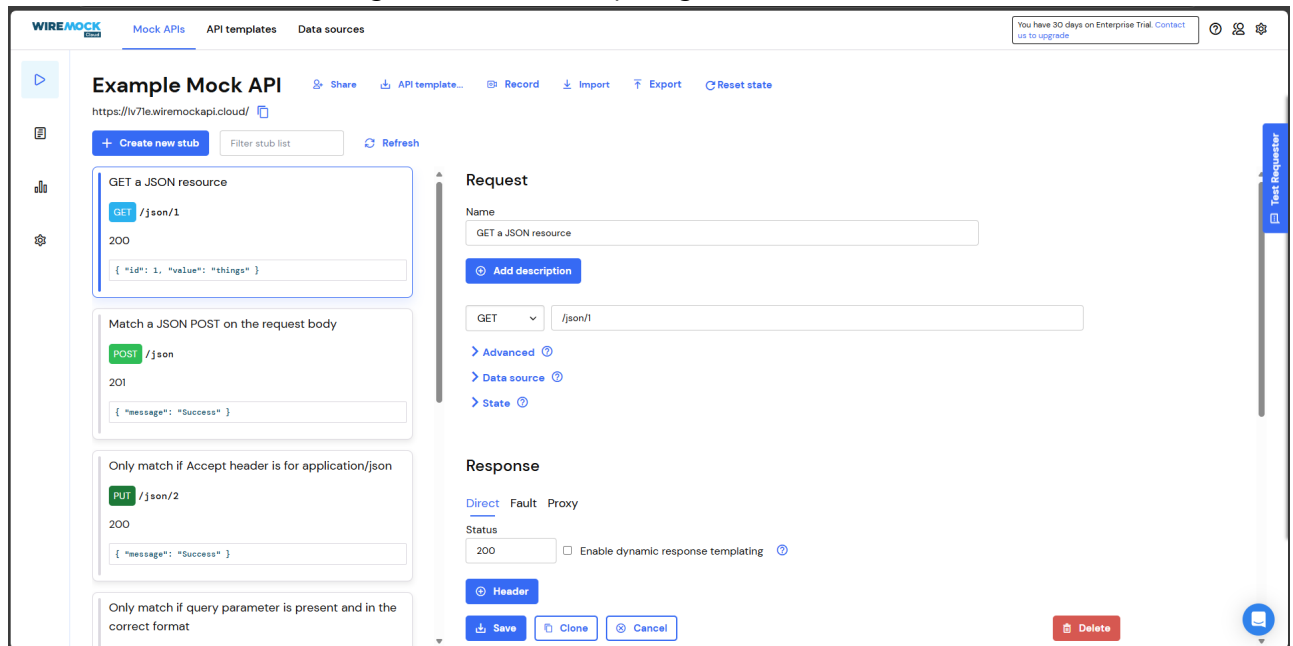


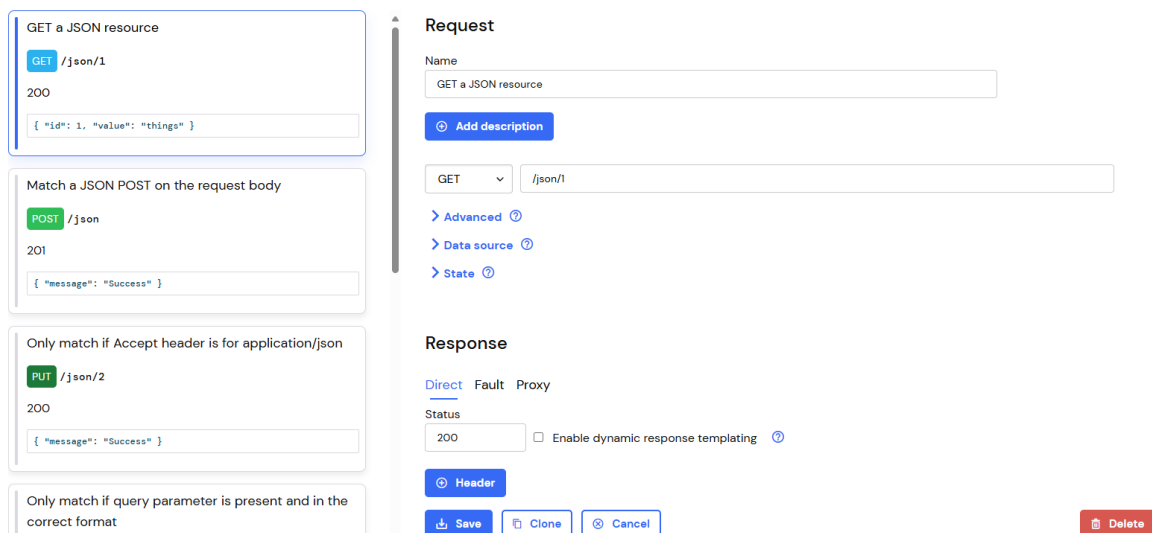
pertemuan 15

Praktikum 1

1. Mendaftarlah ke layanan Lab Mock di <https://app.wiremock.cloud/>. Bisa anda gunakan akun google untuk mendaftar. Jika berhasil mendaftar dan login, akan muncul seperti gambar berikut. Mendaftarlah ke layanan Lab Mock di <https://app.wiremock.cloud/>. Bisa anda gunakan akun google untuk mendaftar. Jika berhasil mendaftar dan login, akan muncul seperti gambar berikut.



2. Di halaman dashboard, klik menu Stubs, kemudian klik entri pertama yaitu "GET a JSON resource". Anda akan melihat layar yang mirip dengan berikut.



3. Klik "Create new stub". Di kolom sebelah kanan, lengkapi data berikut. Namanya adalah "Pizza List", kemudian pilih GET dan isi dengan "/pizzalist". Kemudian, pada bagian Response, untuk status 200, kemudian pada Body pilih JSON sebagai formatnya dan isi

konten JSON dari <https://bit.ly/pizzalist>. Perhatikan gambar berikut.

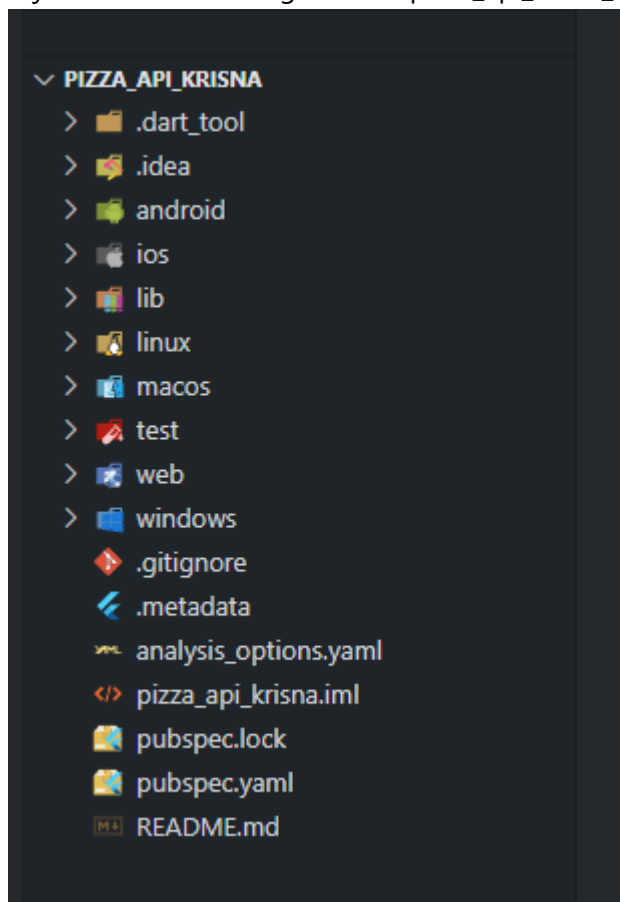
The screenshot shows a REST client interface with the following sections:

- Name:** A text input field containing "Pizza List".
- Add description:** A blue button with a plus icon and the text "Add description".
- Method:** A dropdown menu showing "GET".
- URL:** A text input field containing "/pizzalist".
- Advanced:** A link with a right arrow and a question mark icon.
- Data source:** A link with a right arrow and a question mark icon.
- State:** A link with a right arrow and a question mark icon.
- Response:**
 - Direct:** A link with a blue underline.
 - Fault:** A link.
 - Proxy:** A link.
 - Status:** A text input field showing "200".
 - Enable dynamic response templating:** A checkbox that is unchecked, followed by a question mark icon.
 - Header:** A blue button with a plus icon and the text "Header".
- Body:**
 - JSON:** A dropdown menu showing "JSON".
 - Clear:** A button with a blue border and a close icon.
 - Indent:** A button with a blue border and a double arrow icon.
 - JSON Content:** A text area showing a JSON array with one object:

```
1 [
2   {
3     "id": 1,
4     "pizzaName": "Margherita",
5     "description": "Pizza with tomato, fresh mozzarella and basil",
6     "price": 8.75,
```

4. Tekan tombol SAVE di bagian bawah halaman untuk menyimpan Mock ini. Jika berhasil tersimpan, maka Mock API sudah siap digunakan.

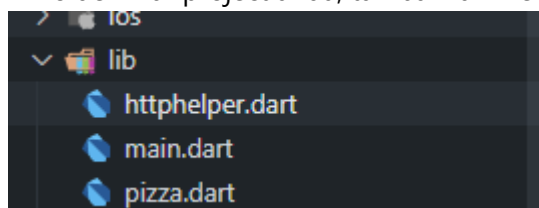
5. Buatlah project flutter baru dengan nama pizza_api_nama_anda, tambahkan depedensi "http" melalui



terminal.

```
PS D:\Kuliahe wong jenius\Semester 5\Mobile\Pertemuan 15\pizza_api_krisna> flutter pub add http
Resolving dependencies...
Downloading packages...
  async 2.11.0 (2.12.0 available)
  boolean_selector 2.1.1 (2.1.2 available)
  characters 1.3.0 (1.3.1 available)
  clock 1.1.1 (1.1.2 available)
  collection 1.18.0 (1.19.1 available)
  fake_async 1.3.1 (1.3.2 available)
  flutter_lints 4.0.0 (5.0.0 available)
+ http 1.2.2
+ http_parser 4.0.2 (4.1.1 available)
  leak_tracker 10.0.5 (10.0.8 available)
  leak_tracker_flutter_testing 3.0.5 (3.0.9 available)
  lints 4.0.0 (5.1.0 available)
  matcher 0.12.16+1 (0.12.17 available)
  material_color_utilities 0.11.1 (0.12.0 available)
  meta 1.15.0 (1.16.0 available)
  path 1.9.0 (1.9.1 available)
  stack_trace 1.11.1 (1.12.0 available)
  string_scanner 1.2.0 (1.4.0 available)
  test_api 0.7.2 (0.7.4 available)
```

6. DI folder "lib" project anda, tambahkan file dengan nama "httphelper.dart".



7. Isi `httphelper.dart` dengan kode berikut. Ubah `"02z2g.mocklab.io"` dengan URL Mock API anda.

```
lib > httphelper.dart > Httphelper > getPizzaList
1  import 'dart:convert';
2  import 'dart:io';
3
4  import 'package:http/http.dart' as http;
5  import 'package:pizza_api_krisna/pizza.dart';
6
7  class Httphelper {
8    final String authority = 'https://lv71e.wiremockapi.cloud/';
9    final String path = 'pizzalist';
10
11    Future<List<Pizza>> getPizzaList() async {
12      final Uri url = Uri.https(authority, path);
13      final result = await http.get(url);
14      if (result.statusCode == HttpStatus.ok) {
15        final jsonResponse = jsonDecode(result.body);
16
17        List<Pizza> pizzas =
18          jsonResponse.map<Pizza>((i) => Pizza.fromJson(i)).toList();
19
20        return pizzas;
21      } else {
22        return [];
23      }
24    }
25  }
26
```

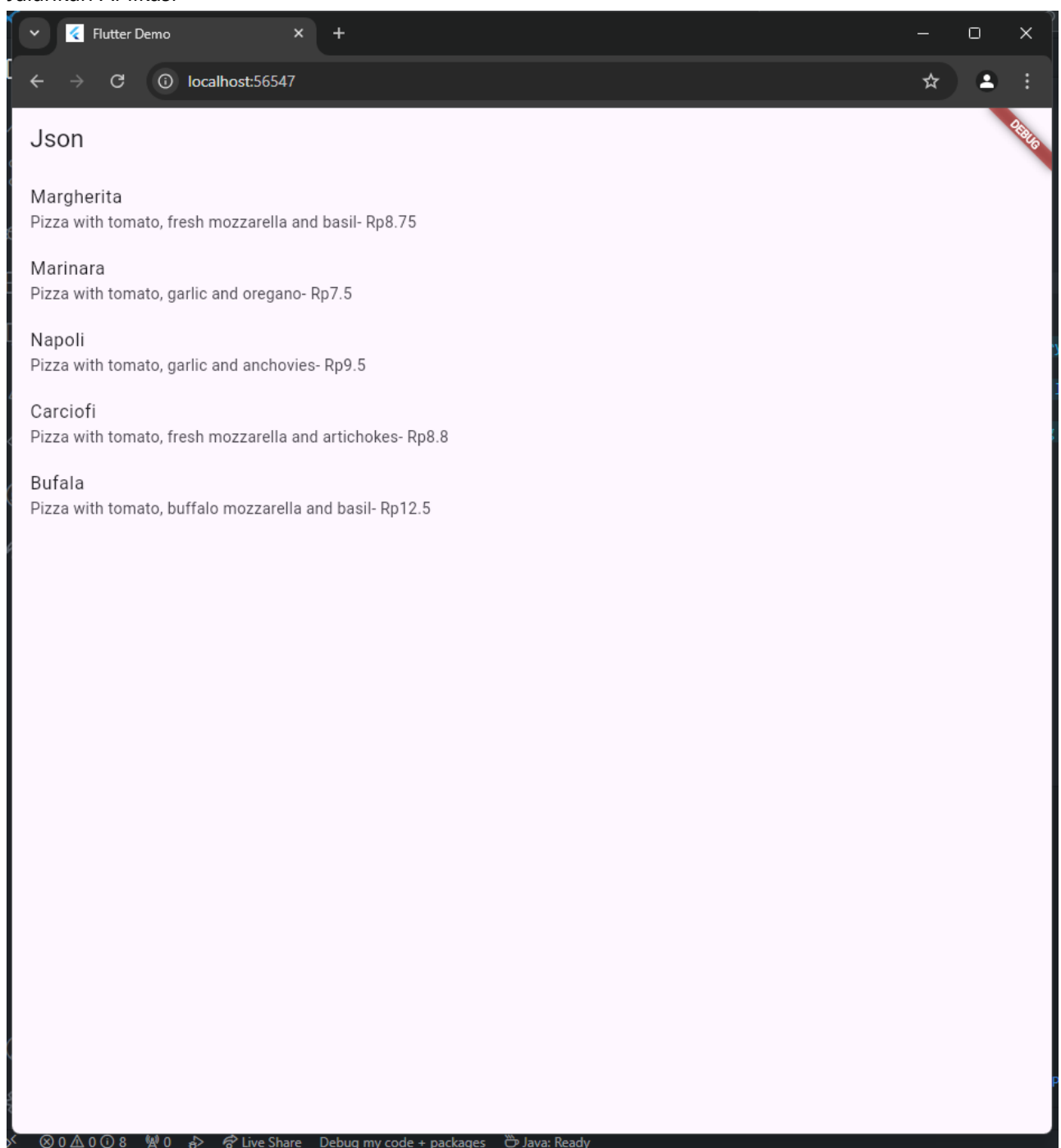
8. Di file `"main.dart"`, di class `_MyHomePageState`, tambahkan metode bernama `"callPizzas"`. Metode ini mengembalikan sebuah Future dari daftar objek Pizza dengan memanggil metode `getPizzaList` dari kelas `HttpHelper`, dengan kode sebagai berikut:

```
47
48  class _MyHomePageState extends State<MyHomePage> {
49    Future<List<Pizza>> callPizzas() async {
50      Httphelper helper = Httphelper();
51      List<Pizza> pizzas = await helper.getPizzaList();
52      return pizzas;
53    }
54
55    @override
56    Widget build(BuildContext context) {
57      // TODO: implement build  TODO: implement build
58      throw UnimplementedError();
59    }
60  }
```

9. Pada metode `build` di class `_MyHomePageState`, di dalam body Scaffold, tambahkan `FutureBuilder` yang membuat `ListView` dari widget `ListTile` yang berisi objek Pizza:

```
    }  
    return ListView.builder(  
      itemCount: (snapshot.data == null)? 0 : snapshot.data!.length,  
      itemBuilder: (context, position){  
        final pizza = snapshot.data![position];  
        return ListTile(  
          title: Text(pizza.pizzaName),  
          subtitle: Text(pizza.description + '- Rp' + pizza.price.toString()),  
        ); // ListTile  
      }  
    ); // ListView.builder  
  ), // FutureBuilder  
); // Scaffold
```

10. Jalankan Aplikasi



Praktikum 2