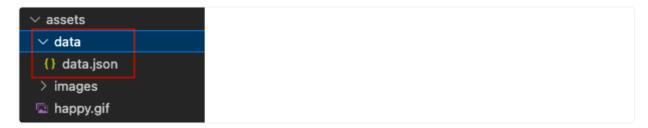
kaufmann_kennzahlen

Change of Data Source

Please follow the below steps:

- Create directory name data under the assets folder.
- Drag & drop the data.json file from the zip which I provided.

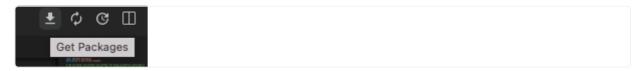


Add one line (assets/data/) to get the data.json file from assets in pubspec.yaml file.

```
1 assets:
2 - assets/happy.gif
3 - assets/images/
4 - assets/data/
```

```
# To add assets to your application, add an assets section, like this:
assets:
   - assets/happy.gif
   - assets/images/
   - assets/data/
```

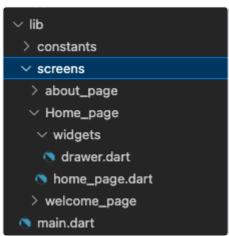
 After then click the down arrow to get packages in pubspec.yaml (it will get data.json from the assets folder)



• Remove the data directory from the lib folder.







home_page.dart

• Remove unnecessary library which is not used in the home_page.dart.

```
Ф
        EXPLORER
                                                lib > screens > Home_page > ♠ home_page.dart > .

1 import 'dart:convert';
       V KAUFMANN_KENNZAHLEN
                                                  import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
fmport './../data/data.dart';
import 'widgets/drawer.dart';
        > android
        > assets
> build
> ios
                                                   > about_page
                                                           HomePageState createState() => HomePageState();
           widgets
o drawer.dart
           home_page.dart
welcome_page
         main.dart
                                                         List _items = [];
final _controller = TextEditingController();
@override
        > web

≡ .flutter-plugins
                                                           void initState() {
                                                            super.initState();
readJson();
          gitignore.
                                                        // Fetch content from the json file
Future<void> readJson() async {
    final String response = await rootBundle.loadString('assets/data/data.json');
    final data = await json.decode(response);
        ! pubspec.yaml

① README.md
        OUTLINE
```

```
EXPLORER

★ Get Started

                                      home_page.dart
> .dart_tool
                            1 import 'dart:convert';
                               import 'package:flutter/material.dart';
> android
                               import 'package:flutter/services.dart';
                               import 'widgets/drawer.dart';
 {} data.json
 > images
                               class HomePage extends StatefulWidget {
happy.gif
                               const HomePage({Key? key}) : super(key: key);
                                 @override
∨ lib
                                 HomePageState createState() => HomePageState();
> constants
 > about_page
                           13

√ Home_page

                           14 class HomePageState extends State<HomePage> {
  ∨ widgets
                                List _items = [];
  drawer.dart
                                 final _controller = TextEditingController();
  home_page.dart
                                 @override
 > welcome_page
                                 void initState() {
 main.dart
                                super.initState();
```

Add _items variable in the home_page.dart file.

```
1 List _items = [];

class HomePageState extends State<HomePage> {
   List _items = [];
   final _controller = TextEditingController();
```

- Add readJson() function in the home_page.dart file.
- Call this function in the initState() function.

```
1
     @override
      void initState() {
         super.initState();
3
4
         readJson();
5
      }
6
    // Fetch content from the json file
       Future<void> readJson() async {
8
         final String response = await
9
    rootBundle.loadString('assets/data/data.json');
10
         final data = await json.decode(response);
         setState(() {
11
```

```
12    _items = data["items"];
13    });
14 }
```

```
class HomePageState extends State<HomePage> {
 List _items = [];
 final _controller = TextEditingController();
 @override
 void initState() {
   super.initState();
   readJson();
 Future<void> readJson() async {
   final String response = await rootBundle.loadString('assets/data/data.json');
   final data = await json.decode(response);
   setState(() {
     _items = data["items"];
   });
 @override
 Widget build(BuildContext context) {
     appBar: AppBar(
       title: const Text("Kaufleute - Begriffe"),
       centerTitle: true,
       scrolledUnderElevation: 0,
     ), // AppBar
     drawer: const AppDrawer(),
     body: Column(
```

 Add search functionality when the user types in the search box. It will return titlematched words and display them in the list.

```
onChanged: (t) {
                    if(_controller.text == ""){
2
3
                       readJson();
4
                    }else {
                       _items = _items.where((element) =>
5
6
    element['title'].toString().toLowerCase().contains(t.toLowerCase
    ())).toList();
                       setState(() {});
7
                    }
8
9
                  }
```

Written in **slite**

```
body: Column(
 children: [
   Padding(
     padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 5),
     child: TextField(
       controller: _controller,
       onChanged: (t) {
          if(_controller.text == ""){
           readJson();
          }else {
           _items = _items.where((element) =>
                element['title'].toString().toLowerCase().contains(t.toLowerCase())).toList
            setState(() {});
       decoration: InputDecoration(
          border: const OutlineInputBorder(),
          hintText: 'Schreibe etwas...',
          prefixIcon: const Icon(Icons.search),
          isDense: true,
          suffixIcon: IconButton(
```

- When the user clicks on the cross button that time data.json file is read again and displays all data from data.json in the list.
- so call readJson() again on onPressed() method.

```
onPressed: () {

readJson();

FocusScope.of(context).unfocus();

setState(() {});

_controller.clear();
},
```

```
suffixIcon: IconButton()
onPressed: () {
    readJson();
    FocusScope.of(context).unfocus();
    setState(() {});
    _controller.clear();
},
icon: const Icon(Icons.close),
), // IconButton
```

• Add items.length in the itemCount property of ListView.builder widget.

```
1 itemCount: _items.length,
```

Written in **slite** 5

- Change CircleAvatar, title & subtitle properties of the ListTile widget with the following code.
- CircleAvatar displays abbreviation, title display title & subtitle display subtitle from the data.json file.

```
child: ListTile(
 2
                          leading: Padding(
 3
                            padding: const EdgeInsets.only(left:
     18.0),
                            child: CircleAvatar(
 4
                              backgroundColor:
 5
     Theme.of(context).colorScheme.primary,
                              child: Text(_items[i]['abbreviation']),
 6
 7
                            ),
 8
                          ),
                          title: Text(
9
                              _items[i]['title'],
10
                              style: const TextStyle(
11
                                fontWeight: FontWeight.bold, fontSize:
12
     20.0),
13
                          ),
                          subtitle: Text(
14
                            _items[i]['subtitle'],
15
                            style: const TextStyle(fontSize: 16.0),
16
17
                          ),
                        ),
18
```

Written in **slite**

```
itemBullder: (context, 1)
 return Card(
   margin: const EdgeInsets.symmetric(horizontal: 20)
        .copyWith(top: 10),
   elevation: 10,
   shadowColor: □Colors.black.withOpacity(0.2),
   child: ListTile(
     leading: Padding(
       padding: const EdgeInsets.only(left: 18.0),
       child: CircleAvatar(
         backgroundColor: Theme.of(context).colorScheme.primary,
         child: Text(_items[i]['abbreviation']),
       ), // CircleAvatar
     ), // Padding
     title: Text(
         _items[i]['title'],
         style: const TextStyle(
           fontWeight: FontWeight.bold, fontSize: 20.0), // TextStyle
     ), // Text
     subtitle: Text(
       _items[i]['subtitle'],
       style: const TextStyle(fontSize: 16.0),
     ), // Text
   , // ListTile
  ); // Card
```

Written in **Slite** 7