

Table des matières

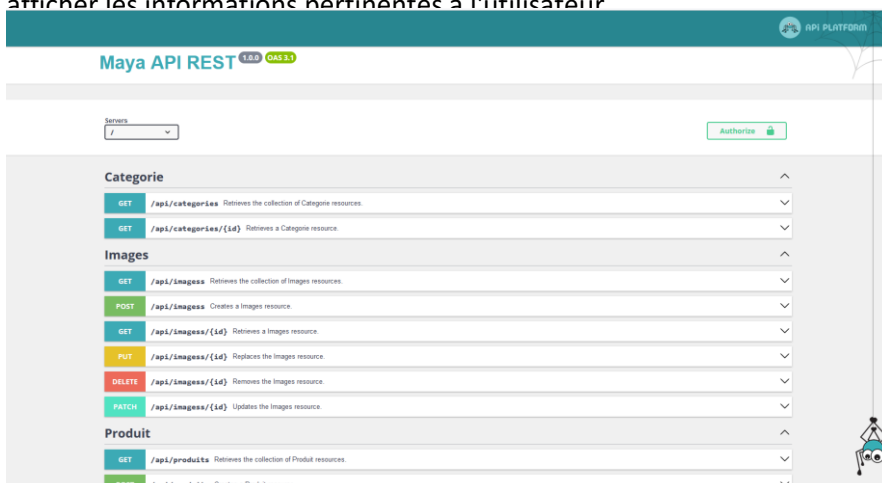
- I. Introduction..... 1
- II. Fonctionnement de l'API 1
- III. Les requêtes HTTP 2
 - 1) Catégorie 2
 - 2) Recette 3
- IV. Conclusion..... 4

I. Introduction

Ce projet est une application mobile qui utilise une API pour récupérer des données sur les catégories, les produits et les recettes. L'API est conçue pour fournir des informations structurées sur ces éléments, permettant à l'application de les afficher de manière conviviale pour l'utilisateur. L'API est accessible via des requêtes HTTP et renvoie des réponses au format JSON, ce qui permet à l'application de facilement extraire les données nécessaires pour les afficher.

II. Fonctionnement de l'API

L'API est basée sur le protocole HTTP, ce qui signifie qu'elle utilise des requêtes et des réponses pour communiquer avec les clients. Pour récupérer des données, l'application envoie des requêtes HTTP à des endpoints spécifiques de l'API, tels que /api/categories, /api/produits, ou /api/recettes. L'API traite ces requêtes et renvoie des réponses contenant les données demandées, généralement au format JSON. L'application analyse ensuite ces réponses et utilise les données récupérées pour afficher les informations pertinentes à l'utilisateur.



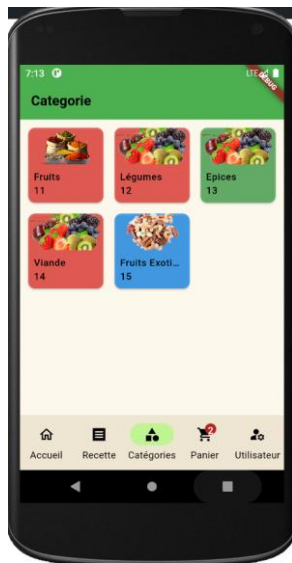
III. Les requêtes HTTP

Les requêtes HTTP sont des messages envoyés par l'application à l'API pour demander des données spécifiques. Ces requêtes contiennent généralement des informations telles que l'URL de l'endpoint à interroger, la méthode HTTP à utiliser (GET, POST, etc.), et éventuellement des paramètres supplémentaires tels que des filtres ou des données à envoyer dans le corps de la requête. Une fois la requête envoyée, l'API traite la demande et renvoie une réponse contenant les données demandées, ainsi qu'un code d'état HTTP indiquant si la requête a réussi ou échoué.

```
▼ 0:
  id:      11
  libelle: "Fruits"
  picture: "/api/imagess/4"
▼ 1:
  id:      12
  libelle: "Légumes"
▼ 2:
  id:      13
  libelle: "Epices"
▼ 3:
  id:      14
  libelle: "Viande"
▼ 4:
  id:      15
  libelle: "Fruits Exotique"
  picture: "/api/imagess/2"
```

1) Catégorie

Dans ce projet, les données sont organisées en différentes catégories, et pour récupérer ces catégories, l'application utilise l'API avec le chemin `/api/categorie`. Les catégories agissent comme des groupes organisateurs pour les produits disponibles dans l'application, permettant aux utilisateurs de naviguer et de découvrir les articles qui les intéressent plus facilement. Chaque catégorie représente un ensemble spécifique de produits apparentés, regroupant des articles similaires sous une même rubrique pour une organisation efficace. Par exemple, les catégories peuvent inclure des classifications telles que "Fruits et Légumes", "Produits Laitiers", "Viandes et Poissons", etc. L'API renvoie les données des catégories au format JSON, fournissant des détails sur chaque catégorie, tels que son nom, son identifiant unique et éventuellement d'autres informations pertinentes. Une fois récupérées par l'application, ces données sont utilisées pour afficher les différentes catégories disponibles à l'utilisateur, lui permettant ainsi de parcourir et de sélectionner facilement les produits qui l'intéressent.

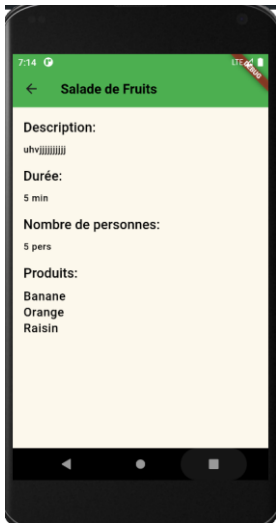


2) Recette

Dans recette, nous avons une page de détail de recette qui affiche les informations détaillées d'une recette spécifique. Lorsque cette page est affichée, elle récupère les détails des produits associés à la recette à partir de l'API. Voici comment cela fonctionne :

- Initialisation des détails des produits : Lorsque la page est initialisée (`initState()`), la méthode `loadProduitsDetails()` est appelée. Cette méthode est responsable de charger les détails des produits associés à la recette.
- Chargement des détails des produits : La méthode `loadProduitsDetails()` itère à travers chaque lien de produit dans la liste `widget.recette.produits`. Pour chaque lien, une requête HTTP est envoyée à l'API pour récupérer les détails du produit correspondant.
- Traitement de la réponse : Lorsque la réponse de l'API est reçue avec succès (code de statut 200), les détails du produit sont extraits de la réponse JSON et ajoutés à la liste `produitsDetails`.
- Affichage des détails de la recette : Une fois que tous les détails des produits ont été récupérés et ajoutés à la liste `produitsDetails`, la page de détail de la recette est construite. Les informations principales de la recette, telles que sa description, sa durée et le nombre de personnes pour lesquelles elle est prévue, sont affichées. Ensuite, la liste des produits associés est affichée, affichant les détails de chaque produit sous forme de texte.

En résumé, cette page de détail de recette récupère dynamiquement les détails des produits associés à partir de l'API lors de son initialisation, puis les affiche aux côtés des autres informations de la recette. Cela permet aux utilisateurs d'avoir une vue complète des ingrédients nécessaires pour chaque recette présentée dans l'application.



IV. Conclusion

L'utilisation de l'API dans cette application permet de fournir des données dynamiques et actualisées à partir d'une source externe, facilitant ainsi la gestion et la mise à jour des informations présentées à l'utilisateur. Grâce à des requêtes HTTP, l'application peut interagir avec l'API pour récupérer différentes ressources telles que les catégories, les produits et les recettes, offrant ainsi une expérience utilisateur enrichie.

Les requêtes HTTP jouent un rôle crucial dans le fonctionnement de l'application, car elles permettent d'établir une communication bidirectionnelle entre l'application et l'API. Ces requêtes sont utilisées pour récupérer des données à afficher à l'utilisateur, mais aussi pour envoyer des informations à l'API, par exemple lors de la création ou de la modification de données.

En résumé, l'utilisation de l'API et des requêtes HTTP dans cette application permet une intégration fluide avec des sources de données externes, offrant ainsi à l'utilisateur une expérience interactive et dynamique. Cela ouvre également la possibilité d'extension et de mise à jour facile de l'application, car les données peuvent être facilement modifiées ou étendues du côté de l'API sans nécessiter de modifications importantes dans l'application elle-même.