



<https://github.com/Pfkob/BlindreaderApp>

```
// ignore_for_file: depend_on_referenced_packages

import 'dart:async';
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

Future<Album> createAlbum(String title) async {
  final response = await http.post(
    Uri.parse('https://jsonplaceholder.typicode.com/albums'),
    headers: <String, String>{
      'Content-Type': 'application/json; charset=UTF-8',
    },
    body: jsonEncode(<String, String>{
      'title': title,
    }),
  );
  if (response.statusCode == 201) {
    return Album.fromJson(jsonDecode(response.body));
  } else {
    // If the server did not return a 201 CREATED response,
    // then throw an exception.
    throw Exception('Failed to create album.');
```

```
}

class Album {
  final int id;
  final String title;
  const Album({required this.id, required this.title});
  factory Album.fromJson(Map<String, dynamic> json) {
    return Album(
      id: json['id'] as int,
      title: json['title'] as String,
    );
  }
}
```

```
void main() {
  runApp(const MyApp());
}
```

```
class MyApp extends StatefulWidget {
  const MyApp({Key? key}) : super(key: key);
```

```

@override
State<MyApp> createState() {
  return _MyAppState();
}
}

```

```

class _MyAppState extends State<MyApp> {
  final TextEditingController _controller = TextEditingController();
  Future<Album>? _futureAlbum;

```

```

@override
Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Piangfa 6488201',
    theme: ThemeData(
      colorScheme: ColorScheme.fromSwatch(primarySwatch: Colors.deepPurple),
    ),
    home: Scaffold(
      appBar: AppBar(
        title: const Text('Create Data Example'),
      ),
      body: Container(
        alignment: Alignment.center,
        padding: const EdgeInsets.all(8),
        child: (_futureAlbum == null) ? buildColumn() : buildFutureBuilder(),
      ),
    ),
  );
}

```

```

Column buildColumn() {
  return Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      TextField(
        controller: _controller,
        decoration: const InputDecoration(
          labelText: 'Enter Title',
        ),
      ),
      ElevatedButton(
        onPressed: () {
          setState(() {
            _futureAlbum = createAlbum(_controller.text);

```

```
    });  
  },  
  child: const Text('Create Data'),  
),  
],  
);  
}
```

```
FutureBuilder<Album> buildFutureBuilder() {  
  return FutureBuilder<Album>(  
    future: _futureAlbum,  
    builder: (context, snapshot) {  
      if (snapshot.hasData) {  
        return Text(snapshot.data!.title);  
      } else if (snapshot.hasError) {  
        return Text('${snapshot.error}');  
      }  
      return const CircularProgressIndicator();  
    },  
  );  
}
```