

# Contexte

---

Cette API permet de rechercher et fournir les détails sur un livre. Elle est utilisée par plusieurs site web commercialisant des livres :fnac, darty, amazon ..

L'URL de documentation de l'API est le suivant : <https://developers.google.com/books/docs/overview>

Nous utiliserons principalement deux endpoints de cet API:

- **GET /volumes?q={search terms}** : Rechercher dans l'API avec des mots clés.  
Prend le paramètre : **maxResults** pour limiter le nombre de résultats. On utilisera la nomination **LIST Books** dans le reste du projet.
- **GET /volumes/volumeId** : Récupère un livre à partir de son Identifiant. On utilisera la nomination **GET Book** dans le reste du projet. ⚠ Pour utiliser **GET Book**, il faut récupérer l'ID present dans la réponse de **LIST Books**

Les paramètres permettant de filtrer les différents résultats de recherche sont détaillées sur la documentation de l'API.

Le maire d'Arcachon, utilisant cette API pour donner de la visibilité sur les livres disponible dans la bibliothèque municipale, fait appel à vos services pour mettre en place une stratégie d'automatisation des tests de Web Service. Votre travail est très important, étant donné que nous allons faire beaucoup de modifications dans l'avenir. Afin d'éviter d'éventuels régressions, nous comptons sur vous.

## Résultat final

---

L'objectif est de créer une collection sur Postman permettant de faire des **tests automatiques de non régression** sur l'API. A l'issue de ce projet, vous auriez des tests sur chaque endpoint vérifiant le fonctionnement : résultat de recherche, nombre de résultats, code de réponse .. etc Ainsi que des vérifications sur la cohérence des réponses des deux endpoints (à vous de choisir les points de comparaison).

## Démarches à suivre :

---

⚠ Rappel : vous êtes libres sur le projet mais on vous guide quand même.

Avant de vous lancer dans le bain de la partie technique. Prenez du temps à écrire vos scénarios de test en Gherkin. Ainsi qu'inclure ces tests dans [la documentation de chaque requête](#) . Ensuite suivez les différentes étapes ↩

## La découverte :

---

Commencez à tester **unitairement**, avec des données **en dur** les différentes requêtes pour les endpoints **LIST Books** et **GET Book**.

## La construction :

---

Ensuite, implémentez dans Postman les tests que vous avez définis précédemment.

Exécutez votre collection sur le Collection Runner.

Jusqu'à cette étape, tous les tests sont exécutés manuellement. L'objectif étant de les mettre sur une chaîne d'Intégration Continue (CI). Exécutez ces tests avec **newman**.

## Reporting :

---

Le maire voulant avoir une vue constante sur les résultats de tests. Il exige la mise en place d'un rapport **personnalisé** et détaillé. Comme vu lors des différentes interventions, il est possible de générer des rapports de différents types grâce à **newman**.

N'hésitez pas à consulter les ressources transmises lors des cours.

## Bonus :

---

Mystère, si vous avez finis avant la date limite contactez moi, j'ai de belles surprises pour vous.