

Download the Zip File from the google drive

<https://drive.google.com/file/d/1eW0Q8e5gOZ2Q87coPfYRC0PWl0cHFTs/view?usp=sharing>

Go to your desired directory where you have saved the simple_calculator folder by unzipping the zip file

After doing the above steps follow the below commands in the vs code terminal

```
python3 -m venv venv
```

```
source venv/bin/activate
```

```
pip install starlette uvicorn
```

Install the iOS 16.1 simulator on your system(Xcode)

Run Python Server (Terminal 1)

```
source venv/bin/activate
```

```
python main.py
```

Run Flutter App (Terminal 2)

1. Terminal → New Terminal
2. Run:

```
open -a Simulator
```

```
flutter run
```

Test the calculator app

The screenshot shows the Android Studio interface with the Dart code for `main.dart` in the center editor window. The code defines a Flutter application with a main entry point and a stateless widget `CalcScreen` that handles button presses and performs calculations via a POST request to a local server.

```
lib > main.dart > ...
main.dart 1
lib > main.dart
1 import 'package:flutter/material.dart';
2 import 'package:http/http.dart' as http;
3 import 'dart:convert';
4
5 Run | Debug | Profile
6 void main() => runApp(const MyApp());
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
10  @override
11  Widget build(BuildContext context) {
12    return MaterialApp(home: const CalcScreen());
13  }
14
15 class CalcScreen extends StatefulWidget {
16   const CalcScreen({super.key});
17   @override
18   State<CalcScreen> createState() => _CalcScreenState();
19 }
20
21 class _CalcScreenState extends State<CalcScreen> {
22   String output = "0";
23   String a = "0", b = "0", op = "";
24
25   void press(String text) async {
26     setState(() {
27       if (text == "C") {
28         output = "0"; a = "0"; b = "0"; op = "";
29       } else if ("+-*/".contains(text)) {
30         a = output; op = text; output = "0";
31       } else if (text == "=") {
32         b = output;
33         http.post(
34           Uri.parse("http://127.0.0.1:8000/calc"),
35           headers: {"Content-Type": "application/json"},
36           body: jsonEncode({"a": double.parse(a), "b": double.parse(b), "op": op}),
37         ).then((resp) {
38           final data = jsonDecode(resp.body);
39           setState(() => output = data["result"]);
40         }).catchError((e) => setState(() => output = "Error"));
41       } else {
42         output = output == "0" ? text : output + text;
43       }
44     }
45   }
46 }
47
48 Ln 4, Col 1  Spaces: 2  UTF-8  LF  () Dart  ⚙ Finish Setup  ⚙ Go Live  macOS (darwin)
```

The screenshot shows the PyCharm interface with the Python code for `main.py` in the center editor window. The code uses the Starlette framework to handle HTTP requests, specifically for a "/calc" endpoint that performs arithmetic operations based on the provided operators (+, -, *, /).

```
main.py > ...
main.py 1
main.py
1 # main.py
2 from starlette.applications import Starlette
3 from starlette.responses import JSONResponse
4 from starlette.routing import Route
5 import uvicorn
6
7 async def calculate(request):
8   data = await request.json()
9   a = float(data["a"])
10  b = float(data["b"])
11  op = data["op"]
12
13  if op == "+": result = a + b
14  elif op == "-": result = a - b
15  elif op == "*": result = a * b
16  elif op == "/": result = a / b if b != 0 else "Error"
17  else: result = "Error"
18
19  return JSONResponse({"result": str(result)})
20
21 app = Starlette(routes=[Route("/calc", calculate, methods=["POST"])])
22
23 if __name__ == "__main__":
24   uvicorn.run(app, host="127.0.0.1", port=8000)
25
26 Ln 24, Col 50  Spaces: 4  UTF-8  LF  () Python  ⚙ Finish Setup  3.13.7 (venv)  ⚙ Go Live  macOS (darwin)
```

