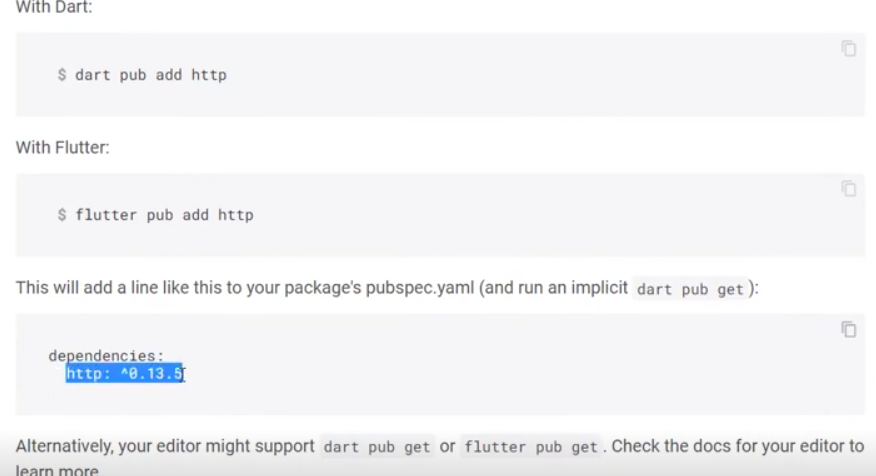
Oque é? Vamos fazer uma conexão fora e retornar para a gente.

Para fazer teste de HTTP vamos precisar de um pacote do Dart.



Ficando dessa forma

name: meu\_app\_testes

description: A sample command-line application.

version: 1.0.0

# repository: https://github.com/my\_org/my\_repo

environment:

  sdk: ^3.8.0

# Add regular dependencies here.

dependencies:

  # path: ^1.8.0

  http: ^1.4.0

dev\_dependencies:

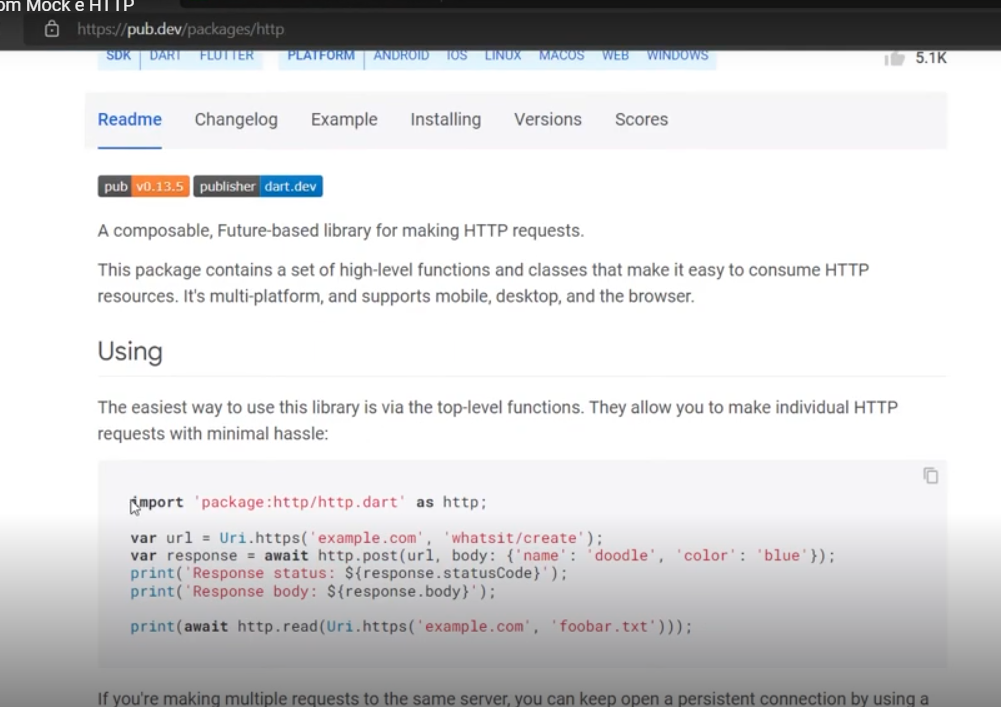
  lints: ^5.0.0

  test: ^1.24.0

Agora podemos seguir

Primeiro vamos importar o pacote do http

https://pub.dev/packages/http/install



import 'package:http/http.dart' as http;

classe com o endPoint

import 'dart:convert';

import 'package:http/http.dart' as http;

class ViaCep {

  Future<Map<dynamic, dynamic>> retornarCEP(String cep) async {

    var url = Uri.parse("https://viacep.com.br/ws/$cep/json/"); // nossa URL

    var retorno = await http.get(

      url,

    ); // manda ele esperar ate que o get esteja com os dados

    var decodedResponse = jsonDecode(utf8.decode(retorno.bodyBytes)) as Map;

    print(decodedResponse);

    return decodedResponse; // Retornar o corpo

  }

}

Teste

  test('Retornar CEP', () async {

    ViaCep viacep = ViaCep();

    var body = await app.retornarCEP("01001000");

    expect(body["bairro"], equals("Sé"));

    expect(body["logradouro"], equals("Praça da Sé"));

  });

}

# Agora vamos Mockar esse teste.

Primeiro precisamos de imports

dependencies:

  # path: ^1.8.0

  http: ^1.4.0

  mockito: ^5.3.0 // Responsavel por mockar os dados

dev\_dependencies:

  lints: ^5.0.0

  test: ^1.24.0

  build\_runner: ^2.4.15 // Pra girar a classe intermedia que vai ajudar a mockar os dados.

Vamos precisar de uma classe intermediaria.

class MockViaCep extends Mock implements ViaCep {

}

Agora vamos precisar de uma anotação

import 'package:http/http.dart' as http;

import 'package:mockito/annotations.dart';

import 'package:mockito/mockito.dart';

@GenerateMocks([MockViaCep])

class ViaCep {

  Future<Map<dynamic, dynamic>> retornarCEP(String cep) async {

    var url = Uri.parse("https://viacep.com.br/ws/$cep/json/"); // nossa URL

    var retorno = await http.get(

      url,

    ); // manda ele esperar ate que o get esteja com os dados

    var decodedResponse = jsonDecode(utf8.decode(retorno.bodyBytes)) as Map;

    print(decodedResponse);

    return decodedResponse; // Retornar o corpo

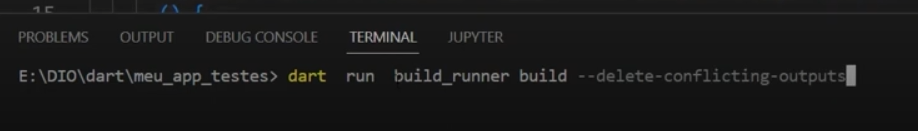
  }

}

class MockViaCep extends Mock implements ViaCep {

}

Agora vamos precisar de um comando para isso



dart run build\_runner build --delete-conflicting-outputs

Com esse código ele vai gerar algumas classes para a gente, inclusive a de testes mock