

Préparez votre projet

9–11 minutes

La société Les Bonnes Pièces vous a sollicité pour développer sa nouvelle page web et afficher une vitrine de pièces détachées automobiles. Votre client souhaite afficher les produits disponibles et les commentaires publiés par les visiteurs de la page.

Pour y parvenir, nous allons dans un premier temps identifier les données existantes sur le site actuel des Bonnes Pièces, et sélectionner celles dont nous aurons besoin pour créer notre page web.

Identifiez les différents formats de données et leur emplacement dans un projet web

Les formats

Que ce soit pour échanger ou stocker des données, les programmes informatiques ont des **formats** pour les représenter. Il existe différents types de formats, je vous présente ici les plus courants d'entre eux.

Le format **CSV** (Comma Separated Values, ou valeurs séparées par des virgules, en français), permet de stocker des données simples sous forme de texte. Il est souvent utilisé pour représenter les données d'un tableau (de type Excel) avec des lignes et des colonnes.

{Voici un exemple :

Nom de la pièce,	Prix en euros,	Catégorie
------------------	----------------	-----------

Ampoule LED,	90,	Optiques
--------------	-----	----------

Pédale d'accélérateur,150,Habitacle

Le format **XML** (eXtensible Markup Language, ou langage de balisage extensible, en français) est souvent utilisé pour la configuration technique des logiciels. C'est un format textuel qui utilise des balises ouvrantes et fermantes pour structurer l'information qu'il contient (un peu comme le HTML).

Voici un exemple :

```
<ListePieces>
  <Piece>
    <Nom>Ampoule LED</Nom>
    <Prix>90</Prix>
    <Categorie>Optiques</Categorie>
    <Description>Avec la technologie OC Motors,
ces ampoules augmentent votre visibilité de 200 %
!</Description>
  </Piece>
  <Piece>
    <Nom>Pédale d'accélérateur</Nom>
    <Prix>150</Prix>
    <Categorie>Habitacle</Categorie>
  </Piece>
</ListePieces>
```

Enfin, le **JSON** (JavaScript Object Notation) est plus moderne que les deux précédents, et de plus en plus répandu. Il est utilisé pour transmettre, stocker et configurer des données. Bref, c'est le langage à tout faire !

Voici un exemple :

[

```
{
  "nom": "Ampoule LED",
  "prix": 90,
  "categorie": "Optiques",
  "description": "Avec la technologie OC Motors,
ces ampoules augmentent votre visibilité de 200 % !"
},
{
  "nom": "Pédale d'accélérateur",
  "prix": 150,
  "categorie": "Habitacle"
}
]
```

C'est le cas des **tableurs** de type Excel, OpenOffice ou LibreOffice, qui permettent de structurer les informations dans des feuillets, sous forme de lignes, de colonnes et de cellules. Ce format est très utilisé en entreprise pour stocker et transmettre des données, mais il ne permet pas facilement d'avoir des informations cohérentes et qui respectent les règles de gestion de l'entreprise.

Enfin, le format **SQL** (Structured Query Language) est le plus adapté pour construire un projet web de zéro. Il permet de structurer un nombre important de données (lignes et colonnes), leurs types (nombre, texte court, texte long, numéro de téléphone, etc.), et d'instaurer des règles de gestion.

Les emplacements

Une fois que les données sont représentées sous un format particulier, il faut bien les stocker quelque part ! Et là aussi, plusieurs solutions s'offrent à vous. Vous pouvez les stocker dans des **fichiers** : ceux que vous trouvez dans vos dossiers sur votre

ordinateur. Ils ont un nom et une extension.

Par exemple : “pieces-auto.json”. Ici, le fichier comporte des informations sur les pièces automobiles au format JSON.

Vous pouvez également stocker vos données dans des **serveurs de base de données**. Ces serveurs sont gérés par des intégrateurs et administrateurs afin de garantir leur intégrité et leur bon fonctionnement. Les développeurs interagissent avec ces serveurs à l’aide du langage SQL et d’applications de gestion spécialisées (Oracle, pgAdmin).

Enfin, les données peuvent être manipulées à travers une **API en ligne**.

Une API en ligne ?

Une API (Application Public Interface) est un service web accessible 24 heures sur 24. Son but est de fournir un accès à des données mises à jour régulièrement. Elle accepte également de nouvelles données pour les stocker ou pour déclencher des calculs.

Par exemple, vous demandez à une API de météo le temps qu’il fera demain à 8 heures du matin : elle annonce que le temps sera pluvieux toute la journée. Deux heures plus tard, le service de météo s’est ravisé et annonce finalement du beau temps dans l’après-midi. Si vous refaites une demande le lendemain matin, l’API vous annoncera du beau temps dans l’après-midi.

Identifiez les solutions de stockage en fonction de votre situation

Face à ce grand nombre de possibilités, vous vous sentez peut-être un peu perdu ! Mais ne vous inquiétez pas, en réalité les choses ne sont pas aussi complexes.

Pour développer un site web depuis zéro, vous utiliserez le format SQL et vous stockerez les données sur un serveur de base de données.

Pour manipuler des données exportées d'un logiciel ou des fichiers de sauvegarde d'un projet, vous utiliserez plutôt les formats CSV, XML ou JSON, et vous stockerez les données dans des fichiers.

Enfin, pour interagir avec des API en ligne, vous devrez utiliser le format JSON pour les API récentes. Certaines API utilisent aussi le format XML, mais leur nombre reste assez limité. Le JSON a l'avantage d'être plus concis que les autres formats textuels, et d'être facilement compréhensible par les navigateurs.

Choisissez la solution adaptée à votre projet

Après ce tour d'horizon, intéressons-nous maintenant aux technologies que nous utiliserons dans ce cours ! 😊

Utilisez les fichiers JSON existants pour créer une interface statique

Dans les deux premières parties de ce cours, nous utiliserons des fichiers au format JSON issus d'une sauvegarde de l'ancienne version du site web des Bonnes Pièces. Nous nous servirons des données existantes sur les pièces automobiles pour créer notre page web.

Utilisez une API en ligne au format JSON pour créer une interface interactive

Dans la troisième partie du cours, nous utiliserons une API en ligne de manière à ce que les pages web puissent se mettre à jour automatiquement. Cette API utilisera le format JSON pour envoyer et recevoir des informations.

Ahh super ! Mais si j'ai bien compris, la deuxième solution est plus performante. Pourquoi ne pas l'utiliser directement ?

Tout simplement pour vous permettre d'appréhender progressivement les enjeux de création d'une page web dynamique, et de vous familiariser avec différentes manipulations

du DOM. 😊

Installez les fichiers du projet

Pour démarrer votre mission dans les meilleures conditions, vous devez récupérer les fichiers et lancer le projet.

Récupérez les fichiers du projet

Je vous invite à cloner ce [dépôt Git](https://github.com/OpenClassrooms-Student-Center/7697016-Front-End.1.git) pour récupérer les fichiers nécessaires au développement de notre page web.

Puis, déplacez-vous dans le répertoire créé, grâce à la commande `cd` :

```
git clone https://github.com/OpenClassrooms-Student-Center/7697016-Front-End.1.git site-web
cd site-web
```

Parmi les fichiers du dépôt, quatre nous intéressent :

Le **fichier principal “index.html”** comporte déjà quelques balises. Les balises `link` et `script` sont préremplies de manière à faire le lien avec les styles de la page et le fichier JavaScript que l’on remplira au fur et à mesure.

Le **fichier “pieces-autos.json”** contient des données de pièces automobiles au format JSON. Nous les utiliserons dans la première partie du cours pour générer notre page web avec du JavaScript.

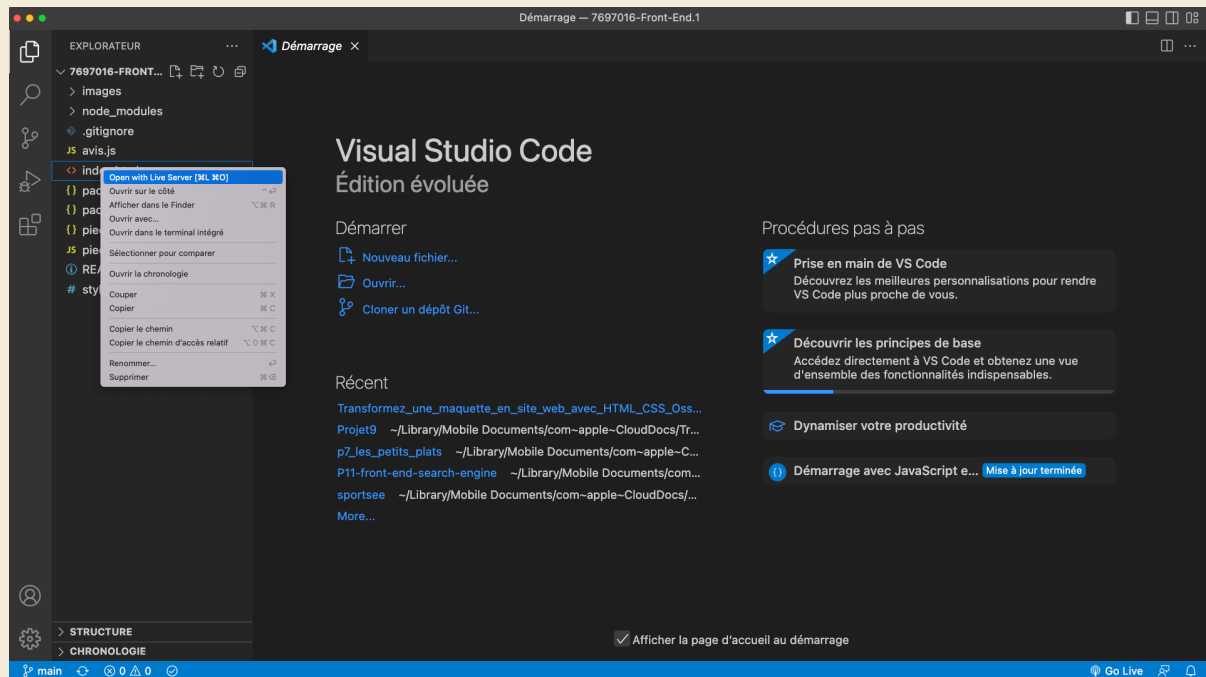
Enfin, les **fichiers “pieces.js” et “avis.js”** sont vides. Nous les remplirons progressivement de manière à rendre notre page web interactive et dynamique.

Démarrez le projet

Il existe deux méthodes différentes pour lancer le projet.

Si vous utilisez Visual Studio Code (VSCode) et l’extension `live-server` ou toute autre extension équivalente, il vous suffit de lancer le fichier `index.html` avec l’extension. Dans le cas de `live-server`,

faites un clic droit sur le fichier index.html, et choisissez l'option 'Open with Live Server'.



Choisissez l'option 'Open with Live Server' pour lancer le fichier index.html

Si vous n'avez pas d'extension pour lancer un serveur web, la deuxième option est d'installer les dépendances présentes dans le projet. Dans ce cas, vous devez installer Node.js et le gestionnaire de package npm fourni lors de l'installation. Pour y parvenir, rendez-vous sur le [site web de Node.js](https://nodejs.org/), et suivez les instructions d'installation adaptées.

Une fois que vous avez installé Node.js, lancez les commandes suivantes dans votre terminal :

```
npm install
```

```
npm run dev
```

Enfin, rendez-vous dans votre navigateur à l'adresse <http://localhost:8080>.

Heu.. Ça fait beaucoup d'étapes pour ouvrir une page web, non ?

Certes ! Mais la manière dont nous lions les fichiers du projet nécessite la syntaxe *import* du JavaScript. Pour que cela fonctionne sur votre ordinateur, il nous faut démarrer un serveur web. C'est ce qui permet d'activer le fonctionnement de la syntaxe

import.

En résumé

- Un projet web peut contenir différents formats de données :
- des formats textuels (CSV/XML/JSON) ;
- des formats structurés (SQL/tableurs).
- Les données peuvent être “en ligne” (base de données, API en ligne) ou sous forme de fichiers.
- Vous devez choisir une solution pour stocker vos données, en fonction des contraintes de votre projet et des objectifs de développement.

Pfiou... cela fait beaucoup d'informations ! Mais vous voilà fin prêt à coder ! Rendez-vous dans le prochain chapitre pour exploiter vos données au format JSON ! 😎