Operation on Rest API

Thursday, March 28, 2024 11:18 PM

The main working through API between server and client site are as follow:
1) GET
2) POST
3) PUT
4) DELETE

Application request to the API through the API request
In response the server sends result in the form of the json, HTML or XML format

Integrate APIs into Flutter App

To integrate an API, we have a few steps that we can follow for our ease:

Step 1: Get API URL and endpoints.

Step 2: Add relevant packages into the app (http, dio, chopper, etc.).

Step 3: Create a constant file that stores URLs and endpoints.

Step 4: Create a model class to parse the JSON.

Step 5: Create a file that handles the API call and write specific methods to fetch data and parse it.

Step 6: Use the data in your app.

Step 3: Create a constant file that stores URLs and endpoints.

Now, it's time to create a simple file named `constants.dart` that will hold all your URLs and endpoints. In our case, we only have one endpoint, but it's a good practice to have a separate file. Your file will look something like the following:

```
class ApiConstants {
  static String baseUrl = 'https://jsonplaceholder.typicode.com';
  static String usersEndpoint = '/users';
}
```

Here, we have created a class called ApiConstants and created 2 static variables so that we can access them without creating an instance of the class like `ApiConstants.baseUrl` .

We have to create the model for the API but with the help of quicktype we can create the any model very quick:

The model format look like this:

```
import 'dart:convert';

List<UserModel> userModelFromJson(String str) =>
List<UserModel>.from(json.decode(str).map((x) => UserModel.fromJson(x)));

String userModelToJson(List<UserModel> data) =>
json.encode(List<dynamic>.from(data.map((x) => x.toJson())));

```

```
class UserModel {
UserModel({
required this.id,
required this.name,
required this.username,
required this.email,
required this.address,
required this.phone,
required this.website,
required this.company,
});

int id;
String name;
String username;
String email;
Address address;
String phone;
String website;
Company company;

factory UserModel.fromJson(Map<String, dynamic> json) => UserModel(
id: json["id"],
name: json["name"],
username: json["username"],
email: json["email"],
address: Address.fromJson(json["address"]),
phone: json["phone"],
website: json["website"],
company: Company.fromJson(json["company"]),
);

Map<String, dynamic> toJson() => {
"id": id,
"name": name,
"username": username,
"email": email,
"address": address.toJson(),
"phone": phone,
"website": website,
"company": company.toJson(),
};
}

class Address {
Address({
required this.street,
required this.suite,
required this.city,
required this.zipcode,
required this.geo,
});

String street;
String suite;
String city;
String zipcode;
Geo geo;

factory Address.fromJson(Map<String, dynamic> json) => Address(
street: json["street"],
suite: json["suite"],
city: json["city"],
zipcode: json["zipcode"],
geo: Geo.fromJson(json["geo"]),
);

Map<String, dynamic> toJson() => {
"street": street,
"suite": suite,
"city": city,
"zipcode": zipcode,
"geo": geo.toJson(),
};
}

class Geo {
Geo({
required this.lat,
required this.lng,
});

String lat;
String lng;

factory Geo.fromJson(Map<String, dynamic> json) => Geo(
lat: json["lat"],
lng: json["lng"],
);

Map<String, dynamic> toJson() => {
"lat": lat,
"lng": lng,
};
}

class Company {
Company({
required this.name,
required this.catchPhrase,
required this.bs,
});

String name;
String catchPhrase;
String bs;

factory Company.fromJson(Map<String, dynamic> json) => Company(
name: json["name"],
catchPhrase: json["catchPhrase"],
bs: json["bs"],
);
```

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

After the model layer we have to use the service layer where we check the reponse code and status code and the many other things:

The api service layer look like this:

import 'dart:developer';	
	import 'package:http/http.dart' as http;
	import 'package:rest_api_example/constants.dart';
	import 'package:rest_api_example/model/user_model.dart';
	class ApiService {
	Future<List<UserModel>?> getUsers() async {
	try {
	var url = Uri.parse(ApiConstants.baseUrl + ApiConstants.usersEndpoint);
	var response = await http.get(url);
	if (response.statusCode == 200) {
	List<UserModel> _model = userModelFromJson(response.body);
	return _model;
	}
	} catch (e) {
	log(e.toString());
	}
	}
	}

Now we have to use the actual data to the UI screen:

import 'package:flutter/material.dart';	
	import 'package:rest_api_example/model/user_model.dart';
	import 'package:rest_api_example/services/api_service.dart';
	class Home extends StatefulWidget {
	const Home({Key? key}) : super(key: key);
	@override
	_HomeState createState() => _HomeState();
	}
	class _HomeState extends State<Home> {
	late List<UserModel>? _userModel = [];
	@override
	void initState() {
	super.initState();
	_getData();
	}
	void _getData() async {

	_userModel = (await ApiService().getUsers())!;
	Future.delayed(const Duration(seconds: 1)).then((value) => setState(() {}));
	}
	@override
	Widget build(BuildContext context) {
	return Scaffold(
	appBar: AppBar(
	title: const Text('REST API Example'),
	),
	body: _userModel == null || _userModel!.isEmpty
	? const Center(
	child: CircularProgressIndicator(),
	)
	: ListView.builder(
	itemCount: _userModel!.length,
	itemBuilder: (context, index) {
	return Card(
	child: Column(
	children: [
	Row(
	mainAxisAlignment: MainAxisAlignment.spaceEvenly,
	children: [
	Text(_userModel![index].id.toString()),
	Text(_userModel![index].username),
	],
	),
	const SizedBox(
	height: 20.0,
	),
	Row(
	mainAxisAlignment: MainAxisAlignment.spaceEvenly,
	children: [
	Text(_userModel![index].email),
	Text(_userModel![index].website),
	],
	),
	],
	),
	);
	},
	),
	);
	}
	}