DEFOSSE Samuel
THIVET Simon

2021-2022

EasyCook

Développement d’application mobile

Introduction

Ce projet consiste à développer une application web mobile hybride de recettes. EasyCook est en réalité un outil pour découvrir, trier et trouver de nouvelles recettes. Cette application utilise les librairies VueJS et Capacitor et reprend les concepts vus en cours.

Choix techniques

Cette partie présente les choix de conception effectués lors du développement de l'application.

- API

Nous avons choisi **Spoonacular** qui est une API REST bien structurée et bien documentée. Elle comporte un grand volume de données avec plus de 5 000 recettes et de nombreux ingrédients. Cette API nous permet de simplifier le code car son modèle de données est très complet. Elle offre des possibilités de filtrage personnalisé au vue des paramètres des différentes requêtes possibles (filtrer en fonction de la diet, cuisine ou bien ingrédient). De plus, la documentation est claire et exhaustive avec des exemples pour effectuer les différents types d'appels API.
Cependant, Spoonacular n'est pas totalement gratuite, elle offre un quota d'utilisation et de requêtage journalier gratuit après création d'un compte. Une fois ce quota dépassé, il faut passer à une version payante pour continuer à l'utiliser. Le nombre de requêtes journalières est limité à 150 ce qui est suffisant pour une utilisation scolaire. Un quota de réponses API nous oblige à limiter le nombre de réponses par requête afin de ne pas dépasser le quota. Les données retournées sont en anglais si bien que notre application sera en anglais pour garder une cohérence générale.

- Text-To-Speech

Text-To-Speech est un plugin Capacitor qui permet de faire appel à la commande vocale du téléphone afin d'utiliser le haut parleur du mobile pour lire des informations, en l'occurrence le détail des recettes. Il nous a semblé judicieux de l'implémenter à notre application pour permettre à l'utilisateur de cuisiner en même temps que les instructions lui soient dictées ou pour permettre aux personnes ayant des problèmes de vue de suivre plus facilement la recette.
Nous avons implémenté ce plugin sur la partie détail des recettes, avec un bouton disponible sur chaque carte des étapes permettant d'effectuer la lecture des instructions de celles-ci. A noter que la lecture est plus fluide et agréable à l'écoute sur la version mobile que sur la version web.

- KeepAwake

  KeepAwake est un plugin capacitor permettant d'empêcher l'écran d'un smartphone de s'atténuer ou de se verrouiller tant que l'appli est en plan principal.
  Nous avons décidé d'implémenter ce plugin pour permettre à l'utilisateur de ne pas avoir constamment à devoir déverrouiller son écran pendant la réalisation d'une recette et ce sans avoir à modifier les options de son smartphone. Son implémentation est effectuée via un bouton disponible sur la partie droite du menu de l'application qui permet de choisir d'empêcher le verrouillage de l'écran ou bien de le permettre à nouveau.

Application et fonctionnalités

Cette partie présente les fonctionnalités ainsi que les différents visuels de EasyCook.

Fonctionnalités

L'application EasyCook a deux fonctionnalités principales faisant appel à une requête API chacune :

- La génération d'une recette aléatoire, cette fonctionnalité permet via une requête API de générer une recette aléatoire à chaque fois que le composant est appelé. Un bouton "Générer une nouvelle recette" permet de recharger le composant avec une nouvelle recette.
- La recherche de recettes, cette fonctionnalité propose de rechercher une recette en fonction de deux filtres, "Location" qui filtre sur la localité des recettes (exemple: Europe) et "Diet" qui filtre sur le régime (exemple: végétarien). Il n'est pas nécessaire de choisir une valeur pour les deux filtres, un seul suffit.

Chaque recette est affichée sous forme de carte avec le nom de la recette ainsi qu'une image représentant le visuel attendu. Sur chaque carte, un lien "Recipes step" est disponible permettant de charger les différentes étapes de celle-ci. Chaque étape correspond à une carte qui contient les ingrédients nécessaires, l'équipement ainsi que les instructions pour réaliser l'étape. Chacune de ses étapes met à disposition un bouton "Read the instructions" qui permet de lancer une lecture audio des instructions de l'étape.

Une autre fonctionnalité plus générale de l'application est de permettre d'empêcher l'atténuation ainsi que le verrouillage de l'écran via le bouton "Keep the screen awake".
Pour enlever cette fonctionnalité, il suffit de cliquer à nouveau sur ce bouton qui est renommé "Allow the screen to sleep". Ce bouton est situé à la fin de la barre de menu de l'application.

Architecture

Dans un premier temps, nous avons décidé de créer un service “easyCookServices” qui va regrouper les différents appels API utilisés au sein de l’application. Elles ne sont qu’au nombre de 3 aujourd’hui mais cela permet de rendre plus léger les différents composants qui ont besoin d’un appel API et regrouper les différents appels à un seul endroit.

L’application se découpe en différents composant, nous allons uniquement présentés ceux en lien avec les recettes qui sont les plus intéressants à détaillés :

- Recipe, ce composant permet de représenter les recettes sous forme de carte. Il a pour propriétés “recipe” qui est un objet contenant les informations d’une recette. A partir de ce composant nous avons implémenté un “router-link” qui permet de rediriger l’utilisateur vers les étapes de la recette, en passant en paramètre l’id ainsi que le nom de la recette.
- RecipeDetails, représente les détails de la recette et notamment les différentes étapes de réalisation de la recette sous forme de cartes. Elle a pour propriétés un id, un nom ainsi qu’une liste de recettes. Pour récupérer les différentes étapes, un appel API est passé qui va se servir de l’id et du nom de la recette qui ont été récupérés auparavant via le composant “Recipe”.
- RandomRecipe, représente la génération aléatoire d’une recette. Il a pour propriété une liste de recettes qui va permettre de stocker la recette aléatoire ainsi qu’un booléen “loading” qui permet de gérer le chargement de la recette. Ce composant utilise un appel API qui récupère une recette aléatoire qui sera ensuite stockée dans la liste des recettes (la réponse de l’appel est une liste d'objets). On affiche ensuite à l’écran la recette en faisant appel au composant “Recipe” en lui passant en paramètre la recette contenue dans la liste.
- ResearchRecipe, représente la recherche de recettes via des filtres. Il a pour propriété une liste de recettes qui va permettre de stocker les recettes récupérées suite à la recherche, locationFilter et dietFilter des listes de chaînes de caractères contenant les filtres de la recherche et enfin filterSelected et dietSelected qui sont deux chaînes de caractères permettant de stocker le choix de l’utilisateur sur les listes des filtres. Afin de récupérer les recettes filtrées, un appel API est passé avec en paramètre locationFilter et dietFilter. On stocke le résultat dans le paramètre de liste des recettes et comme pour RandomRecipe, on fait appel au composant “Recipe” sur chaque recette pour pouvoir les afficher à l’écran.

Le choix de séparer une recette en deux composants pour afficher la recette dans un premier temps et ensuite les détails de celle-ci est motivé par l’aspect visuel. Au départ nous affichions les détails de la recette dans la même carte que la recette, mais nous nous sommes rendus compte que cela produisait une carte de taille conséquente ou qui nécessiterait de scroller à l’intérieur. C’est pour cela que nous avons fait le choix d’afficher de à l’écran au départ uniquement le nom de la recette ainsi que l’image, puis de permettre à l’utilisateur de visualiser les détails de la recette via un autre composant permettant de produire une carte par étape et ainsi rendre cela plus agréable à la visualisation.

Conclusion

Pour conclure, ce projet nous a permis de mettre en application l'ensemble des notions que nous avons abordées en cours et ainsi avoir un premier aperçu de comment se développe une application android/web. Dans l'ensemble le projet s'est bien déroulé, avec peu de difficultés rencontrées et nous avons réussi à mettre en place tout ce que nous avions planifié lors du lancement du projet.