

Curso Técnico Superior Profissional em: Tecnologias e Programação de Sistemas de Informação

Unidade Curricular: Aplicações Centradas em Redes

2.º Ano/1.º Semestre

Docente: Michael Silva / Hugo Perdigão

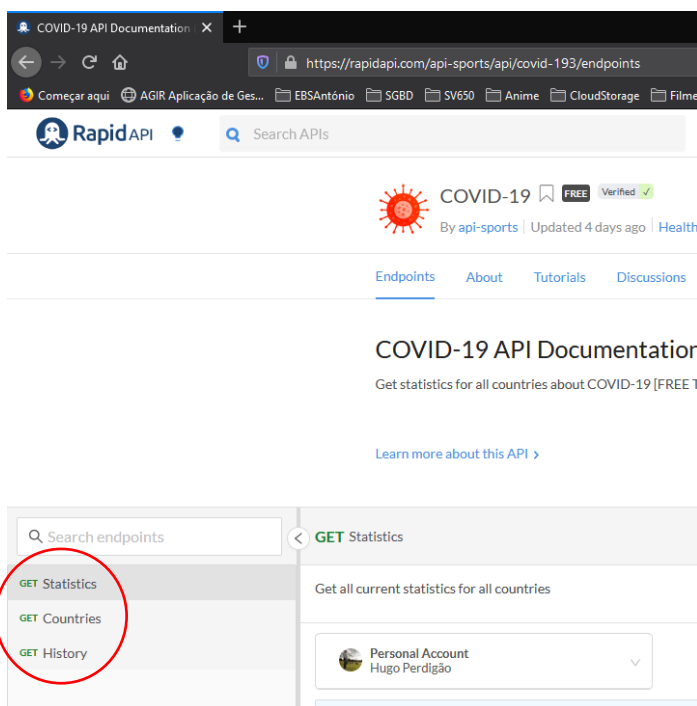
Flutter - Exemplo 6

Objetivos:

- Exemplo prático – Criar uma aplicação para aceder a uma API.

Para este exemplo, vai ser utilizada a API Covid-19 que pode ser acedida em <https://rapidapi.com/api-sports/api/covid-193/> (Requer Registo).

- 1) Definir As Ações da APP, e explorar a API para saber quais os EndPoints a utilizar. A App em questão terá duas funcionalidades
 - a. Listar Países;
 - b. Mostrar estatísticas do Covid no País selecionado;



Esta Api suporta 3 operações como é possível verificar no site:

- Statistics
- Countries
- History

As únicas que interessam é Countries (para listar os Países) e Statistics (para obter as estatísticas).

Para cada Operação estão disponíveis no site snippets de código para várias linguagens, infelizmente não aparece para dart/flutter, mas não é complicado adaptar.

Para utilizar esta aplicação é necessário uma chave que deve ser enviada para o servidor no header.

Isso também pode ser observado nos exemplos do código.

No exemplo que aparece (Node.js) é possível observar qual o url e os parâmetros que devem ser passados nos headers, e independentemente da linguagem esses serão sempre os mesmos.

Code Snippets

(Node.js) Unirest

Install SDK

Copy Code

```
var unirest = require("unirest");

var req = unirest("GET", "https://covid-193.p.rapidapi.com/statistics");

req.headers({
  "x-rapidapi-key": "6d402f4c38msh36eb872a41f08d5p13a97ajsn8758e4fb5e1d",
  "x-rapidapi-host": "covid-193.p.rapidapi.com",
  "useQueryString": true
});
```

O primeiro passo é testar a API, um bom software para fazer isso é o insomnia (<https://insomnia.rest/>) basta colocar o URL e especificar os headers para obter a resposta da API:

- 2) Criar o Projeto Covid-19
- 3) Criar classes para os Objetos baseados nos objetos da API. Neste Caso serão necessários dois Country e Statistics, consultar a resposta da api para identificar as propriedades.
 - a. No caso dos Países é apenas uma String:

The left screenshot shows a JSON response with a list of country names: "Afghanistan", "Albania", "Algeria", "Andorra", and "Angola". The right screenshot shows the Dart code for the `Country` class, which has a single `String` property named `name`.

```

Country.dart X
lib > Country.dart > ...
1 class Country {
2   String name;
3
4   Country({this.name});
5 }

```

- b. No caso das estatísticas tem algumas propriedades:

The left screenshot shows a detailed JSON response for a country's statistics, including continent, country name, population, cases (new, active, critical, recovered, 1M pop, total), deaths (new, 1M pop, total), tests (1M pop, total), day, and time. The right screenshot shows the Dart code for the `Statistics` class, which has many properties corresponding to the JSON fields.

```

statistics.dart X
lib > statistics.dart > Statistics > Statistics
1 class Statistics {
2   String continent;
3   String country;
4   int population;
5   String day;
6   String time;
7   String casesNew;
8   int casesActive;
9   int casesCritical;
10  int casesRecovered;
11  String cases1MPop;
12  int casesTotal;
13  String deathsNew;
14  String deaths1MPop;
15  int deathsTotal;
16  String tests1MPop;
17  int testsTotal;
18
19  Statistics({
20    this.continent,
21    this.country,
22    this.population,
23    this.day,
24    this.time,
25    this.casesNew,
26    this.casesActive,
27    this.casesCritical,
28    this.casesRecovered,
29    this.cases1MPop,
30    this.casesTotal,
31    this.deathsNew,
32    this.deaths1MPop,
33    this.deathsTotal,
34    this.tests1MPop,
35    this.testsTotal,
36  });
37 }

```

- 4) Definir Variáveis Globais. Criar um ficheiro `globals.dart` com uma classe `Globals` que inclui uma série de variáveis static com as definições do programa, o serviço, o url e headers da API, etc.

NOTA: a variáveis `service` poderá ter que ser definida mais tarde uma vez que o ficheiro que a contem ainda não foi criado.

The screenshot shows the `globals.dart` file with the `Globals` class. It includes static variables for the `C19Service` instance, API URL, headers, and the next route.

```

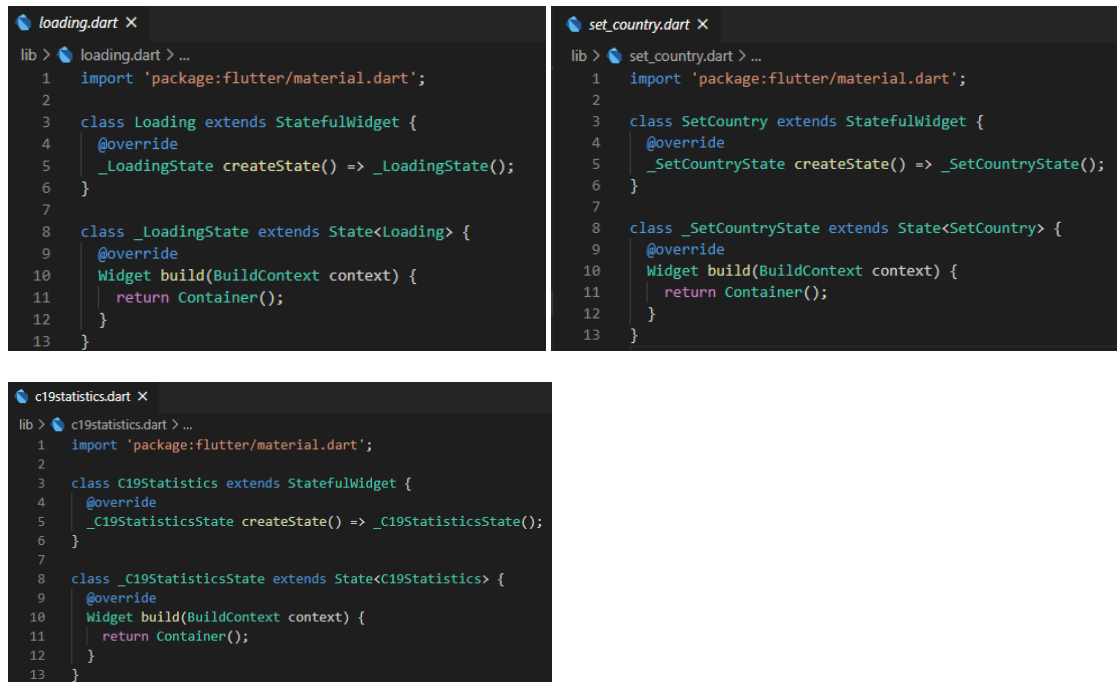
globals.dart X
lib > globals.dart > ...
1 import 'package:exemplo_06/c19service.dart';
2
3 class Globals {
4   static C19Service service = C19Service();
5   static String apiUrl = 'https://covid-193.p.rapidapi.com';
6   static Map<String, String> apiHeaders = {
7     'x-rapidapi-key': '6d402f4c38msh36eb872a41f08d5p13a97ajsn8758e4fb5e1d',
8     'x-rapidapi-host': 'covid-193.p.rapidapi.com',
9     'useQueryString': 'true'
10  };
11
12  static String nextRoute = "/Countries";
13 }

```

A utilização deste ficheiro evita a passagem de variáveis entre ecrãs, bastando em qualquer ecrã aceder à classe `Globals` para ir buscar o serviço por exemplo.

5) Definir as Ecrãs e Routes

- a. Criar um Stateful Widget para cada ecrã (Loading, SetCountry e C19Statistics)



```

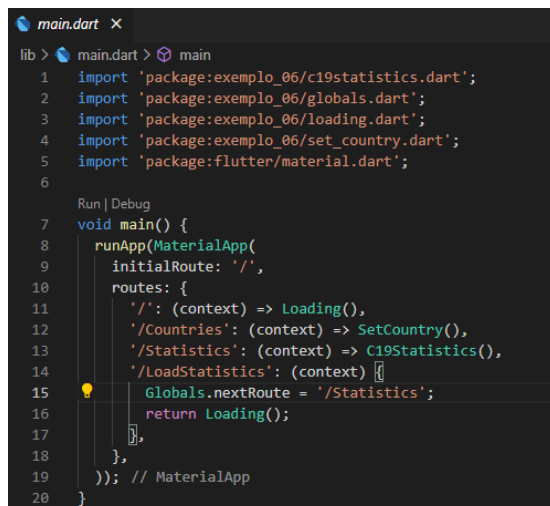
loading.dart
lib > loading.dart > ...
1 import 'package:flutter/material.dart';
2
3 class Loading extends StatefulWidget {
4   @override
5   _LoadingState createState() => _LoadingState();
6 }
7
8 class _LoadingState extends State<Loading> {
9   @override
10  Widget build(BuildContext context) {
11    return Container();
12  }
13 }

set_country.dart
lib > set_country.dart > ...
1 import 'package:flutter/material.dart';
2
3 class SetCountry extends StatefulWidget {
4   @override
5   _SetCountryState createState() => _SetCountryState();
6 }
7
8 class _SetCountryState extends State<SetCountry> {
9   @override
10  Widget build(BuildContext context) {
11    return Container();
12  }
13 }

c19statistics.dart
lib > c19statistics.dart > ...
1 import 'package:flutter/material.dart';
2
3 class C19Statistics extends StatefulWidget {
4   @override
5   _C19StatisticsState createState() => _C19StatisticsState();
6 }
7
8 class _C19StatisticsState extends State<C19Statistics> {
9   @override
10  Widget build(BuildContext context) {
11    return Container();
12  }
13 }

```

- b. Criar as Routes e ecrãs na função main. A variável global nextRoute está definida inicialmente com o valor 'Countries' pelo que ao iniciar a aplicação será mostrado o ecrã Loading que será substituído pelo SetCountry, na route '/LoadStatistics' é carregado na mesma o ecrã Loading mas antes disso a variável nextRoute é definida para '/Statistics' para que seja este ecrã a substituir o Loading nesta situação.



```

main.dart
lib > main.dart > main
1 import 'package:exemplo_06/c19statistics.dart';
2 import 'package:exemplo_06/globals.dart';
3 import 'package:exemplo_06/loading.dart';
4 import 'package:exemplo_06/set_country.dart';
5 import 'package:flutter/material.dart';
6
7 void main() {
8   runApp(MaterialApp(
9     initialRoute: '/',
10    routes: {
11      '/': (context) => Loading(),
12      '/Countries': (context) => SetCountry(),
13      '/Statistics': (context) => C19Statistics(),
14      '/LoadStatistics': (context) => Loading(),
15    },
16    Global.nextRoute = '/Statistics';
17    return Loading();
18  ),
19  ); // MaterialApp
20 }

```

6) Criar o serviço que irá aceder à API no ficheiro c19service.dart

- Contem as variáveis: países que irá conter a lista de países, estatísticas que irá guardar as estatísticas do país selecionado e paisAtual que define o país selecionado.

```
c19service.dart X
lib > c19service.dart > C19Service > getStatistics
1  import 'dart:convert';
2
3  import 'package:exemplo_06/country.dart';
4  import 'package:exemplo_06/globals.dart';
5  import 'package:exemplo_06/statistics.dart';
6  import 'package:http/http.dart';
7
8  class C19Service {
9    List<Country> pais = List<Country>();
10   Statistics estatisticas;
11   Country paisAtual;
12 }
```

b. A função para preencher a lista de países

```
12
13 //Função para aceder à API e preencher a lista de países
14 Future<void> getCountries() async {
15   if (this.paises.isNotEmpty) return;
16
17   String route = Globals.apiUrl + "/countries";
18   Response response = await get(route, headers: Globals.apiHeaders);
19
20   Map rawData = jsonDecode(response.body);
21   List<dynamic> data = rawData['response'];
22
23   paises.clear();
24
25   for (int i = 0; i < data.length; i++) {
26     Country pais = Country(name: data[i]);
27     paises.add(pais);
28   }
29 }
```

Como a lista de países vai ser preenchida no início, este pedido só precisa ser executado uma vez, se estiver preenchido, nada é executado.

É necessário analisar a resposta do servidor para conseguir um tipo de dados que seja compatível neste caso é um Map, mas pode variar dependendo da API.

c. A função que preenche as estatísticas

```
30
31 //Função para aceder à API e preencher as estatísticas de um país
32 Future<void> getStatistics() async {
33   if (this.paisAtual == null) {
34     return;
35   }
36
37   String route =
38     Globals.apiUrl + "/statistics?country=" + this.paisAtual.name;
39   Response response = await get(route, headers: Globals.apiHeaders);
40
41   Map rawData = jsonDecode(response.body);
42   List<dynamic> data = rawData['response'];
43
44   //como só existe um resultado não é necessário ciclo for
45   Map paisData = data[0];
46   Map casesData = paisData['cases'];
47   Map deathsData = paisData['deaths'];
48   Map testsData = paisData['tests'];
49
50   this.estatisticas = Statistics(
51     continent: paisData['continent'],
52     country: paisData['country'],
53     population: paisData['population'],
54     casesNew: casesData['new'],
55     casesActive: casesData['active'],
56     casesCritical: casesData['critical'],
57     casesRecovered: casesData['recovered'],
58     cases1MPop: casesData['1M_pop'],
59     casesTotal: casesData['total'],
60     deathsNew: deathsData['new'],
61     deaths1MPop: deathsData['1M_pop'],
62     deathsTotal: deathsData['total'],
63     tests1MPop: testsData['1M_pop'],
64     testsTotal: testsData['total'],
65     day: paisData['day'],
66     time: paisData['time']);
67 }
68 }
```

Se não existir país atual, não é possível ir buscar estatísticas, por isso nada é executado.

7) Criar o Widget CountryCard que cria um widget para mostrar um país

```
country_card.dart X
lib > country_card.dart > ...
1 import 'package:exemplo_06/country.dart';
2 import 'package:flutter/material.dart';
3
4 class CountryCard extends StatelessWidget {
5   final Country pais;
6   final Function showStats;
7
8   CountryCard({this.pais, this.showStats});
9
10  @override
11  Widget build(BuildContext context) {
12    return Card(
13      margin: EdgeInsets.all(4),
14      color: Colors.lightBlueAccent[600],
15      child: ListTile(
16        title: Text(this.pais.name),
17        leading: Icon(Icons.flag),
18        onTap: showStats,
19      )); // ListTile // Card
20  }
21 }
```

8) Criar o Widget StatisticsContainer

```
StatisticsContainer.dart X
lib > StatisticsContainer.dart > StatisticsContainer > build
1 import 'package:exemplo_06/statistics.dart';
2 import 'package:flutter/material.dart';
3
4 class StatisticsContainer extends StatelessWidget {
5   final Statistics statistics;
6
7   StatisticsContainer({this.statistics});
8
9   @override
10  Widget build(BuildContext context) {
11    return Container(
12      child: Column(
13        crossAxisAlignment: CrossAxisAlignment.stretch,
14        children: <Widget>[
15          Padding(
16            padding: const EdgeInsets.all(8.0),
17            child: Text(
18              "${this.statistics.country} ( ${this.statistics.continent})",
19              style: TextStyle(
20                fontSize: 22,
21                fontWeight: FontWeight.bold,
22                color: Colors.blueAccent[900]), // TextStyle
23            ), // Text
24          ), // Padding
25          SizedBox(
26            height: 5,
27          ), // SizedBox
28          Padding(
29            padding: const EdgeInsets.all(8.0),
30            child: Text("População: ${this.statistics.population}",
31              style: TextStyle(fontSize: 19, color: Colors.black)), // Text
32          ), // Padding
33          SizedBox(
34            height: 20,
35          ), // SizedBox
36          Padding(
37            padding: const EdgeInsets.all(8.0),
38            child: Text("${this.statistics.day}",
39              style: TextStyle(
40                fontSize: 25,
41                color: Colors.black,
42                fontWeight: FontWeight.bold)), // TextStyle // Text
43          ), // Padding
44          SizedBox(
45            height: 20,
46          ), // SizedBox
47          Padding(
48            padding: const EdgeInsets.all(8.0),
49            child: Text(
50              "Casos Ativos: ${this.statistics.casesActive} " +
51              "${this.statistics.casesNew}",
52              style: TextStyle(fontSize: 18, color: Colors.red)), // Text
53          ), // Padding
54          SizedBox(
55            height: 5,
56          ), // SizedBox
57          Padding(
58            padding: const EdgeInsets.all(8.0),
59            child: Text("Em estado Crítico: ${this.statistics.casesCritical}",
60              style: TextStyle(fontSize: 18, color: Colors.red)), // Text
61          ), // Padding
62          SizedBox(
63            height: 5,
64          ), // SizedBox
65          Padding(
66            padding: const EdgeInsets.all(8.0),
67            child: Text("Recuperados: ${this.statistics.casesRecovered}",
68              style: TextStyle(fontSize: 18, color: Colors.green)), // Text
69          ), // Padding
70          SizedBox(
71            height: 5,
72          ), // SizedBox
73          Padding(
74            padding: const EdgeInsets.all(8.0),
75            child: Text(
76              "Mortes: ${this.statistics.deathsTotal} " +
77              "${this.statistics.deathsNew}",
78              style: TextStyle(fontSize: 18, color: Colors.red)), // Text
79          ), // Padding
80          SizedBox(
81            height: 5,
82          ), // SizedBox
83          ], // <Widget>[]
84        )); // Column // Container
85  }
86 }
87 }
```

9) Finalizar os ecrãs, Loading, SetCountry e C19statistics

```

loading.dart X
lib > loading.dart > _LoadingState
1 import 'package:exemplo_06/globals.dart';
2 import 'package:flutter/material.dart';
3 import 'package:flutter_spinkit/flutter_spinkit.dart';
4
5 class Loading extends StatefulWidget {
6   @override
7   _LoadingState createState() => _LoadingState();
8 }
9
10 class _LoadingState extends State<Loading> {
11   Future<void> setupService() async {
12     await Globals.service.getCountries();
13     await Globals.service.getStatistics();
14
15     Navigator.pushReplacementNamed(context, Globals.nextRoute);
16   }
17
18   @override
19   void initState() {
20     super.initState();
21     setupService();
22   }
23
24   @override
25   Widget build(BuildContext context) {
26     return Scaffold(
27       backgroundColor: Colors.amber[300],
28       body: Center(
29         child: SpinKitCubeGrid(
30           color: Colors.red,
31           size: 50.0,
32         )); // SpinKitCubeGrid // Center // Scaffold
33   }
34 }

c19statistics.dart X
lib > c19statistics.dart > ...
1 import 'package:exemplo_06/StatisticsContainer.dart';
2 import 'package:exemplo_06/globals.dart';
3 import 'package:flutter/material.dart';
4
5 class C19Statistics extends StatefulWidget {
6   @override
7   _C19StatisticsState createState() => _C19StatisticsState();
8 }
9
10 class _C19StatisticsState extends State<C19Statistics> {
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold(
14       backgroundColor: Colors.amber[300],
15       appBar: AppBar(
16         title: Text("Estatísticas Covid19"),
17         backgroundColor: Colors.lightBlueAccent[600],
18       ), // AppBar
19       body: Column(
20         children: <Widget>[
21           Expanded(
22             child:
23               StatisticsContainer(statistics: Globals.service.estatisticas))
24         ], // <Widget>[]
25       ), // Column
26     ); // Scaffold
27   }
28 }

```

```

set_country.dart X
lib > set_country.dart > _SetCountryState > build
1 import 'package:exemplo_06/country.dart';
2 import 'package:exemplo_06/country_card.dart';
3 import 'package:exemplo_06/globals.dart';
4 import 'package:flutter/material.dart';
5
6 class SetCountry extends StatefulWidget {
7   @override
8   _SetCountryState createState() => _SetCountryState();
9 }
10
11 class _SetCountryState extends State<SetCountry> {
12   Future<void> showStats(Country pais) async {
13     Globals.service.paisAtual = pais;
14     await Globals.service.getStatistics();
15
16     Navigator.pushNamed(context, '/LoadStatistics');
17   }
18
19   @override
20   Widget build(BuildContext context) {
21     return Scaffold(
22       backgroundColor: Colors.amber[300],
23       appBar: AppBar(
24         title: Text("Escolha o Pais"),
25         backgroundColor: Colors.lightBlueAccent[600],
26       ), // AppBar
27       body: Column(
28         children: <Widget>[
29           Expanded(
30             child: ListView.builder(
31               itemCount: Globals.service.países.length,
32               itemBuilder: (context, index) => CountryCard(
33                 pais: Globals.service.países[index],
34                 showStats: () {
35                   this.showStats(Globals.service.países[index]);
36                 }, // CountryCard
37             ), // ListView.builder // Expanded
38         ], // <Widget>[]
39       ), // Column
40     ); // Scaffold
41   }
42 }

```