demonstrate the use of RESTAPI:

Code:

Main.dart:

```
import 'package:flutter/material.dart';
import 'package:restkedar/user.dart';
import 'package:restkedar/remote_services.dart';
void main() {
runApp(const MyApp());
class MyApp extends StatelessWidget {
 const MyApp({super.key});
 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   title: 'Flutter Demo',
   theme: ThemeData(
    colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
    useMaterial3: true,
   home: const MyHomePage(title: 'Flutter Demo Home Page'),
  );
 }
class MyHomePage extends StatefulWidget {
 const MyHomePage({super.key, required this.title});
 final String title;
 @override
State<MyHomePage> createState() => _MyHomePageState();
class MyHomePageState extends State<MyHomePage> {
 List<User>? users:
 var isLoaded = false:
 @override
 void initState() {
  super.initState();
  getData();
 getData() async {
  users = await remoteservice().getUsers();
  if (users != null) {
   setState(() {
    isLoaded = true;
```

```
}
}
@override
Widget build(BuildContext context) {
return Scaffold(
  appBar: AppBar(
   backgroundColor: Theme.of(context).colorScheme.inversePrimary,
   title: Text(widget.title),
  body: Visibility(
   visible: isLoaded,
   child: ListView.builder(
    itemCount: users?.length,
    itemBuilder: (context, index) {
     return Container(
       padding: const EdgeInsets.all(8.0),
       child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
         Text(users![index].name),
         Text(users![index].emailid),
        ],
       ),
      );
   replacement: const Center(
    child: CircularProgressIndicator(),
   ),
  ),
);
```

user.dart:

```
import 'dart:convert';
List<User> userFromJson(String str) => List<User>.from(json.decode(str).map((x) =>
User.fromJson(x)));
String userToJson(List<User> data) => json.encode(List<dynamic>.from(data.map((x) =>
x.toJson())));
class User {
    String name;
    String emailid;

User({
    required this.name,
        required this.emailid,
    });
factory User.fromJson(Map<String, dynamic> json) => User(
    name: json["name"],
```

```
emailid: json["emailid"],
);

Map<String, dynamic> toJson() => {
   "name": name,
   "emailid": emailid,
};
}
```

Remote service.dart:

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

class remoteservice{
   Future<List<User>?> getUsers() async
   {
      var client = http.Client();
      var uri = Uri.parse("https://mocki.io/v1/e4adeb48-dcdb-4287-a72b-c15f52a98e90");
      var response = await client.get(uri);
      if (response.statusCode == 200)
      { var data = response.body;
      return userFromJson(data);
      }
    }
}
```

OUTPUT:

```
🚜 main.dart 🗵 🧥 pubspec.yaml 🔟 🚜 user.dart 🗵 🚜 remote_services.dar
README.md
                                                        Pub get Pu
Flutter commands
       sdk: ^3.5.4
    # Dependencies specify other packages that your package needs in or
     # To automatically upgrade your package dependencies to the latest
# consider running 'flutter pub upgrade --major-versions'. Alternat
     # dependencies can be manually updated by changing the version numb
     # the latest version available on pub.dev. To see which dependencie
     # versions available, run 'flutter pub outdated'
    dependencies:
      flutter:
         sdk: flutter
       # The following adds the Cupertino Icons font to your application
       W Use with the Cupertinolcons class for 10S style icons.
      ecupertino_icons: ^1.0.8
    http:
     dev_dependencies:
       flutter_test:
         sdk: flutter
```

```
JSON GENERATOR

""ecords": [
"nam": "Sankep Valoy",
"emailid": "namyyatanty/Agganil.com"

"name": "John Due",
"emailid": "jahn.dooglenample.com"

"name": "John Due",
"emailid": "jahn.dooglenample.com"

"name": "John Smith",
"emailid": "jahn.smith@comple.com"

"name": "John Smith",
"emailid": "jahn.smith@comple.com"

"name": "Thilly Davis",
"emailid": "enly davis@enample.com"

"name": "Michael Brown",
"emailid": "michael Brown",
"emailid: "michael Brown",
"emailid: "michael Brown",
"emailid: "m
```

```
Trecords": [
{
    "name": "Sandeep Yadav",
    "emailid":

Success! Your API is available
at
    https://macki.io/v1/e4219579-
382b-489b-bf7d-
1c0a5e894475

Create API
```

```
@ quicktype
                                                                                                                                   2004 B 34
                                                                                                                                                                                                                                          Deport 'dort (connect')
                                                                                                                                                                                                                                               ier sierfivalism(tring str) -- liter.frunktin()sur.lecole(str));
                                                                                                                                                                                                                                               tring userToJace(Uter detw) >= jace.eecode;detw.teJace(1);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          C NA Seley
                                                                                                                                                                                                                                              City Speed only
                                                                                                                                                                                                                                                      required this records,
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          (3) he cover known in these
                                                                                                                                                                                                                                                      factory User-from Excepting dynamics [see] as User( excepts [ist Maxcord from [see] "records"] .eep((x) as Accord from [see] (x) as Accord from [
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          CIR General Country restor
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          (I) the appares passed (i)
                                                                                                                                                                                                                                                      Papeloring, dynamic tolone; == (
"records": Liebodynamics, from(records.map((s) == s.tolenn(()),
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 Copy Code
                                                                                                                                                                                                                                                 ons Record (
String tume;
String smalls;
                                                                                                                                                                                                                                                     factory Record from Sun Office density (Sun) \rightarrow Records
                                                                                                                                                                                                                                                    nume: jump['emmilie'],
smaller: jump['emmilie'],
                                                                                                                                                                                                                                                     Pagestring, dynamic talkan() or ( 'vame') came, 'remailed'; smalled, );
```

Flutter Demo Home Page

Name: Sandeep Yadav Email ID: sandyyadav578@gmail.com

Name: Jane Smith

Email ID: jane.smith@example.com

Name: Jane Smith Email ID: jane.smith@example.com

Name: Robert Johnson Email ID: robert.johnson@example.com

Name: Emily Davis

Email ID: emily.davis@example.com

Name: Michael Brown Email ID: michael.brown@example.com