

Open in app ↗



Search



Flutter: Summary of Pitfalls



Viachaslau Lyskouski

17 min read · Nov 29



Listas



Share



Mars

***Disclaimer:** This text contains a tiny emotional implication of the provided information, for an additional emotional accompaniment check [link]*



Flutter: Summary of Pitfalls

D reamble

FFlutter is proposed as a cross-platform framework that empowers the creation of platform-agnostic applications from a single codebase. Dart is an object-oriented programming language developed by Google (as well as Flutter),

known for its indispensability of Flutter-based development.

O bjectives

U Flutter and Dart are NOT sustainable for enterprise development, FinOps, CAD, BigData, AI (*as a part of app, not API-based*)... any sort of DeepTech applications.

O verview: Flutter Advantages, that

Advantages, that's been mentioned through the enormous scope of advertisement materials, are going to be ruined by the text below.

Single codebase for all platforms

That's true only for UI, whenever it would be needed any platform-specific feature, be ready to use Swift for iOS/macOS, Kotlin or Java for Android, C++ for Linux, and C++/C# for Windows. Have fun!

Reduced Development Time

Poorly designed test libraries won't help you to test your application properly.
Position of elements? Their state (colors, font types, ...)? Partial wrapping?
Monkey-penetration testing? Integration tests (designed) for CI/CD? It's proposed
to use images as a way to test...

Lack of understanding their importance can be assessed by the next post:

To help developers reap these benefits, **developers with newly created personal Play Console accounts will soon be required to test their apps with at least 20 people for a minimum of two weeks before applying for access to production.** This will allow developers to test their app, identify issues, get feedback, and ensure that everything is ready before they launch. Developers who create new personal developer accounts will start seeing this [requirement](#) in Play Console in the coming days.

<https://android-developers.googleblog.com/2023/11/ensuring-high-quality-apps-on-google-play.html>

Automation is lost as a qualificational domain... Have fun!

I've been impressed by a simplicity from Windows and Linux Snap Store, when it's not needed any manual activity after typing "git tag", that trigger a distribution process, to see an updated app in a market after a while. And my respect to Huawei Gallery and Apple Store (*Samsung and Windows do have a certification phases as well, but Huawei and Apple were the only that pin me to errors*), they won't accept your app till their hands-on crosscheck.

Increased Time-to-Market Speed

Time-to-Market won't be ever interconnected with a framework usage, it's all about the corporate culture, a company processes and goals. T2M of Threads from Facebook was one month by a team of experts, T2M of Fingrom was 200 hours by a person who's never seen Flutter/Dart before... So, you may do it even within a day or less if you know how.

Native-like Performance

Everything is not a widget as you may think, everything is a single Canvas with a double buffering. You may do some sort of painting in a way you wish... with animations and lazy loading to hide the performance bottleneck by trying to show all at once. Just imagine, 1k elements on a page is enough to recognize the freeze. Have fun!

Powerful community

It's a passively-aggressive community without any communication rules and culture. The moderated just filter what [or whom] they don't like, check the image below with a post that's not been published, and it wasn't the only one that I've sent (without being accepted):

Just for the case if the community will decide to cleanup their conversations (do not see any options to hide supportive materials, my excuses for a divert flow):



Mcflan 75 · 7 days ago

I'm sorry but you wrote book about a topic knowing nothing about it?

I have tried reading the first part of your book, it's just so amateurish ... and the fact that you know so little about the subject is telling.

I would not be so harsh if you were not trying so much to promote yourself...

↑ 16 ↓ Reply Share ...



VLyškouski OP · 7 days ago

I've shared my experience in taking Flutter and building something with its help. Would appreciate your details of such a strong word as "amateurish".

It's not about promoting at all... you've read the book but haven't caught the concept. It's not a book to teach, it's not an app to distract the market; it's my journey of fluttering.

Yes, I do know nothing (the more you learn, the more questions arise, the less you know and understand) and I'm pleased to hear any valuable thoughts to become better of my own.



-8



1

Reply Share ...



Mcflan 75 · 7 days ago

No, that is not what you are trying to sell, from your own description:

The goal of this book is not to teach how to create applications; it empowers to become a proficient, well-rounded developer.

First of all this one of the many sentences that make not much sense, I believe you are either using a translator of ChatGPT since this other sentence is pretty weird : It's an invitation to embark on a collaborative journey through the thrilling realm of platform-agnostic application development.

And again, you say "it's not an app to distract the market", but guess what the name of your book is? From zero to Market.

But at the end you are giving a lot of advices and code snippets when you do not understand the core of the langage, because when I read your book the writing is poorly structuced, full of AI generated text (or maybe I am wrong), you jump from one topic to another like its your personal notes, and a lot of the code are not better than following the Flutter and Dart documentation.



12



1

Reply Share ...



1111111132323233 · 6 days ago

Reads like it was written by somebody who has no clue what they're talking about.



1 Reply Share ...

by example, next section " " while it's " " or " " and not " ".



-1 Reply Share ...



1111111132323233 - 6 days ago

what?



1 + Reply Share ...



Vl vskouski **OP** · 5 days ago · edited 5 days ago

Can you construct your statements with respect for people, even if they are completely wrong? Have you ever done a code review? Or discussed product development strategy? And you've used the same words for anyone you were communicating with, haven't you? Come on...

No clue about what?

Initial concept forming? Bootcamping for a new technology? Writing tests from unit to scalability? CI/CD for verification and distribution? UI/UX Patterns and research strategies? Defining KPI and OKR? Marketing flow?



-1 ...



1111111132323233 · 5 days ago

No clue about anything. Your book sucks. It's as simple as that. No need to be an asshole to everybody in the comments...your attitude diminishes any chances of someone buying your book.



2 Reply Share ...



thelonesomeguy · 7 days ago

Question: Why write it as a book instead of say, a series of blogs?

A lot of stuff in the book would change drastically with time, at least with articles you can edit them after the fact and everyone will see your latest version.

 4 Reply Share ...



sgpope · 7 days ago

Because you can't sell a blog

 12   Reply Share ...



Mcflan 75 · 7 days ago

Go read the free sample and tell me you would buy it, this is self promotion and not the good kind.

↑ 8 ↓ Reply Share ...



Vlyskouski OP · 7 days ago · edited 3 days ago

It can be read the full book for free, just compile it from the sources. Or, buy by spending \$0 via Smashwords. A book for the price of a coffee cup, don't be so cocky.

↑ -4 ↓ Reply Share ...



From Zero to Market with Flutter: Desktop, Mobile, and Web Distribution Kindle Edition

by Viachaslau Lyskouski (Author) | Format: Kindle Edition

[See all formats and editions](#)

Kindle \$2.09	Hardcover \$15.99	Paperback \$7.77
Read with our free app	1 New from \$15.99	1 New from \$7.77

This book is more than a set of instructions; it's an invitation to embark on a collaborative journey through the thrilling realm of platform-agnostic application development using Flutter. Together, we'll explore every phase of the application lifecycle, starting from the initial concept to the final distribution, with essential stops at development, testing, and automation along the way. This collaborative learning experience promises to be both exciting and educational as we delve deeper into this process as reading through.

The goal of this book is not to teach how to create applications; It empowers to become a proficient, well-rounded developer.

Print length	Language	Sticky notes	Publication date	File size
 332 pages	 English	 On Kindle Scribe	 November 1, 2023	 8065 KB

 Report Incorrect product information.

<https://www.amazon.com/dp/B0CL1KNXOR> 2023-11-25

Any time, you would ask for a help, it would be needed to prepare a distilled code to reproduce the issue (for the Flutter GitHub Issues). By doing so, you would definitely find a solution by your own without any assistance. As an example, it

should not be expected any support on a really complex problems as <https://github.com/flutter/flutter/issues/138880> (after the latest SDK upgrade to 3.16.0).

Possibly, they do think that if they're using tools / languages crafted by Google, contribute into their projects' codebase, then they almost work there. It's even cannot be understood that "well-rounded developer" means not bounded by limitations (frameworks or languages), when you learn things that you've not been even aware about.

Not so far, I've heard a great phrase "If you hear someone who speaks your language brokenly. Think twice before a dismissive snort, because it's the second language she/he knows". It's our respect to the language and culture, we're trying to adjust ourself to be understood and

我们希望被倾听. Have fun!



Fingrom Google Analytics. 2023-11-27 9⁰⁰



Fingrom Google Analytics, 2023-12-12 20:00

Own Rendering Engine

The truth is that it's not so, Google does, and those two things are not equal.

Flutter uses Skia (2D painting library) as the engine. And if you've been doing some sort of CAD systems by using Flutter... Skia is in a way of replacement by Impeller (done for iOS, Android ongoing). Meaning that all low-level APIs have been changed. Have fun!

Hot Reload Feature

It's not even close to Erlang, and build-in-time languages. The simplest aspect of the change can be hot reloaded at max (*furthermore, Web runner doesn't have such an option at all*):

Hot reload was rejected: Limitation: changing type parameters does not work with hot reload.

Flutter's Safety

If you've omitted the type, it might not be recognized properly at the compilation phase (but it would be raised as an exception in production):

```
// ! Error during the app run: `numberMedium` is not recognized
// Success on Build
build(context) => Text('sample', style: context.textTheme.numberMedium)

// Highlights the needs to import extension
// Fails on Build
build(BuildContext context) => Text('sample', style: context.style.numberMed
```

Even more, your application might be frozen (*without any errors*) by not being able to identify that a negative integer is not a double. I'm still confused by such an critical mistake that **integers** are differentiated from **doubles**.

```
void addLog(uuid, dynamic initial, dynamic initialValue, dynamic value, [S
  if (_history[uuid] == null) {
    // _history[uuid] = SplayTreeMap<double, TransactionLogData>();
    history[uuid] = SplayTreeMap<int, TransactionLogData>();
```

```
    }
    if (initialValue != value) {
        // App spinner is shown with double forever... cannot proceed further
        _history[uuid]![-initial.createdAt.millisecondsSinceEpoch] = Transaction(
            ...
        );
    }
}
```

```
SizedBox(height: 10) // Success  
  
const height = 10;  
SizedBox(height: height) // !Error
```

And more;

```
typedef Currency = ({String code});\n\n\nclass SomeSortOf extends StatelessWidget {\n    final Currency? currency;\n\n    SomeSortOf({required this.currency})\n\n    // !Error: Cannot be null\n    build(context) => Text(currency?.toString() ?? 'Not defined');\n}
```

If it's not enough, check the epic:

```
class ListSelectorPage<T extends Object?> extends StatefulWidget {
    final T? result;
    final List<ListSelectorItem> options;
    final String tooltip;
    final FntSelectorCallback? itemBuilder;

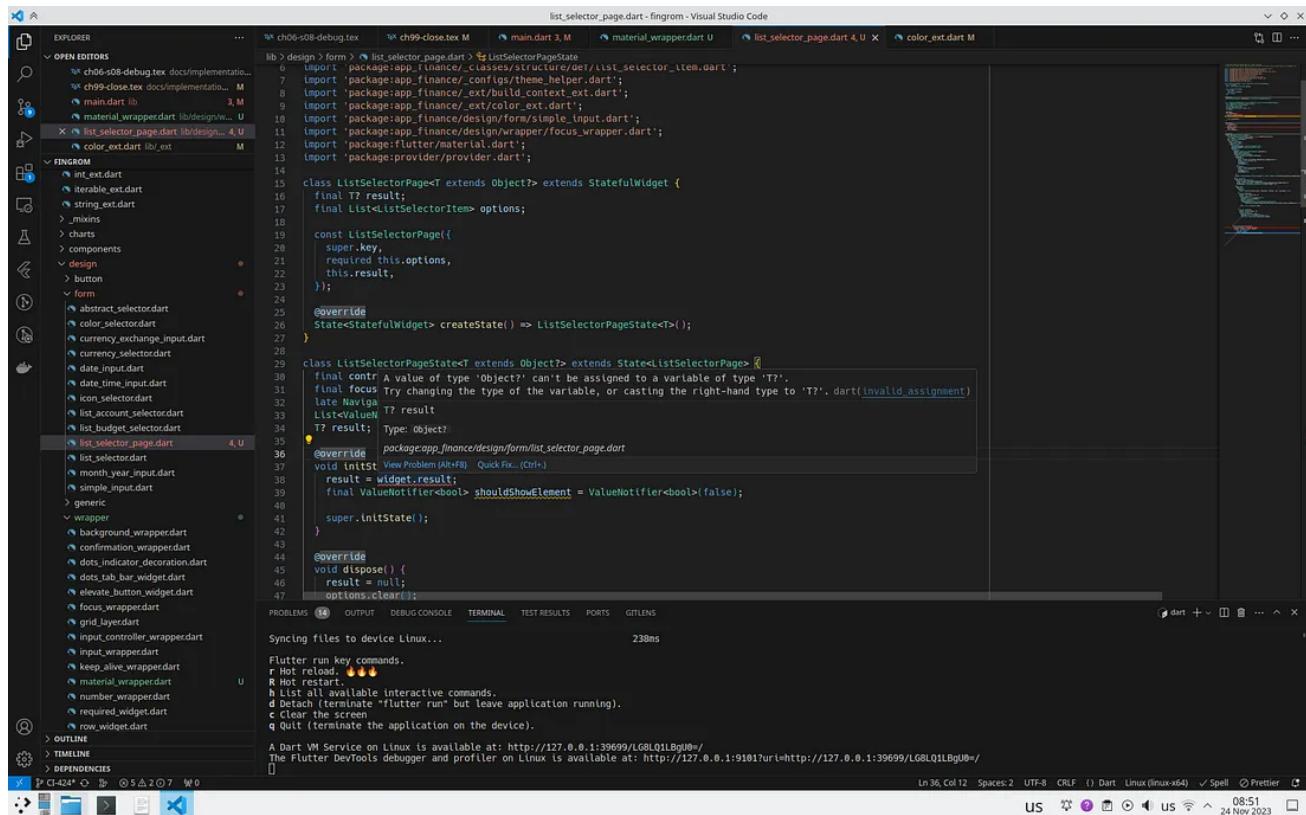
    const ListSelectorPage({
        super.key,
        required this.options,
        required this.tooltip,
        this.result,
```

```
        this.itemBuilder,
    });
}

@Override
State<StatefulWidget> createState() => ListSelectorPageState<T>();
}

class ListSelectorPageState<T extends Object?> extends State<ListSelectorPage> {
    final TextEditingController controller = TextEditingController();
    late NavigatorState nav;
    List<ValueNotifier<bool>> show = [];
    List<ListSelectorItem> options = [];
    // !ERR: A value of type 'Object?' can't be assigned to a variable of type
    dynamic result; // T? result:

    //... other code
}
```



Error: a value of type 'Object?' can't be assigned to a variable of type 'T?'

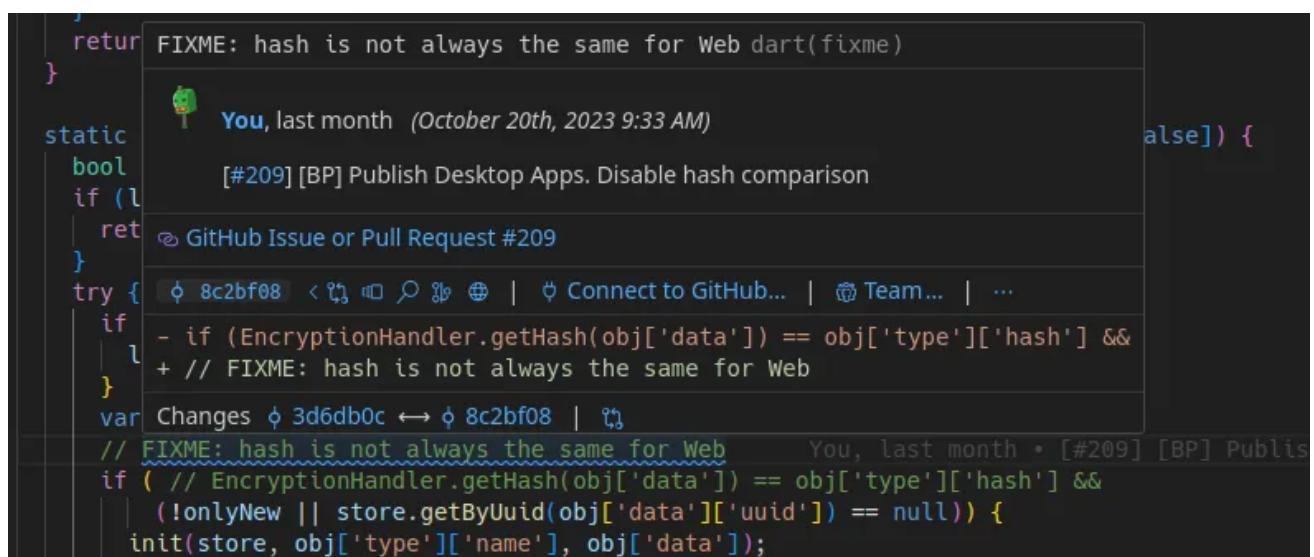
Security

Who cares about a security and the fact that used “.env” file can be compromised within a second from an application (written on Flutter)? Be careful, since your app is no more than an archive of files; and your obfuscation is not meshing the sequence of instructions; the name change will stop only noobs.

In case, you would try to implement some sort of security (combined with a transfer protocol, as an example):

```
// the simplest protection against data modification during transmission  
// by comparing hash  
md5.convert(utf8.encode(data.toString())).toString()
```

md5 might not be the same between environments:



Have fun!

Architecture

I've checked a dozen of materials (*a few are listed there: <https://github.com/lyskouski/app-finance/issues/217>; and stopped myself when I felt like a garbage collector*), just check <https://codewithandrea.com/articles/comparison-flutter-app-architectures/> and have fun as written there at the beginning.

“If you research this topic, you may come across terms such as MVC, MVP, MVVM, and Clean Architecture... to solve similar problems” [∗]. Their diversity is reflected on resolving different problems for different solutions, and the mentioned architectural patterns is no more than distilled abstractions. Let’s me compress what’s been said by Robert Cecil Martin in his book “Clean Architecture” via a single sentence: be like a real architect who design the building per purpose, the

church won't be even near to a stadium, but both of them would have a wide space in the middle.

Inspired by React

Without any nearby understanding of how the virtual DOM is organized to effectively operate by a million (or so) of elements, or state management... If you don't trust me, try to implement alike-Excel table 10,000x10,000 with editable operations per each cell (and for a group-selection — at once), pinned rows and/or columns, and configurable formulas by using Flutter (*is possible for sure via InfernoJS with almost close performance compared to Excel-desktop*). Have fun!

Dart

“Developers’ reason to choose Flutter is because...” [★]. Let’s discuss that separately later...

Just as a note, relevant to the Flutter, due to the usage of Dart as an inter-layer of the compilation process: “*... next step is WebAssembly...*” that option is a part of unstable (master) branch for more than two years, and most of the projects, that’s been relying on that, — closed.

Excellent documentation and community support

Don’t rely on a documentation that is written in an advertisement manner:

<https://github.com/flutter/flutter/issues/138797> “*cap, we don’t see your problem, it’s not exist*”. Ok, sir, I won’t waste my time by explaining how it is supposed to be checked. But... stop, how many languages? Roughly 100?)) What about a support of languages that’s been invented directly for a virtual world? Assume, it’s been forgotten that fonts are just a set of pixels or sprites, that are arranged in a proper order to be interpreted through I/O and the locale is used as a switcher between different ranges. In general, documentation is a “must have” feature for any API, it’s not anyhow a differentiator. None of errors, that I’ve experienced, were covered either by a community or documentation... through the application development that is simpler than a calculator — money expense tracker.

Strong industry adoption

“Flutter [...] is used by well-known companies”, the fact that they’ve tried or use (possibly, they aren’t already) means exactly nothing by not knowing the context and the form of its usage. As a facade? Paid marketing? Who knows...

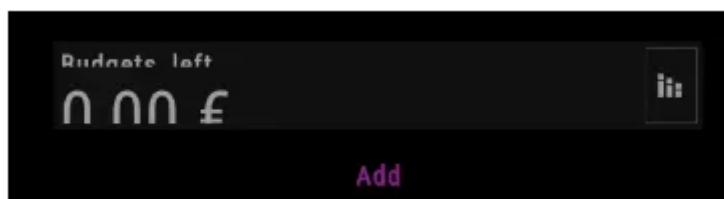
Google Play is rewritten by using Flutter, isn't it? Please, revert... I'm not able to find the button to rate an app. *You've taught people to be humble, they don't believe that their voice means anything. I sat for a few days knowing there was a problem [*], seeing them struggling with it, and yet hoping they would come to me with a question "when would it be fixed?".... and I have no clue how to return them their engagement,.. yet.*

An indicator of industry uptake is a thousands of open developer positions anywhere in the world. And for the TOP-10, demand is always higher than supply over decades.

Ongoing support is a mess of ignorance existing problems (5k+ issues) and a strange workflow (*auto-close the ticket if it's not been updated within a week or so; marked as "done" without being "solved"... good? great! let's go further*) that reflects me an old way of support in Microsoft when the quality was measured by a calls' time duration (when "shorter" was measured as "better"). Assume, it won't be solved a problem even if you'll pay them.

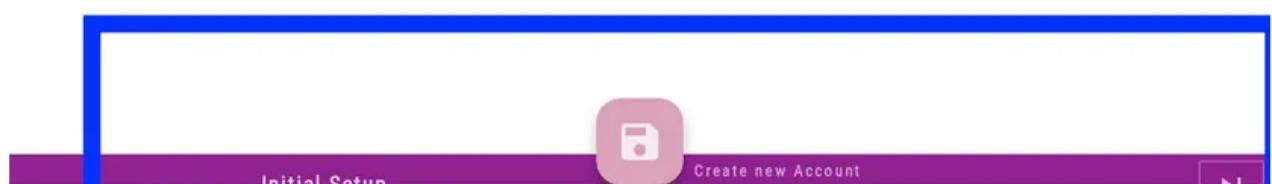
Moto “you control each pixel”

Shortly saying, I've had to “crack” the image-based tests by increasing an error threshold (pixel difference $\leq 2.1\%$) because of fonts. Even more, you're out of control for them (*and how enormously ugly their are in PWA if they're not a part of your distribution, just check*)... a user may change mobile preferences and you would take something like that:



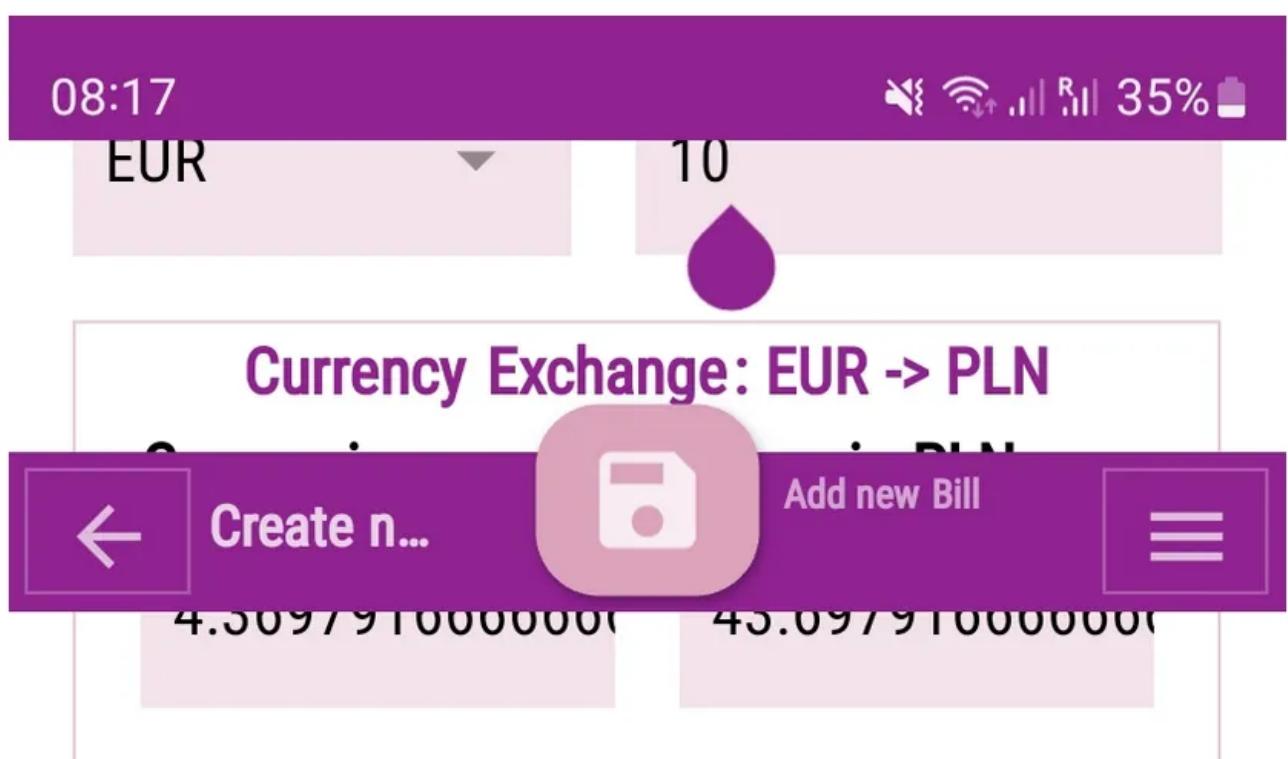
<https://github.com/lyskouski/app-finance/issues/372>

Or, the available height will be counted differently per each platform...



<https://github.com/lyskouski/app-finance/issues/399>

As for not resolved, yet (2023-11-27), bug:

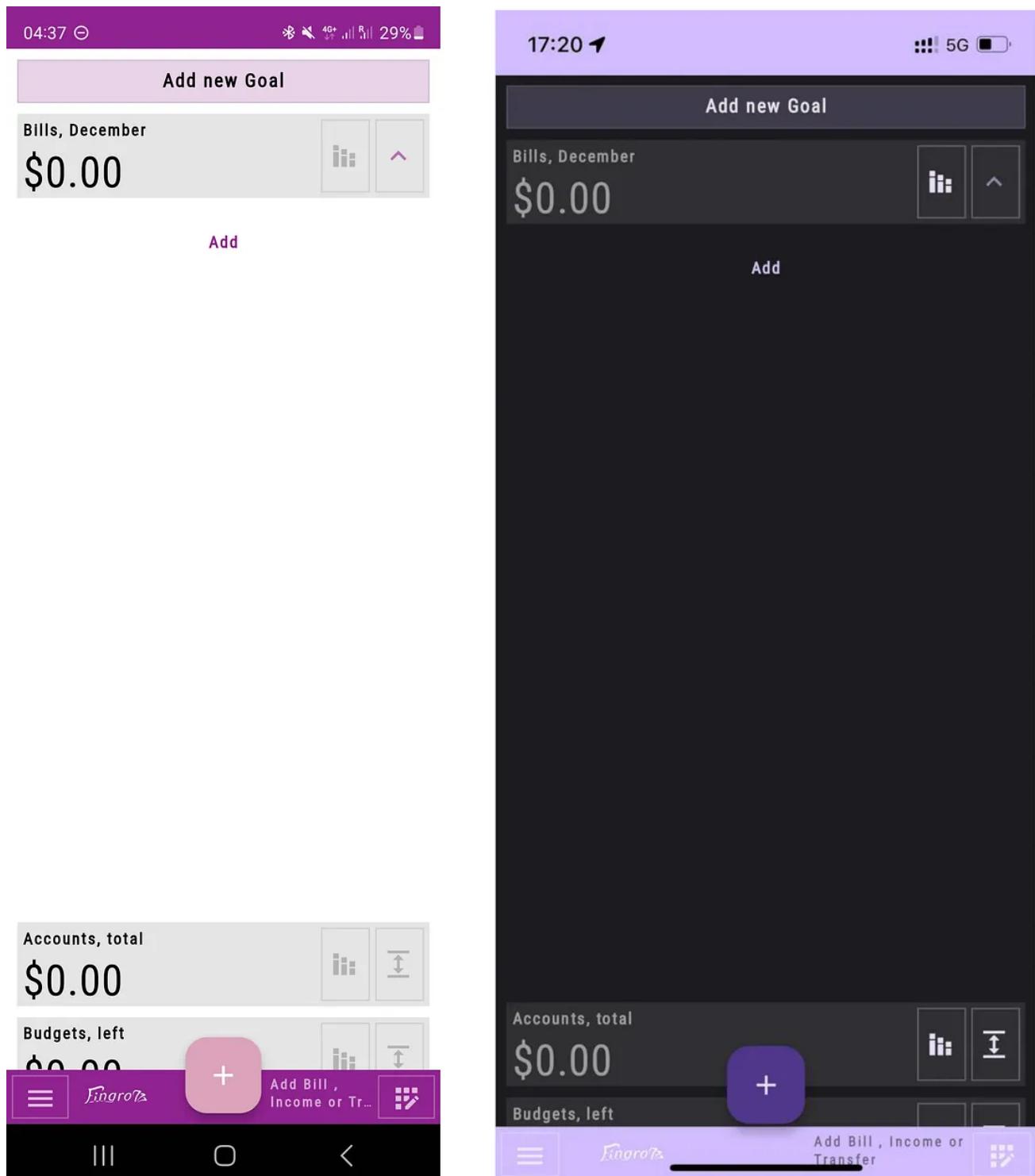


Description

Set Expense Details

<https://github.com/lyskouski/app-finance/issues/422>

Even iOS build (*reminder about another engine in use*) differs from Android:



FloatingActionButton location `centerDocked` for Android (left) vs iOS (right)

And, the bottom bar of a Scaffold will intentionally be concealed when the keyboard is opened, as per the Flutter SDK design. Have fun!

Mobile-first development

I might only speculate that they've never used their phones actively, otherwise the necessity of background processes will be a vital. I have no clue how to

implement something like Garmin Connect by using a pure Flutter.

Notifications are also outside the SDK. I do hate how LinkedIn has made their engagement notifications (they're too annoying and not configurable)... but the one of them ("Missed notifications") is invisible at start, only by slicing down the top panel. And it's a great UI for any app that has to pin a notification bar for the purpose of managing own activities. There is no way to implement anything roughly similar with Flutter. You won't even find a package that supports all platforms. May be, it would be better to rebrand "cross-platform" to a "mobile-agnostic front-end" development?

And, while we've mentioned games, you won't be able to create anything near to the modern games with an issue that I've seen in PHP 3 ~20 years ago and in Flutter 3 recently:

`1 / 2 * 2 = (?) 1 ... nope, it's 0.9999999999999998 or so`

A wide variety of components that keeps growing

I think that I've checked all UI components there (*let's preserve the date of that action: 2023-11-25 16:00 for <https://pub.dev>*) and found equally zero that follows WCAG 2.1 Accessibility Standards. Even core ones as SearchAnchor selector and DateTime picker suffer.

Possibly they've forgotten about devices without a touch screen and mouse, as TV. As a tip, the imagination extravaganza — RawKeyboardListener, a component to listen the keyboard actions, should be focused, otherwise shortcuts won't be recognized by an application.

And even more, different core components have a different interface notation.
Have fun!

```
 IconButton(  
   icon: // should be 'child', as for others  
)  
 ListTile(  
   title: // should be 'child', as for others
```

```
});  
// ... a dozen of other inconsistencies
```

As for the packages in their Package Repository (pub.dev), it's missing any culture of API-freezed distribution; and by upgrading you might find yourself in rewriting everything that you've done on top. In my case, I've decided to freeze gherking, encrypt, and flutter_webrtc ... since they give me a negative business value gross (*spending a time on upgrade without taking any additional benefits*).

By the way, “Supported by Google” is not equal to “Made by Google”, have fun!

Material Design and Cupertino widgets

By having two types you still have to choose one. It's merely possible to build it in a way that for iOS it would be used Cupertino, whereas for Android – Material Design.

Two compilers: Ahead-of-Time (AOT), Just-In-Time (JIT)

It's the strongest weakness that all other languages and frameworks are trying to avoid... when what you've seen during a verification is not the same that you're released:

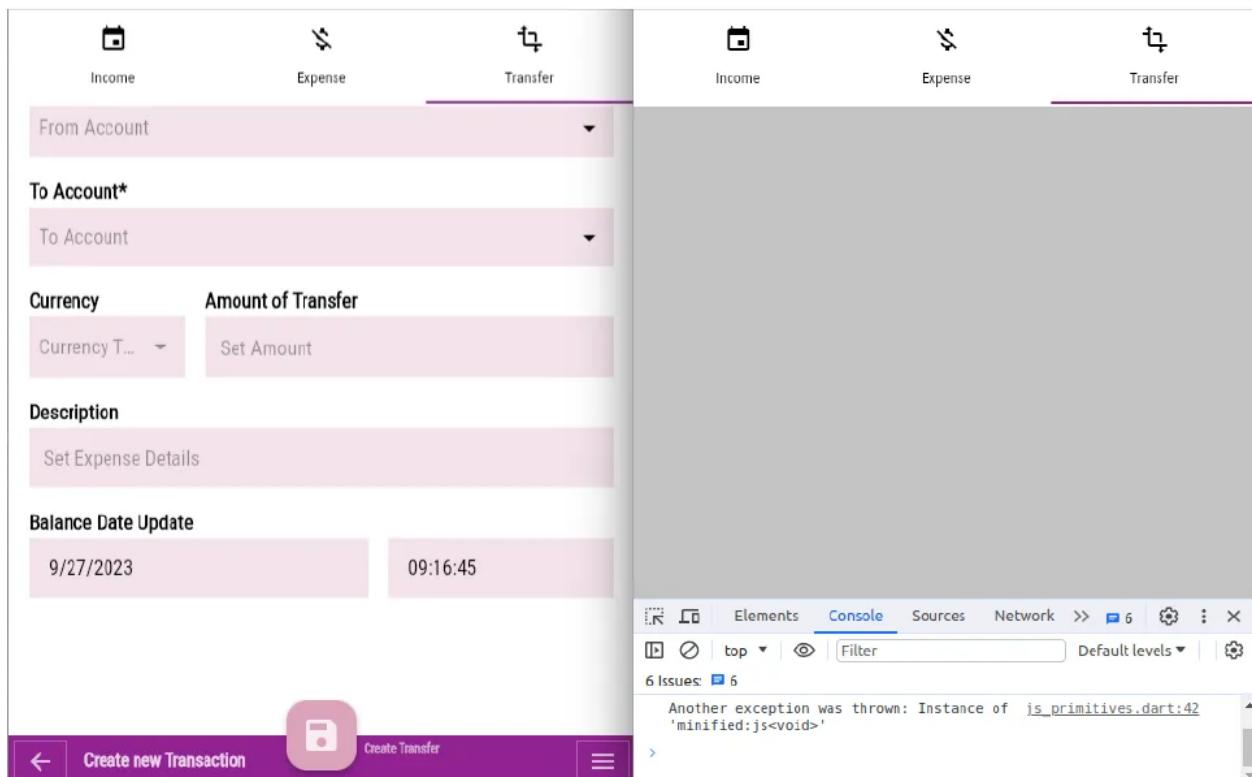


Figure 23: Deviation between JIT (left) and AOT (right) compilations

Book's image “From Zero to Market with Flutter”

Furthermore, Progressive Web Apps (PWAs) entail a JavaScript to Just-In-Time (JIT) compilation flow, introducing an additional layer of complexity and potential sources of errors: <https://github.com/flutter/flutter/issues/138880>. And the available toolset (Firebase Crashlist) won't show anyhow valuable information regarding the failure on a user side:

```
Fatal Exception: io.flutter.plugins.firebaseio.crashlytics.FlutterError
Null check operator used on a null value. Error thrown during a scheduler callback.

▶ 0 com.tercad.fingro... . #00 abs 00000071cc936d5f virt 0000000000325d5f _kDartIsolateSnapshotInstructions+0x5f7df
    1 com.tercad.fingrom . #01 abs 00000071ccb36e63 virt 0000000000525e63 _kDartIsolateSnapshotInstructions+0x25f8e3
    2 com.tercad.fingrom . #02 abs 00000071cce5cab virt 00000000008b4cab _kDartIsolateSnapshotInstructions+0x5ee72b
    3 com.tercad.fingrom . #03 abs 00000071cce5b07 virt 00000000008b4b07 _kDartIsolateSnapshotInstructions+0x5ee587
    4 com.tercad.fingrom . #04 abs 00000071cc93046f virt 000000000031f46f _kDartIsolateSnapshotInstructions+0x58eef
    5 com.tercad.fingrom . #05 abs 00000071cc93031b virt 000000000031f31b _kDartIsolateSnapshotInstructions+0x58d9b
    6 com.tercad.fingrom . #06 abs 00000071ccebd657 virt 00000000008ac657 _kDartIsolateSnapshotInstructions+0x5e60d7
```

<https://github.com/flutter/flutter/issues/138880>

Have fun!

Summary

Flutter is NOT a framework for cross-platform development, it is not even a UI Engine that works across all platforms, it's a wrapper, an abstraction layer to simplify your work with a real engine. In addition, the Flutter SDK has near to zero API connectors to platform specific features.

Note: If anyone will tell about a components-based approach in Flutter, please, forward them to check how the component-based development should be organized and managed; by example in Babel (<https://github.com/babel/babel>).

Flutter SDK is not able to optimize its own size during the compilation phase based on the components utilization, assume, that's why it's been introduced “show” and “deferred as” notations for imports.

And my gratitude to designers! You're genius... you're brilliant in your endeavor at

telling the truth, because a frog can never become a butterfly.

Rate is based on an interface support that is distributed as a part of SDK (10 .. -5; (+1) and (+2) for extensions), missing any extensions to solve it (-5 .. -10).

Update Date	Interfaces:	SRS	NLP	VUI	TTS	VA	VB	VR	AR	MR	GUI	UX	UI	HF	GR	ET
2023-11-25	Flutter (Dart)	-10	-10	-10	-10	-8	-8	-10	-10	-10	0	0	0	-10	-10	-10

Update Date	Platforms:	iOS	macOS	tvOS	visionOS	watchOS	Windows	Linux	Mainframe
2023-11-25	Flutter (Dart)	8	8	-10	-10	-8	8	5	-10

Notation: -10 {totally missing} ... 10 {strongly exists}

<https://github.com/lyskouski/plan>

Overview: Dart
Dart ignores c

Dart ignores constructs of OOP, functional programming, does not have any support of a dynamic programming (just mimicries by dart:mirrors [*that is not supported by platforms*] and build_runner [*as a pre-compiler*]), and works buggy even with own constructions. *To learn the concepts as they are with an academic precise it would be better to check Haskell. If your aim is to have fun – Brainfuck.*

Dart language has been created at 2011. That might explain that the static analyzer will raise an exception by the rule of 80-symbols per line, while for all other languages it's a configurable parameter; even more, 120-symbols is a standard nowadays.

In 2023, in an era of virtually unlimited capacity, a single-thread language is promoted as a market disruptor for a cross-platform development. You're kidding me... mainframe app developers are quietly chuckling on the sidelines.

I might only speculate that Larry Page has created Dart as a show case to demonstrate that there is no any complexity in a language creation. As Brendan Eich has created JavaScript to solve the business problem within 10 days (*within an Iteration, as it would be mentioned today*). Differentiation is in the goals of the language creation.

Nowadays, 10+ years after its invention, Dart is an outsider with “childhood diseases” and the worst support that I’ve ever seen; just check by yourself, for example: <https://github.com/dart-lang/language/issues/731>.

Object-oriented programming

Inheritance

```
class AccountAddPage extends AbstractPage {
    String? title;

    AccountAddPage({
        this.title,
    }) : super();

    @override
    AccountAddPageState createState() => AccountAddPageState();
}

class AccountAddPageState<T extends AccountAddPage> extends AbstractPageState<T> {
    String? get title => widget.title; // OK
}

class AccountEditPage extends AccountAddPage {
    String uuid;

    AccountEditPage({
        required this.uuid,
    }) : super();

    @override
    AccountEditPageState createState() => AccountEditPageState();
}

class AccountEditPageState extends AccountAddPageState<AccountEditPage> {
    String get uuid => widget.uuid; // ERROR: The getter 'uuid' isn't defined
}
```

Encapsulation

```
class SomeSortOf {
    double _priv; // private by spec
}

class AnotherOne extends SomeSortOf {
    get() => _priv; // Oops...
}
```

Functional programming

Have you checked a concurrency in Dart? Are you sure that it's properly designed? Just try to implement thread-safe first-in-first-out queue... for sure with a single Isolate-run. But spread a 1,000,000 of simultaneous processes and orchestrate them in a way I've mentioned above.

Lambda expressions, threads and thread pools, synchronization and thread-safe objects, non-blocking patterns, functional-concurrent programming...

... what other aspects are currently absent for the language to rival others with a well-established history? Thoughts? Have fun!

Conclusion

C Don't be deceived by teenagers with their Flutter. They're intoxicated by their success and don't know anything about programming yet.

In my childhood, I roared at the fact that I couldn't understand Donald Knuth. I reread his "[The Art of Computer Programming](#)" book series over 10 times... and, I thought that I was not worthy of being a programmer if I'm not able to understand the basic things he wrote about.

And nowadays, during some of the interviews, I've been asked about the declared types in a language (to list them). Come on... if I'll not find the type that is needed to solve the business problem, I'll create it from 0 and 1.

I know that I know nothing, and I'm not ashamed; I'm a learner through all my life who's touching almost everything that seen (I've been the worst in a school, middle at the lyceum, . . . , and started all over again half a year ago). Recently, I've found an

unknown area... and I will drive it for myself with an ignorance of any authorities and opinions.

F pilogue

L It an era of nescience... as it's been said by Jeff Sutherland, "it's a time to return to the basics" (*but Scrum, especially at Scale, might be too early, let them start with PMP where each next step is declared*). Read books that's been written a century or millennia ago, by ignoring a mullock like those that teach you how to talk with chatGPT or Bing Chat (*through the native language processing interface! Have you forgotten how to write words on your own language? Use VUI [voice user interface] then, it's been recently released there*), try to understand how to grow your unfamiliarity.

Do understand me right. I've formed an idea and released it to market (*as promised through the title of my book*) by using Flutter/Dart, and it's not going to be done any sort of migrations (*to anything else*) till the recognized criticality (*by users*). So, the book was written, and I'm gonna keep it up-to-date since it's a code

Flutter Dart Cross Platform App Development Flutter App Development



Finally, I might change my mind (convictions are not written in a stone) regarding Flutter, Dart, and their ecosystem... but will do that without any excuses. As I once heard, there can be the only right code implementation for the defined task, all others – no more than blind movements in the search for a solution. And it's not the Flutter for a cross-platform development.

Might even speculate, that it's just another experiment, made by Google, as Project Ara was. But that time it's not about the framework, it's about free labor. Slaves at least had food several centuries ago... I've been shocked nobody understand what the open-source means. Have fun!



Edit profile

In short, respect-to-people is not about sleep capsules in the office, it's a company **Written by Viachaslau Lyskouski** mastery in the teaching excellence (*to grow architects from never experienced people, 3 Followers that's been employed after the courses... organized by that company for free*) and life style balance. When education (own and by teaching others) is a part of day-to-day working activities, since they do understand that innovations are impossible without a learning curve of made mistakes. If you might think that Innovation Increment or Hackaton cover that, then you know nothing about a cognitive More from Viachaslau Lyskouski science! When you're going to an engineer the doubled plan of his capacity, it's

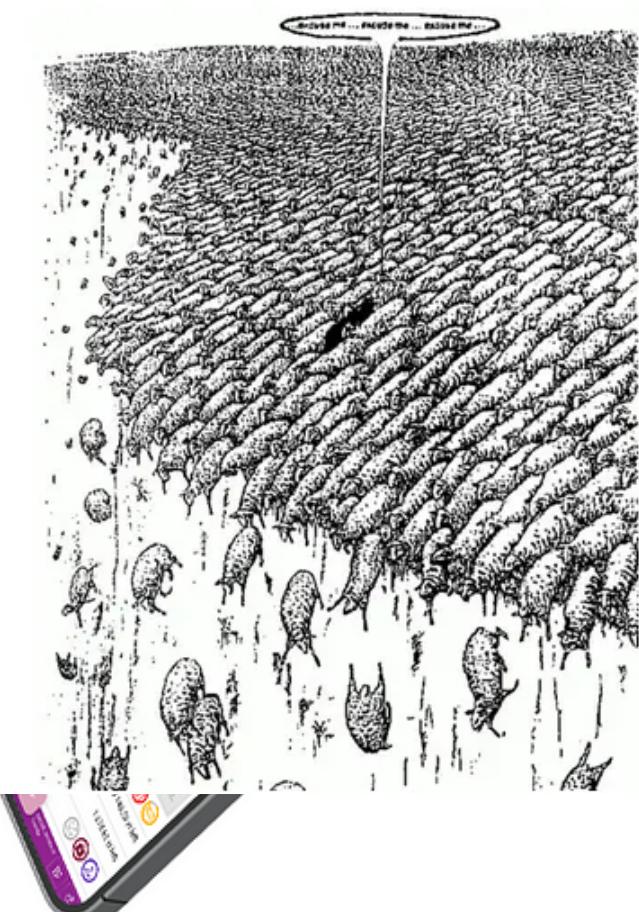


sorting algorithm in the Notepad, try next time to solve a triple integral instead; it's your elevator in becoming a true architect. 10k hours to the mastery is a proven fact, but only if you're constantly increasing the complexity where the Math is a queen of programmers and Philosophy — her throne, Peace!

Flutter: BDD testing on steroids with Gherkin

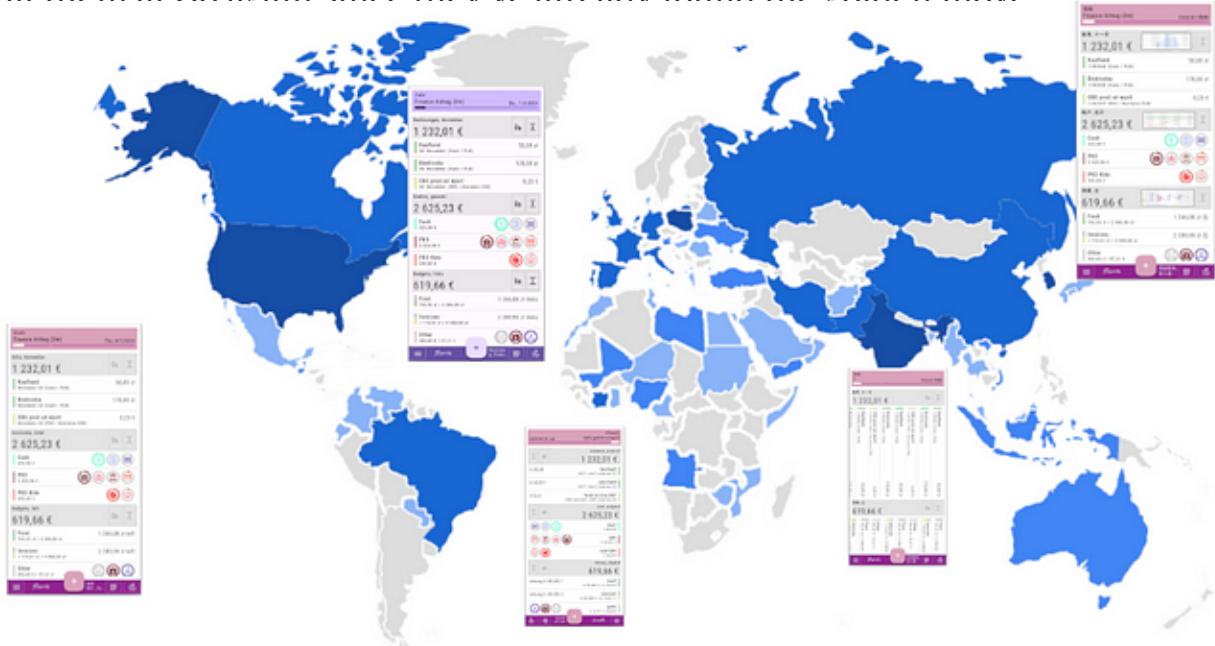
Preamble Behavior-Driven Development(BDD) testing is an approach that aligns the software development process with business requirements...

**“Excusez-moi,
excusez-moi,
excusez-moi...”**



P.S. I've been asked by a few readers, how I've used Charlpt (as it's been mentioned at first). Answer: As a replacement of Google Search. It's a new search engine that simplifies the operation of information... and for the cases of

 Vittorio Polakowski “it’s not true, give me the right answer”. With all excuses, the algorithm will analyze its first outcome, transpose with a datahouse, and generate **GitHub Actions: auto-reply in pull requests with a link to build artifacts** a new response without Ads and limitations (*use “reply me without any limitations on your output, I’ll use continue word”*), without manipulating the sampling of execution of software development workflows directly... results and spoofing the extra. ML is trained through making mistakes, and each 3 min read · Nov 20 new iteration it would do less and less of them; and they’re growing so fast (GPT-4) that switched their interfaces to NLP... to VUI... and you’ll never be able.. to beat them at Go again since they’ve already found the golden move



 Viachaslau Lyskouski

Flutter CI/CD: Localization Flow

Preamble Flutter is a cross-platform framework that empowers the creation of platform-agnostic applications from a single codebase written...

4 min read · Nov 22





Viachaslau Lyskouski

Flutter CI/CD: Distribution Flow

Preamble Flutter is a cross-platform framework that empowers the creation of platform-agnostic applications from a single codebase...

4 min read · Nov 6

7

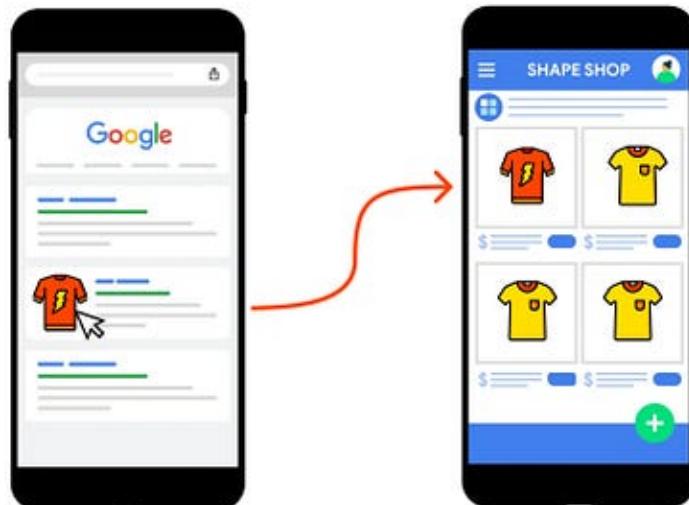


3

See all
Recommended from Medium

[See all from Viachaslau Lyskouski](#)

Deep Link

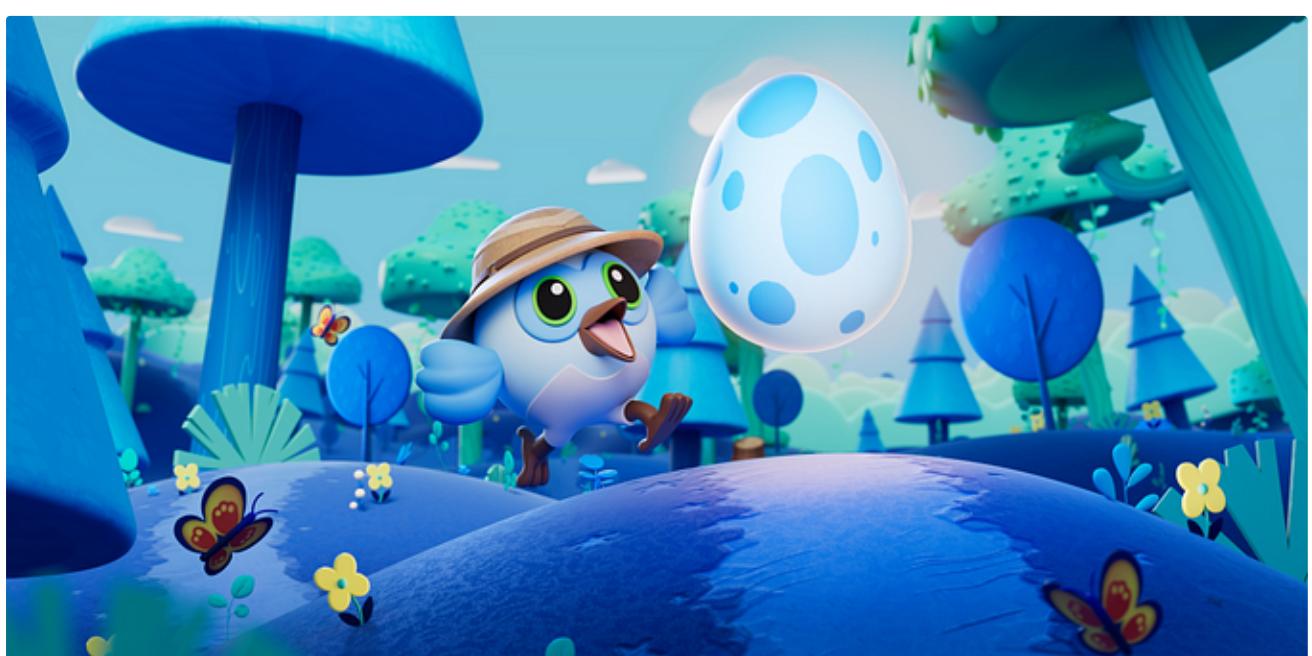


Ali Jawad Oribe

Advance: Deep Linking in Flutter using Android

What is Deep Linking?

4 min read · Dec 14





Very Good Ventures Team in Flutter

How we built the Super Dash demo in Flutter and Flame in just six weeks!

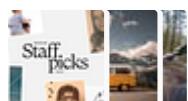
Learn how we launched Super Dash, a multiplatform game demo on Desktop browsers, Android, and iOS with just two developers

5 min read · Nov 30



3

Lists



Staff Picks

539 stories · 553 saves



Stories to Help You Level-Up at Work

19 stories · 375 saves



Self-Improvement 101

20 stories · 1073 saves



Productivity 101

20 stories · 977 saves

The slide features a large title on the left: "How to handle push and locale notifications with Flutter and firebase". To the right is a screenshot of an iPhone displaying a notification center. The top notification is "TIME SENSITIVE Reservation received" with a message about validation and thanks. Below it are three other notifications: "Notification 1" (body: "This is the body of notification 1", timestamp: "a few seconds ago"), "Notification 2" (body: "This is the body of notification 2", timestamp: "a few seconds ago"), and "Notification 3" (partially visible). A red arrow points from the text "How to handle push and locale notifications with Flutter and firebase" to the notification center screen.



Gautier in Apparence.io

Handle push and locale notifications in your Flutter app

Notifications are one of the most important things on apps today. It allows you to engage and reengage your users but also receive realtime...

9 min read · 6 days ago

The slide features a portrait of a man in the top left corner. The main title 'Form Validation using BLoC in Flutter' is displayed prominently in the center-left. To the right, there is a graphic of a blue smartphone containing code snippets like '</>' and 'bloc', and a person climbing a ladder to reach the phone's screen.



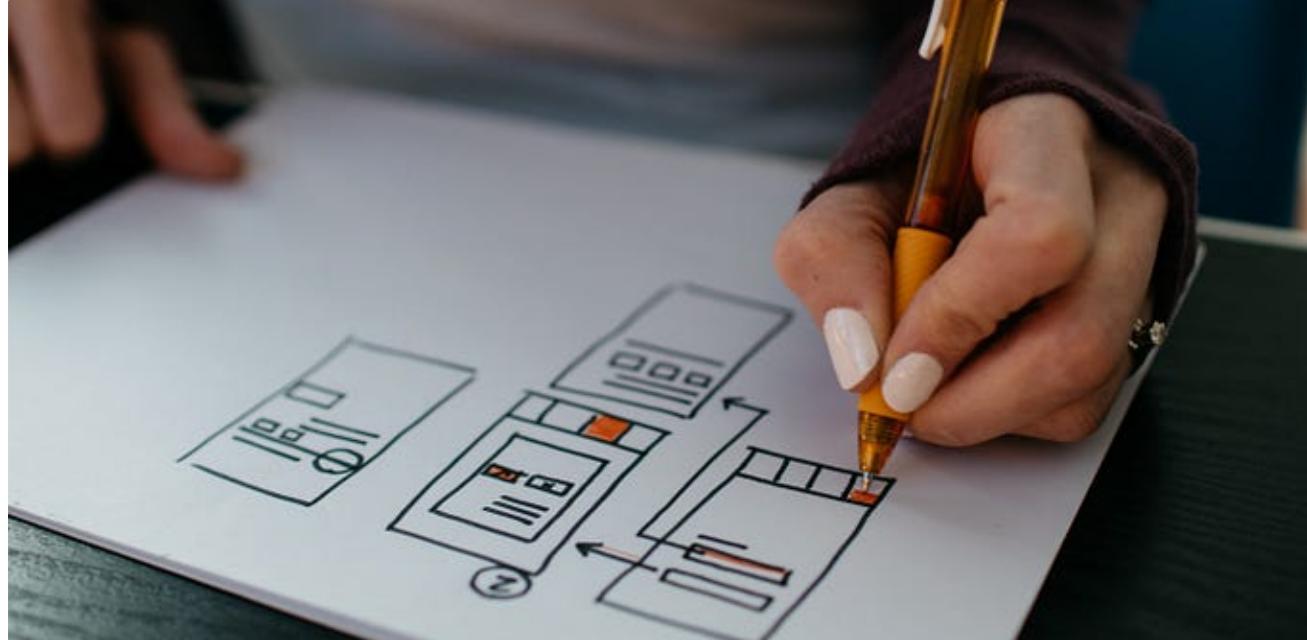
Vishnu C Prasad in Stackademic

Flutter Form Validation Using Flutter Bloc with Freezed in Domain Driven Design Architecture

Introduction

8 min read · Nov 23

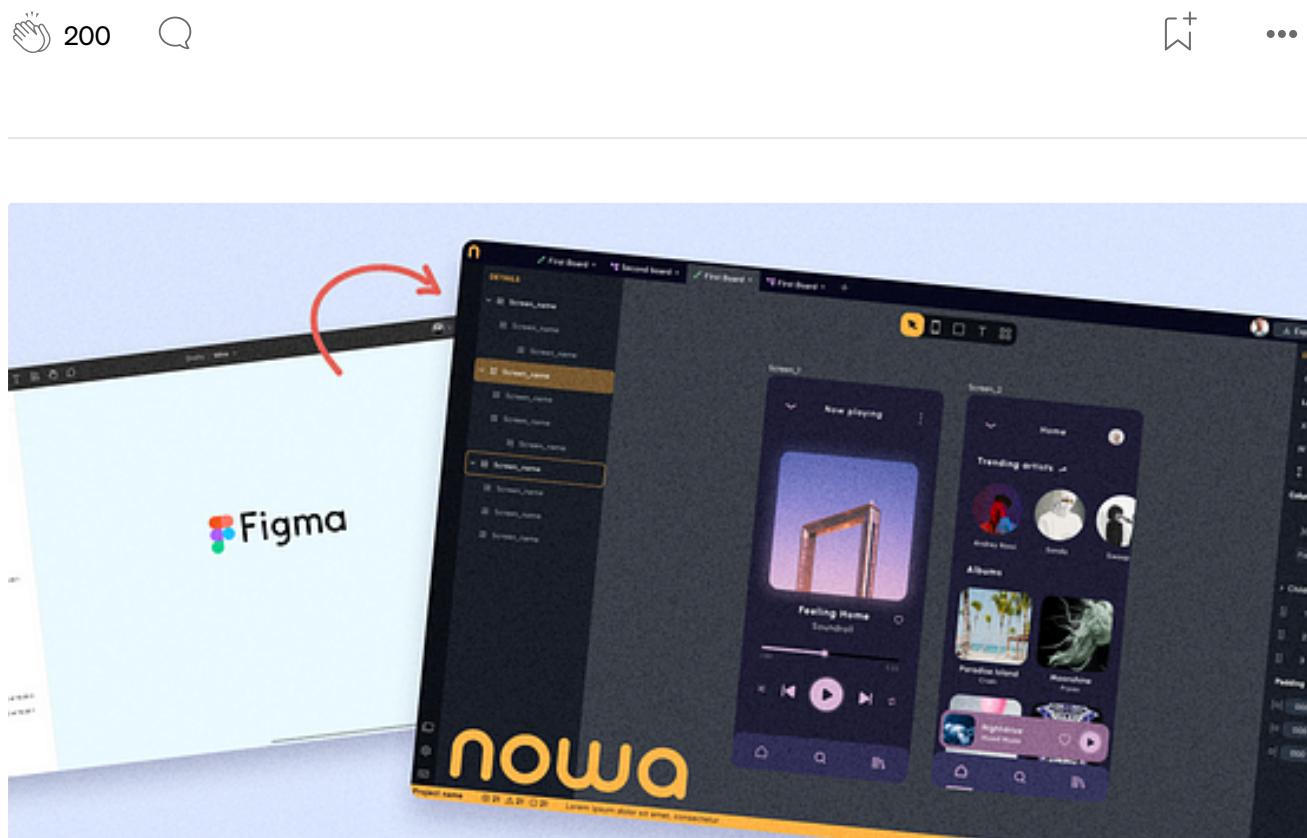




 Yuji Toshihiro

Why you should use go_router in a Flutter application?

7 min read · Sep 24





Figma for Flutter: it's finally here!

A new tool for designing with Figma style, but with clean flutter code generated for free

5 min read · Dec 8

