

CurrenSee

# Developer Guide

**Preared by:**

[Aptech Students]

***CurrenSee***

|  |  |  |
| --- | --- | --- |
| **Supervisor.** | Sir Imran Ali | |
| **Batch.** | 2202C2 | |
| **Serial No.** | **Enrollment Number** | **Student Name** |
| 1. | Student 1365824 | Atya |
| 2. | Student1362560 | MUHAMMAD BILAL |
| 3. | Student1365428 | Azhar Shah |
| 4. | Student1365416 | Abdul Mughni Siddiqui |

Developer Guide for CurrenSee  
(Currency Conerter App)

### Introduction

Welcome to the developer guide for the ABC Finance Ltd. Currency Converter App. This document provides a comprehensive overview of the app's architecture, setup instructions, features, and functionalities to assist developers in understanding and contributing to the project

### Project Overview

The Currency Converter App name as Curren is a Flutter-based mobile application for Android that provides users with real-time currency conversion rates. The app supports multiple currencies and allows users to quickly and easily perform conversions between them.

## **Setup Instructions**

1. Prerequisties:

* Flutter SDK
* Android Studio or Visual Studio Code
* An Android device or emulator

1. **Install Dependencies**

Flutter pub get

1. **Run the App:**

Flutter run

## **Folder Structure**

|  |  |
| --- | --- |
| **Folder** | **Description** |
| lib/ | Contains the main source code for the app |
| assets/ | Contains images, fonts, and other static assets |

## Dependencies

The Currency Converter App utilizes the following major dependencies:

* Real-time currency conversion rates
* Support for multiple currencies
* The list is searchable and filterable
* The app stores a history of currency conversion transactions for users
* User-friendly interface
* Users can set their default base currency.
* Users can provide feedback and report issues through the app

## Key Features

* Real-time currency conversion rates
* Support for multiple currencies
* The list is searchable and filterable
* The app stores a history of currency conversion transactions for users
* User-friendly interface
* Users can set their default base currency.
* Users can provide feedback and report issues through the app

## Architecture

The app follows the Flutter which ensures a clear separation of concerns and makes the application more maintainable and testable.

## API Integration

The Currency Converter App fetches real-time exchange rates from the currency exchange API. Below is an example of how to make a request to the API:

Future<void> \_getCurrencies() async {

try {

var response = await http

.get(Uri.parse('https://api.exchangerate-api.com/v4/latest/USD'));

if (response.statusCode == 200) {

var data = json.decode(response.body);

setState(() {

currencies = (data['rates'] as Map<String, dynamic>).keys.toList();

rate = data['rates'][toCurrency];

\_updateTotal();

});

} else {

throw Exception('Failed to load currencies');

}

} catch (e) {

print('Error fetching currencies: $e');

}

}

## Swap Currencies

## Currency conversion, a standard practice in global markets, entails exchanging one currency for another and requires awareness of fluctuating exchange rates. Below is an example of how to make a request to the swap:

void \_swapCurrencies() {

setState(() {

String temp = fromCurrency;

fromCurrency = toCurrency;

toCurrency = temp;

\_getRate();

});

}

## API Services

The Currency Converter App fetches real-time exchange rates from the currency API. Below is an example of how to make a request to the API

const base\_url = "https://api.currencyapi.com/v3/";

const apikey = "cHi5rGdt1B9qK3hmyox5KhKuR1SVf48m0jOMseAJ";

class ApiService {

Future<List<CurrencyModel>> getLatest(String baseCurrrency) async {

List<CurrencyModel> currencyModelList = [];

String url =

'${base\_url}latest?apikey=$apikey&base\_currency=$baseCurrrency';

try {

final response = await http.get(Uri.parse(url));

if (response.statusCode == 200) {

Map<String, dynamic> json = jsonDecode(response.body);

Map<String, dynamic> body = json['data'];

body.forEach((key, value) {

CurrencyModel currencyModel = CurrencyModel.fromJson(value);

currencyModelList.add(currencyModel);

});

return currencyModelList;

} else {

return [];

}

} catch (e) {

throw e.toString();

}

}

## Slide to Delete

Our platform offers a highly customizable experience, allowing users to set their preferred currency effortlessly. To enhance convenience. and also delete it

void \_removeCurrency(Country country) async {

setState(() {

\_selectedCountries.remove(country);

\_currencyDetails.remove(country.currencyCode);

});

await \_saveSelectedCurrencies();

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(

backgroundColor: Colors.teal,

content: Center(child: Text("${country.currencyName} removed")),

duration: Duration(seconds: 6),

),

);

}

**Thank You.**