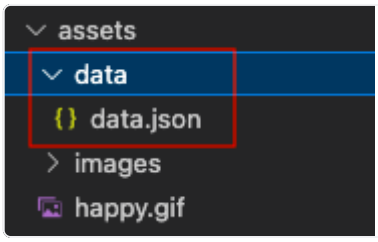
kaufmann_kennzahlen

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.

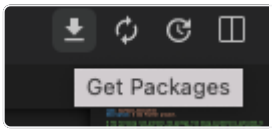


- Add one line (assets/data/) to get the data.json file from assets in pubspec.yaml file.

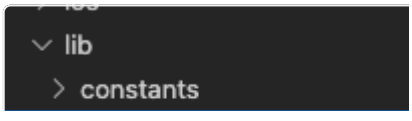
```
assets:
    - assets/happy.gif
    - assets/images/
    - assets/data/
```

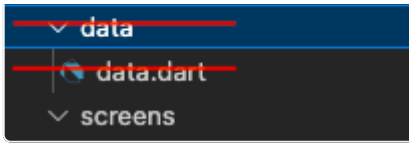
```
# To add assets to your application, add an assets section, like this:
assets:
  - assets/happy.gif
  - assets/images/
  - assets/data/
```

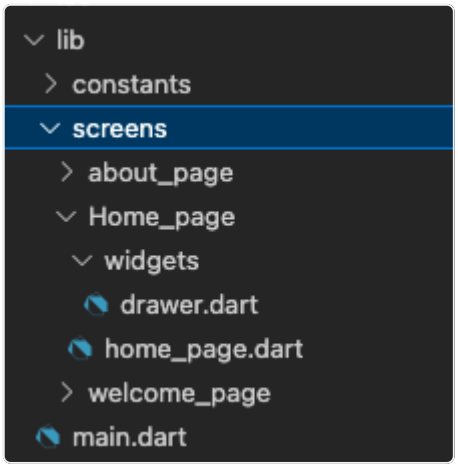
- After then click the down arrow to get packages in pubspec.yaml (it will get data.json from the assets folder)



- Remove the data directory from the lib folder.

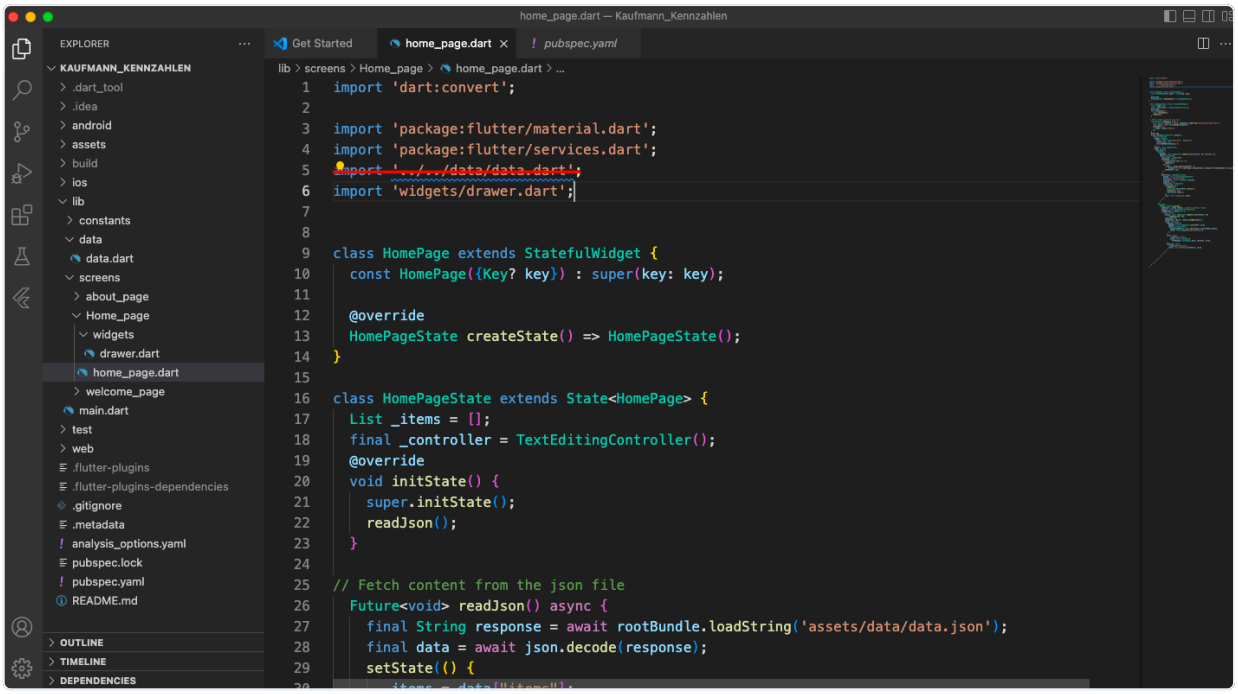


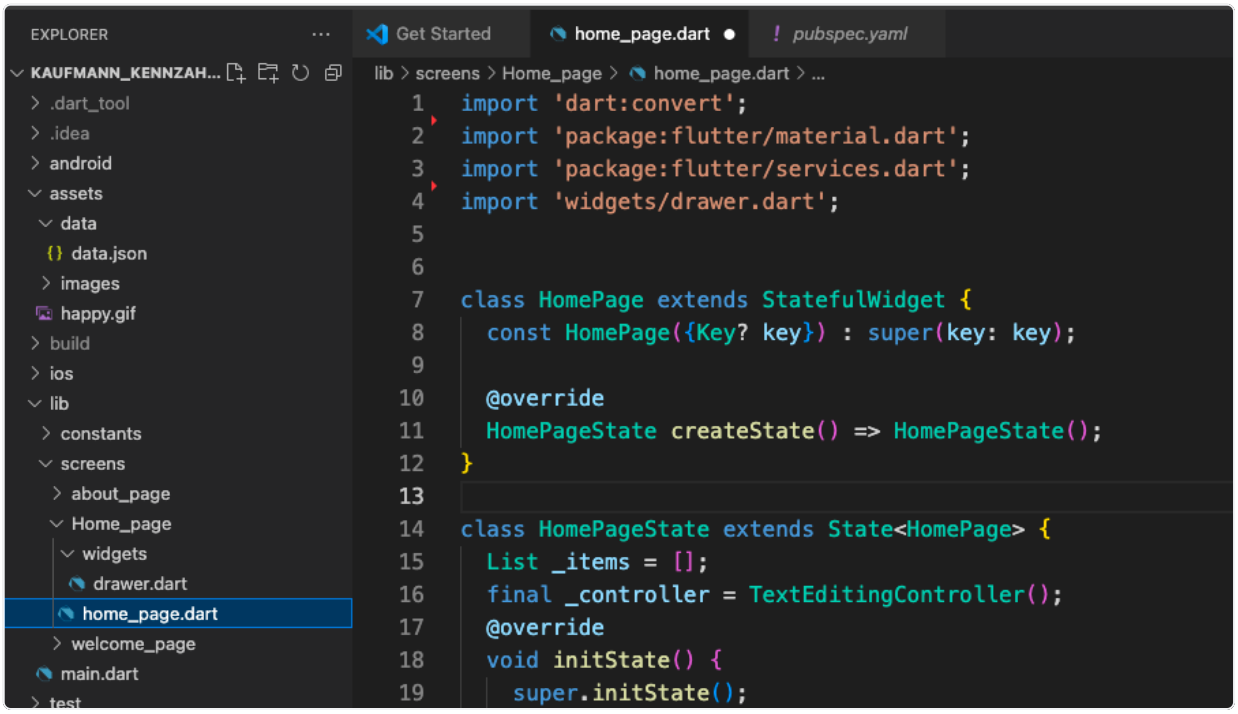




home_page.dart

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





- Add _items variable in the home_page.dart file.

```
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.

```
 @override
  void initState() {
    super.initState();
    readJson();
  }

// 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);
    setState(() {
```

```
        _items = data["items"];
      });
    }
```

```
class HomePageState extends State<HomePage> {
  List _items = [];
  final _controller = TextEditingController();
  @override
  void initState() {
    super.initState();
    readJson();
  }

// 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);
    setState(() {
      _items = data["items"];
    });
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      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 title-matched words and display them in the list.

```
onChanged: (t) {
                  if(_controller.text == ""){
                    readJson();
                  }else {
                    _items = _items.where((element) =>

element['title'].toString().toLowerCase().contains(t.toLowerCase
()())).toList();
                    setState(() {});
                  }
                }
```

```
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.

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

```
Flexible(
  child: ListView.builder(
    itemCount: _items.length,
    physics: const BouncingScrollPhysics(),
    itemBuilder: (context, i) {
      return Card(
        margin: const EdgeInsets.symmetric(horizontal: 20)
            .copyWith(top: 10),
```

- 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(
                    leading: Padding(
                      padding: const EdgeInsets.only(left: 18.0),
                      child: CircleAvatar(
                        backgroundColor: Theme.of(context).colorScheme.primary,
                        child: Text(_items[i]['abbreviation']),
                      ),
                    ),
                    title: Text(
                        _items[i]['title'],
                        style: const TextStyle(
                          fontWeight: FontWeight.bold, fontSize: 20.0),
                    ),
                    subtitle: Text(
                      _items[i]['subtitle'],
                      style: const TextStyle(fontSize: 16.0),
                    ),
                  ),
```

```
itemBuilder: (context, i) {
  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
}
```