import 'dart:convert';

import 'package:flutter/material.dart';

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

void main() {

runApp(const CommentsApp());

}

class CommentsApp extends StatelessWidget {

const CommentsApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'User Comments',

theme: ThemeData(primarySwatch: Colors.blue),

home: const CommentsScreen(),

);

}

}

class Comment {

final int id;

final String name;

final String email;

Comment({

required this.id,

required this.name,

required this.email,

});

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

return Comment(

id: json['id'],

name: json['name'],

email: json['email'],

);

}

}

class CommentsScreen extends StatefulWidget {

const CommentsScreen({super.key});

@override

State<CommentsScreen> createState() => \_CommentsScreenState();

}

class \_CommentsScreenState extends State<CommentsScreen> {

late Future<List<Comment>> \_commentsFuture;

Future<List<Comment>> fetchComments() async {

final response =

await http.get(Uri.parse('https://jsonplaceholder.typicode.com/comments'));

if (response.statusCode == 200) {

final List<dynamic> data = json.decode(response.body);

return data.map((json) => Comment.fromJson(json)).toList();

} else {

throw Exception('Failed to load comments');

}

}

@override

void initState() {

super.initState();

\_commentsFuture = fetchComments();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: const Text('User Comments')),

body: FutureBuilder<List<Comment>>(

future: \_commentsFuture,

builder: (context, snapshot) {

if (snapshot.connectionState == ConnectionState.waiting) {

return const Center(child: CircularProgressIndicator());

} else if (snapshot.hasError) {

return Center(child: Text('Error: ${snapshot.error}'));

} else if (!snapshot.hasData || snapshot.data!.isEmpty) {

return const Center(child: Text('No comments found.'));

} else {

final comments = snapshot.data!;

return ListView.builder(

itemCount: comments.length,

itemBuilder: (context, index) {

final comment = comments[index];

return ListTile(

leading: CircleAvatar(

child: Text(comment.name[0].toUpperCase()),

),

title: Text(comment.name),

subtitle: Text(comment.email),

);

},

);

}

},

),

);

}

}