

# Recipe Vault API

## 1. Fonctionnalités

- Authentification :
    - Processus d'authentification sans mot de passe oublié et vérification d'email (Utiliser Laravel Sanctum).
    - L'ensemble du site n'est accessible que si l'utilisateur est connecté.
  - Database :
    - Création d'une table pour stocker les recettes
    - Les images des recettes sont stockées sur le serveur
    - Liaison d'une recette vers un utilisateur qui sera son auteur
    - Liaison d'un utilisateur vers plusieurs recettes pour ses favoris
    - La recette est secrète et doit être cryptée sur la base de données (Création d'un trait Encryptable qui permet d'ajouter un attribut \$encryptable contenant l'ensemble des champs à crypter)
  - Seeding :
    - Utilisation de jzonta/faker-restaurant pour les noms des recettes
    - Passer par un Factory
    - Utilisation de xvladqt/faker-lorem-flickr pour les images (<https://loremflickr.com/320/240/dish>)
  - Validation :
    - Tous les champs sont requis
    - La description doit faire maximum 2048 lettres
  - Permissions :
    - Utilisation de spatie/laravel-permissions
    - Permissions :
      - Admin :
        - create dishes
        - update dishes
        - delete dishes
- Seuls les admins peuvent créer / modifier / supprimer un plat
- Notification :
    - Lors de la création d'une recette l'utilisateur reçoit un mail pour lui spécifier que sa recette est en ligne
  - Recherche :
    - Mettre en place un système de recherche à l'aide de Laravel Scout et Meilisearch
-

## 2. Recommandations

- Pour le mail utiliser les notifications Laravel qui permettent de créer des mails à partir du template de base Laravel par défaut
  - Attention à la manière d'upload l'image du plat
  - Utilisation de <https://mailspoons.com/> pour le test des mails
- 

## 3. Contraintes techniques

- **Technologies** : Utiliser le package RestAPI (<https://laravel-rest-api.lomkit.com/>)
-