

@DIY App

Code guide
2021

<https://github.com/danielgoncalves25/diyapp>

Table of Contents

Overview	3
Screens	4
Onboarding.....	5
home_page.dart	6
view_article.dart	7
add_article.dart.....	8
search_page.dart	9
widgets	10
drawer.dart	11
article_card.dart.....	12
Ordering & License.....	13

Overview

This app is built using the atsign Company's backend tools. All user accounts should be tested using an @sign or using the virtual environment testing accounts. for more see <https://atsign.dev/>

Created with Dr.Explain
Unregistered version

Screens

Structurally this app uses 5 screens and one onboarding screen:

- add_article.dart
- edit_article.dart
- home_screen.dart
- search_page.dart
- view_article.dart
- main.dart (onboarding)

The code for these screens can be found at <https://github.com/danielgoncalves25/diyapp/tree/main/lib/screens>

Each screen serves a purpose for the app allowing the viewer to navigate via the snackbar from any screen to any other screen.

The exception being the view_article.dart screen which is accessed anytime an existing article is clicked from home or search.

Onboarding

(main.dart)

This page consists mostly of boilerplate onboarding code straight from the dev site of the atsign company.
<https://atsign.dev/>

The only changes were done to style the page and include a small credit to the authors: Whitney Hamnett, Daniel Goncalves and Katherine Elia.



Created with Dr.Explain
Unregistered version

Profile Pic/ Username:

This is an object instance of an image which is stored in the @sign backend similarly to a simplified article class username for atsign is displayed below profile pic in large centered text.

Article Scrollable:

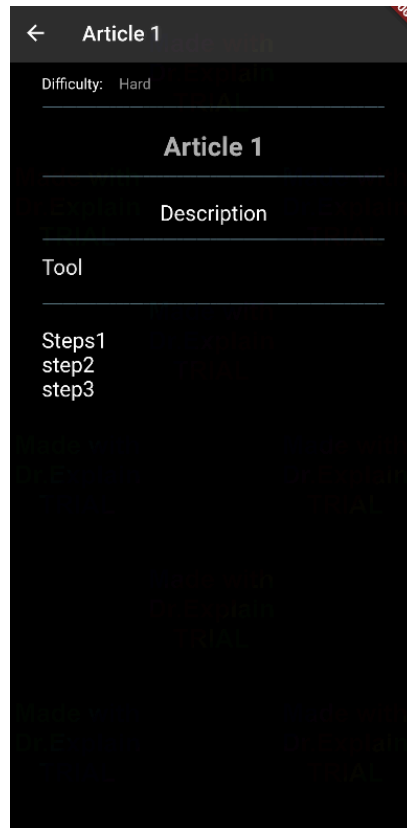
Below the profile information is a scrollable list of articlecards using FutureBuilder widget. See the articlecard widget under widgets/articlecard for more about how we use this in homescreen as well as search to display a list containing some article information, which upon click takes users to view_article, which will display the full article contents, stylized.



view_article.dart

All of the information we store in the backend from `add_article` is accessed to display the contents of each article in a meaningful way. The way we display the content is a bit primitive and absolutely could use some improvement to be more compatible and pretty on a wider array of devices.

Each attribute in the instance of the article class is accessed in the scrollview after an instance of an article is created using the required key. `article.description!` for example returns the description (if any) of the article being created. Try it out with a print statement!



add_article.dart (and) edit_article.dart

createArticle() is called after all fields are entered, which creates an instance of the article class and saves it with a key to the backend using the atsign protocol. This page consists of several textfieldwidgets, in rows with toggles for privacy as well as a tag field and dropdown, and an image button for uploading multiple images. All this will be saved to an article object when the user clicks the create article button at the bottom of the scrollable (not shown in photo). Edit article is much the same as add_article and thus not provided seperate documentation.

The screenshot shows a mobile application interface for adding an article. At the top, a note states: "NOTE: Information is public by default! Toggle privacy on for all or part of your article". Below this, there are three main input sections, each with a toggle switch on the right:

- Enter Article name:** A single-line text input field.
- Enter Article description:** A multi-line text input field with a toggle switch set to "on" (blue).
- Enter tool nessesary (seperated by ,):** A multi-line text input field with a toggle switch set to "on" (blue).

At the bottom of the screen, a standard Android keyboard is visible, showing the letters 'q' through 'p' on the first row, 'a' through 'l' on the second row, and 'z' through 'm' on the third row, along with a checkmark button.

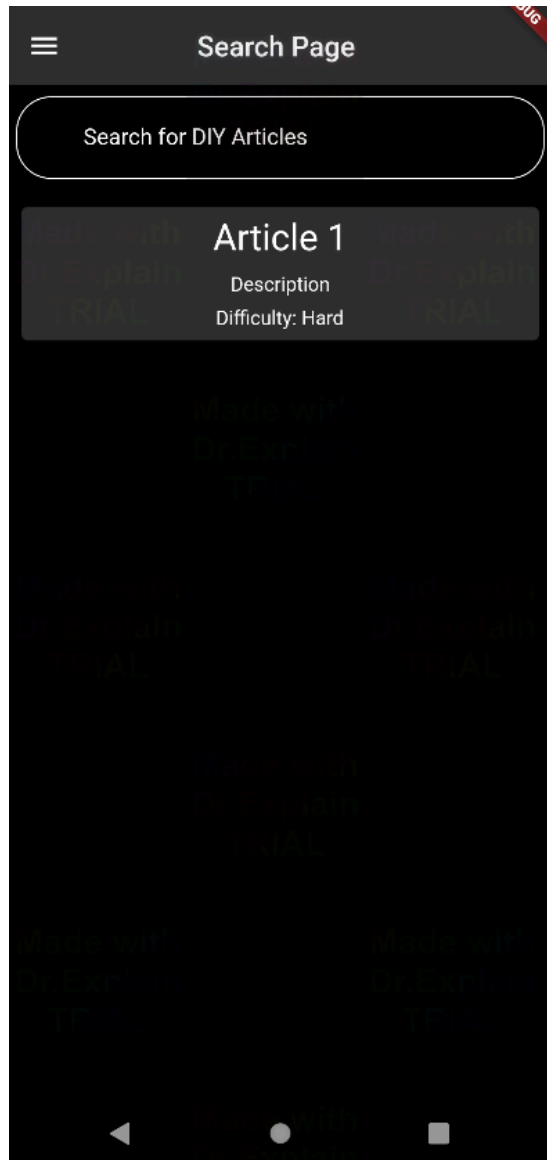
search_page.dart

Here users may search for articles both locally and from other users

There is a filter which is run upon users search entry and is then applied to a future list containing all articles (either locally or on the app, depending on user choice*).

A textfield takes user entry and applies the runfilter on the list it has built using future scanYourArticles()

*public search not yet completed as of 12/21/2021 local only for now.



widgets

There are two custom widgets used in this app:

- the drawer which is just a stylized typical drawer implementation in flutter
- the article_card is a crucial widget that we use in a couple places whenever we want to list articles such that users may see some information displayed in a clickable icon that leads to a large article viewing screen (article_view.dart)

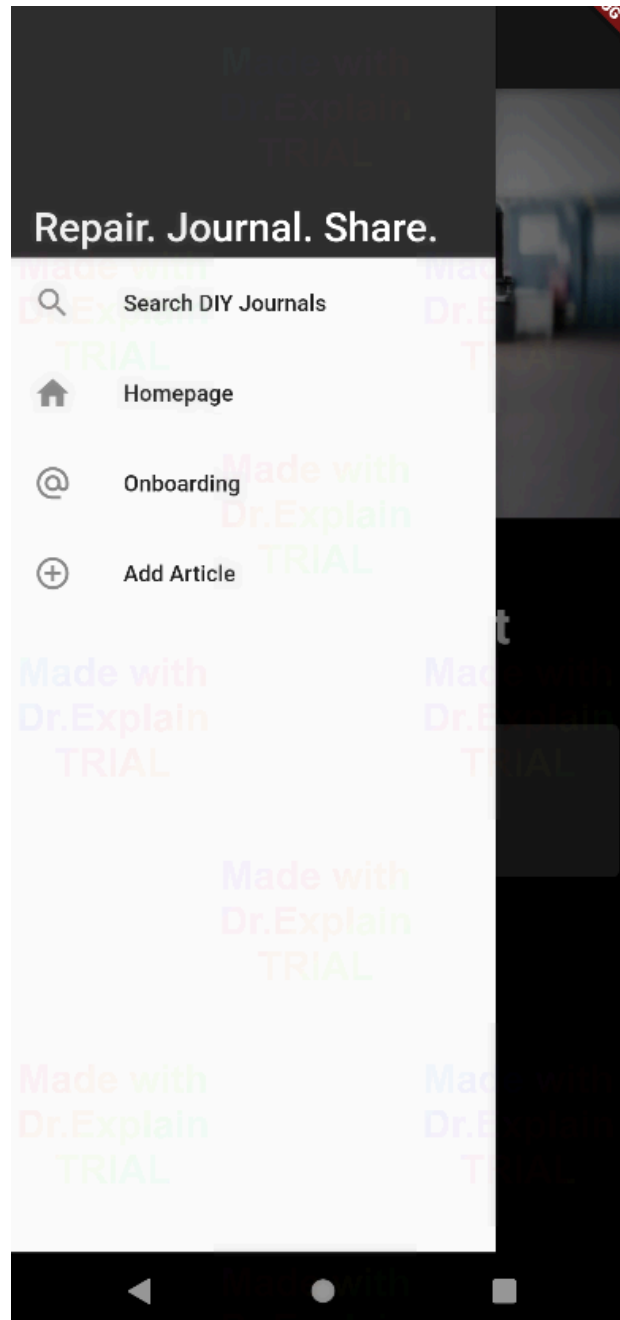
drawer.dart

Navigate the various pages on click:

- search
- onboarding
- create_article
- home_page

For code see lib/widgets/drawer

Just a simple drawer implementation with a title and some links to navigate the pages of our app



article_card.dart

This is a custom widget (lib/widgets/article_card.dart)
article_card requires to be passed an article (of the class type article) and then it creates a widget which displays some relevant information about the article in a clickable card that when clicked will navigate to view_article.dart passing the article.



Created with Dr.Explain
Unregistered version

Ordering & License

All development for this app is open source code available at:

<https://github.com/danielgoncalves25/diyapp>

On completion this app will be compatible with both android and ios devices as well as future plans for a webapp version.

Created with Dr.Explain

Unregistered version