//main.

import 'package:app\_pokedex/pokemon.dart';

import 'package:flutter/material.dart';

import 'home\_page.dart';

import 'tela\_detalhes.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

return MaterialApp(

theme: ThemeData(

primarySwatch: Colors.blue,

),

debugShowCheckedModeBanner: false,

initialRoute: '/',

routes:{

'/':(context) => const HomePage(),

'/teladetalhes':(context) => TelaDetalhes(detalhes: ModalRoute.of(context)?.settings.arguments as Pokemon),

},

);

}

}

----------------------------------------------

//home

import 'dart:convert';

import 'package:flutter/material.dart';

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

import 'pokemon.dart';

import 'pokemon\_widgwet.dart';

class HomePage extends StatefulWidget {

const HomePage({Key? key}) : super(key: key);

@override

State<HomePage> createState() => \_HomePageState();

}

class \_HomePageState extends State<HomePage> {

List<Pokemon> pokemonList = [];

bool loading = true;

@override

void initState() {

getPokemon();

super.initState();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Pokedex'),

actions: [

Padding(

padding: const EdgeInsets.all(10),

child: Image.network(

'https://raw.githubusercontent.com/RafaelBarbosatec/SimplePokedex/master/assets/pokebola\_img.png'),

),

],

),

body: ListView.builder(

itemCount: pokemonList.length,

itemBuilder: (context, index) {

return PokemonWidget(item: pokemonList[index]);

}),

);

}

void getPokemon() {

var url = Uri.parse('http://104.131.18.84/pokemon/?limit=50&page=0');

http.get(url).then((response) {

if (response.statusCode == 200) {

Map json = const JsonDecoder().convert(response.body);

setState(() {

pokemonList = json['data'].map<Pokemon>((item) {

return Pokemon.fromJson(item);

}).toList();

});

} else {

return [];

}

});

}

}

--------------------------------------------

//teladetalhes

import 'package:app\_pokedex/pokemon.dart';

import 'package:flutter/material.dart';

class TelaDetalhes extends StatelessWidget {

final Pokemon detalhes;

const TelaDetalhes({Key? key, required this.detalhes}) : super(key: key);

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text(detalhes.name ?? ''),

),

body: Card(

elevation: 15,

color: Colors.white12,

margin: const EdgeInsets.only(bottom: 90, top: 20, right: 25, left: 30),

child: Column(

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Container(

margin: const EdgeInsets.all(20),

decoration: BoxDecoration(

color: Colors.brown[50],

borderRadius: BorderRadius.circular(15),

),

child: Stack(

alignment: Alignment.topRight,

children: [

Padding(

padding: const EdgeInsets.all(15.0),

child: Align(

alignment: Alignment.topRight,

child: Text('#${(detalhes.number ?? '')}'),

),

),

Image.network(

detalhes.thumbnailImage ?? '',

width: 300,

height: 200,

),

],

),

),

Container(

margin: const EdgeInsets.all(20),

child: Text(

detalhes.description ?? '',

style: const TextStyle(

fontSize: 16.0,

),

),

),

const Divider(

indent: 25,

endIndent: 25,

thickness: 1,

color: Colors.green,

),

Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

crossAxisAlignment: CrossAxisAlignment.stretch,

children: [

Container(

padding: const EdgeInsets.only(left: 25),

margin: const EdgeInsets.all(10),

child: const Text(

'Type',

style:

TextStyle(fontSize: 20.0, fontWeight: FontWeight.bold),

),

),

Container(

padding: const EdgeInsets.only(left: 20),

margin: const EdgeInsets.all(10),

child: Text(

'${(detalhes.type ?? '')}',

style: const TextStyle(

fontSize: 1,

),

),

),

const Divider(

indent: 25,

endIndent: 25,

thickness: 2,

color: Colors.green,

),

Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

crossAxisAlignment: CrossAxisAlignment.stretch,

children: [

Container(

padding: const EdgeInsets.only(left: 25),

margin: const EdgeInsets.all(10),

child: const Text(

'Type',

style: TextStyle(

fontSize: 20.0, fontWeight: FontWeight.bold),

),

),

Container(

padding: const EdgeInsets.only(left: 20),

margin: const EdgeInsets.all(5),

child: Text(

'${(detalhes.weakness ?? '')}',

style: const TextStyle(

fontSize: 15.0,

),

),

),

],

),

],

),

],

),

),

);

}

}

------------------------------------

//PokemonWidget

import 'package:flutter/material.dart';

import 'package:app\_pokedex/pokemon.dart';

import 'tela\_detalhes.dart';

class PokemonWidget extends StatelessWidget {

const PokemonWidget({

Key? key,

required this.item,

}) : super(key: key);

final Pokemon item;

@override

Widget build(BuildContext context) {

void goToTelaDetalhes() {

Navigator.of(context).push(MaterialPageRoute(

builder: (context) => TelaDetalhes(detalhes: item)));

}

return Card(

margin: const EdgeInsets.all(20),

child: ListTile(

title: Text(item.name ?? ''),

leading: Image.network(item.thumbnailImage ?? ''),

trailing: Text('#${item.number ?? ''}'),

onTap: () {

goToTelaDetalhes();

},

),

);

}

}

--------------------------------------------

class Pokemon {

List<String>? abilities;

String? detailPageUrl;

int? weight;

List<String>? weakness;

String? number;

int? height;

String? collectiblesSlug;

String? featured;

String? slug;

String? description;

String? name;

String? thumbnailAltText;

String? thumbnailImage;

int? id;

List<String>? type;

Pokemon(

{this.abilities,

this.detailPageUrl,

this.weight,

this.weakness,

this.number,

this.height,

this.collectiblesSlug,

this.featured,

this.slug,

this.description,

this.name,

this.thumbnailAltText,

this.thumbnailImage,

this.id,

this.type});

Pokemon.fromJson(Map<String, dynamic> json) {

this.abilities =

json["abilities"] == null ? null : List<String>.from(json["abilities"]);

this.detailPageUrl = json["detailPageURL"];

this.weight = json["weight"];

this.weakness =

json["weakness"] == null ? null : List<String>.from(json["weakness"]);

this.number = json["number"];

this.height = json["height"];

this.collectiblesSlug = json["collectibles\_slug"];

this.featured = json["featured"];

this.slug = json["slug"];

this.description = json["description"];

this.name = json["name"];

this.thumbnailAltText = json["thumbnailAltText"];

this.thumbnailImage = json["thumbnailImage"];

this.id = json["id"];

this.type = json["type"] == null ? null : List<String>.from(json["type"]);

}

Map<String, dynamic> toJson() {

final Map<String, dynamic> data = new Map<String, dynamic>();

if (this.abilities != null) data["abilities"] = this.abilities;

data["detailPageURL"] = this.detailPageUrl;

data["weight"] = this.weight;

if (this.weakness != null) data["weakness"] = this.weakness;

data["number"] = this.number;

data["height"] = this.height;

data["collectibles\_slug"] = this.collectiblesSlug;

data["featured"] = this.featured;

data["slug"] = this.slug;

data["description"] = this.description;

data["name"] = this.name;

data["thumbnailAltText"] = this.thumbnailAltText;

data["thumbnailImage"] = this.thumbnailImage;

data["id"] = this.id;

if (this.type != null) data["type"] = this.type;

return data;

}

}