

B5 - Application Development

B-DEV-500

OTAKUBOARD

**All your information in the blink
of an eye**

DOCUMENTATION

Technique

Sommaire

I - Context du projet

II - Architecture

1. Arborescence des fichiers
2. Système de routing
3. Utilité des différents fichiers

III - Composants

1. Diagramme de composants UML
2. Liste des composants

I - Context du projet

Ce projet a pour objectif la mise en place d'un dashboard. Dashboard est la traduction anglaise du terme « tableau de bord ». Ainsi, c'est une représentation visuelle des informations importantes sur un même écran.

Ce Dashboard affiche donc des informations venant de différentes API. Il propose 3 services différents et chacun de ces services possède au moins 2 widgets qui sont les informations. Ces widgets sont régulièrement mis à jour afin que le dashboard n'affiche pas d'information erronée.

Ce projet a été programmé avec le framework React avec Firebase comme base de données.

II - Architecture

1 - Arborescence des fichiers

En termes d'organisation de nos fichiers, nous avons un dossier par catégories de fichiers. C'est-à-dire que nous trions nos fichiers en fonction de leur nature (javascript, css, ...). Ainsi il est facile de retrouver le fichier que l'on cherche.

2 - Système de routing

Tout d'abord le routing, dans un site web, permet d'identifier et de différencier toutes les pages du site. Nous avons choisi de regrouper toutes les "routes" (donc pages) de notre site dans un seul fichier principal. Cela nous permet de gérer toutes les pages de la même manière et depuis qu'un seul fichier. Il contient donc le système de routing ainsi que des fonctions toutes simples qui ont pour seul but d'afficher la page correspondant à la route. Ainsi notre fichier principal est extrêmement simple et chacun de nos autres fichiers correspondent à une seule page.

3 - Utilité des différents fichiers

Client

Les pages principales :

App.js : c'est le fichier principal. Il nous permet de définir toutes les routes du site ainsi que d'afficher ce que l'on souhaite sur ces routes. C'est ici que sont regroupées toutes les parties graphiques de notre projet.

Auth.js : il s'agit de la page d'authentification à notre Dashboard. C'est la première page que voit l'utilisateur. Cette page relie Firebase à notre site.

Les pages widgets :

YoutubeWidget.js : ce fichier contient notre widget de recherche de vidéo Youtube. Nous appelons ce fichier dans App.js afin de d'afficher ce widget.

ChannelStats.js : ce fichier contient notre widget de d'affichage de statistiques de chaîne Youtube. Nous appelons ce fichier dans App.js afin de d'afficher ce widget.

TradEng.js : ce fichier contient notre widget de traduction de mot Anglais en Kanji Japonais. Nous appelons ce fichier dans App.js afin de d'afficher ce widget.

TradKanji.js : ce fichier contient notre widget de traduction de Kanji Japonais en mot Anglais. Nous appelons ce fichier dans App.js afin de d'afficher ce widget.

MangaWidget.js : ce fichier contient notre widget d'affichage de caractéristiques de manga. Nous appelons ce fichier dans App.js afin de d'afficher ce widget.

TendencyWidget.js : ce fichier contient notre widget d'affichage du classement des animés ou mangas du service Kitsu. Nous appelons ce fichier dans App.js afin de d'afficher ce widget.

Les fichiers utilitaires :

Navbar.js : ce fichier contient l'affichage de notre barre de navigation que nous retrouvons sur la grande partie de notre Dashboard.

ApiYoutube.js : ce fichier contient la requête à l'api de youtube afin de rechercher et de récupérer les ID des vidéos à afficher dans le widget de vidéo youtube.

VideoPlayer.js : c'est le fichier qui permet l'intégration et l'affichage de la vidéo trouver par le widget vidéo youtube.

ApiKitsu.js : ce fichier contient la page d'authentification au service kitsu ainsi que la requête afin de récupérer le token indispensable pour nos requêtes à cette API.

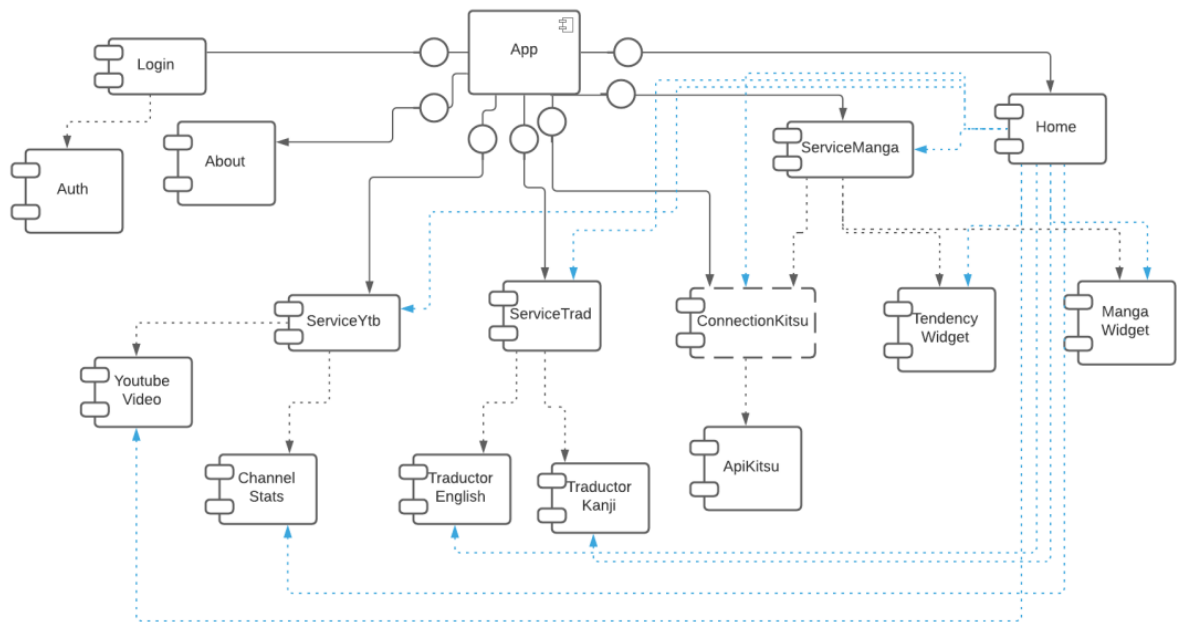
Serveur

index.js : il contient toutes nos routes de stockage et de récupération de différentes informations. C'est aussi lui qui contient le About.json qui affiche beaucoup de renseignements concernant le Dashboard ou le serveur.

III - Composants

1- Diagramme de composants UML

Diagramme de composants UML: côté client



[OBJ]

2- Liste des composants

Globale

App: Gestionnaire des routes

Affichage

Login: Affichage de l'authentification

About: Affiche le about.json

Service Youtube: Affichage des widgets du service Youtube

Service Traduction: Affichage des widgets du service Traduction

Service Manga: Affichage des widgets du service Manga

Connection Kitsu: Affichage le formulaire pour obtenir le token Kitsu

Home: S'occupe de générer les différents widgets présents sur la page principale, de les supprimer et de les sauvegarder via le serveur.

Mise en place

Auth:(paramètres: IsConnected -> Bool) Gère les formulaires d'authentification, les requêtes avec Firebase, et le sauvegarde via le serveur

Youtube Video: Affichage du widget Youtube video + Gestion des requêtes à l'API Youtube, gestion des paramètres, changement de l'affichage (pour ce widget)

Channel Stats: Affichage du widget Channel Stats + Gestion des requêtes à l'API Youtube, gestion des paramètres, changement de l'affichage (pour ce widget)

Traduction English: Affichage du widget Traduction English + Gestion des requêtes à l'API Kanjialive, gestion des paramètres, changement de l'affichage (pour ce widget)

Traduction Kanji: Affichage du widget Traduction Kanji + Gestion des requêtes à l'API Kanjialive, gestion des paramètres, changement de l'affichage (pour ce widget)

Api Kitsu: Gère le formulaire d'authentification Kitsu, fait la requêtes avec Kitsu, et le sauvegarde via le serveur

Tendency Widget:(paramètres: token -> String) Affichage du widget Tendency Widget + Gestion des requêtes à l'API Kitsu, gestion des paramètres, changement de l'affichage (pour ce widget)

Manga Widget:(paramètres: token -> String) Affichage du widget Manga Widget + Gestion des requêtes à l'API Kitsu, gestion des paramètres, changement de l'affichage (pour ce widget)