);</code>	<code>[5, 14, 21, 6]</code>
Supprimer un élément	<code>let numbers = [5, 14, 7, 6];</code> <code>numbers.splice(2, 1);</code> <code>console.log(numbers);</code>	<code>[5, 14, 6]</code>
Trier	<code>[3, 9, 7].sort();</code>	<code>[3, 7, 9]</code>
Taille	<code>["Bonjour", "Bonsoir", "Bye"].length;</code>	<code>3</code>
Concaténer	<code>[2, 1].concat([5, 7]);</code>	<code>[2, 1, 5, 7]</code>
Élément depuis l'index	<code>["Bonjour", "Bonsoir", "Bye"].at(1);</code>	<code>"Bonsoir"</code>

OBJET

COLLECTION DE PAIRES CLEF/VALEUR

JSON

```
{
  "fruits": [
    {
      "name": "banana",
      "colors": ["yellow"]
    },
    {
      "name": "apple",
      "colors": ["green", "red", "yellow"]
    },
    {
      "name": "orange",
      "colors": ["orange"]
    },
    {
      "name": "mandarin",
      "colors": ["orange"]
    }
  ]
}
```

fruits.json

Les objets sont très utiles pour représenter les éléments du monde réel. En JSON, ils se déclarent à l'aide d'accolades et les clefs sont entourées de guillemets obligatoires dit "double quotes" en anglais.



MANIPULER DES OBJETS

JAVASCRIPT

JS

```
let car = {  
  brand: "Kia",  
  model: "Rio",  
  color: "Black",  
  year: 2013  
};
```

#	EXEMPLES	RÉSULTATS
Tableau des clefs	<code>Object.keys(car);</code>	<code>['brand', 'model', 'color', 'year']</code>
Tableau des valeurs	<code>Object.values(car);</code>	<code>['Kia', 'Rio', 'black', 2013]</code>
Ajouter une propriété	<code>car.doors = 3;</code>	--
Modifier une propriété	<code>car.model = "Sportage";</code>	--
Supprimer une propriété	<code>delete car.color;</code>	--



TRAVAUX PRATIQUES



CRÉER UN FICHER JSON

A l'aide de tableaux et d'objets, trouver le meilleur moyen de stocker toutes les données ci-dessous dans un fichier JSON. [customers.json]

Client 1

Nom : Doe
Prénom : John
Age : 21
Email : john.doe@xyz.com
Hobbies : Karaté, Tennis

Client 2

Nom : Stewart
Prénom : Jane
Age : 26
Email : [Email non renseigné]
Hobbies : Danse, Peinture, Chant

Client 3

Nom : Tardieu
Prénom : Olivier
Age : 32
Email : olivier.tardieu@xyz.com
Hobbies : [Hobbies non renseignés]



En partant du document CSV ci-dessous, créer le fichier json équivalent, products.json.

Pour une meilleure lisibilité du document CSV, vous pouvez ouvrir le fichier products.csv avec un tableur, Google Sheet ou Excel.



```
id,name,price,rate
1,Lightbulb,3.79,4
2,Knife,8.52,3.8
3,Mug,9.99,4.2
4,Pen,15.23,4.7
5,Sweatshirt,26.51,4.6
```

products.csv



CSV VERS JSON

PART. I





OUTILS

JSON LINT

SAVOIR SI UN DOCUMENT JSON EST VALIDE

JSON

```
{
  "fruits": [
    {
      "name": "banana",
      "colors": ["yellow"]
    },
    {
      "name": "apple",
      "colors": ["green", "red", "yellow"]
    },
    {
      "name": "orange",
      "colors": ["orange"]
    },
    {
      "name": "mandarin"
      "colors": ["orange"]
    }
  ]
}
```

fruits.json

Grâce au site web jsonlint.com, vous pourrez aisément savoir si un document JSON est valide ou invalide et sachant pourquoi.

Validate JSON

Error: Parse error on line 15:

```
...ame": "mandarin"      "color": ["orange"]
-----^
```

Expecting 'EOF', '}', ':', ',', ']', got 'STRING'

JSON FORMATTER

LISIBILITÉ

JSON Formatter fournit un ensemble d'outils permettant de manipuler des formats de données textuelles tels que JSON et XML.

```
{"cars":[{"brand": "Kia","model":  
"Sportage","year": 2016},{"brand":  
"Mazda","model": null,"year": 2011}]}
```

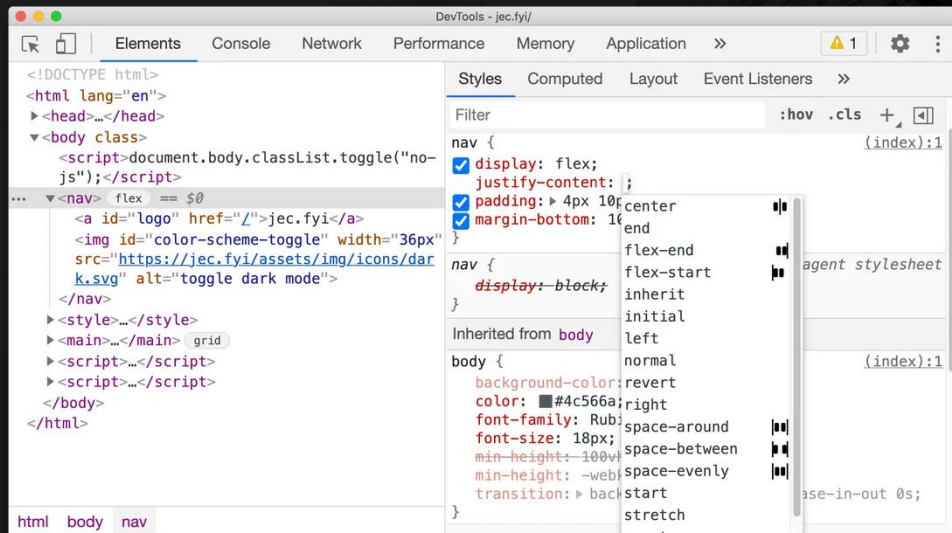
```
{  
  "cars": [  
    {  
      "brand": "Kia",  
      "model": "Sportage",  
      "year": 2016  
    },  
    {  
      "brand": "Mazda",  
      "model": null,  
      "year": 2011  
    }  
  ]  
}
```

→
FORMAT / BEAUTIFY



CHROME DEVTOOLS

F12 SUR LE NAVIGATEUR CHROME



DevTools



CHROME DEVTOOLS

F12 SUR LE NAVIGATEUR CHROME

NOM DE L'ONGLET

BRÈVE DESCRIPTION

ELEMENTS

Inspecter le code source HTML et le CSS associé

CONSOLE

Sortie standard du code JavaScript, c'est ce qui nous intéresse ici.

SOURCES

Consulter les différents fichiers sources, html, css, js et plus...

NETWORK

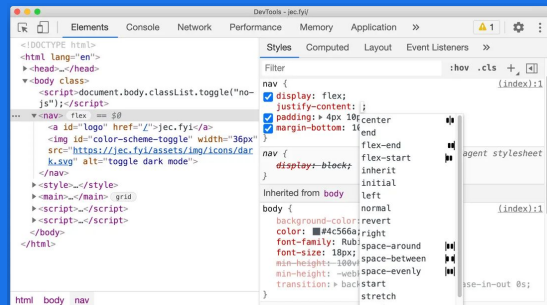
Vérifier le bon chargement des éléments du site.

PERFORMANCE

Monitoring, Benchmarking...

Il y a énormément de choses à dire sur cet outil, pour en savoir plus je vous invite à vous rendre sur le lien ci-dessus.

Dans ce cours, on utilisera le langage JavaScript et l'onglet "Console" pour lire et écrire dans un document JSON.



DevTools





JAVASCRIPT : LECTURE



CAS D'UN FICHIER DANS LE CLOUD

FONCTION FETCH

JS

```
let url = 'https://...'  
  
fetch(url)  
.then((response) => { return response.json() })  
.then((data) => {  
    // Affichage des données  
    console.log(data)  
})
```

js/main.js

Écrire le fichier "main.js" identique à l'exemple ci-contre, avec l'url ci-dessus affectée à la variable "url". Ouvrir le fichier index.html dans un navigateur et à l'aide du raccourci "F12", ouvrir DevTools. et aller dans l'onglet console.



CAS D'UN FICHER LOCAL

MODULE FS – READFILE

JSON

```
{
  "cars": [
    {
      "brand": "Kia",
      "model": "Sportage",
      "year": 2016
    },
    {
      "brand": "Mazda",
      "model": null,
      "year": 2011
    }
  ]
}
```

db/cars.json

JS

```
// Module File System
const fs = require('fs')

fs.readFile('app/db/cars.json', (err, data) => {
  if (err) throw err

  let dataStr = data.toString()
  let dataJSON = JSON.parse(dataStr)

  // Affichage des données
  console.log(dataJSON)
})
```

js/main.js

Une fois les fichiers cars.json et main.js identiques à l'exemple ci-contre, utiliser l'extension code runner et exécuter le code JavaScript.





CAS D'UN FICHER LOCAL

MODULE FS – READFILE

JS

```
// Module File System
const fs = require('fs')

fs.readFile('app/db/cars.json', (err, data) => {
  if (err) throw err

  let dataStr = data.toString()
  let dataJSON = JSON.parse(dataStr)

  // Affichage des données
  console.log(dataJSON)
})
```

js/main.js

La méthode "readFile" prend en paramètres le chemin du fichier à lire ainsi qu'une fonction callback dans laquelle on retrouve la variable "data".

Cette variable "data" correspond au contenu binaire du fichier, c'est pourquoi à l'aide de la méthode "toString", on transforme le binaire en chaîne de caractères.

Enfin, à l'aide de la méthode "parse" de la classe JSON, on transforme la chaîne de caractères en un objet JavaScript manipulable.



ACCÈS AUX DONNÉES

ACCESSEURS

JSON

```
{
  "cars": [
    {
      "brand": "Kia",
      "model": "Sportage",
      "year": 2016
    },
    {
      "brand": "Mazda",
      "model": null,
      "year": 2011
    }
  ]
}
```

db/cars.json

JS

```
// Première voiture
console.log(dataJSON.cars[0])

// Marque de la seconde voiture
console.log(dataJSON.cars[1].brand)
```

js/main.js

```
{ brand: 'Kia', model: 'Sportage'...
```

Mazda

SORTIE CONSOLE

ACCÈS AUX DONNÉES

ACCESSEURS

JS

```
// Première voiture
console.log(dataJSON.cars[0])

// Marque de la seconde voiture
console.log(dataJSON.cars[1].brand)
```

js/main.js



```
{ brand: 'Kia', model: 'Sportage'...
}
```

Mazda

SORTIE CONSOLE

En JavaScript, pour accéder à la valeur d'un tableau, on utilise les crochets "[]" en précisant l'indice / index à l'intérieur.

Pour accéder à la propriété d'un objet, on utilise le point "." suivi du nom de la propriété.



JAVASCRIPT : ÉCRITURE

CAS D'UN FICHIER DANS LE CLOUD

CE N'EST PAS POSSIBLE SANS UN SYSTÈME D'AUTHENTIFICATION

Pouvoir écrire dans un fichier JSON contenu dans le cloud sans authentification au préalable serait une faille de sécurité. Le format JSON est utilisé dans certains fichiers de configuration de sites web ou comme stockage de données.

STOP

Halte là ! Non je ne lancerai pas ton script JS me demandant de modifier un de mes fichiers JSON, si cela était possible, il y aurait une grosse faille de sécurité !

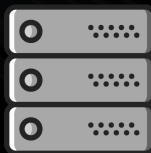


Client

Requête HTTP



index.html



Serveur



CAS D'UN FICHER LOCAL

MODULE FS - WRITEFILESYNC / SUPPRESSION DU SECOND UTILISATEUR

JSON

```
{
  "cars": [
    {
      "brand": "Kia",
      "model": "Sportage",
      "year": 2016
    },
    {
      "brand": "Mazda",
      "model": null,
      "year": 2011
    }
  ]
}
```

db/cars.json

JS

```
// Module File System
const fs = require('fs')

fs.readFile('app/db/cars.json', (err, data) => {
  if (err) throw err

  let dataStr = data.toString()
  let dataJSON = JSON.parse(dataStr)

  // Suppression de la seconde voiture
  dataJSON.cars.splice(1, 1)
  fs.writeFileSync('app/db/cars.json', JSON.stringify(dataJSON))
})
```

js/main.js



TRAVAUX PRATIQUES



LECTURE

À l'aide de la fonction `fetch`, lire le fichier JSON ci-dessus et afficher dans la console les informations suivantes :

1. Afficher la réponse de la deuxième question des "maths".
2. Afficher la troisième option de la première question de "sport".
3. Afficher l'intitulé de la première question de "sport".
4. Afficher l'intitulé de la deuxième question des "maths".

BONUS

Afficher la dernière option de la première question de "sport".

BONUS

Afficher le nombre d'options de la deuxième question de "maths".



Récupérer le fichier JSON ci-dessus en local, et faire les instructions suivantes.

1. Modifier la valeur de "sea_level" en 1156.
2. Supprimer la propriété "description" de "weather".
3. Trouver à quelle date correspond "sunrise" puis enlever 15 minutes.
4. Ajouter dans "main" la température en Celsius.

BONUS

Calculer dans le format hh:mm:ss, la durée de la journée.



ÉCRITURE





CSV VERS JSON

PART. II

A l'aide de ce que l'on a appris et de vos recherches, écrire un programme qui convertit products.csv en products.json.

JavaScript ou Python recommandé.



```
id,name,price,rate
1,Lightbulb,3.79,4
2,Knife,8.52,3.8
3,Mug,9.99,4.2
4,Pen,15.23,4.7
5,Sweatshirt,26.51,4.6
```

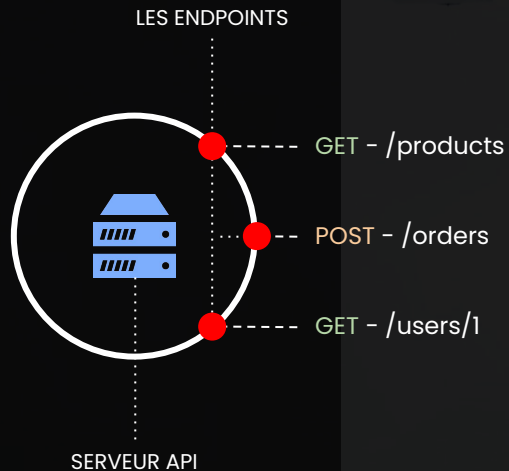
products.csv



API

FONCTIONNEMENT D'UNE API REST

API : APPLICATION PROGRAMMING INTERFACE



L'API est une solution informatique qui permet à des applications de communiquer entre elles et de s'échanger mutuellement des services ou des données.

Nous pouvons faire l'analogie avec une télécommande [outil de requête normalisé] et un téléviseur proposant différentes chaînes, l'API et ses endpoints.

ROUTES REST

CAS DE LA RESSOURCE ARTICLE

VERBE HTTP	URL - ROUTE	ACTION	DESC.
GET	/articles	Consulter	Liste tous les articles
GET	/articles/create	Consulter (<i>Créer</i>)	Affiche le formulaire de création d'articles
POST	/articles	Ajouter	Ajout de l'article après soumission du formulaire
GET	/articles/:id	Consulter	Affiche l'article spécifiée par l'id
GET	/articles/:id/edit	Éditer	Affiche le formulaire d'édition de l'article spécifiée par l'id
PATCH	/articles/:id	Modifier	Modifie l'article spécifiée par l'id
PUT	/articles/:id	Modifier	Remplace l'article spécifiée par l'id
DELETE	/articles/:id	Supprimer	Supprime l'article spécifiée par l'id



JSON PLACEHOLDER

COMPRENDRE LE FONCTIONNEMENT D'UNE API / FAUSSE API REST

ENDPOINTS

NOMBRE DE RESSOURCES

<https://jsonplaceholder.typicode.com/posts>

100 articles

<https://jsonplaceholder.typicode.com/comments>

500 commentaires

<https://jsonplaceholder.typicode.com/albums>

100 albums

<https://jsonplaceholder.typicode.com/photos>

5000 photos

<https://jsonplaceholder.typicode.com/todos>

200 todos

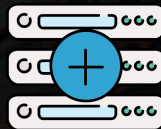
<https://jsonplaceholder.typicode.com/users>

10 utilisateurs

A noter que les ressources ont des relations, les articles ont des commentaires et les albums ont des photos.

JSON PLACEHOLDER





JSON SERVER



JSON-SERVER

AVOIR UNE FAUSSE API REST RAPIDEMENT

```
npm install -g json-server
```

1. Installer globalement json-server.

```
https://github.com/aunimhsn/json-server/blob/main/app/db.json
```

2. Créer le fichier db.json et suivre le lien pour avoir le contenu.

```
json-server --watch db/db.json
```

3. Lancer le serveur local de notre fausse API REST.

```
http://localhost:3000/cars
```

4. Ouvrir votre navigateur et se rendre sur le lien du serveur local.

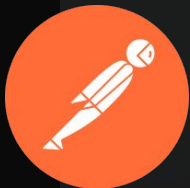
CLIENT REST

OUTIL INDISPENSABLE POUR TRAVAILLER AVEC LES APIs



Insomnia

Kong Inc.



Postman

Postman Inc.

Un client REST est un outil de développement et de débogage.

Il permet de personnaliser et garder en mémoire les requêtes envoyées à une API REST.



INSOMNIA

CLIENT REST

GET http://localhost:3000/cars/1

Send ▼

Preview ▼

```
--  
  
{  
  "id": 1,  
  "brand": "Kia",  
  "model": "Rio III",  
  "color": "white",  
  "year": 2014  
}
```



INSOMNIA



ALLER PLUS LOIN

ALLER PLUS LOIN

CONTINUER D'APPRENDRE



Le format de données XML



NoSQL avec MongoDB



Le langage JavaScript



La bibliothèque TinyDB en Python



MERCI DE VOTRE ATTENTION