Aplicació Flutter. API CALL

Objectiu

En aquesta pràctica volem recollir la informació de l'API i mostrar-la al detall que vam dissenyar a la sessió anterior

Documentació

https://docs.flutter.dev/cookbook/networking/fetch-data

Model

El primer que farem serà crear el model de dades que rebrem de l'API

1. Dins el directori lib crea un nou directori que es digui model
2. Dins d'aquest nou directori crea un fitxer .dart (Ex: apod.dart)

   En aquest fitxer definirem el model de dades de l'API. Per a fer-ho ens hem de fixar en l'estructura de l'api.

   https://api.nasa.gov/planetary/apod?api_key=6u8YizBXHawwVvOgDaJXNXJoUoxQ2xTZzxXWC11Y

   També hem de pensar quins atributs necessitem. En aquest cas, com a mínim, copyright, date, explanation, title i url.

   En el fitxer .dart definirem la classe, els atributs i el constructor. Fixa't en l'exemple i acaba'l de completar amb la resta d'atributs.

```
class Apod{
  String copyright;
  String date;
```

```
 Apod({
    required this.copyright,
    required this.date,
 });
}
```

En aquesta classe també definirem el mètode que transforma el Json en objecte de dades.

```
factory Apod.fromJson(Map<String, dynamic> json) {
    return Apod(
      copyright: json["copyright"].toString(),
      date: json["date"].toString(),
    );
  }
```

Fixa't que el que fa és associar la classe Apod (definida anteriorment) al json de dades que li passem. Aquest json té una estructura <String, dynamic> és a dir, string per la clau i un valor dinàmic pel valor.

El que fa el mètode és retornar un objecte Apod en el que assigna als atributs els valors del json.

API

1. Crearem un nou directori que es dirà apiService
2. En aquest hi afegirem un nou fitxer .dart que es dirà ApiService.
3. Com que volem fer una petició http hem d'habilitar aquesta dependència a l'arxiu *pubspec.yalm* afegint la línia http:

```
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  http:
```

Alhora, a l'Android Manifest (android → app → src → main → AndroidManifest.xml) afegirem els permisos d'Internet

```
<uses-permission android:name="android.permission.INTERNET" />
```

4. Al fitxer apiService.dart hi importarem el package http

```
import 'package:http/http.dart' as http;
```

Crearem la classe ApiService amb el mètode getData()

```
class ApiService {

  Future<Apod> getData() async {
```

```
    var url = 'api.nasa.gov';
    var urlExtension = '/planetary/apod';
    final Map<String, String> queryParameters = <String, String>{
      'api_key': '6u8YizBXHawwVvOgDaJXNXJoUoxQ2xTZzxXWC11Y',
    };
    final api = Uri.https(url, urlExtension, queryParameters);
    print(api);

    final response = await http.get(api);

    if (response.statusCode == 200) {
      return Apod.fromJson(jsonDecode(response.body));
    } else {
      throw Exception('Failed to load album');
    }
  }
```

Fixeu-vos que definim el mètode Future<Apod>getData() async{

Això vol dir que definim un mètode asíncron (per no bloquejar el thread principal) i per a fer-ho utilitzem la classe Future.

Dins el mètode definim la URL base i l'extensió, i creem els paràmetres per poder-li passar l'API_KEY. Amb aquests tres paràmetres definits creem la URL i fem un print per assegurar-nos que el que hem creat és la URL que realment volem.

Una vegada tenim la URL ja podem fer la crida http mitjançant el mètode http.get(api)

Si el statusCode == 200 podrem analitzar el json de resposta (response.body) decodificant-lo i passant-li al mètode que hem definit anteriorment a la classe Apod.

Vista

1. Crearem un nou Stateful Widget. Per a fer-ho crea un nou arxiu .dart i escriu `stful` això crearà una nova plantilla d'un stateful widget. Fixa't però, que hauràs de determinar el nom de la classe i automàticament s'escriuran els noms del constructor i del state.

2. Dins la classe _State definirem una instància de la classe ApiService

```
final ApiService apiService = ApiService();
```

Fixeu-vos en el següent codi.

```
return MaterialApp(
      title: 'Fetch Data Example',
      home: Scaffold(
        body: Center(
          child: FutureBuilder<Apod>(
            future: apiService.getData(),
            builder: (context, snapshot) {
              if (snapshot.hasData) {
                return Text(snapshot.data!.title);
```

```
            } else if (snapshot.hasError) {
              return Text('${snapshot.error}');
            }else{
              return const CircularProgressIndicator();
            }
          },
        ),
      ),
    ),
  );
```

Per a fer la crida a l'api el que fa és crear un objecte FutureBuilder<Apod> això vol dir que aquest objecte no es mostrarà fins que l'acció de l'atribut future no s'hagi completat.

En aquesta acció el que fem és cridar al mètode getData de l'api service.

Quan es completi anirà a l'atribut builder i guardarà tota la informació a la variable snapshot (com a firebase).

Tot seguit comprova si snapshot té informació i si en té podem fer un return de la visibilitat que vulguem. En aquest cas només volem mostrar un text amb el títol.

Per agafar el títol farem snapshot.data!.title. Aquests atributs han de coincidir amb els que haguem definit a la classe Apod.

Exercici

Si tots aquests passos t'han funcionat correctament pots fer els següents exercicis:

1. Si la setmana passada vas acabar el disseny del detall crea un nou widget en el que li passis per paràmetre els diferents camps (imatge, títol, autor, data i explicació)
2. Adapta el widget perquè en comptes de mostrar un text estàtic agafi els que li passis per paràmetre
3. Substitueix la línia del `return Text(snapshot.data!.title);` per a que el return retorni una instància del nou widget creat en el que li passis totes les dades
4. Crea una nova funció a l'Api per a que et torni un llistat d'Apod

   Pots utilitzar la següent URL: https://api.nasa.gov/planetary/apod?api_key=6u8YizBXHawwVvOgDaJXNXJoUoxQ2xTZzxXWC11Y&count=10

   Documentació: https://docs.flutter.dev/cookbook/networking/background-parsing

   I si vols també el següent tutorial d'exemple mostra com recollir una array i mostrar-la en format llistat https://www.codewithflutter.com/flutter-fetch-data-from-api-rest-api-example/

5. Crea una vista seguint l'estètica de card view en la que es vegi la imatge i alguns textos

   https://api.flutter.dev/flutter/material/Card-class.html