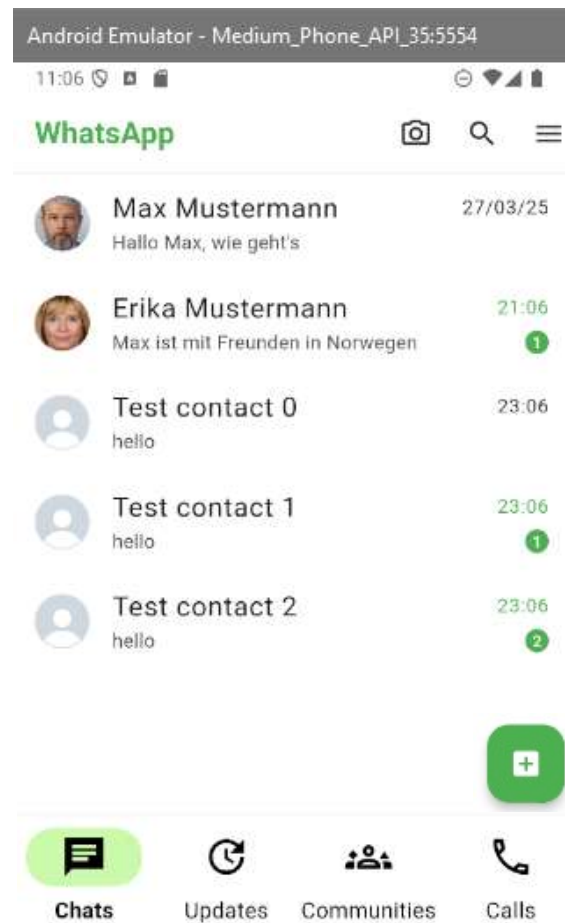




Goal: Implement a first version of the Chats Page



The Contact class



Max Mustermann

Hallo Max, wie geht's

27/03/25



Erika Mustermann

Max ist mit Freunden in Norwegen

20:59

1



```
class Contact {  
    Contact(this.name);  
    String name;  
    String? imagePath;  
    String lastMessage = "hello";  
    DateTime dateOfLastMessage = DateTime.now();  
    int countOfNewMessages = 0;  
  
    simulateNewMessage() {  
        dateOfLastMessage = DateTime.now();  
        lastMessage = "new message at $dateOfLastMessage";  
        countOfNewMessages++;  
    }  
}
```

Note as ToDo:
This could be made easier with
an external package called intl
(for internationalization)

```
String getDateOfLastMessageAsString() {  
    // normally show date as dd/mm/yy  
    var result = "${dateOfLastMessage.day}".padLeft(2, "0") +  
        "/" +  
        "${dateOfLastMessage.month}".padLeft(2, "0") +  
        "/" +  
        "${dateOfLastMessage.year - 2000}";  
  
    var now = DateTime.now();  
    // when last message is from current day, show hh:mm  
    if (now.day == dateOfLastMessage.day &&  
        now.month == dateOfLastMessage.month &&  
        now.year == dateOfLastMessage.year) {  
        result = "${dateOfLastMessage.hour}".padLeft(2, "0") +  
            ":" +  
            "${dateOfLastMessage.minute}".padLeft(2, "0");  
    }  
    return result;  
}
```



The Contact Provider

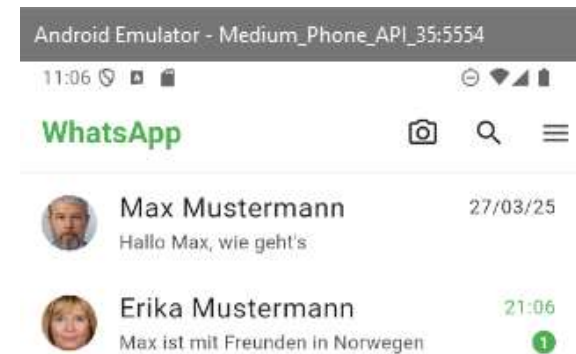
```
import 'contact.dart';

class ContactProvider {
  ContactProvider() {
    initializeContacts();
  }

  List<Contact> contacts = [];

  void initializeContacts() {
    var now = DateTime.now();
    var contact = Contact("Max Mustermann");
    contact.imagePath = "assets/images/mm.jpg";
    contact.lastMessage = "Hallo Max, wie geht's";
    contact.dateOfLastMessage = now.add(const Duration(days: -1));
    contacts.add(contact);

    contact = Contact("Erika Mustermann");
    contact.imagePath = "assets/images/em.jpg";
    contact.lastMessage = "Max ist mit Freunden in Norwegen";
    contact.dateOfLastMessage = now.add(const Duration(hours: -2));
    contact.countOfNewMessages = 1;
    contacts.add(contact);
  }
}
```



The ContactListTile



```
class ContactListTile extends StatelessWidget {  
  const ContactListTile({super.key, required this.contact});  
  
  final Contact contact;  
  
  @override  
  Widget build(BuildContext context) {  
    var imagePath = contact.imagePath ?? "assets/images/default_image.jpg";  
  
    return ListTile(  
      leading: CircleAvatar(foregroundImage: AssetImage(imagePath)),  
      title: Row(  
        children: [  
          Expanded(  
            child: Text(  
              contact.name,  
              style:  
                const TextStyle(fontSize: 20, fontWeight: FontWeight.normal),  
            ), // Text  
          ), // Expanded  
          const SizedBox(width: 15),  
          Text(  
            contact.getDateOfLastMessageAsString(),  
            style: TextStyle(  
              fontSize: 14,  
              color: contact.countOfNewMessages > 0 ? Colors.green : null,  
            ) // Text  
          ),  
        ],  
      ), // Row  
    );  
  }  
}
```



Max Mustermann

Hallo Max, wie geht's

27/03/25



Erika Mustermann

Max ist mit Freunden in Norwegen

20:59

1

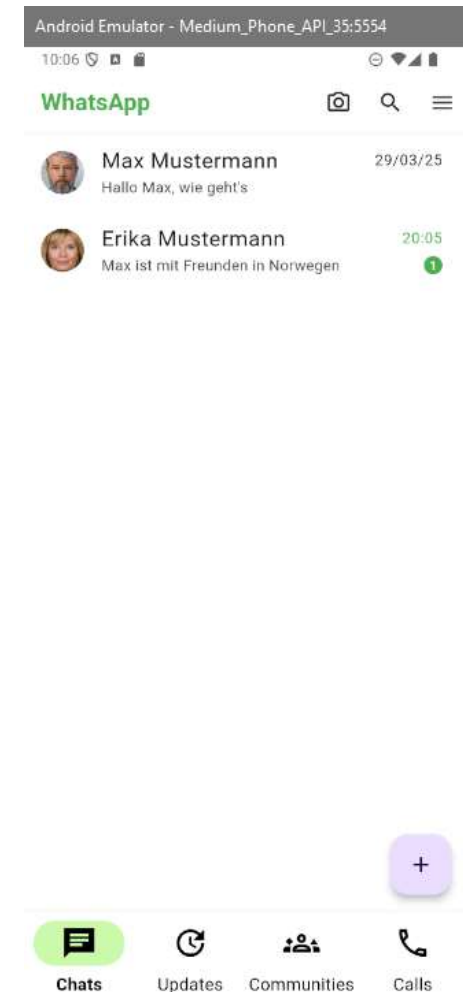
We will see on Slide 7 why "Expanded" is used here.

```
      subtitle: Row(  
        children: [  
          Expanded(  
            child: Text(  
              contact.lastMessage,  
            ), // Text  
          ), // Expanded  
          if (contact.countOfNewMessages > 0)  
            Badge(  
              label: Text("${contact.countOfNewMessages}",  
                style: const TextStyle(fontSize: 12)), // Text  
              backgroundColor: Colors.green,  
            ) // Badge  
          ],  
        ), // Row  
      ); // ListTile
```

The ChatsWidget in the _pages list

```
class ChatsWidget extends StatelessWidget {  
  const ChatsWidget({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    var provider = ContactProvider();  
    var contacts = provider.contacts;  
  
    return Column(  
      children: [  
        ContactListTile(contact: contacts[0]),  
        ContactListTile(contact: contacts[1]),  
      ],  
    ); // Column  
  }  
}
```

```
class _MyHomePageState extends State<MyHomePage> {  
  int _selectedIndex = 0;  
  
  static final List<Widget> _pages = <Widget>[  
    const ChatsWidget(),  
    const Center(child: Text("Updates page", style: TextStyle(fontSize: 20))),  
    const Center(  
      child: Text("Communities page", style: TextStyle(fontSize: 20)), // Center  
    ),  
    const Center(child: Text("Calls page", style: TextStyle(fontSize: 20))),  
  ]; // <Widget>[]  
}
```



Contacts with long names and messages



```
contact = Contact("Contact with a very very loooooong name");
contact.lastMessage = "a very very very long loooooong "
    "loooooooooooooooooooooooooooooooooooooooooooooooooooooong message";
contact.dateOfLastMessage = DateTime(2025, 03, 27, 00, 11);
contacts.add(contact);
```

```
Expanded(
  child: Text(
    contact.name,
    style:
      const TextStyle(fontSize: 20, fontWeight: FontWeight.normal),
    softWrap: false,
    overflow: TextOverflow.ellipsis,
  ), // Text
), // Expanded
```

```
Expanded(
  child: Text(
    contact.lastMessage,
    softWrap: false,
    overflow: TextOverflow.ellipsis,
  ), // Text
), // Expanded
```



Erika Mustermann

Max ist mit Freunden in Norwegen

20:17

1

Contact with a very very
loooooong name

27/03/25

a very very very long loooooong loooooooooooooo
ooong
message



Erika Mustermann

Max ist mit Freunden in Norwegen

20:19

1

Contact with a very very...

27/03/25

a very very very long loooooong loooooooooooooo...



Without Expanded widget -> overflow

```
title: Row(  
  children: [  
    Expanded(  
      child: Text(  
        widget.contact.name,  
        style:  
          const TextStyle(fontSize: 20, fontWeight: FontWeight.normal),  
        softWrap: false,  
        overflow: TextOverflow.ellipsis,  
      ), // Text  
    ), // Expanded  
    const SizedBox(width: 15),  
    Text(  
      widget.contact.getDateOfLastMessageAsString(),
```



```
title: Row(  
  mainAxisAlignment: MainAxisAlignment.spaceBetween,  
  children: [  
    Text(  
      widget.contact.name,  
      style: const TextStyle(fontSize: 20, fontWeight: FontWeight.normal),  
      softWrap: false,  
      overflow: TextOverflow.ellipsis,  
    ), // Text  
    Text(  
      widget.contact.getDateOfLastMessageAsString(),
```



Reset countOfNewMessages when tile is tapped



```
class ContactListTile extends StatefulWidget {  
  const ContactListTile({  
    super.key,  
    required this.contact,  
  });  
  
  final Contact contact;  
  
  @override  
  State<ContactListTile> createState() => _ContactListTileState();  
}
```

```
class _ContactListTileState extends State<ContactListTile> {  
  @override  
  Widget build(BuildContext context) {  
    var imagePath =  
      widget.contact.imagePath ?? "assets/images/default_picture.jpg";  
  
    return ListTile(  
      onTap: () {  
        setState(() {  
          widget.contact.countOfNewMessages = 0;  
        });  
      },  
      leading: CircleAvatar(foregroundImage: AssetImage(imagePath)),  
      title: Row(  
        children: [  
          Expanded(  
            child: Text(  
              widget.contact.name,  

```



Erika Mustermann
Max ist mit Freunden in Norwegen

20:56



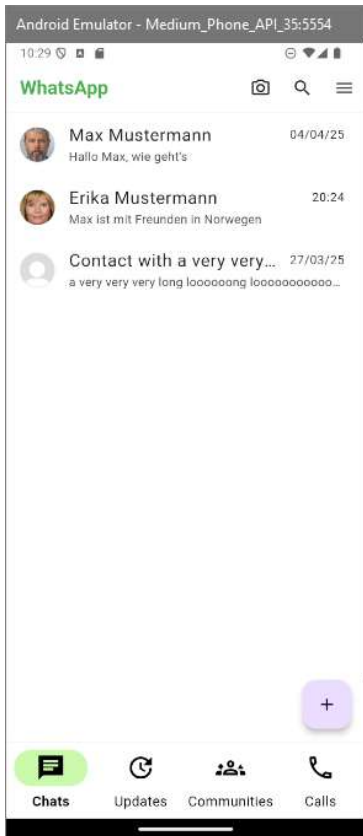
Erika Mustermann
Max ist mit Freunden in Norwegen

20:56

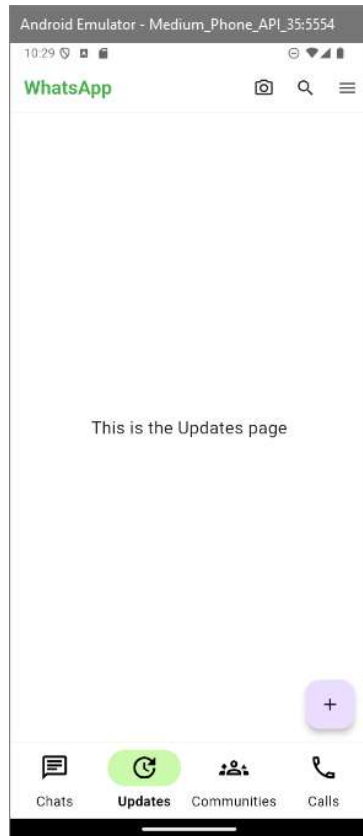
ContactListTile must be stateful

contact is accessed via widget

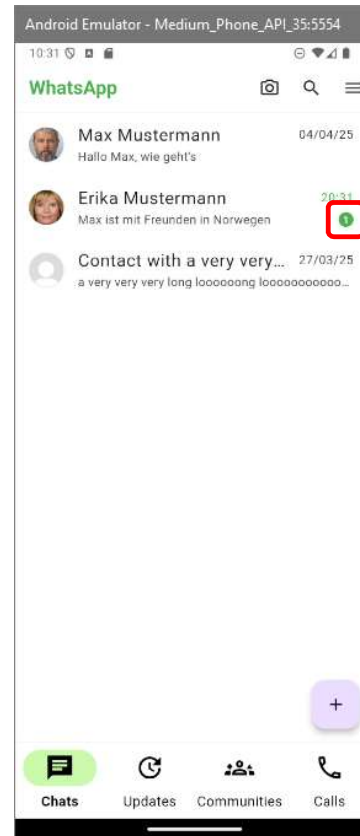
Build method gets re-called after page change



No more “new message”



Open Updates page



Back to Chats page

```
5 class ChatsWidget extends StatelessWidget {  
6   const ChatsWidget({super.key});  
7  
8   @override  
9   Widget build(BuildContext context) {  
10    var provider = ContactProvider();  
11    var contacts = provider.contacts;  
12  
13    return Column(  
14      children: [  
15        ContactListTile(contact: contacts[0]),  
16        ContactListTile(contact: contacts[1]),  
17        ContactListTile(contact: contacts[2]),  
18      ],  
19    ); // Column  
20  }  
21 }
```

A new provider is created and
it has a new list of contacts
with new messages

First workaround: use a global ContactProvider

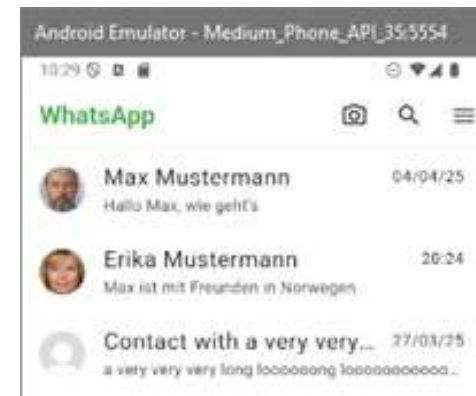


```
var globalContactProvider = ContactProvider();

class ChatsWidget extends StatelessWidget {
  const ChatsWidget({super.key});

  @override
  Widget build(BuildContext context) {
    var contacts = globalContactProvider.contacts;

    return Column(
      children: [
        ContactListTile(contact: contacts[0]),
        ContactListTile(contact: contacts[1]),
        ContactListTile(contact: contacts[2]),
      ],
    ); // Column
  }
}
```



No more new message after returning from Updates page.

Simulate a new message for a random contact



```
floatingActionButton: FloatingActionButton(  
  foregroundColor: Colors.white,  
  backgroundColor: Colors.green,  
  onPressed: () {  
    setState(() {  
      var contacts = globalContactProvider.contacts;  
      var random = Random();  
      var index = random.nextInt(contacts.length);  
      print("random index is $index");  
      contacts[index].simulateNewMessage();  
    });  
  },  
  tooltip: 'simulate a new message',  
  child: const Icon(Icons.add_box),  
), // FloatingActionButton
```

Although we call setState in
main.dart, the chats page is
not updated ?!
(no badges appear after
pressing FloatingActionButton)

Build method of ChatsWidget
is not called in this case.

```
6  
7 class ChatsWidget extends StatelessWidget {  
8   const ChatsWidget({super.key});  
9  
10  @override  
11  Widget build(BuildContext context) {  
12    var contacts = globalContactProvider.contacts;  
13  
14    return Column(  
15      children: [  
16        ContactListTile(contact: contacts[0]),  
17        ContactListTile(contact: contacts[1]),  
18        ContactListTile(contact: contacts[2]),  
19      ],  
20    ); // Column  
21  }  
22 }
```

Solution: move pages list to build method



```
class _MyHomePageState extends State<MyHomePage> {  
  int _selectedIndex = 0;  
  
  final List<Widget> _pages = [  
    const ChatsWidget(),  
    const Text("This is the Updates page", style: TextStyle(fontSize: 20)),  
    const Text("This is the Communities page", style: TextStyle(fontSize: 20)),  
    const Text("This is the Calls page", style: TextStyle(fontSize: 20)),  
  ];  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      drawerScrimColor: Colors.transparent,  
    );  
  }  
}
```

C-tor of ChatsWidget is called outside the build method of _MyHomePageState.

```
class _MyHomePageState extends State<MyHomePage> {  
  int _selectedIndex = 0;  
  
  @override  
  Widget build(BuildContext context) {  
    List<Widget> pages = [  
      ChatsWidget(),  
      const Text("This is the Updates page", style: TextStyle(fontSize: 20)),  
      const Text("This is the Communities page",  
        style: TextStyle(fontSize: 20)), // Text  
      const Text("This is the Calls page", style: TextStyle(fontSize: 20)),  
    ];  
  
    return Scaffold(  
      drawerScrimColor: Colors.transparent,  
    );  
  }  
}
```

Important:
do not use the const c-tor for ChatsWidget

```
var globalContactProvider = ContactProvider();

class ChatsWidget extends StatelessWidget {
  const ChatsWidget({super.key});

  @override
  Widget build(BuildContext context) {
    var contacts = globalContactProvider.contacts;
  }
}
```



Should global variables be used in flutter



It's generally not recommended to heavily rely on global variables in Flutter for managing application state. While Dart allows for global variables, their use can lead to several issues, especially in larger and more complex applications.

Here's a breakdown of why and what alternatives you should consider:

Why Global Variables Are Generally Discouraged in Flutter:

- **Tight Coupling:** Global variables create dependencies throughout your codebase. If a global variable changes, it can be difficult to track down all the parts of the application that are affected, leading to unexpected behavior and making debugging harder.
- **Hard to Track Changes:** When multiple parts of your application can modify a global variable, it becomes difficult to understand how and why its value changes over time. This can make debugging and understanding the flow of data much harder.
- **Potential for Naming Conflicts:** As your application grows, you risk accidentally using the same name for a global variable in different parts of your code, leading to conflicts and errors. ▾

BTW:
C# does not allow
global variables.

Avoid a global ContactProvider



```
class _MyHomePageState extends State<MyHomePage> {  
  int _selectedIndex = 0;  
  var contactProvider = ContactProvider();  
  
  @override  
  Widget build(BuildContext context) {  
    List<Widget> pages = [  
      ChatsWidget(contactProvider: contactProvider),  
      const Text("This is the Updates page", style: TextStyle(fontSize: 20)),  
      const Text("This is the Communities page",  
        style: TextStyle(fontSize: 20)), // Text  
      const Text("This is the Calls page", style: TextStyle(fontSize: 20)),  
    ];  
  
    return Scaffold(  
      drawerScrimColor: Colors.transparent,  
    );  
  }  
}
```

```
class ChatsWidget extends StatelessWidget {  
  const ChatsWidget({super.key, required this.contactProvider});  
  
  final ContactProvider contactProvider;  
  
  @override  
  Widget build(BuildContext context) {  
    var contacts = contactProvider.contacts;  
  
    return Column(  
      children: [  
        ContactListTile(contact: contacts[0]),  
      ],  
    );  
  }  
}
```

This is called
“constructor based dependency injection”

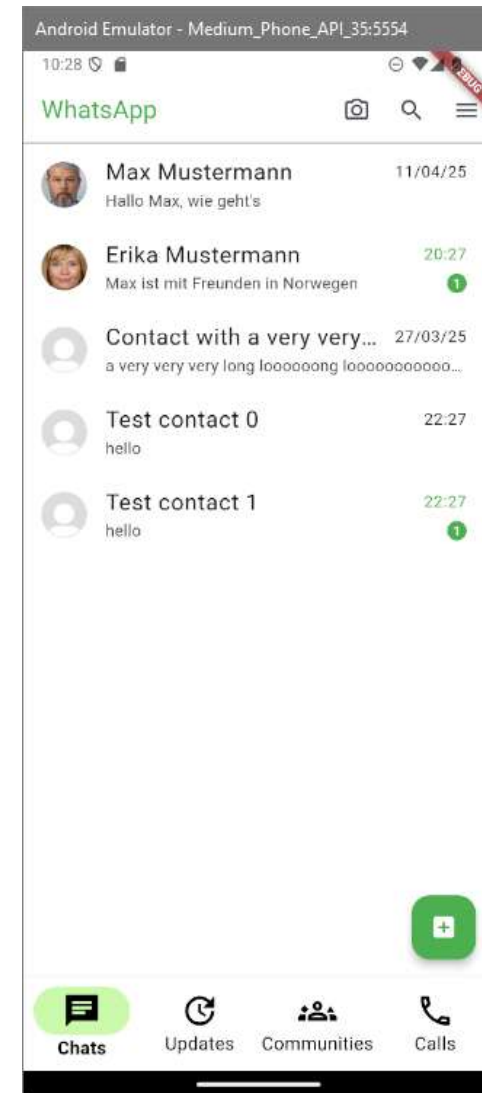
Displaying many contacts

in ContactProvider:

```
for (var i = 0; i < 20; i++) {  
    contact = Contact("Test contact $i");  
    contact.countOfNewMessages = i;  
    contacts.add(contact);  
}
```

in ChatsWidget:

```
return Column(  
    children: [  
        ContactListTile(contact: contacts[0]),  
        ContactListTile(contact: contacts[1]),  
        ContactListTile(contact: contacts[2]),  
        ContactListTile(contact: contacts[3]),  
        ContactListTile(contact: contacts[4]),  
    ],  
); // Column
```



Collection for in Dart

Using `for` in Collections (Loop Elements)

You can use `for` inside collections to generate elements dynamically.

```
dart

void main() {
  List<int> doubled = [for (int i = 1; i <= 5; i++) i * 2];

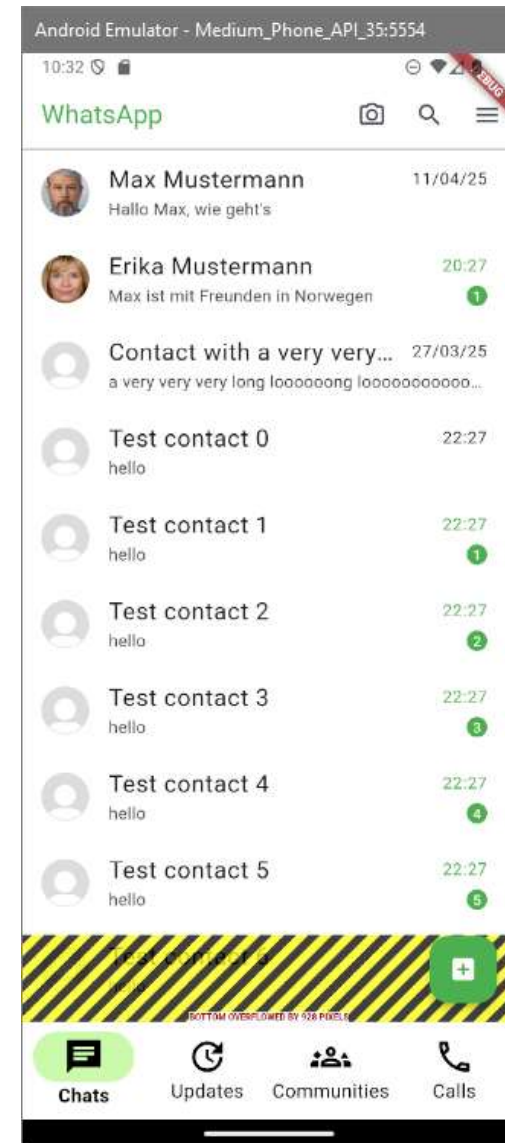
  print(doubled); // Output: [2, 4, 6, 8, 10]
}
```

```
@override
Widget build(BuildContext context) {
  var contacts = contactProvider.contacts;

  return Column(
    children: [
      for (int i = 0; i < contacts.length; i++)
        ContactListTile(contact: contacts[i])
    ],
  ); // Column
}
```

Alternative:

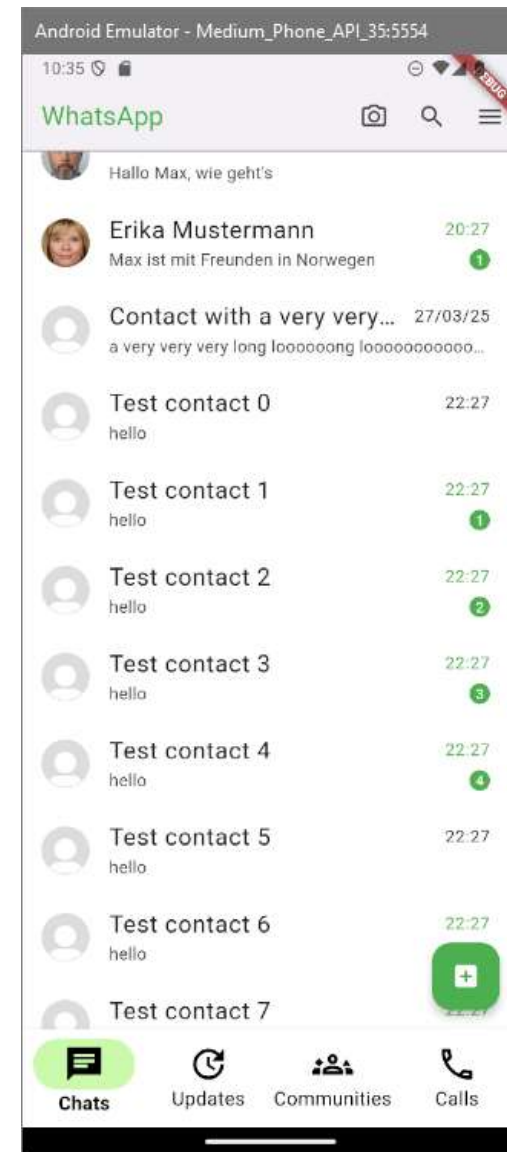
```
return Column(
  children: [
    for (var contact in contacts)
      ContactListTile(contact: contact)
  ],
); // Column
```



With SingleChildScrollView

```
@override
Widget build(BuildContext context) {
  var contacts = contactProvider.contacts;

  return SingleChildScrollView(
    child: Column(
      children: [
        for (int i = 0; i < contacts.length; i++)
          ContactListTile(contact: contacts[i])
      ],
    ), // Column
  ); // SingleChildScrollView
}
```



Alternative: ListView



```
@override
Widget build(BuildContext context) {
  var contacts = contactProvider.contacts;

  // return SingleChildScrollView(
  //   child: Column(
  //     children: [
  //       for (int i = 0; i < contacts.length; i++)
  //         ContactListTile(contact: contacts[i])
  //     ],
  //   ),
  // );

  return ListView(
    children: [
      for (int i = 0; i < contacts.length; i++)
        ContactListTile(contact: contacts[i])
    ],
  ); // ListView
}
```

A ListView handles scrolling internally.
It does not need a SingleChildScrollView.

We already used a ListView in our Drawer
(this was proposed by ChatGPT):

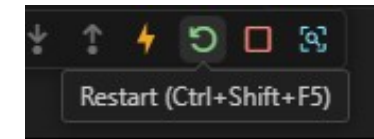
```
child: Drawer(
  backgroundColor: Colors.transparent,
  child: ListView(
    padding: EdgeInsets.zero,
    children: [
      ListTile(
        title: const Text('New group'),
        onTap: () {
          Navigator.pop(context); // Close the drawer
        },
      ), // ListTile
      ListTile(
        title: const Text('New broadcast'),
        onTap: () {
          Navigator.pop(context); // Close the drawer
        },
      ), // ListTile
    ],
  ), // Drawer
),
```


Performance in Chrome with about 500 Contacts



```
contact = Contact("Contact with a very very loooooong name");
contact.lastMessage = "a very very very long loooooong "
| | "loooooooooooooooooooooooooooooooooooooooooooooooooooooong message";
contact.dateOfLastMessage = DateTime(2025, 03, 27, 00, 11);
contacts.add(contact);

for (var i = 0; i < 500; i++) {
    contact = Contact("Test contact $i");
    contact.countOfNewMessages = i;
    contacts.add(contact);
}
```



A Restart of the App on Chrome takes:


- with ListView **less than 1 second**
- with Column & SingleChildScrollView **about 5 seconds.**

Realistic performance measurements should be done with the **Release version** of the App.



Build Debug and Release versions for Android

◆ AI Overview

An **APK file**, or Android Package Kit, is **the file format used by the Android operating system for distributing and installing mobile apps and middleware**. It's essentially a compressed archive containing everything needed to install and run an Android app, including the app's code, resources, and metadata. Think of it like a ZIP file for Android apps. 

Run the build command for the desired build mode:

- **Debug APK:** This build is faster to compile and is useful for testing on emulators or physical devices. It includes debugging symbols, which make it larger.

Bash 

```
flutter build apk --debug
```

- **Release APK:** This build is optimized for distribution to users. It's smaller, stripped of debugging symbols, and often obfuscated.

Bash 

```
flutter build apk --release
```

Build Debug and Release versions for Android



```
PS C:\flutter\repos\whatsapp_ui> flutter build apk --release

Font asset "MaterialIcons-Regular.otf" was tree-shaken, reducing it from 1645184 to 3484 bytes
(99.8% reduction). Tree-shaking can be disabled by providing the --no-tree-shake-icons flag whe
n building your app.
Running Gradle task 'assembleRelease'...                               30,9s
✓ Built build\app\outputs\flutter-apk\app-release.apk (18.9MB)
PS C:\flutter\repos\whatsapp_ui> flutter build apk --debug

Running Gradle task 'assembleDebug'...                               2.502ms
✓ Built build\app\outputs\flutter-apk\app-debug.apk
```

Local Disk (C:) > flutter > repos > whatsapp_ui > build > app > outputs > flutter-apk	
Name	Size
app-debug.apk	109.022 KB
app-debug.apk.sha1	1 KB
app-release.apk	19.354 KB
app-release.apk.sha1	1 KB

Apk files are zip files. To prove this, make a copy e.g. of app-release.apk and change its file extension to zip:

build > app > outputs > flutter-apk > app-release - Copy.apk.zip	
Name	Type
assets	File folder
kotlin	File folder
lib	File folder
META-INF	File folder
res	File folder
AndroidManifest.xml	Microsoft Edge HTML Document
classes.dex	DEX File
DebugProbesKt.bin	BIN File
kotlin-tooling-metadata.json	JSON File
resources.arsc	ARSC File

What are .sha1 files ?

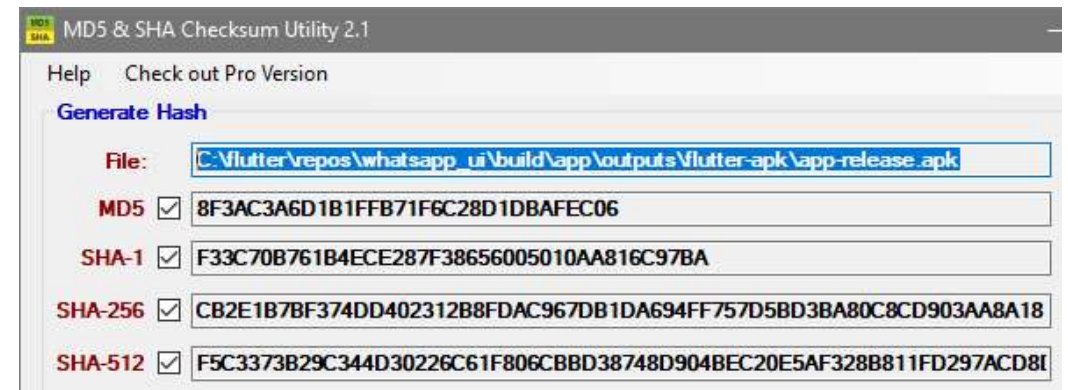
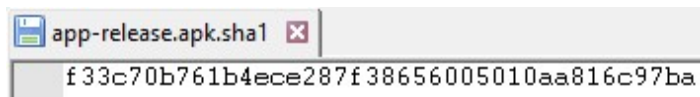


SHA-1 (Secure Hash Algorithm 1) is a cryptographic hash function designed by the United States National Security Agency (NSA) and published in 1995 by the National Institute of Standards and Technology (NIST) as part of the Secure Hash Standard (SHS).

Here's a breakdown of what that means:

- **Cryptographic Hash Function:** It's a mathematical algorithm that takes an input (or "message") of any size and produces a fixed-size output called a hash, message digest, or fingerprint.
- **Collision Resistance (Weakened):** Ideally, a hash function should be collision-resistant, meaning it should be extremely difficult to find two different inputs that produce the same hash output. While SHA-1 was initially considered secure in this regard, vulnerabilities have been discovered, significantly weakening its collision resistance.

app-debug.apk	109.022 KB
app-debug.apk.sha1	1 KB
app-release.apk	19.354 KB
app-release.apk.sha1	1 KB



Install release versions of your app on an Android device



```
PS C:\flutter\repos\whatsapp_ui> flutter devices
Found 5 connected devices:
  SM A336B (mobile) • RZCW200FZAW • android-arm64 • Android 14 (API 34)
  sdk gphone64 x86 64 (mobile) • emulator-5554 • android-x64 • Android 15 (API 35) (emulator)
  Windows (desktop) • windows • windows-x64 • Microsoft Windows [Version 10.0.19045.4291]
  Chrome (web) • chrome • web-javascript • Google Chrome 135.0.7049.85
  Edge (web) • edge • web-javascript • Microsoft Edge 135.0.3179.73

PS C:\flutter\repos\whatsapp_ui> flutter install --release -d emulator-5554
Installing app-release.apk to sdk gphone64 x86 64...
Uninstalling old version...
Installing build\app\outputs\flutter-apk\app-release.apk... 1.440ms
```

Don't forget to save all your code changes and to re-build the apk of the release version before installing it.

When you debug in VS Code by pressing F5, saving and re-building is done automatically before the install.

Performance measurements on Android emulator



Tests were done with :

```
for (var i = 0; i < 5000; i++) {  
    contact = Contact("Test contact $i");  
    contact.countOfNewMessages = i;  
    contacts.add(contact);  
}
```

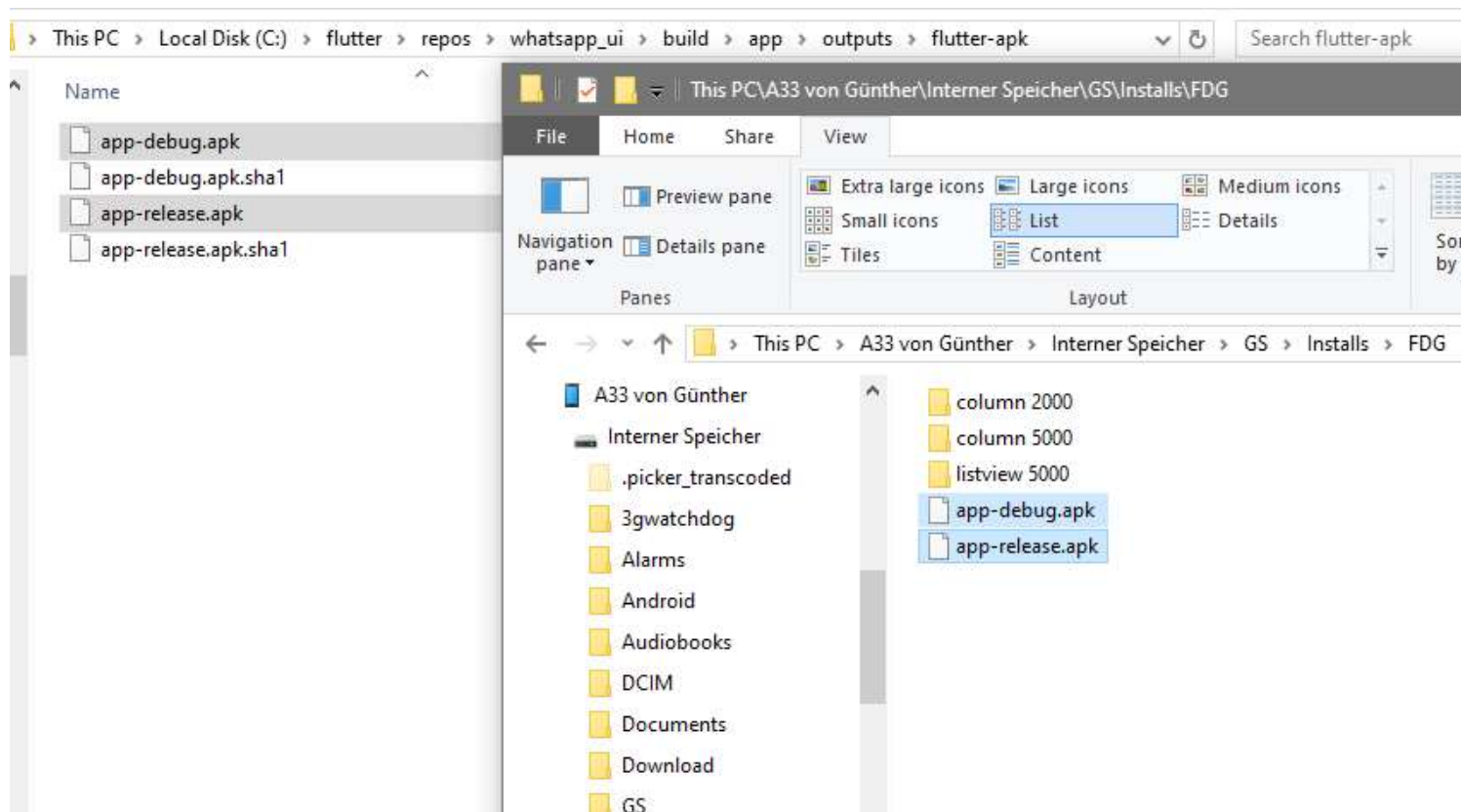
Application launch times	Release	Debug
with Column and SingleChildScrollView	10 s *)	18 s
with ListView	1s	4 s

*) These measurements had a wide range (two times 4s and four times about 15s).
BTW: Before repeating a measurement, I always closed all running apps on emulator.

Other ways to install apk files on your Android phone



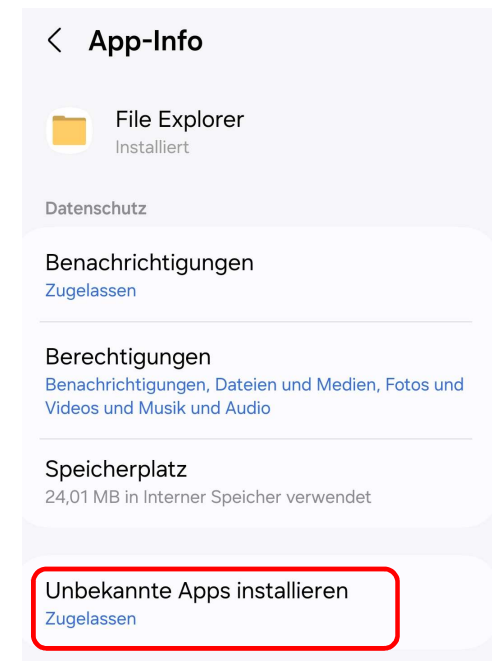
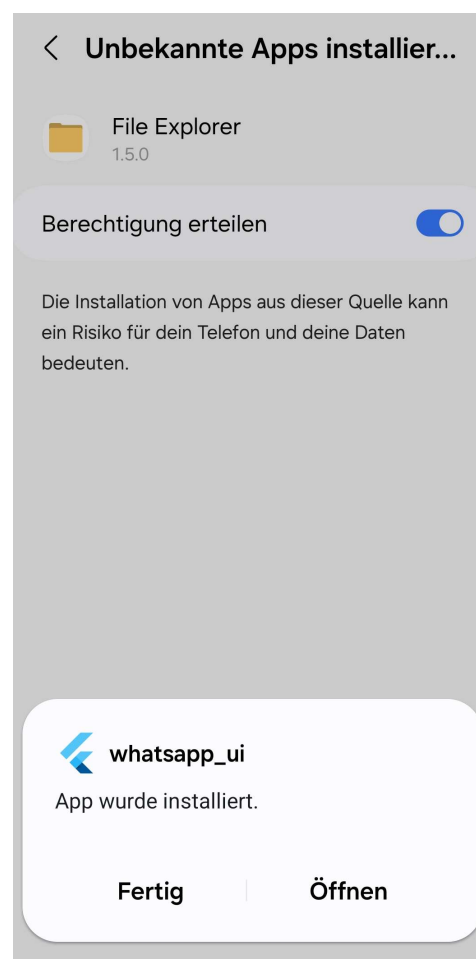
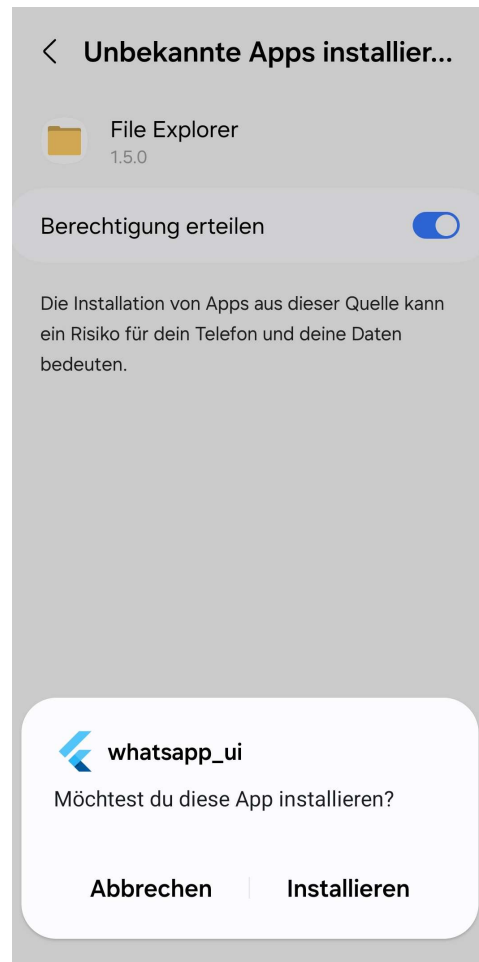
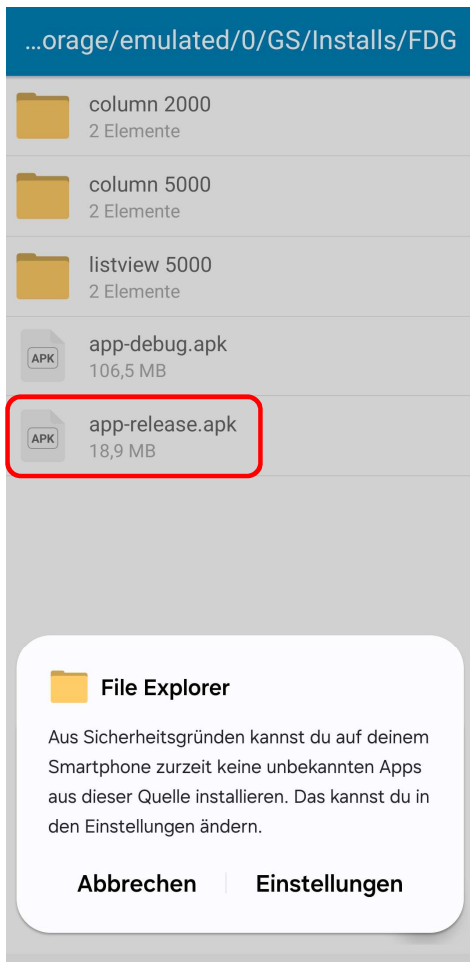
When your Android device is connected to your PC, you can open the Android file system in your Windows Explorer and drag&drop files from Windows to Android:



Install apk files via File Explorer



I used one of these two Explorers on my phone.



For security reasons you should reset this permission later in the App settings.



Publish your apk files on GitHub

The screenshot shows the GitHub interface for the repository 'fdg2425 / apks'. The repository is public and has 1 branch (main) and 0 tags. The commit history shows three commits: 'Update README.md' (1 hour ago) and two commits for '2 versions of our whatsapp_ui for performance comparison (listview vs...)' (17 minutes ago). The file list includes 'README.md', 'whatsapp_ui_column_5000_release.apk', and 'whatsapp_ui_listview_5000_release.apk'. The README content is visible below the file list, showing the repository name 'apks' and a description: 'apk files for some of the apps we have developed during a flutter training at FDG in 2024/2025'.

fdg2425 / apks

Type to search

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

apks Public Pin Unwatch 1

main 1 Branch 0 Tags Go to file + Add file <> Code

fdg2425 2 versions of our whatsapp_ui for performance comparison (listview vs... 7dcd111 · 17 minutes ago 3 Commits

File	Commit	Time
README.md	Update README.md	1 hour ago
whatsapp_ui_column_5000_release.apk	2 versions of our whatsapp_ui for performance compariso...	17 minutes ago
whatsapp_ui_listview_5000_release.apk	2 versions of our whatsapp_ui for performance compariso...	17 minutes ago

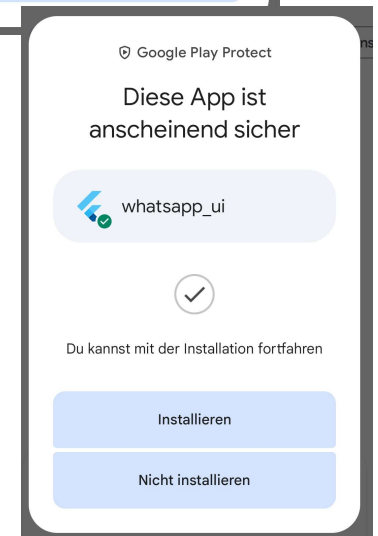
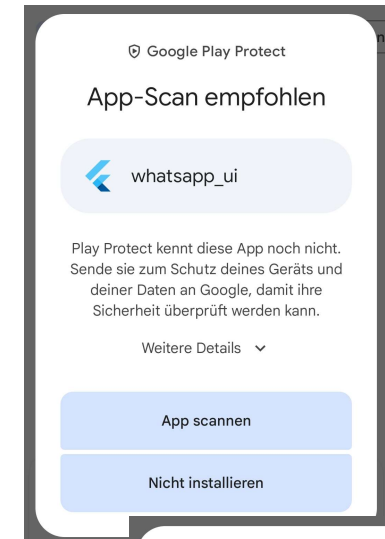
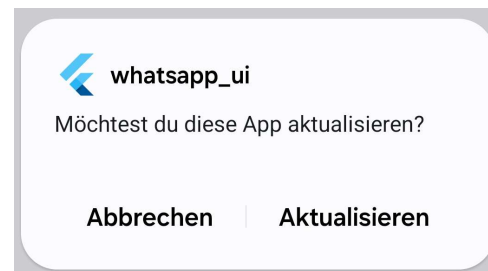
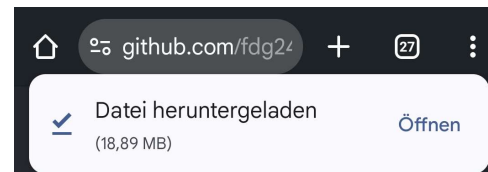
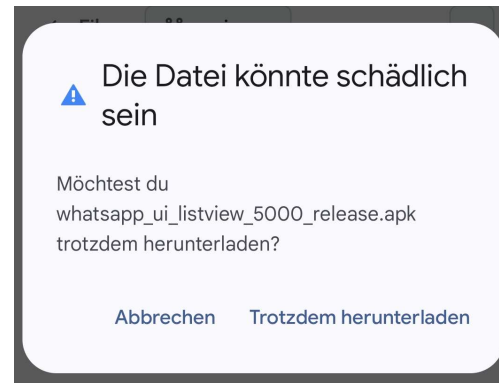
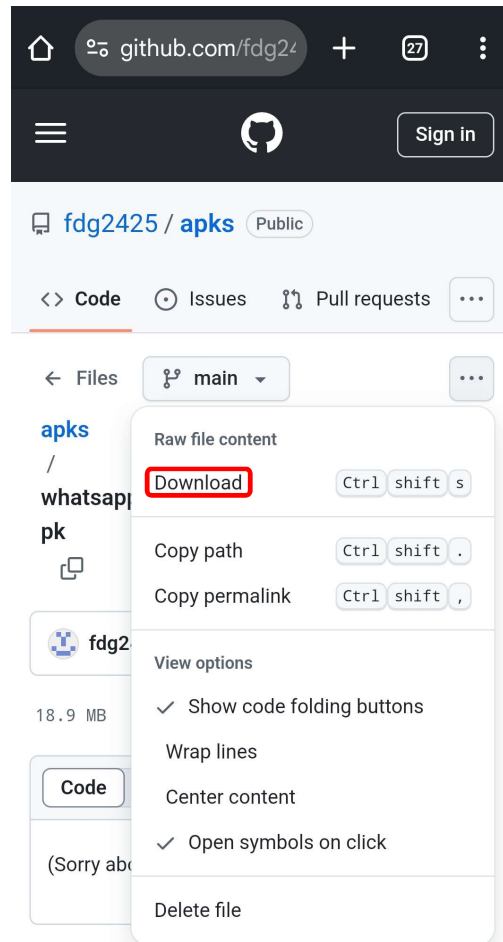
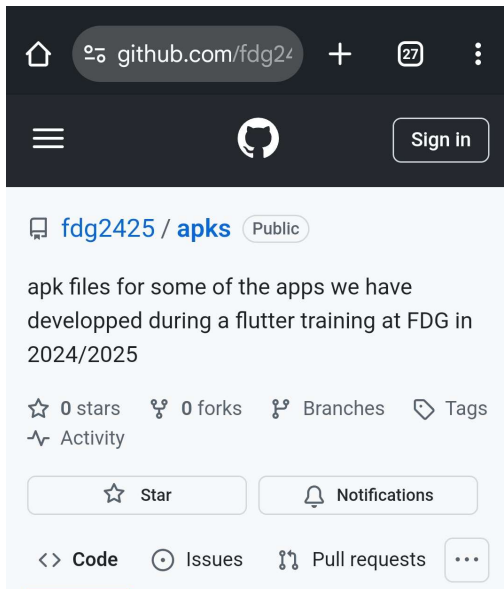
README

apks

apk files for some of the apps we have developed during a flutter training at FDG in 2024/2025

Link to GitHub: <https://github.com/fdg2425/apks>

Download & Install from GitHub on your phone





Publishing an app on Google Play Store

what are the preconditions to upload apps to Google Play store

✦ To upload apps to the Google Play Store, you need to meet several preconditions. These can be broadly categorized into account-related requirements, app-related requirements, and store listing requirements. Here's a detailed breakdown:

1. Google Play Developer Account:

- **Registration:** You need a Google account and must register for a Google Play Developer account through the Google Play Console.
- **Age Requirement:** You must be at least 18 years of age to sign up.
- **Developer Distribution Agreement:** You'll need to review and accept the Google Play Developer Distribution Agreement.
- **One-time Registration Fee:** There is a one-time registration fee of \$25 USD. This can be paid using major credit or debit cards (MasterCard, Visa, American Express, Discover (U.S. only), Visa Electron (outside the U.S. only)). Prepaid cards are not accepted.

This list has another 2 pages ...



.apk or .aab files

can apk files be uploaded to Google Play store



Yes, APK (Android Package Kit) files can still be uploaded to the Google Play Store. ▾

While Google prefers and recommends the use of Android App Bundles (.aab) for publishing new apps and updates due to their benefits in terms of app size optimization and dynamic delivery, the Play Store still accepts APK files. ▾

Here's a more detailed explanation:

- **Android App Bundle (.aab):** This is the newer publishing format. ▾ When you upload an .aab file, Google Play's servers generate optimized APKs for different device configurations (screen density, CPU architecture, languages, etc.). This results in smaller app downloads for users, as they only receive the components they need. ▾
- **APK (Android Package Kit):** This is the traditional format for Android apps. ▾ An APK file contains all the necessary elements for an app to be installed on a device. ▾ When an APK is uploaded, all users, regardless of their device configuration, download the same file. This can lead to larger download sizes compared to apps published as app bundles.

Key points to consider:

- New apps submitted to the Google Play Store are required to use the **Android App Bundle** format as of August 2021. ▾



How to build Android App Bundles

```
PS C:\flutter\repos\whatsapp_ui> flutter build appbundle

Warning: SDK processing. This version only understands SDK XML versions up to 3 but an SDK XML file of version 4 was
encountered. This can happen if you use versions of Android Studio and the command-line tools that were released at d
ifferent times.
Font asset "MaterialIcons-Regular.otf" was tree-shaken, reducing it from 1645184 to 3484 bytes (99.8% reduction). Tre
e-shaking can be disabled by providing the --no-tree-shake-icons flag when building your app.
Running Gradle task 'bundleRelease'... 43,7s
✓ Built build\app\outputs\bundle\release\app-release.aab (18.9MB)
```

whatsapp_ui > build > app > outputs > bundle > release	
Name	Size
app-release.aab	19.395 KB

for comparison:

whatsapp_ui > build > app > outputs > flutter-apk	
Name	Size
app-release.apk	19.342 KB
app-release.apk.sha1	1 KB

aab files are also zip files:

app-release - Copy.zip > base > assets > flutter_assets > assets > images	
Name	Compressed size
default_image.jpg	8 KB
em.jpg	26 KB
gs.jpg	103 KB
mm.jpg	26 KB

Günther on Google Play Store



In 2023, I uploaded 2 of my first Flutter Apps to Play Store:




Visualize $z=f(x,y)$
Guenther Schmitt



Don't forget!
Guenther Schmitt

Some weeks ago, Google warned me:

 **Your developer account is at risk of being closed because it's not being used**

Hello Play Console developer,

You have 60 days to fix issues with your Play Console developer account, Guenther Schmitt.

Fix this before June 8, 2025 to prevent your account from being closed. If you need help signing in, reach out to the policy support team. [Contact us](#)

So I have build a new version of the first app with a current Flutter SDK and uploaded it to Play Store (registration of this new app took me more than 2 hours ...):



function_plotter_2025
Guenther Schmitt

Link to Play Store:

https://play.google.com/store/apps/details?id=de.julig.function_plotter_2025

Link to Web version:

https://guentherschmitt.github.io/web_test/function_plotter_2025/

Final sum-up: Comparison Column vs. ListView



ChatGPT ▾

Share

flutter when to use Column compared to ListView

Great question — in Flutter, both `Column` and `ListView` are used to display a vertical list of widgets, but they serve slightly different purposes and have different behaviors. Here's a breakdown to help you decide when to use each:

✅ Use `Column` when:

1. You have a small, fixed number of children.
2. You don't need scrolling — or scrolling is handled by a parent (like in a `SingleChildScrollView`).
3. You want everything laid out immediately (eager rendering).

✅ Use `ListView` when:

1. You have a long or dynamic list of items.
2. You need built-in scrolling.
3. You want to benefit from lazy loading (only rendering visible items).
4. You're working with lists that can grow or come from a backend.