Flutter

Training Assignments

Working with REST APIs

Overview

Fetching data from the internet is necessary for most apps. Luckily, Dart and Flutter provide tools, such as the http package, for this type of work.

This lab uses the following steps:

1. Add the http package.
2. Make a network request using the http package.
3. Convert the response into a custom Dart object.
4. Fetch and display the data with Flutter.

Tasks

1. Create Add the http package: **http**: ^0.13.5

```
pubspec.yaml
Flutter commands
40  # Networking
41  http: ^0.13.5
```

2. Additionally, in your AndroidManifest.xml file, add the Internet permission.

   <!-- Required to fetch data from the internet. -->

   <uses-permission android:name="android.permission.INTERNET" />

```
Project Files
~/Documents/R2S/Training/Workspace/
.dart_tool
.idea
android
.gradle
app
src
debug
main
java
res
AndroidManifest.xml
profile
build.gradle
AndroidManifest.xml
Flutter commands
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2     package="r2s.edu.app_demo">
3     <!-- Required to fetch data from the internet. -->
4     <uses-permission android:name="android.permission.INTERNET" />
5     <application
6         android:name="${applicationName}"
7         android:icon="@mipmap/ic_launcher"
8         android:label="app_demo">
```

3. Convert the response into a custom Dart object

```
todo.dart
1  class Todo {
2    final String? id;
3    final String? title;
4    final String? description;
5
6    Todo({this.id, this.title, this.description});
7
8    factory Todo.fromJson(Map<String, dynamic> json) {
9      return Todo(
10         id: json['id'], title: json['title'], description: json['description']);
11   }
12 }
```

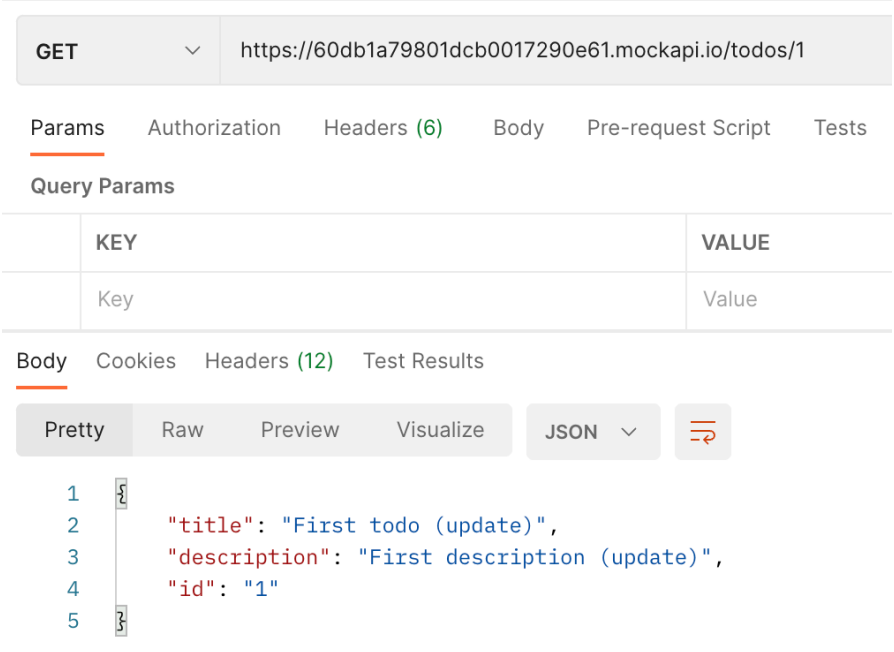
4. Fetch and display the data with Flutter

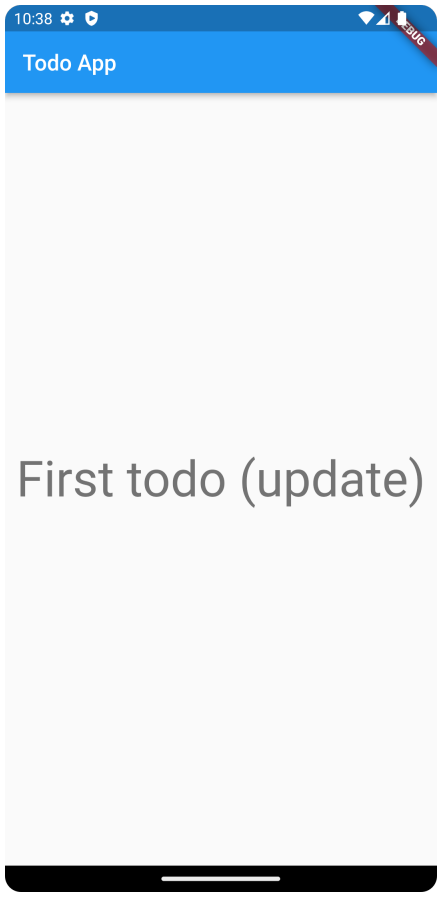
```
5   import 'package:http/http.dart' as http;
6
7   class TodoList extends StatelessWidget {
8     const TodoList({super.key});
9
10    @override
11    Widget build(BuildContext context) {
12      return const MaterialApp(
13        home: _HomePage(),
14      );  // MaterialApp
15    }
16  }
17
18  class _HomePage extends StatefulWidget {
19    const _HomePage({super.key});
20
21    @override
22    State<_HomePage> createState() => _HomePageState();
23  }
24
25  class _HomePageState extends State<_HomePage> {
26    late Future<Todo> futureTodo;
27
28    @override
29    void initState() {
30      super.initState();
31      futureTodo = fetchTodo();
32    }
33
34    @override
35    Widget build(BuildContext context) {
36      return Scaffold(
37        appBar: AppBar(title: const Text('Todo App')),
38        body: Center(
39          child: FutureBuilder(
40            future: futureTodo,
41            builder: (context, snapshot) {
42              if (snapshot.hasError) {
43                return const Text('Retrieve Failed');
44              } else if (snapshot.hasData) {
45                final todo = snapshot.data as Todo;
46                return Text(
47                  '${todo.title}',
```

```
48                  style: Theme.of(context).textTheme.headline3,
49                ); // Text
50              } else {
51                return const CircularProgressIndicator();
52              }
53            },
54          ), // FutureBuilder
55        ), // Center
56      ); // Scaffold
57    }
58
59    Future<Todo> fetchTodo() async {
60      const url = "https://60db1a79801dcb0017290e61.mockapi.io/todos/1";
61      final uri = Uri.parse(url);
62      final response = await http.get(uri);
63
64      if (response.statusCode == 200) {
65        return Todo.fromJson(json.decode(response.body));
66      } else {
67        throw Exception('Failed to load Todo');
68      }
69    }
70  }
```

5. Run the app

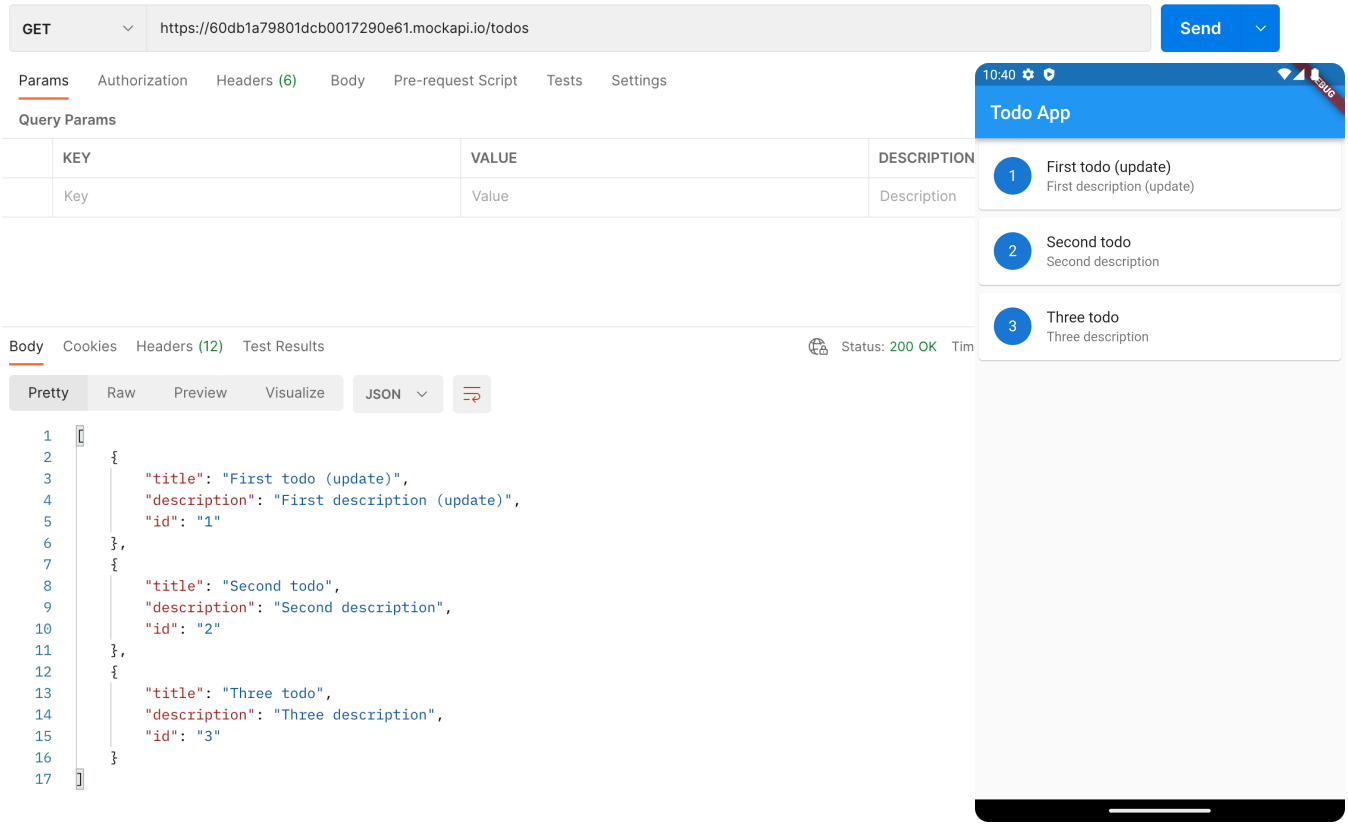
URL: https://60db1a79801dcb0017290e61.mockapi.io/todos/1





Extra tasks

- Display a list of todos
- URL: https://60db1a79801dcb0017290e61.mockapi.io/todos



- Tips:

+ Convert the response into a list of todos

```
List<Todo> parseTodos(String response) {
  final parsed = jsonDecode(response).cast<Map<String, dynamic>>();
  return parsed.map<Todo>((json) => Todo.fromJson(json)).toList();
}

Future<List<Todo>> fetchTodos() async {
  const url = "https://60db1a79801dcb0017290e61.mockapi.io/todos";
  final uri = Uri.parse(url);
  final response = await http.get(uri);

  if (response.statusCode == 200) {
    return parseTodos(response.body);
  } else {
    throw Exception('Failed to load Todo');
  }
}
```

--THE END—