

**PEMROGRAMAN MOBILE
PENGANTAR BAHASA PEMROGRAMAN DART -
BAGIAN 14 PRAKTIKUM 3 & 4**



OLEH:

Nama : Agung Rizky S

NIM : 2241720187

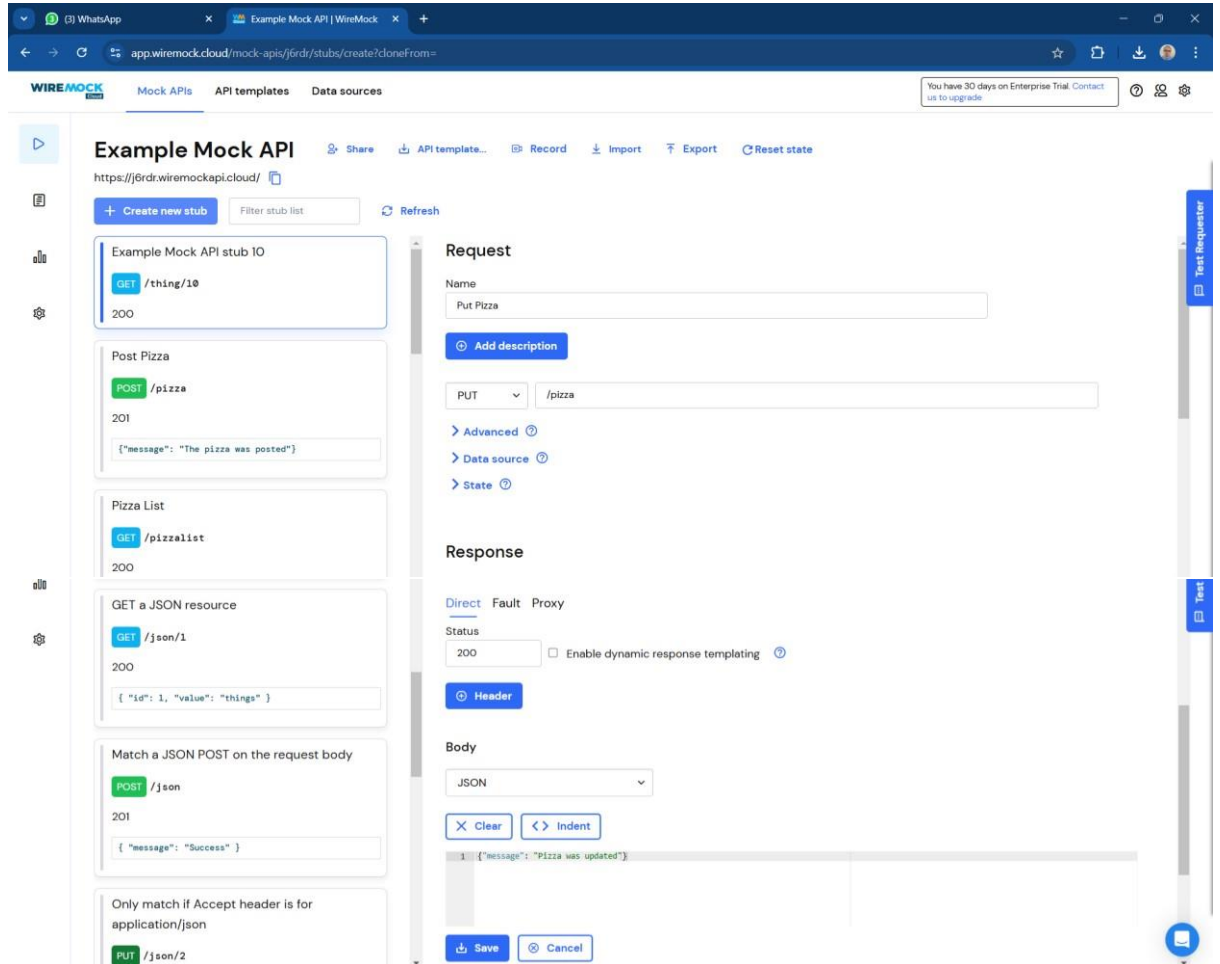
Kelas : TI – 3C

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

2024

Praktikum 3 : PUT-ting data

1. Masuk ke layanan Lab Mock di <https://app.wiremock.cloud/> dan klik bagian Stubs, kemudian, buatlah stub baru.
2. Lengkapi isian seperti gambar berikut:



3. Simpan
4. Di proyek Flutter, tambahkan metode putPizza ke kelas HttpHelper di file http_helper.dart

```
Future<String> putPizza(Pizza pizza) async {
  const putPath = '/pizza';
  String put = json.encode(pizza.toJson());
  Uri url = Uri.https(authority, putPath);
  http.Response r = await http.put(
    url,
    body: put,
  );
  return r.body;
}
```

5. Di kelas PizzaDetailScreen di file pizza_detail.dart, tambahkan dua properti, Pizza dan boolean, dan di konstruktor, atur dua properti tersebut

```
final Pizza pizza;
final bool isNew;
const PizzaDetailScreen(
  {super.key, required this.pizza, required this.isNew});
```

6. Di kelas PizzaDetailScreenState, override metode initState. Bila properti isNew dari kelas PizzaDetail tidak baru, properti ini akan menetapkan konten TextFields dengan nilai objek Pizza yang dilewatkan

```
@override
void initState() {
  if (!widget.isNew) {
    txtId.text = widget.pizza.id.toString();
    txtName.text = widget.pizza.pizzaName;
    txtDescription.text = widget.pizza.description;
    txtPrice.text = widget.pizza.price.toString();
    txtImageUrl.text = widget.pizza.imageUrl;
  }
  super.initState();
}
```

7. Edit metode savePizza sehingga memanggil metode helper.postPizza ketika properti isNew bernilai benar, dan helper.putPizza ketika bernilai salah

```
Future savePizza() async {
  ...

  final result = await (widget.isNew
  ? helper.postPizza(pizza)
  : helper.putPizza(pizza));
  setState(() {
    operationResult = result;
  });
}
```

8. Di file main.dart, di metode build _MyHomePageState, tambahkan properti onTap ke ListTile sehingga saat pengguna mengetuknya, aplikasi akan mengubah rute dan menampilkan layar PizzaDetail, dengan menampilkan data pizza yang ada saat ini dan menjadikan false untuk parameter isNew

```
return ListTile(
  title: Text(pizzas.data![position].pizzaName),
  subtitle: Text(pizzas.data![position].description +
    ' - € ' +
    pizzas.data![position].price.toString()),

  onTap: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => PizzaDetailScreen(
          pizza: pizzas.data![position], isNew: false)),
    );
  });
```

9. Di floatingActionButton, passing data Pizza baru dan menjadikan true untuk parameter isNew ke rute PizzaDetail

```

floatingActionButton: FloatingActionButton(
  child: Icon(Icons.add),
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => PizzaDetailScreen(
          pizza: Pizza(),
          isNew: true,
        )),
    );
  }
),
);

```

10. Jalankan aplikasi. Pada layar utama, ketuk Pizza mana pun untuk menavigasi ke rute PizzaDetail

Edit detail pizza di kolom teks dan tekan tombol Simpan. Anda akan melihat pesan yang menunjukkan bahwa detail pizza telah diperbarui

Jawaban :

Code :

Main.dart :

```

import 'package:flutter/material.dart';
import 'http_helper.dart';
import 'pizza.dart';
import 'pizza_detail.dart';

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

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'API Agung R',
      theme: ThemeData(

```

```

        primarySwatch: Colors.blue,
      ),
      home: const MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key});

  @override
  State<MyHomePage> createState() => _MyHomePage();
}

class _MyHomePage extends State<MyHomePage> {
  Future<List<Pizza>> callPizzas() async {
    HttpHelper helper = HttpHelper();
    List<Pizza> pizzas = await helper.getPizzaList();
    return pizzas;
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Pizza API Agung R'),
      ),
      body: FutureBuilder<List<Pizza>>(
        future: callPizzas(),
        builder: (BuildContext context, AsyncSnapshot<List<Pizza>> snapshot) {
          if (snapshot.hasError) {
            return const Center(
              child: Text('Something went wrong'),
            );
          }
          if (!snapshot.hasData) {
            return const Center(
              child: CircularProgressIndicator(),
            );
          }
          return ListView.builder(
            itemCount: snapshot.data!.length,
            itemBuilder: (BuildContext context, int position) {
              return Dismissible(
                key: Key(snapshot.data![position].id.toString()),

```

[illegible]

```

        builder: (context) => PizzaDetailScreen(
          pizza: Pizza(
            id: 0, // Default values for a new pizza
            pizzaName: '',
            description: '',
            price: 0.0,
            imageUrl: '',
          ),
          isNew: true, // Adding a new pizza
        ),
      ),
    ),
  );
},
),
);
}
}
}

```

Pizza_dettaill.dart :

```

import 'package:flutter/material.dart';
import 'pizza.dart';
import 'httphelper.dart';

class PizzaDetailScreen extends StatefulWidget {
  final Pizza pizza;
  final bool isNew;
  const PizzaDetailScreen(
    {super.key, required this.pizza, required this.isNew}
  );

  @override
  State<PizzaDetailScreen> createState() => _PizzaDetailScreenState();
}

class _PizzaDetailScreenState extends State<PizzaDetailScreen> {
  final TextEditingController txtId = TextEditingController();
  final TextEditingController txtName = TextEditingController();
  final TextEditingController txtDescription = TextEditingController();
  final TextEditingController txtPrice = TextEditingController();
  final TextEditingController txtImageUrl = TextEditingController();
  String operationResult = '';

  @override

```



```

Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Agung R Pizza Detail '),
    ),
    body: Padding(
      padding: const EdgeInsets.all(12),
      child: SingleChildScrollView(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              operationResult,
              style: TextStyle(
                backgroundColor: Colors.green[200],
                color: Colors.black,
              ),
            ),
            const SizedBox(height: 24),
            TextField(
              controller: txtId,
              decoration: const InputDecoration(
                hintText: 'Insert ID',
              ),
            ),
            const SizedBox(height: 24),
            TextField(
              controller: txtName,
              decoration: const InputDecoration(
                hintText: 'Insert Pizza Name',
              ),
            ),
            const SizedBox(height: 24),
            TextField(
              controller: txtDescription,
              decoration: const InputDecoration(
                hintText: 'Insert Description',
              ),
            ),
            const SizedBox(height: 24),
            TextField(
              controller: txtPrice,
              decoration: const InputDecoration(
                hintText: 'Insert Price',
              ),
            ),
          ],
        ),
      ),
    ),
  );
}

```

```

    ),
    const SizedBox(height: 24),
    TextField(
      controller: txtImageUrl,
      decoration: const InputDecoration(
        hintText: 'Insert Image URL',
      ),
    ),
  ),
  const SizedBox(height: 48),
  ElevatedButton(
    onPressed: () {
      postPizza();
    },
    child: const Text('Send Post'),
  ),
],
),
),
),
);
}

@override
void dispose() {
  txtId.dispose();
  txtName.dispose();
  txtDescription.dispose();
  txtPrice.dispose();
  txtImageUrl.dispose();
  super.dispose();
}

@override
void initState() {
  if (!widget.isNew) {
    txtId.text = widget.pizza.id.toString();
    txtName.text = widget.pizza.pizzaName;
    txtDescription.text = widget.pizza.description;
    txtPrice.text = widget.pizza.price.toString();
    txtImageUrl.text = widget.pizza.imageUrl;
  }
  super.initState();
}

Future<void> postPizza() async {

```

```

    HttpHelper helper = HttpHelper();
    Pizza pizza = Pizza(
        id: int.tryParse(txtId.text) ?? 0, // Gunakan default value jika null
        pizzaName: txtName.text,
        description: txtDescription.text,
        price: double.tryParse(txtPrice.text) ??
            0.0, // Gunakan default value jika null
        imageUrl: txtImageUrl.text,
    );

    String result = await helper.postPizza(pizza);

    setState(() {
        operationResult = result;
    });
}

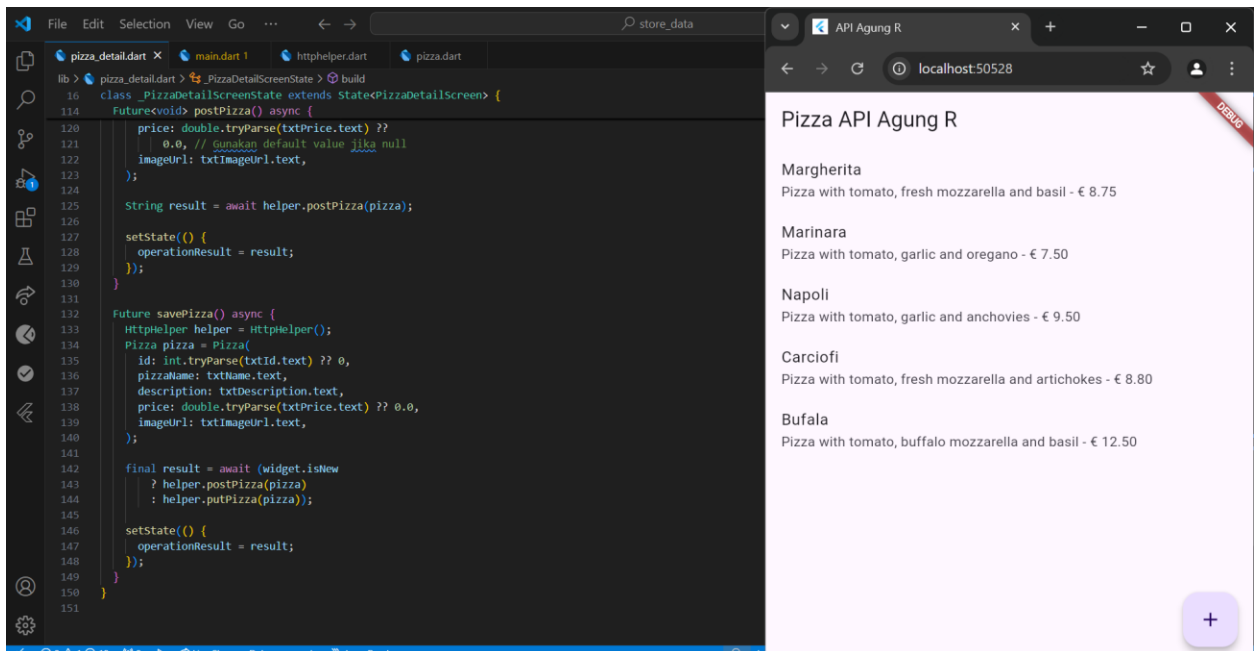
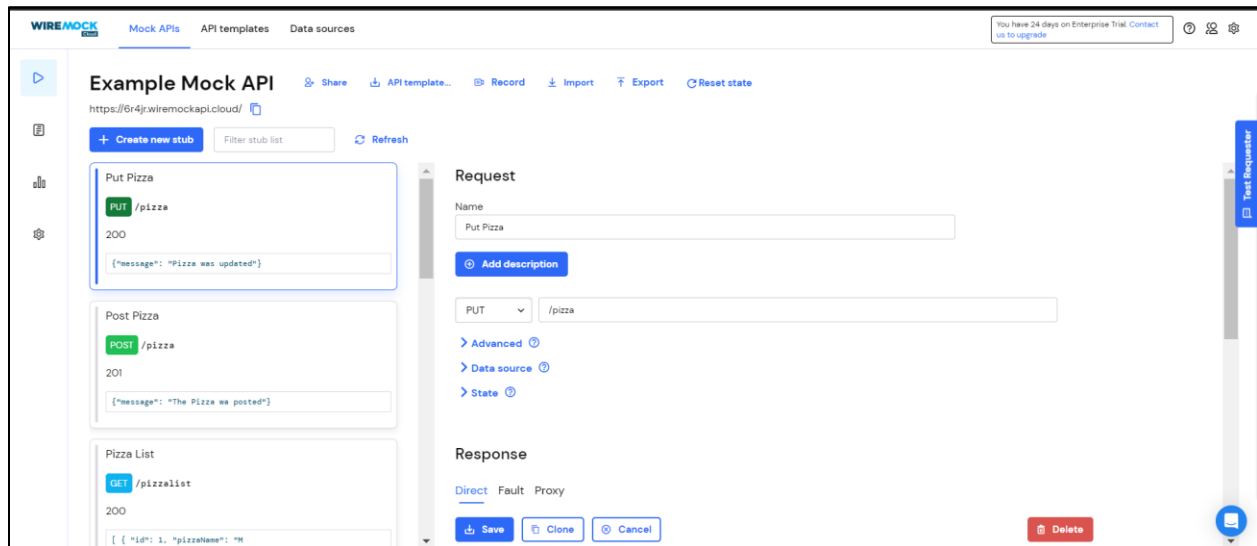
Future savePizza() async {
    HttpHelper helper = HttpHelper();
    Pizza pizza = Pizza(
        id: int.tryParse(txtId.text) ?? 0,
        pizzaName: txtName.text,
        description: txtDescription.text,
        price: double.tryParse(txtPrice.text) ?? 0.0,
        imageUrl: txtImageUrl.text,
    );

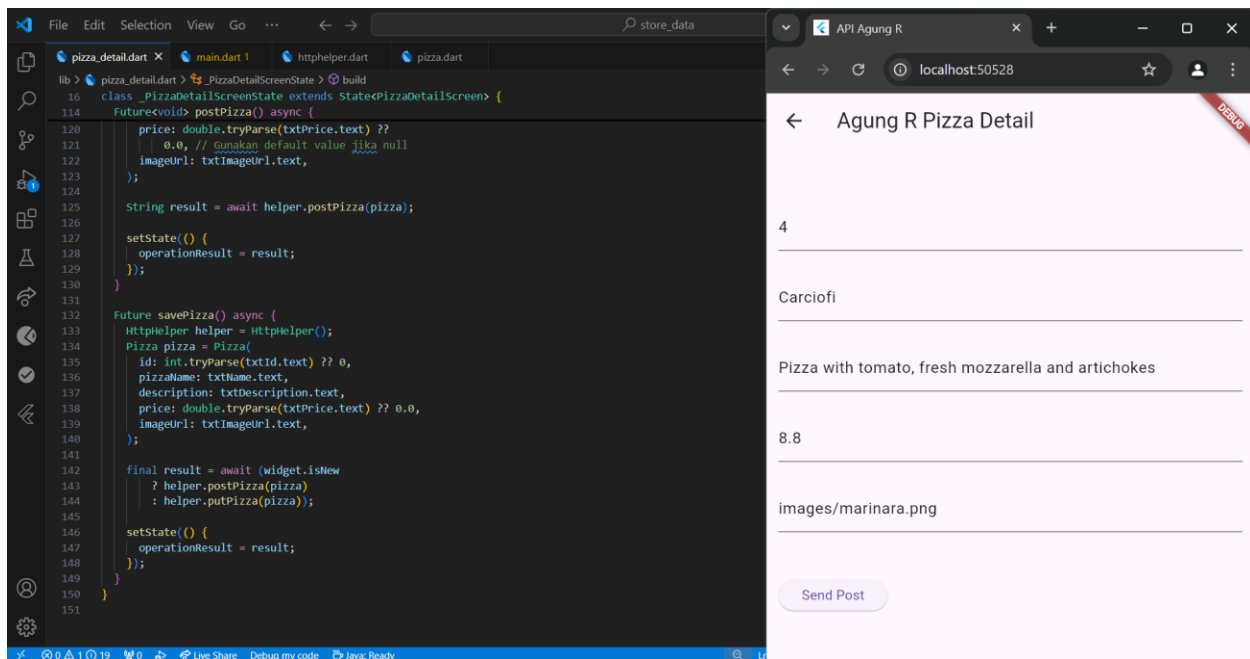
    final result = await (widget.isNew
        ? helper.postPizza(pizza)
        : helper.putPizza(pizza));

    setState(() {
        operationResult = result;
    });
}
}

```

Input :





Praktikum 4 : DELETE-ing data

1. Masuk ke layanan Wiremock di <https://app.wiremock.cloud> dan klik pada bagian Stubs pada contoh API. Kemudian, buatlah sebuah stub baru
2. Lengkapi isian, dengan data berikut:
 - Name: Delete Pizza
 - Verb: DELETE
 - Address: /pizza
 - Status: 200
 - Body Type: json
 - Body: {"message": "Pizza was deleted"}
3. Save the new stub
4. Di proyek Flutter, tambahkan metode deletePizza ke kelas HttpHelper di file http_helper.dart

```
Future<String> deletePizza(int id) async {  
    const deletePath = '/pizza';  
    Uri url = Uri.https(authority, deletePath);  
    http.Response r = await http.delete(  
        url,  
    );  
    return r.body;  
}
```

5. Pada file main.dart, di metode build kelas _MyHomePageState, refaktor itemBuilder dari ListView.builder agar ListTile terdapat dalam widget Dismissible, seperti berikut

```

return ListView.builder(
  itemCount: (pizzas.data == null) ? 0 : pizzas.data.length,
  itemBuilder: (BuildContext context, int position) {
    return Dismissible(
      key: Key(position.toString()),
      onDismissed: (item) {

        HttpHelper helper = HttpHelper();
        pizzas.data!.removeWhere(
          (element) => element.id == pizzas.
data![position].id);

        helper.deletePizza(pizzas.data![position].
id!);

      },
      child: ListTile(...)
    );
  }
);

```

Jalankan aplikasi. Saat Anda menggeser elemen apa pun dari daftar pizza, ListTile akan menghilang

Jawaban :

HttpHelper.main:

```

import 'dart:convert';
import 'dart:io';
import 'package:http/http.dart' as http;
import 'pizza.dart';

class HttpHelper {
  final String authority = '6r4jr.wiremockapi.cloud';
  final String path = 'pizzalist';

  Future<List<Pizza>> getPizzaList() async {
    final Uri url = Uri.https(authority, path);
    final http.Response result = await http.get(url);
    if (result.statusCode == HttpStatus.ok) {
      final jsonResponse = json.decode(result.body) as List;

      List<Pizza> pizzas =
        jsonResponse.map<Pizza>((i) =>
          Pizza.fromJson(i)).toList();
      return pizzas;
    } else {

```

```

        return [];
    }
}

Future<String> postPizza(Pizza pizza) async{
    const postPath = '/pizza' ;
    String post = json.encode(pizza.toJson());
    Uri url = Uri.https(authority, postPath);
    http.Response r = await http.post(
        url,
        body: post,
    );
    return r.body;
}

Future<String> putPizza(Pizza pizza) async{
    const putPath = '/pizza' ;
    String put = json.encode(pizza.toJson());
    Uri url = Uri.https(authority, putPath);
    http.Response r = await http.put(
        url,
        body: put,
    );
    return r.body;
}

Future<String> deletePizza(int id) async {
    const deletePath = '/pizza';
    Uri url = Uri.https(authority, deletePath);
    http.Response r = await http.delete(
        url,
    );
    return r.body;
}
}

```

Main.dart :

```

import 'package:flutter/material.dart';
import 'httphelper.dart';
import 'pizza.dart';
import 'pizza_detail.dart';

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

```



```

}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'API Agung R',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key});

  @override
  State<MyHomePage> createState() => _MyHomePage();
}

class _MyHomePage extends State<MyHomePage> {
  Future<List<Pizza>> callPizzas() async {
    HttpHelper helper = HttpHelper();
    List<Pizza> pizzas = await helper.getPizzaList();
    return pizzas;
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Pizza API Agung R'),
      ),
      body: FutureBuilder<List<Pizza>>(
        future: callPizzas(),
        builder: (BuildContext context, AsyncSnapshot<List<Pizza>> snapshot) {
          if (snapshot.hasError) {
            return const Center(
              child: Text('Something went wrong'),
            );
          }
        }
      )
    );
  }
}

```

```

if (!snapshot.hasData) {
  return const Center(
    child: CircularProgressIndicator(),
  );
}
return ListView.builder(
  itemCount: snapshot.data!.length,
  itemBuilder: (BuildContext context, int position) {
    return Dismissible(
      key: Key(snapshot.data![position].id.toString()),
      onDismissed: (direction) {
        setState(() {
          HttpHelper helper = HttpHelper();
          // Save the deleted pizza details before removing it
          Pizza deletedPizza = snapshot.data![position];

          // Call delete API
          helper.deletePizza(deletedPizza.id);

          // Remove the pizza from the list
          snapshot.data!.removeAt(position);

          // Log the deletion in the console
          print(
            "Deleted Pizza: ${deletedPizza.pizzaName} (ID:
${deletedPizza.id})");
        });
      },
      child: ListTile(
        title: Text(snapshot.data![position].pizzaName),
        subtitle: Text(
          '${snapshot.data![position].description} - €
${snapshot.data![position].price.toStringAsFixed(2)}',
        ),
        onTap: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => PizzaDetailScreen(
                pizza: snapshot.data![position],
                isNew: false,
              ),
            ),
          );
        },
      ),
    );
  },
);

```

```

        ),
      );
    },
  );
),
floatingActionButton: FloatingActionButton(
  child: const Icon(Icons.add),
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => PizzaDetailScreen(
          pizza: Pizza(
            id: 0, // Default values for a new pizza
            pizzaName: '',
            description: '',
            price: 0.0,
            imageUrl: '',
          ),
          isNew: true, // Adding a new pizza
        ),
      ),
    );
  },
),
);
}
}

```

Ouput :

