



PEMROGRAMAN MOBILE

FLUTTER



HTTP REQUEST

- API (*Application Programming Interface*) merupakan suatu aplikasi yang memungkinkan untuk berkomunikasi antar aplikasi lainnya.
- Format dari response API umumnya JSON atau XML, namun format JSON lebih disukai karena sederhana, dan ukuran yang lebih kecil.
- Untuk dapat menggunakan HTTP Request perlu menambahkan `http: ^0.12.2` di dependencies nya pada file `pubspec.yaml`.

Contoh 1

```
1 import 'package:flutter/material.dart';
2 import 'package:http/http.dart' as http;
3 import 'dart:convert';
4
5 Run | Debug | Profile
6 void main() {
7   runApp(MaterialApp(
8     home: BelajarGetData(),
9   )); // MaterialApp
10
11 class BelajarGetData extends StatelessWidget {
12   final String apiUrl = "https://reqres.in/api/users?page=15";
13
14   Future<List<dynamic>> _fetchDataUsers() async {
15     var result = await http.get(apiUrl);
16     return json.decode(result.body)['data'];
17   }
18
19   @override
20   Widget build(BuildContext context) {
21     return Scaffold(
22       appBar: AppBar(
23         title: Text('Belajar GET HTTP'),
24       ), // AppBar
25       body: Container(
26         child: FutureBuilder<List<dynamic>>(
27           future: _fetchDataUsers(),
28           builder: (BuildContext context, AsyncSnapshot snapshot) {
29             if (snapshot.hasData) {
30               return ListView.builder(
31                 padding: EdgeInsets.all(10),
32                 itemCount: snapshot.data.length,
33
34                 itemBuilder: (BuildContext context, int index) {
35                   return ListTile(
36                     leading: CircleAvatar(
37                       radius: 30,
38                       backgroundImage:
39                         NetworkImage(snapshot.data[index]['avatar']),
40                     ), // CircleAvatar
41                     title: Text(snapshot.data[index]['first_name'] + " " +
42                               snapshot.data[index]['last_name']), // Text
43                     subtitle: Text(snapshot.data[index]['email']),
44                   ); // ListTile
45                 }, // ListView.builder
46               ) else {
47                 return Center(child: CircularProgressIndicator());
48               }
49             }, // FutureBuilder
50           ), // Container
51         ); // Scaffold
52       }
53     }
```

```
final String apiUrl = "https://reqres.in/api/users?per_page=15";

Future<List<dynamic>> _fetchDataUsers() async {
  var result = await http.get(apiUrl);
  return json.decode(result.body)['data'];
}
```

```
1 import 'package:flutter/material.dart';
2 import 'package:http/http.dart' as http;
3 import 'dart:convert';
```

- Fungsi `_fetchDataUsers` digunakan untuk memanggil API dari string `apiURL` menggunakan `http.get`. Karena kita ingin menunggu response dari API sebelum melanjutkan ke proses selanjutnya maka kita membutuhkan asynchronous function dengan menambahkan `async` setelah nama function dan `await` pada proses yang ingin kita tunggu.
- `List<dynamic>` digunakan ketika kita tidak ingin mengatur output dari response API maka kita cukup gunakan List dengan tipe `dynamic`. Dalam projek rill sebaiknya kita define terlebih dahulu format output dari mulai tipe data dan struktur datanya untuk meminimalisir error dan memudahkan dalam pembacaan data.
- Karena respon API menggunakan format JSON, maka kita perlu merubahnya menggunakan `json.decode`. Untuk menggunakan `json.decode` wajib import `'dart:convert'`;
- Karena kita hanya membutuhkan data yang ada dalam array `"data"` maka saat return array nya pun hanya mengambil array data saja `(result.body)['data']`;

```
child: FutureBuilder<List<dynamic>>(  
  future: _fetchDataUsers(),  
  builder: (BuildContext context, AsyncSnapshot snapshot) {  
    if (snapshot.hasData) {  
      return ListView.builder(  
        padding: EdgeInsets.all(10),  
        itemCount: snapshot.data.length,  
        itemBuilder: (BuildContext context, int index) {  
          return ListTile(  
            leading: CircleAvatar(  
              radius: 30,  

```

- FutureBuilder, properti future digunakan untuk memanggil fungsi fetchDataUser dan builder untuk menampilkan data API ke UI.
- Dalam penggunaan FutureBuilder sebaiknya selalu tambahkan pengecekan apakah snapshot sudah memiliki data atau belum agar tampilan tidak error jika gagal atau belum mendapatkan feedback dari API yang selanjutnya menampilkan data ke dalam ListView

Contoh 2. main.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:http_request/post_result_model.dart';
3 import 'package:http_request/user_model.dart';
4
5 Run | Debug | Profile
6 void main() {
7   runApp(MyApp());
8 }
9
10 class MyApp extends StatefulWidget {
11   @override
12   _MyAppState createState() => _MyAppState();
13 }
14
15 class _MyAppState extends State<MyApp> {
16   PostResult postResult = null;
17   User user = null; //GET
18   String output = "no data";
19
20   @override
21   Widget build(BuildContext context) {
22     return MaterialApp(
23       home: Scaffold(
24         appBar: AppBar(
25           title: Text("API Demo Http Request"),
26         ), // AppBar
27         body: Center(
28           child: Column(
29             mainAxisAlignment: MainAxisAlignment.spaceEvenly,
30             children: <Widget>[
31               Text((postResult != null) // method POST
32                 ? postResult.id +
33                   " | " +
34                   postResult.name +
35                   " | " +
36                   postResult.job +
37                   " | " +
38                   postResult.created
39                   : "Tidak ada data"), // Text
40               ElevatedButton(
41                 onPressed: () {
42                   PostResult.connectToAPI("Badu", "Dokter").then((value) {
43                     postResult = value;
44                     setState(() {});
45                   });
46                 },
47                 child: Text("POST"),
48               ), // ElevatedButton
49               Text((user != null) //Method GET
50                 ? user.id + " | " + user.name
51                 : "Tidak ada data"), // GET // Text
52               ElevatedButton(
53                 onPressed: () {
54                   User.connectToAPI("2").then((value) {
55                     user = value;
56                     setState(() {});
57                   });
58                 },
59                 child: Text("GET"),
60               ), // ElevatedButton
61               Container(width: 300, child: Center(child: Text(output))), // GET
62               ElevatedButton(
63                 onPressed: () {
64                   User.getUsers("2").then((users) {
65                     output = "";
66                   });
67                 },
68                 child: Text("GET Users"),
69               ), // ElevatedButton
70             ],
71           ),
72         ),
73       ),
74     );
75   }
76 }
```

```
65         for (int i = 0; i < users.length; i++) {
66             output = output + "[ " + users[i].name + " ] ";
67             setState(() {});
68         }
69     });
70 },
71     child: Text("Kumpulan Data"),
72   ), // ElevatedButton
73   ], // <Widget>[]
74 ), // Column
75 ), // Center
76 ), // Scaffold
77 ); // MaterialApp
78 }
79 }
```


Contoh 3. post_result_model.dart

```
1 import 'package:http/http.dart' as http;
2 import 'dart:convert';
3
4 class PostResult {
5   String id;
6   String name;
7   String job;
8   String created;
9
10  PostResult({this.id, this.name, this.job, this.created});
11
12  factory PostResult.createPostResult(Map<String, dynamic> object) {
13    return PostResult(
14      id: object["id"],
15      name: object["name"],
16      job: object["job"],
17      created: object["createdAt"],
18    );
19  }
20  static Future<PostResult> connectToAPI(String name, String job) async {
21    String apiURL = "https://reqres.in/api/users";
22    var apiResult = await http.post(apiURL, body: {"name": name, "job": job});
23    var jsonObject = json.decode(apiResult.body);
24
25    return PostResult.createPostResult(jsonObject);
26  }
27 }
```


Contoh 3. user_model.dart

```
1 import 'package:http/http.dart' as http;
2 import 'dart:convert';
3
4 class User {
5   String id;
6   String name;
7
8   User({this.id, this.name});
9
10  factory User.createUser(Map<String, dynamic> object) {
11    return User(
12      id: object["id"].toString(),
13      name: object["first_name"] + " " + object["last_name"],
14    );
15  }
16
17  static Future<User> connectToAPI(String id) async {
18    String apiURL = "https://reqres.in/api/users/" + id;
19    var apiResult = await http.get(apiURL);
20    var jsonObject = json.decode(apiResult.body);
21    var userData = (jsonObject as Map<String, dynamic>)['data'];
22
23    return User.createUser(userData);
24  }
25
26  //untuk banyak data
27  static Future<List<User>> getUsers(String page) async {
28    String apiURL = "https://reqres.in/api/users?page=" + page;
29
30    var apiResult = await http.get(apiURL);
31    var jsonObject = json.decode(apiResult.body);
32    List<dynamic> listUser = (jsonObject as Map<String, dynamic>)['data'];
33    List<User> users = [];
34
35    for (int i = 0; i < listUser.length; i++) {
36      users.add(User.createUser(listUser[i]));
37    }
38    return users;
39  }
```