

Weather Forecast App Using API

PROJECT REPORT

**Open Source Development for Google
Applications(EXC1081)**

**Submitted To:-
DSC VIT**

By:-

**Mitarth Jain
17BCE0765**



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

March,2019

ACKNOWLEDGEMENTS

A deepest gratitude and sincere thanks to DSC VIT in helping us complete our Project with several learning outcomes. We feel deeply obliged to thank the SCOPE (School of Computer Science and Engineering) Department and the VIT University for their services rendered and for giving us an opportunity to make such projects along with our studies at the University.

(Mitarth Jain)

Reg. No. 17BCE0765

Abstract

The purpose of the project entitled as “Weather Forecast App Using API” is to display the weather of a place using the API. This project will display the current weather and the upcoming days forecast in different activity. It consists of temperature , city , date and time information. This will help the user to easily know about the weather of any place just by entering the city name.

Introduction

The project “Weather Forecast App Using API” is Android or IOS app which displays the weather information for any place. It has been made using Flutter & Dart.

Flutter is an open-source mobile application development framework created by Google. It is used to develop applications for Android and iOS. Dart is general purpose programming language which is optimized to develop client-side web or mobile applications.

Using API from the weather forecast sites we can get the data related to weather such as Temperature, Humidity, Maximum Temperature, Minimum Temperature for a place that the user has input. This application has different activities to show the various result according to user choice. The User can view the current temperature in one activity and the upcoming days forecast in other activity.

Methodology

This app has been made using Flutter framework which allows us to build cross-platform applications and simpler in developing. It is generally used when we want widgets in our application such as text, images, menu or any function to display any output. Flutter uses dart language for its application development, which is also easier to understand and write.

In this application following flutter packages are used:

`import 'package:flutter/material.dart';` - It is material library used for Flutter Widget Material design.

`import 'dart:async';` - It is dart Asynchronous support used to execute Future operations which are scheduled later.

`import 'dart:convert';` - It is converter library which converts the API data to JSON file for further operations.

`import 'package:http/http.dart' as http;` - This library is used to get data from HTTPS site and to get a response from particular URL.

OpenWeatherMap API is used in this application to display the current weather and upcoming days forecast. API (Application program interface) is used to access the particular data or information from another site. It makes things easier because we don't have to write a separate program to just access the data, we can just use the API and perform further operations with the data.

In this flutter application various terminologies are used such as :-

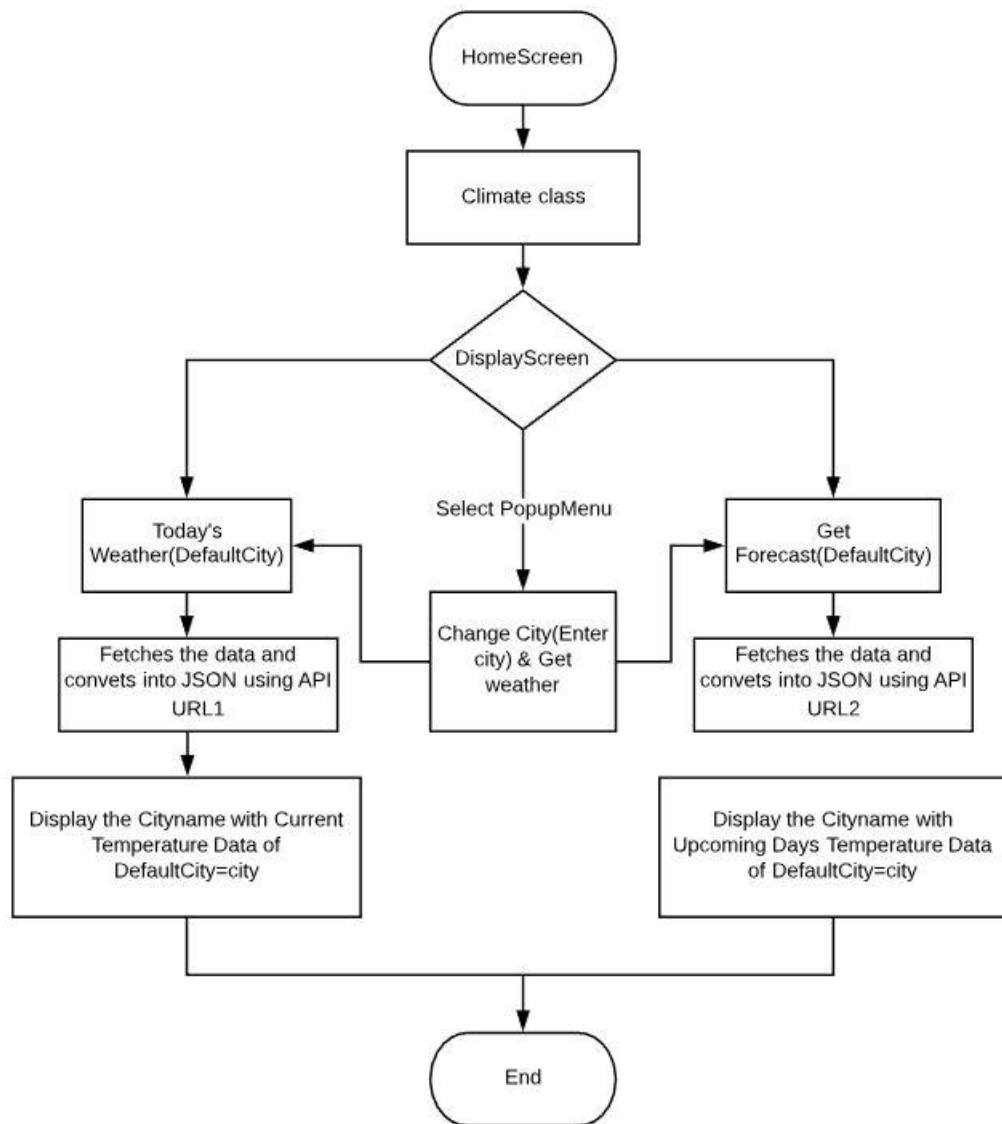
Stateful Widget - A stateful widget is a widget that describes part of the user interface by building a constellation of other widgets that describe the user interface more concretely. The building process continues recursively until the description of the user interface is fully concrete. Stateful widgets are useful when the part of the user interface you are describing can change dynamically. It has mutable state.

Stateless Widget - A stateless widget is a widget that describes part of the user interface by building a constellation of other widgets that describe the user interface more concretely. The building process continues recursively until the description of the user interface is fully concrete. Stateless widget are useful when the part of the user interface you are describing does not depend on anything other than the configuration information in the object itself. It does not require mutable state.

Future - A future is a `Future<T>` object, which represents an asynchronous operation that produces a result of type T. If the result isn't a usable value, then the future's type is `Future<void>`. When a function that returns a future is invoked, two things happen:

1. The function queues up work to be done and returns an uncompleted Future object.
2. Later, when the operation is finished, the Future object completes with a value or with an error.

Scaffold - Implements the basic material design visual layout structure. This class provides APIs for showing drawers, snack bars, and bottom sheets. To display a snackbar or a persistent bottom sheet,



APIURL1

"https://api.openweathermap.org/data/2.5/weather?q=\$city&appid="\$ {util.appId} &units=metric";

APIURL2

"https://api.openweathermap.org/data/2.5/forecast?q=\$city&appid="\$ {util.appId} &units=metric";

```

Future<Map> getWeather(String appId, String city) async{
    String
    apiUrl1='https://api.openweathermap.org/data/2.5/weather?q=$city&appid="$ {util.appId}
&units=metric';

    http.Response response = await http.get(apiUrl);
    return json.decode(response.body);
}

```

This future object Maps the data according to the city and the app ID and used to fetch information from the given API.

API URL:-

```

{"coord":{"lon":79.13,"lat":12.91},"weather":[{"id":801,"main":"Clouds","description":"few
clouds","icon":"02d"}],"base":"stations","main":{"temp":25.58,"pressure":1014.11,"humidity":73,"temp_min":25.58,"temp_max":25.58,"sea_level":1014.11,"grnd_level":978.71},"wind":{"speed":2.41,"deg":191.504},"clouds":{"all":12},"dt":1553224236,"sys":{"message":0.2518,"country":"IN","sunrise":1553215602,"sunset":1553259259},"id":1253286,"name":"Vellore","cod":200}

```

This is data which is accessed by the Future object, and after converting it to JSON :

```

{
  "coord":{
    "lon":79.13,
    "lat":12.91
  },
  "weather":[
    {
      "id":801,
      "main":"Clouds",
      "description":"few clouds",
      "icon":"02d"
    }
  ],
  "base":"stations",
  "main":{
    "temp":25.58,
    "pressure":1014.11,
    "humidity":73,
    "temp_min":25.58,
    "temp_max":25.58,
    "sea_level":1014.11,
    "grnd_level":978.71
  }
}

```



```

    },
    "wind": {
      "speed": 2.41,
      "deg": 191.504
    },
    "clouds": {
      "all": 12
    },
    "dt": 1553224236,
    "sys": {
      "message": 0.2518,
      "country": "IN",
      "sunrise": 1553215602,
      "sunset": 1553259259
    },
    "id": 1253286,
    "name": "Vellore",
    "cod": 200
  }
}

```

It fetches and parses the data which can be used for performing actions to get the data out of the API.

```

Map content = snapshot.data;

return new Container(
  child: new Column(
    children: <Widget>[
      new ListTile(
        title: new Text(content['main']['temp'].toString() + " °C",
          style: new TextStyle(
            fontStyle: FontStyle.normal,
            fontSize: 49.9,
            color: Colors.white,
            fontWeight: FontWeight.w500
          ),
        subtitle: new ListTile(
          title: new Text(
            "Humidity: ${content['main']['humidity'].toString()}\n"
            "Min: ${content['main']['temp_min'].toString()} °C\n"
            "Max: ${content['main']['temp_max'].toString()} °C ",

```

This is a widget class which takes the snapshot of the data, if the snapshot has data then it Maps it with content or we can say it stores in it. Then using basic operations like `content['main']['temp']` the particular value can be find and display. Similarly we can display the humidity & Max_temp, Min_temp values.

```
class ChangeCity extends StatelessWidget {  
  var _cityFieldController = new TextEditingController();  
  @override  
  Widget build(BuildContext context) {  
    return new Scaffold(  
      appBar: new AppBar(  
        backgroundColor: Colors.red,  
        title: new Text('Change City'),  
        centerTitle: true,  
      ),  
      body: new Stack(  
        children: <Widget>[  
          new Center(  
            child: new Image.asset(  
              'images/white_snow.png',  
              width:490.0,  
              height:1200.0,  
              fit:BoxFit.fill,  
            ),  
          ),  
        ],  
      ),  
    );  
  }  
}
```

This is Change City Class which has textfield to enter city and button to get weather which brings back to home screen to display the results for the particular city.

It updates the city in the APIURL and then again it fetches the data and displays the temperature updates.

```
Future<List> fetchData(String apiUrl) async {
    http.Response response = await http.get(apiUrl);
    return(json.decode(response.body))['list'];
}
```

This Future object is used to get the upcoming days forecast, as it divides the data into list and displays it in a column.

```
body: new Center(
    child: new ListView.builder(
        padding: const EdgeInsets.all(15.0),
        itemCount: provider.length,
        itemBuilder: (BuildContext context, int position) {
            if(position.isOdd) return new Divider(
                color: Colors.red,
            );
            final index = position ~/2;
            return new ListTile(
                title: new Text(
                    "Temp:${provider[index]['main']['temp'].toString()} °C",
                    style: new TextStyle(
                        fontSize: 18.9,
                        fontWeight: FontWeight.bold)),

                subtitle: new Text("${provider[index]['dt_txt']}")
            );
        },
    ),
```

This widget function will build a list and display in columns

It will display the temperature for the upcoming days with date and time as its subtitle.

With a divider after every forecast to present the info separately.

Result



7:43



Change City

Enter City

Get Weather



7:43



weather

Temp:23.99 °C

2019-03-21 15:00:00

Temp:20.87 °C

2019-03-21 18:00:00

Temp:18.69 °C

2019-03-21 21:00:00

Temp:18.52 °C

2019-03-22 00:00:00

Temp:25.58 °C

2019-03-22 03:00:00

Temp:33.24 °C

2019-03-22 06:00:00

7:43



weather

Temp:20.53 °C

2019-03-22 21:00:00

Temp:18.43 °C

2019-03-23 00:00:00

Temp:25.75 °C

2019-03-23 03:00:00

Temp:34.03 °C

2019-03-23 06:00:00

Temp:36.44 °C

2019-03-23 09:00:00

Temp:34.83 °C

2019-03-23 12:00:00

7:43



DEBUG

WEATHER

Today's weather

Get Forecast

The problems faced in this application in getting the updated output when any place is input, To update the city in every class and then to display result, but the errors were solved using appropriate methods. The display of the upcoming forecast temperature was showing null then I solved that using Future<List> object and was able to display the result.

Navigation from home to new and new to home with updated actions was one of the problems, that was resolved using BuildContext , Material PageRoute etc other flutter classes.

Conclusion

This flutter application is the implementation of how to use API to get the data and to display it to user. This weather application will access the data of the city that has been input and will display the current weather and upcoming days forecast in different activities.

This is a cross-platform application which can be used for both Android & iOS.

References

1. <https://docs.flutter.io/flutter>
2. <https://openweathermap.org/api>
3. <https://dragosholban.com/2018/07/01/how-to-build-a-simple-weather-app-in-flutter/>
4. <https://itsallwidgets.com/>
5. <https://github.com/flutter/flutter/projects>
6. <https://www.youtube.com/watch?v=NnY4B7VK6e4>