

# Synthèse - Projet Web Client (Book44)

## Introduction

Dans le cadre d'un projet de réalisation d'une application web utilisant les fonctionnalités d'une API REST, nous avons une semaine pour utiliser au mieux l'API que nous avons choisie : Google API Books. Grâce à cette API, nous pouvons récupérer toutes les informations concernant un livre et ainsi connaître les détails de ces derniers.

## Vision fonctionnelle

Cette application web fonctionne de la manière suivante :

- une première page web nous indiquant où nous nous trouvons et comment pourquoi poursuivre sur cette application,
- une seconde page web permettant la recherche de livres via différentes fonctionnalités que nous allons détailler.

Les différentes fonctionnalités de cette application web :

L'interface de cette seconde page web comprend deux barres de recherche

- une première permettant la recherche d'un livre grâce à son titre
- une seconde permettant la recherche d'un livre grâce au nom de l'auteur

Cela permet d'élargir le champ des recherches et ainsi offrir plus de possibilités à l'utilisateur.

L'interface comprend également quatre boutons de type checkbox (eBook, Gratuit, Récent, Auteur). Ces derniers permettent une recherche plus approfondie des résultats selon les critères de ces boutons.

Il est possible d'effectuer une recherche uniquement via le titre d'un livre ou le nom de l'auteur de celui-ci ; pour pouvoir effectuer une recherche avec un nom d'auteur il est nécessaire de cocher la checkbox "auteur" située en dessous du bouton permettant la recherche de livres.

Afin d'effectuer une recherche par date de parution, il est possible de cocher la checkbox "Récent".

Pour pouvoir utiliser la checkbox "Gratuit", il est nécessaire de cocher la checkbox "eBook" qui permet une lecture en ligne du livre et gratuitement si la checkbox "Gratuit" est cochée.

Lorsqu'une recherche est effectuée, l'affichage rendu correspond à une liste avec pour chaque élément un lien cliquable permettant l'affichage des détails du livre sélectionné. Ces détails comprennent le titre du livre, le prix (si celui-ci est payant), l'année de parution, le nombre de pages, le genre du livre, l'auteur de ce dernier et une courte description.

## Vision technique

Dans le développement de ce site web, nous nous sommes aidés du framework Materialize et du framework Vue.Js que nous avons initialisé dans "index.html" afin que le site soit plus ergonomique.

Nous avons organisé notre architecture selon un dossier "src" contenant :

"main.js" qui initialise la vue de l'application,

"gapibook.js", c'est ici que l'on va passer des requêtes à l'API par des méthodes que l'on appellera dans les différents composants afin d'afficher les informations de l'API,

Un dossier "components", qui contient 5 composants Vue.J :

"app.js", c'est la vue principale de l'application qui sera développé par les 4 autres composants "author.js" pour afficher la barre de recherche quand l'utilisateur clique sur "auteur", "book.js" qui sera la carte du livre, "search.js" pour la barre de recherche et le formulaire et "detail.js" afin d'afficher une liste détail quand l'utilisateur clique sur "En savoir plus".

Nous avons un autre dossier "assets", qui va servir pour les 2 frameworks ainsi que les images que nous mettons sur le site comme le logo ou alors une image par défaut quand celle du livre n'est pas disponible.

Nous avons procédé en récupérant tout d'abord les données grâce à l'API dans "app.js" et ensuite manipuler ces données comme nous le voulons dans l'optique de nos objectifs du site.

Voici quelques exemples de points d'entrées de l'API :

- [https://www.googleapis.com/books/v1/volumes?q={search}&key={API\\_KEY}](https://www.googleapis.com/books/v1/volumes?q={search}&key={API_KEY})

"search" contiendra le titre du livre, elle renverra un json de tous les livres contenant ce titre

- <https://www.googleapis.com/books/v1/volumes/{id}>

"id" sera l'id du livre, elle renverra l'objet du livre associé à cet id

- [https://www.googleapis.com/books/v1/volumes?q={search}+inauthor:{theAuthor}&filter=ebooks&orderBy=newest&key={API\\_KEY}](https://www.googleapis.com/books/v1/volumes?q={search}+inauthor:{theAuthor}&filter=ebooks&orderBy=newest&key={API_KEY})

"theAuthor" sert à rechercher un titre écrit par l'auteur donné.

"filter=ebooks" est un paramètre qui va demandé les livres disponibles en lignes

"orderBy=newest" va donné les les livres dans un ordre des plus récents

Ce sont quelques exemples parmi tant d'autres disponibles.

Notre clé d'API : "AlzaSyDnUpxohULaEXA1P5hArRI\_1mg4NvdzDis" donné par Google.

La documentation : <https://developers.google.com/books/docs/v1/reference/volumes/list>

## Conclusion

Ce module “client riche”, nous a beaucoup plu car l’organisation des TDs et des exercices étaient construits de sorte que l’on puisse apprendre au fur et à mesure que l’on développe des fonctionnalités de plus en plus complexes. C’était aussi l’occasion d’apprendre 2 nouveaux frameworks “Materialize” et “Vue.js” qui permettent une ergonomie efficace, apprendre le fonctionnement d’API Rest, MVC, les promesses, etc. Ce dernier projet était plaisant car nous avons pu choisir une API librement et développer notre propre vision d’un site en l’utilisant. Cela nous a permis de bien apprendre comment développer un site de cette façon.