1. **Imports**: Importing necessary packages.

* **flutter/material.dart:** Contains the material design components.
* **currency\_converter\_page.dart:** Imports the **CurrencyConverterPage** which is the main UI of the app.

1. **main() Function**:

* This is the entry point of the app. The runApp function initializes the app and takes the MyApp widget as an argument.
* The runApp function is the entry point of a Flutter application. It takes a Widget and makes it the root of the widget tree, which represents the application's user interface. Let's break down its role and what it does in detail.

1. **MyApp Class**:

* A stateless widget that returns a MaterialApp.
* MaterialApp is the root widget of the app that contains the configuration and the home widget (CurrencyConverterPage).

‘currency\_converter\_page.dart’

**1) Imports**: Importing necessary packages.

* flutter/material.dart: Contains the material design components.
* http: Used to make HTTP requests.
* dart:convert: Used for JSON encoding and decoding.
* dart:io: Provides access to I/O operations.

**2) CurrencyConverterPage Class**:

* A stateful widget that creates the CurrencyConverterPage state.

**3)\_CurrencyConverterPageState Class**:

* This class contains the state and logic for the currency converter.
* \_controller: A text editing controller to manage the input field.
* \_convertedAmount: A double to store the converted amount.
* \_isLoading: A boolean to indicate if a conversion request is in progress.
* \_sourceCurrency: A string to store the source currency (default is 'USD').
* \_targetCurrency: A string to store the target currency (default is 'EUR').
* \_currencies: A list of supported currencies.

In Flutter, a TextEditingController is used to control and manage the content of a TextField. It allows you to read the value of the text field, listen for changes to the value, and programmatically set the value.

1. **\_convertCurrency() Method**:

* This method fetches the exchange rate and converts the entered amount.
* Checks if the entered amount is valid.
* Makes an HTTP GET request to the exchange rate API.
* If the response is successful, it calculates the converted amount.
* Handles potential errors such as no internet connection or other exceptions.

1. **\_showSnackBar() Method**:

* Displays a snackbar with the given message for 3 seconds.

1. **build() Method**:

* This method builds the UI for the currency converter page.
* Scaffold: Provides the structure for the UI, including the app bar and body.
* AppBar: Displays the title of the app.
* Column: Arranges the child widgets vertically.
  + Displays the converted amount.
  + TextField for entering the amount.
  + DropdownButtons for selecting source and target currencies.
  + TextButton for triggering the currency conversion.

**Summary**

* The app consists of two main parts: main.dart (the entry point) and currency\_converter\_page.dart (the main UI and logic).
* The CurrencyConverterPage is a stateful widget that manages the user input, fetches exchange rates, and performs the currency conversion.
* The \_convertCurrency method handles the API call and updates the state with the converted amount.
* The UI includes input fields, dropdowns for currency selection, and a button to trigger the conversion process.
* The app displays a loading indicator while fetching the exchange rates and handles potential errors gracefully with snack bars.