

Rapport TwitterLike

MANGEOT Thomas

Un peu de difficulté à comprendre au début, notamment pour comprendre l'architecture du projet. Mais finalement une fois dedans ça allait. J'ai eu aussi des problèmes au début avec l'importation de certains package et avec l'utilisation du google font en lien dans le TP1.

Tweet.dart :

Dans lequel on va instancier le contenu d'un tweet en allant récupérer les informations nécessaires vers le fichier json qui contient les tweets.

J'avais un problème que vous avez résolu, le code ne marchait pas comme lors de ma récupération des images pour les tweets je ne prenais pas en compte qu'il pouvait ne pas y en avoir

```
text: json['text'],
tags: json['tags'] != null ? List<String>.from(json['tags']) : [],
// ignore: prefer_if_null_operators
image: json['image']!=null ? json['image'] : null,); // Tweet
```

Tweet Master :

Pour afficher tweetpreview, pas grand-chose à dire ici à par que c'est ici que je fais en sorte que les tweets affichés dans tweetpreview soient aussi un bouton qui redirige vers TweetDetailsPage (que j'explique plus bas)

```
final Tweet currentTweet = snapshot.data![index];

return ListTile(
  title: TweetPreview(
    currentTweet: currentTweet,
    index: index,
  ), // TweetPreview
  onTap: () => Navigator.push(context, MaterialPageRoute(builder: (context) => TweetDetailsPage(tweet: currentTweet, index: index,
```

TweetPreview :

Donc pour tweetPreview et tweetDetails j'utilise les tweets instancié dans mon tweet.dart (avec les currentTweet)

Je retourne donc en ListStile (qui s'est avéré ne pas forcément être le meilleur choix car ça ne collait pas pour le tweetDetail)

Sinon je me suis un peu amusé, pour « le front », j'ai mis les tweets en jaune avec écrite noire, j'ai changé de couleur et arrondi les bordures de mes tweets. J'ai changé le format de la date qui n'était vraiment pas beau, et j'y ai gardé juste l'année, le mois et le jour)

Juste aussi un commentaire au niveau de l'image ou je mets une icone d'image en grey quand il n'y a pas d'image (currentTweet.image = null)

Tweet_details_page :

pour afficher le widget TweetDetails. J'ai un peu eu de mal lors de la création de la 2^e page pour afficher les détails. Je ne voyais pas comment afficher un widget comme étant une page donc j'ai créé cette nouvelle classe qui vient récupérer les infos (ou utiliser plutôt) de mon widget tweetDetails

TweetDetails : je voulais essayer d'afficher comme vous les éléments les uns en dessous des autres, pour ça je n'étais pas sûr qu'une liste marche donc pour afficher les éléments les uns en dessous des autres j'ai mis une card (j'ai trouvé ça sur internet). Et ensuite j'ai l'attribue column pour placer les trucs en colonnes. Cross axciss alignement pour que les éléments prenne toute la taille de la colonne. Les padding servent à définir les différents étages de ma colonne, et donc en premier j'affiche l'image, en dessous la date et liked, en dessous le texte (title) et en dessous l'auteurs et les # (juste séparé par un retour à la ligne \n cette fois ci)

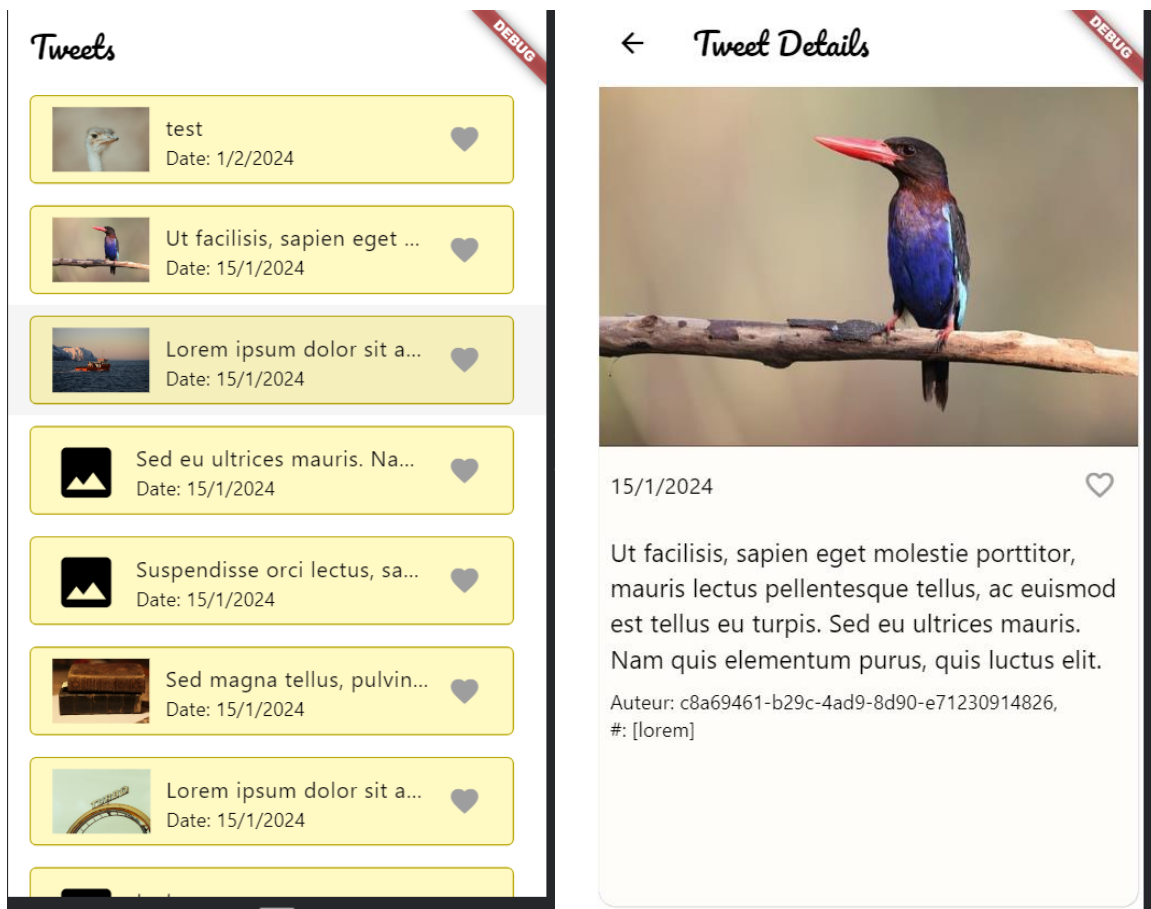
Remarque : du coup c'est très joli au format application, mais très moche au format site. Je pars du principe que le but du projet est de créer une application.

Dans tweetDetails liked est un button donc je retiens l'état avec mon setState. J'avais aussi rencontré un problème pour cette étape que vous avez résolu (je n'arrivais pas à sauvegarder le changement d'état de liked), c'était bien le widget qui devait être en StateFul et non en StateLess

Au niveau de l'image on vérifie si liked est true et dans ce cas-là un cœur rouge apparait en haut à droite de l'image (bon j'ai juste repris l'icone que j'avais mis pour liked par faute de temps)

Par faute de temps je n'ai pas regardé comment sauvegarder aussi le changement d'état de liked dans TweetPreview

Normalement si tout se passe bien quand je run le code ça donne ça :



Je sais pas si ça se ressent dans mes explications mais je trouve que sur la fin je comprenais vraiment bien ce que je faisais, je suis content

