

# Projet – Groupie Tracker

## Objectif

Vous allez devoir réaliser un site web en individuel, qui exploite une API pour récupérer des données dans l'objectif d'alimenter votre site web en contenu. Pour réaliser cette demande, vous devez au minimum exploiter 3 endpoints différents d'une API mais également implémenter les différentes fonctionnalités listées ci-dessous. Pour le choix du thème, du contenu puis de l'API, vous êtes libre. Cependant, vous devez faire valider votre choix par un encadrant (thème, contenu, puis enfin l'API avec les différents endpoints choisis).

L'API doit respecter quelques contraintes :

- Elle doit avoir une architecture REST.
- Elle doit retourner des données au format JSON.
- Elle ne doit pas être sous à quotas d'utilisation.
- Elle doit pouvoir délivrer des collections ou bien des ressources permettant de répondre aux différentes fonctionnalités.

Pour la question de l'accessibilité de l'API vous êtes de prendre une API avec contrainte d'accès ou pas. En cas d'API sous à des contraintes d'accès il faudra penser à gérer cette contrainte (gestion du Token avec renouvellement automatique de celui-ci) des points bonus seront ajoutés.

Vous trouvez une API au lien suivant : <https://github.com/public-api-lists/public-api-lists>

Vous pouvez également trouver une liste d'API sur moodle, pour les étudiants qui n'ont pas d'idée ou bien qui ne trouve pas d'API.

**Rendu :** Vous devez le lien du dépôt GitHub contenant le code source de votre solution ainsi que les documentations demandés sur moodle, avant le 10/03/25 à 23h59.

## Fonctionnalités

Votre solution devra embrasser les fonctionnalités demandées ci-dessous. Attention l'inégalité des fonctionnalités doivent être implémentées à la main avec l'aide du langage de programmation Golang.

### *FT1 – Un système de recherche*

Il faudra permettre à un utilisateur de pouvoir effectuer une recherche sur une collection. Cette recherche doit pouvoir être effectué sur au minimum deux propriétés de la ressource. Par exemple pour un site web proposant une collection de musiques, lorsque l'utilisateur va entrer une query, les deux propriétés tester seront le titre de la musique, puis le nom de l'artiste ou bien encore le nom de l'album.

### *FT2 – Un système de filtre*

Il faudra permettre à un utilisateur de pouvoir filtrer les différentes ressources d'une collection dans l'objectif de pouvoir sélectionner uniquement certaines ressources correspondantes aux filtres appliquées. Il faudra avoir la possibilité d'appliquer de pouvoir filtrer sur trois propriétés différentes, les filtres doivent être cumulable. Par exemple pour un site web proposant une collection de films, l'utilisateur pourra filtrer sur le genre, la date de sortie, puis la durée.

### *FT3 – Un système de pagination*

Votre solution devra embrasser un système de pagination, lors de l'affichage d'un nombre important de ressources sur une page dans l'objectif d'afficher les ressources par lot de 10, 20 ou bien 30.

### *FT4 – Un système de favoris*

L'utilisateur devra pouvoir ajouter des ressources de votre site web à une liste de favoris, mais il devra également pouvoir gérer cette liste en ayant la possibilité de pouvoir supprimer une ressource. Cette liste de favoris devra être persistante, donc vous allez devoir utiliser des fichiers txt ou bien json pour faire persister cette liste.

Vous pouvez rajouter des fonctionnalités bonus, dans l'objectif d'améliorer votre site web. Cependant penser à ne pas négliger celle obligatoires ! les bonus rapportes moins que les demandes du sujet.

## L'arborescence du site web

Le design de l'interface de l'application, devra être en accord avec les données renvoyées par l'API que vous avez choisie. De plus, chaque page doit contenir un header contenant un menu de navigation qui permettra d'accéder aux autres pages, puis enfin un footer. Votre solution va devoir contenir une multitude de pages. Voici quelques indications sur les attentes de l'arborescence de votre site web :

Page d'accueil : cette page permettra d'afficher l'univers, le thème... de votre site web.

Page collection : cette page permettra d'afficher une collection de ressources.

Page ressource (détails) : cette page permettra d'afficher les détails d'une ressource.

Page catégorie : cette page permettra d'afficher les ressources en fonction de leur catégorie.

Page favoris : cette page permettra d'afficher et de gérer la liste des favoris.

Page recherche : cette page permet d'afficher les résultats d'une recherche.

Page à-propos : cette page permettra d'avoir un retour projet, elle devra répondre aux demandes de la section « documentation ».

Vous pouvez ajouter des pages ou bien réorganiser celle-ci.

## Les contraintes techniques

Pour mener à bien ce projet, vous êtes soumis aux contraintes indiquées ci-dessous :

- Exploitation d'une API REST pour récupérer des données
- Mise en place d'une gestion des erreurs, l'application web ne doit pas planter en cas de service offline (gestion des erreurs grâce au code de réponse http, page d'erreur...)
- Utilisation uniquement des langages de programmation : Golang, HTML et CSS
- Utilisation des librairies standards de Golang
- Interdiction d'utilisation de Framework HTML/JS/CSS
- Une interface en accord avec les données, mais également user-friendly

## Documentation

Un fichier README doit être présent à la racine de votre projet. Ce fichier doit présenter votre projet (thème, fonctionnalité, objectif...), expliquer comment installer et lancer votre projet, ainsi que détailler l'ensemble des routes implémentées. Enfin, il faudra mentionner l'API avec les différents endpoints exploités de celle-ci.

En plus de cela, une synthèse du déroulement du projet de votre point de vue est attendue. Seuls les aspects liés à la gestion de projet seront évalués. Voici les éléments qui doivent être abordés en détail :

- Comment avez-vous décomposé le projet ? Quelles ont été les phases clé ?
- Comment avez-vous réparti les tâches entre les membres de l'équipe ? Avez-vous utilisé une stratégie particulière pour organiser le travail ?
- Comment avez-vous géré votre temps ? Avez-vous défini des priorités ?
- Quelle stratégie avez-vous adoptée pour vous documenter ?

Ce rendu devra être fait sur votre page à-propos, sous forme de FAQ.

## Rendus / Livrables

### *Code source et documentation*

Vous devez fournir un dépôt Git qui contiendra le code source du projet ainsi que tous les éléments nécessaires à son bon fonctionnement. Le dépôt devra également inclure les documentations demandées.

**Date limite de rendu sur GitHub : le 10/03/25 avant 23h59.**

## Les étapes clé

Voici pour ceux qui ont encore des difficultés à identifier les grandes étapes qui vont vous permettre de mener à bien ce projet.

Étapes 01 : Choix de l'API et préparation des endpoints exploités avec l'outil Postman

Étapes 02 : Préparation des wireframes ainsi que des maquettes du site web avec Figma

Étapes 03 : Implémentation et développement du site web

Étapes 04 : Test et correction des fails du site web

Étapes 05 : Livraison de la solution web !