#KARSILZ

BPF.LI/S

KARSTEN SILZ

JANUARY 20, 2021

HOW SHOULD JAVA DEVELOPERS BUILD FRONT-ENDS FOR WEB, MOBILE, AND DESKTOP TODAY?

SLIDES & VIDEOS

GET STARTED

HOW TO BUILD JAVA APPS TODAY?

BPF.LI/PAT

YOU GET A NEW PRESIDENT DURING MY TALK!

BACK TO OUR REGULARLY SCHEDULED PROGRAM

I GIVE ADVICE. . .

. . . SO WHAT'S MY ANGLE?

JAVA DEVELOPER FOR 22 YEARS

MOSTLY BACK-END

CAN DESIGN FRONT-END MYSELF

CAN BE EARLY ADOPTER

I GIVE YOU OPTIONS FOR YOUR NEXT PROJECT!

APPLY YOUR OWN CRITERIA
(TECHNICAL, LEGAL, ETHICAL, . . .)

EVALUATE IN YOUR ENVIRONMENT

MAKE YOUR OWN CHOICES

WHAT YOU NEED TO DO

HOW CAN WE
BUILD FRONT-ENDS TODAY?

HOW SHOULD JAVA DEVELOPERS
BUILD FRONT-ENDS TODAY?

DATA ENTRY & RETRIEVAL

DRILLING DOWN INTO DATA DETAILS

NAVIGATION BETWEEN SCREENS

NO GAMES OR MEDIA APPLICATIONS

APPLICATION REQUIREMENTS

TO DO APP

EXPENSE FILING

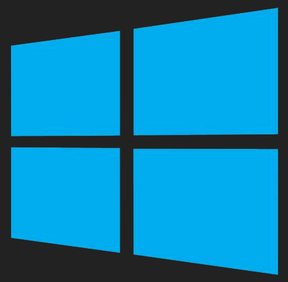
CHAT

SOCIAL NETWORK

EXAMPLE APPLICATIONS

APPLIES TO THESE DEVICES...

PC



MOBILE

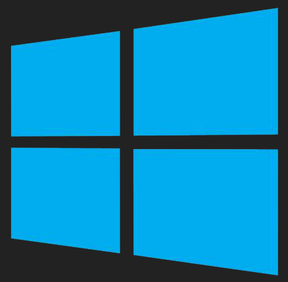
. . . BUT NOT TO THESE DEVICES!

WEARABLES

TV


SONY
8K 4K 120 HDR
PS5
PlayStation®5
CFI-1016A
7
www.pegi.info

EXPENSE FILING OR CHAT APPLICATION




SUMMARY: APPLICATIONS

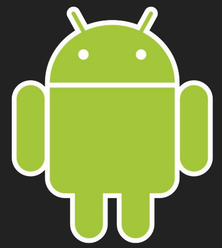
HOW CAN WE BUILD FRONT-ENDS TODAY?

HOW SHOULD JAVA DEVELOPERS BUILD FRONT-ENDS TODAY?

WE CAN GO SAFE

SAFE?

STILL HERE IN 5 YEARS – FOR SURE



WEB

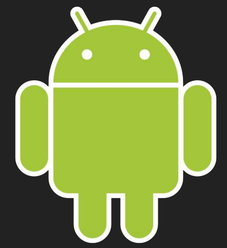
NATIVE

NATIVE

NATIVE

NATIVE

NATIVE



- ONE CODE BASE
- SLOW
- LESS FEATURES
- JAVASCRIPT
- BAD DEBUGGING

WEB

- 5 CODE BASES
- FAST
- ALL FEATURES
- 5 LANGUAGES
- GOOD DEBUGGING

NATIVE

- ONE CODE BASE
- FAST
- ALL FEATURES
- GOOD DEBUGGING

CROSS-PLATFORM

THIS TALK: NO PURE NATIVE FRAMEWORKS!

ANGULAR (GOOGLE)

REACT (FACEBOOK)

VUE.JS (PERSON: EVAN YOU)

WEB FRAMEWORKS

FLUTTER (GOOGLE)

JAVAFX (GLUON)

REACT NATIVE (FACEBOOK)

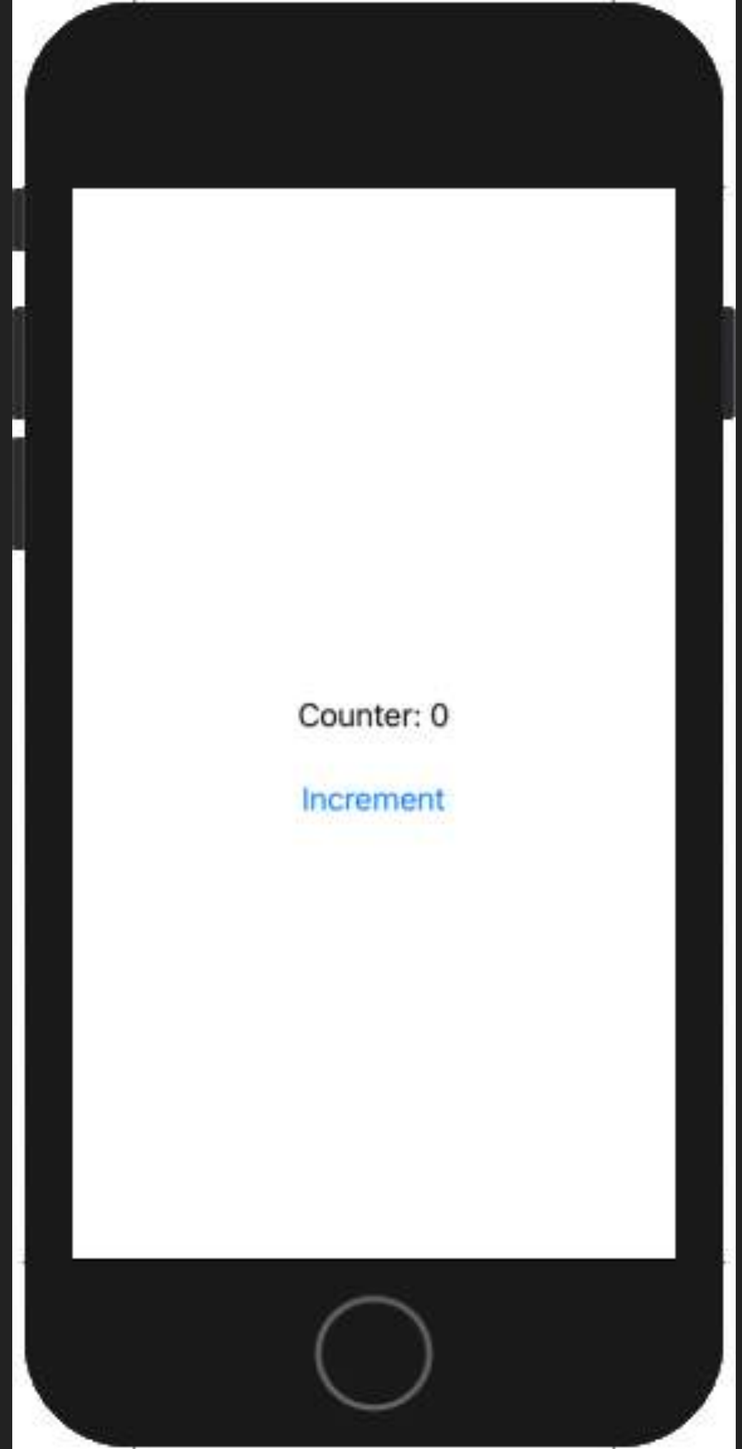
XAMARIN (MICROSOFT)

X-PLATFORM FRAMEWORKS

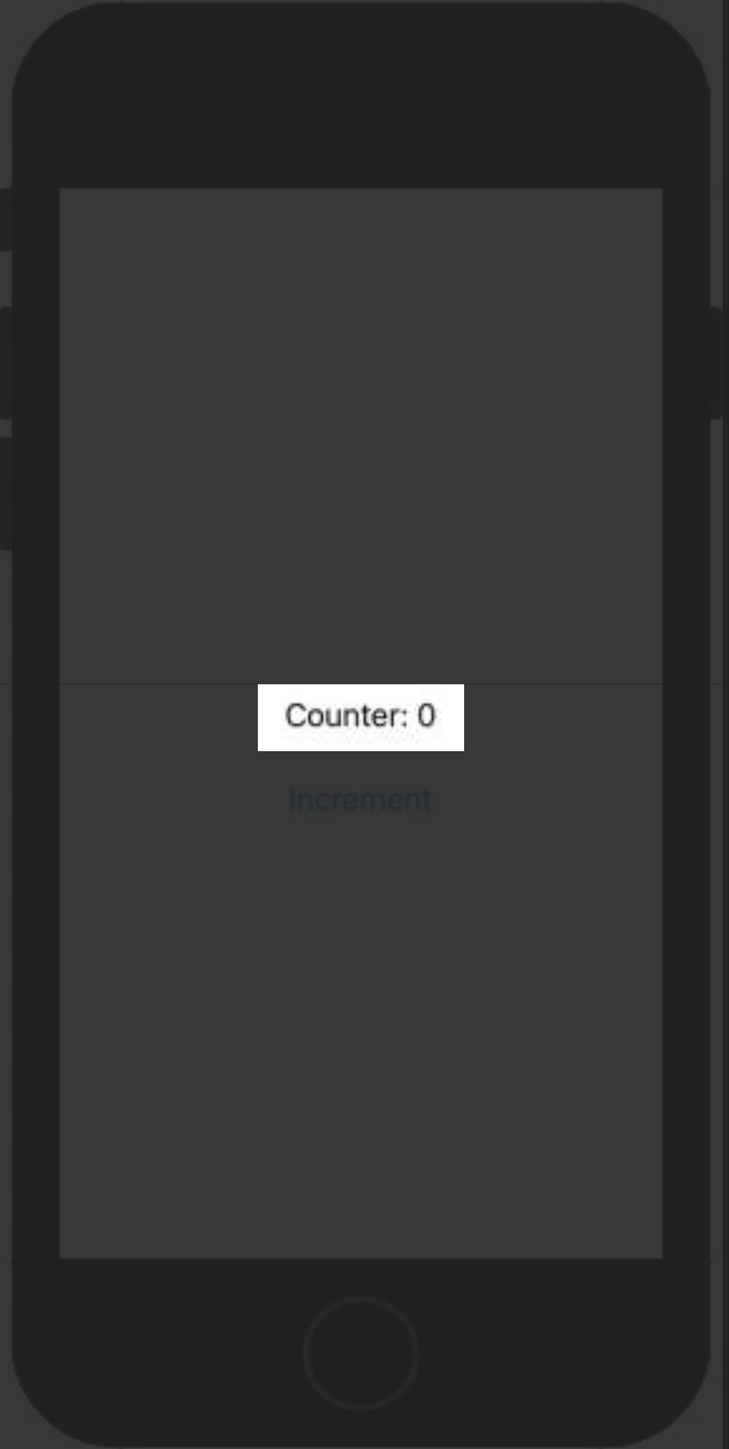
STATE OF THE ART:
DECLARATIVE UIS

EXAMPLE:
APPLE SWIFTUI

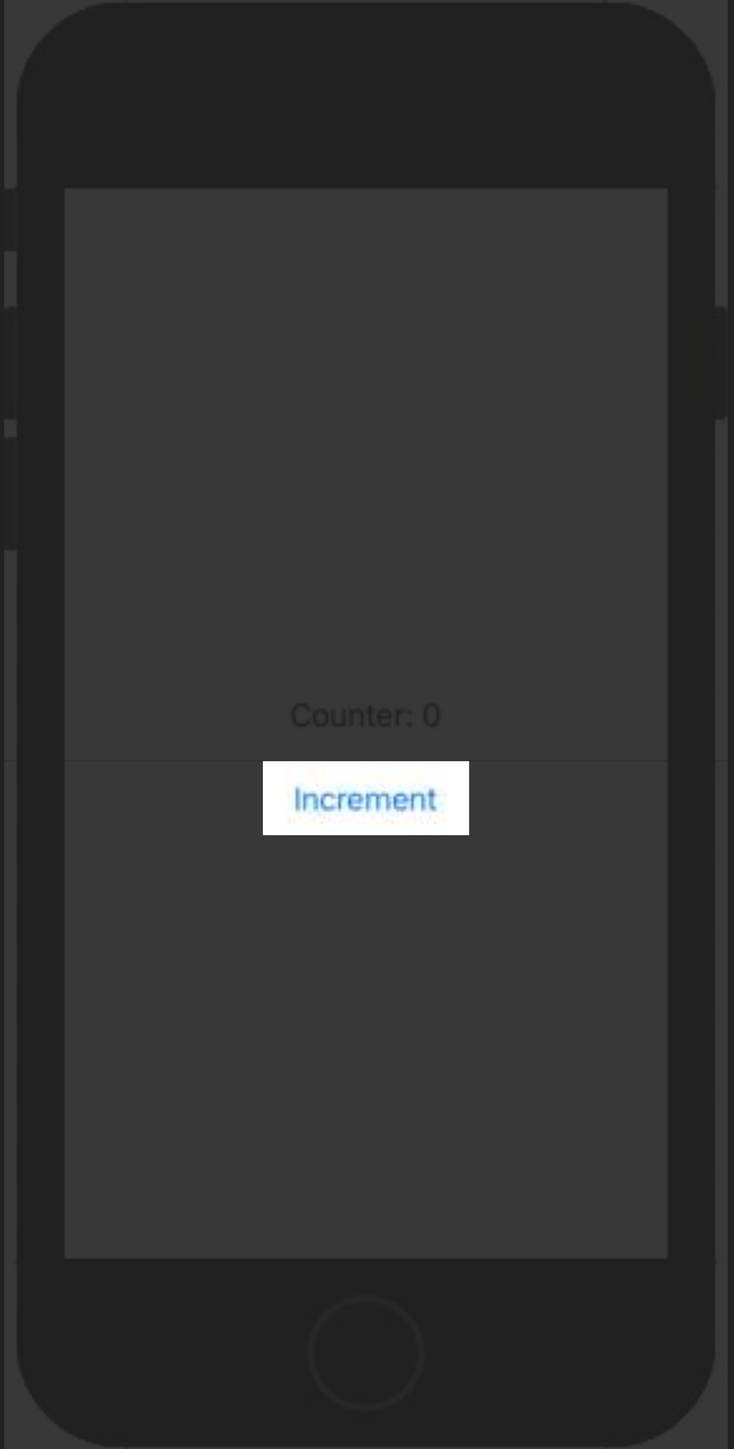
```
@State var count: Int = 0

var body: some
View {
  VStack(alignment: .center,
    content: {
      Text("Counter: \(count)")
        .padding()
      Button(action: {
        self.count += 1
      }, label: {
        Text("Increment")
      })
  })
}
```



```
@State var count: Int = 0

var body: some
View {
  VStack(alignment: .center,
    content: {
      Text("Counter: \(count)")
        .padding()
      Button(action: {
        self.count += 1
      }, label: {
        Text("Increment")
      })
  })
}
```

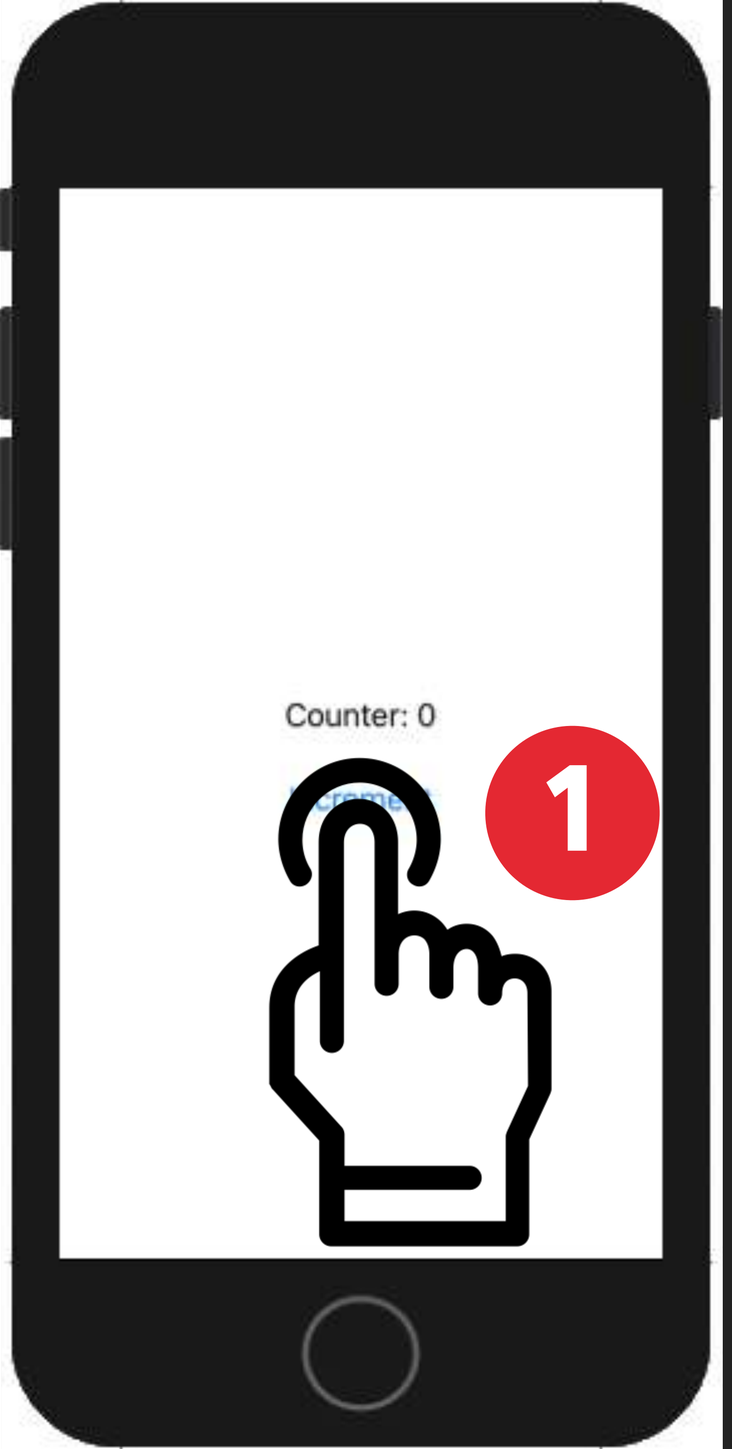


```
@State var count: Int = 0

var body: some
View {
  VStack(alignment: .center,
    content: {
      Text("Counter: \(count)")
        .padding()
      Button(action: {
        self.count += 1
      }, label: {
        Text("Increment")
      })
  })
}
```



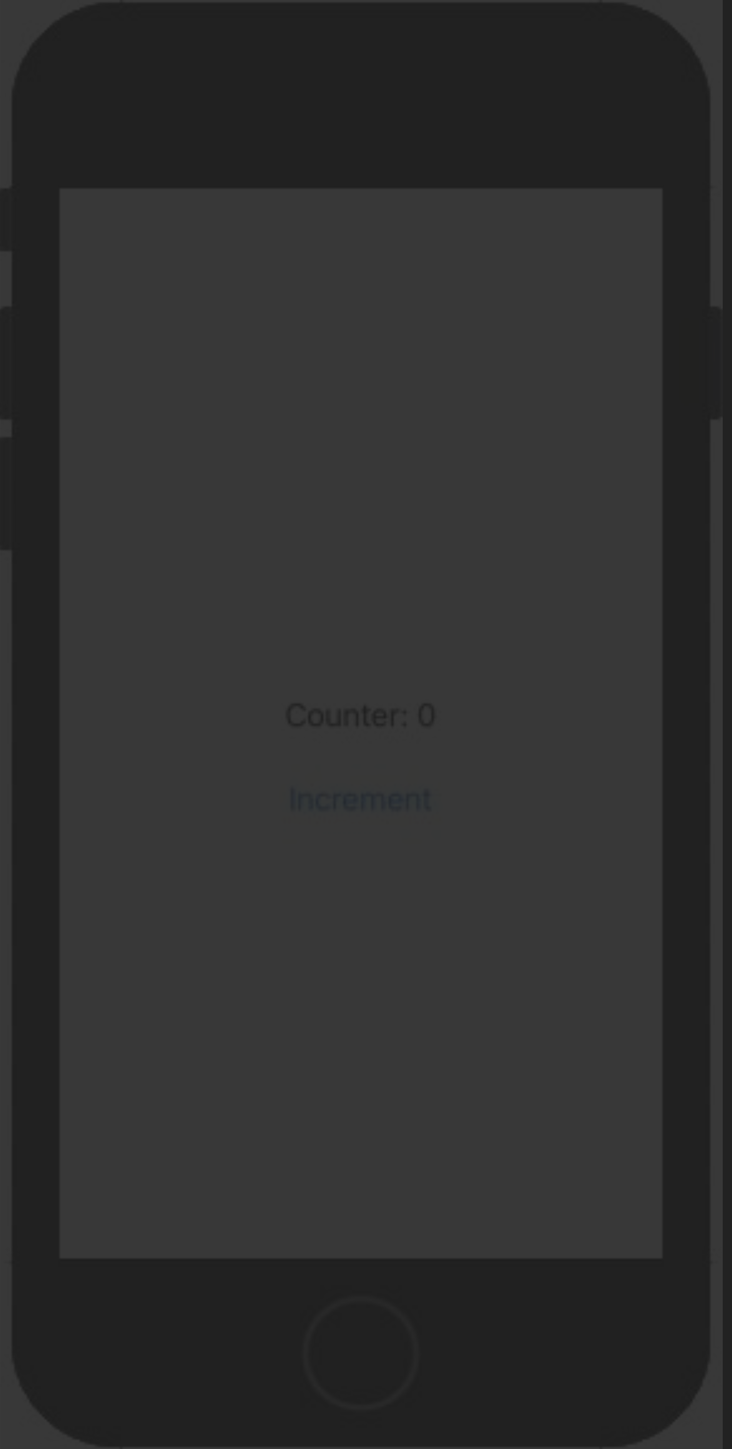
HOW DOES IT WORK?

```
@State var count: Int = 0

var body: some
View {
  VStack(alignment: .center,
    content: {
      Text("Counter: \(count)")
        .padding()
      Button(action: {
        self.count += 1
      }, label: {
        Text("Increment")
      })
  })
}
```



```
Button(action: {
    self.count += 1
```

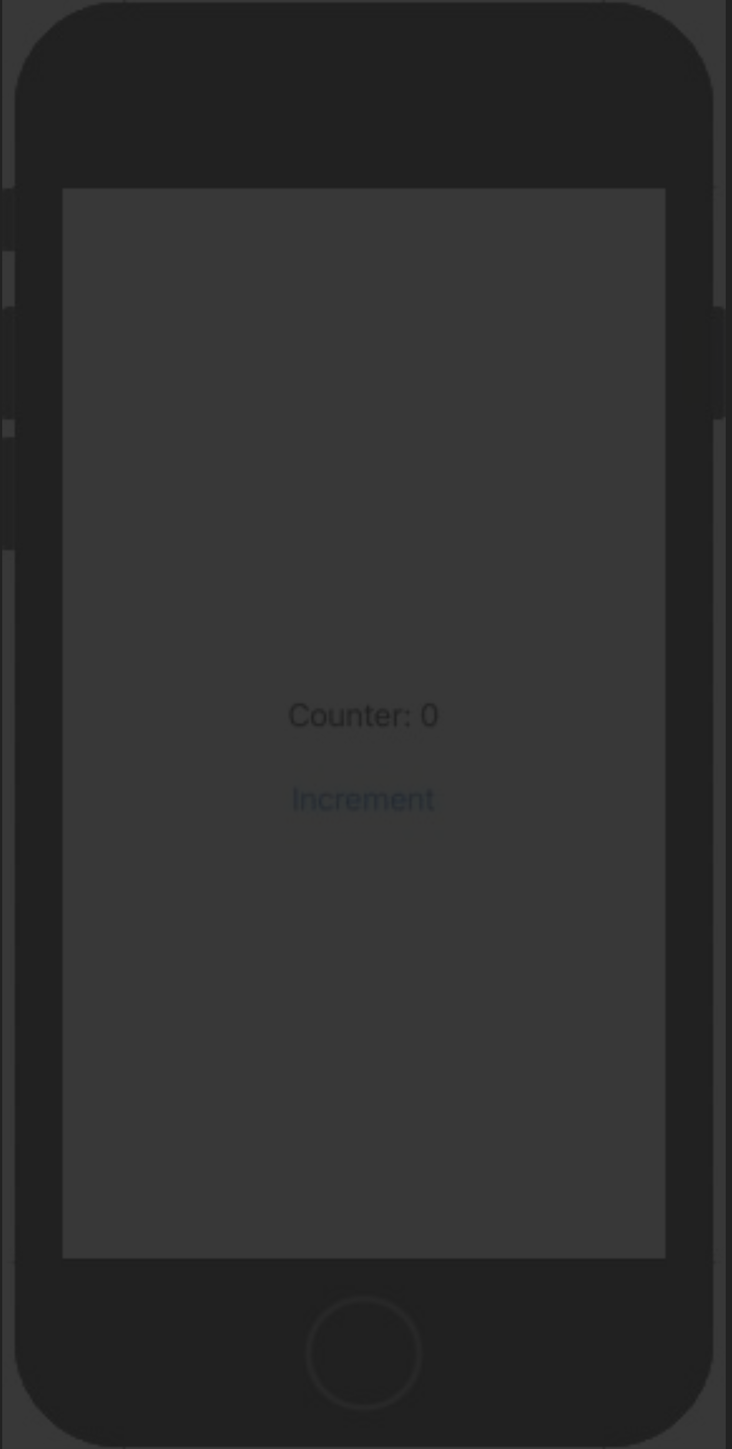
2



```
@State var count: Int = 0
```

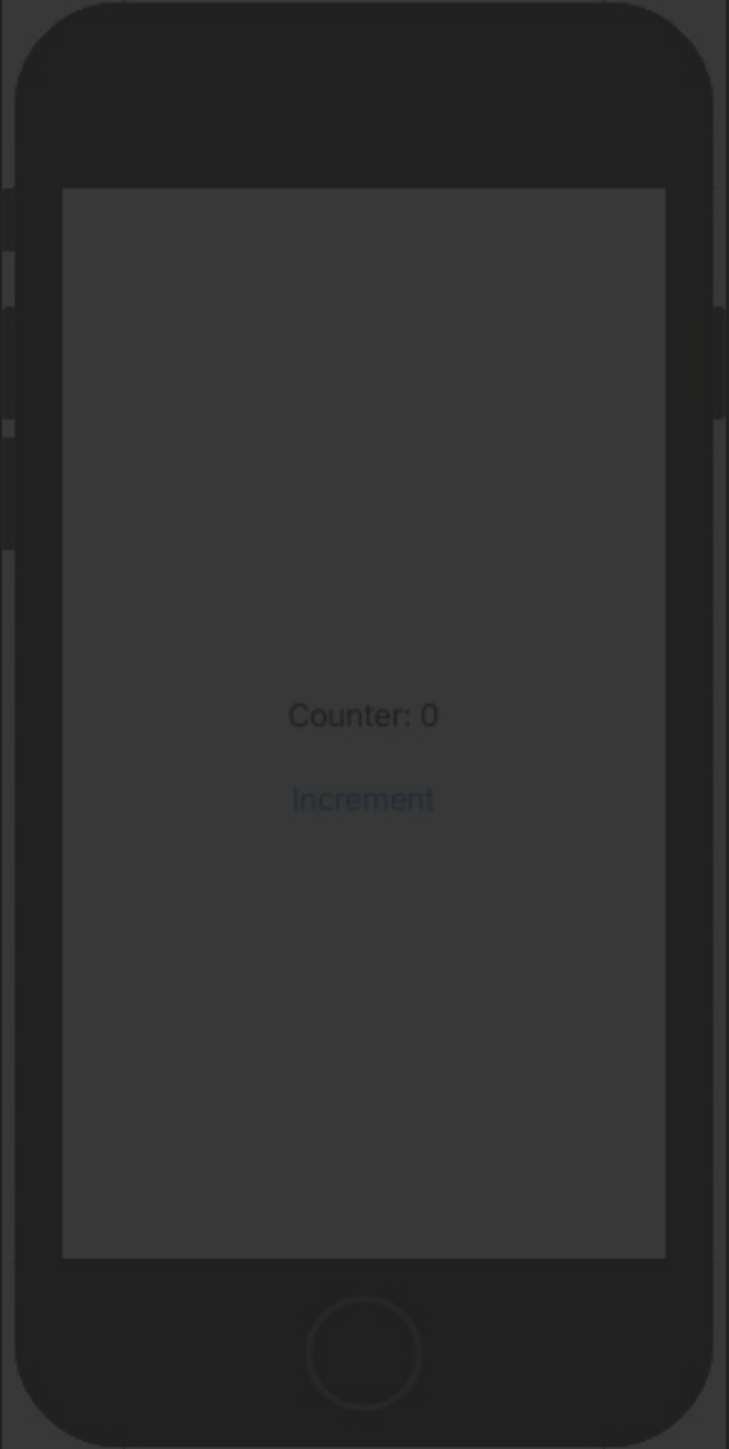
3

```
var body: some
View {
  VStack(alignment: .center,
    content: {
      Text("Counter: \(count)")
        .padding()
      Button(action: {
        self.count += 1
      }, label: {
        Text("Increment")
      })
  })
}
```



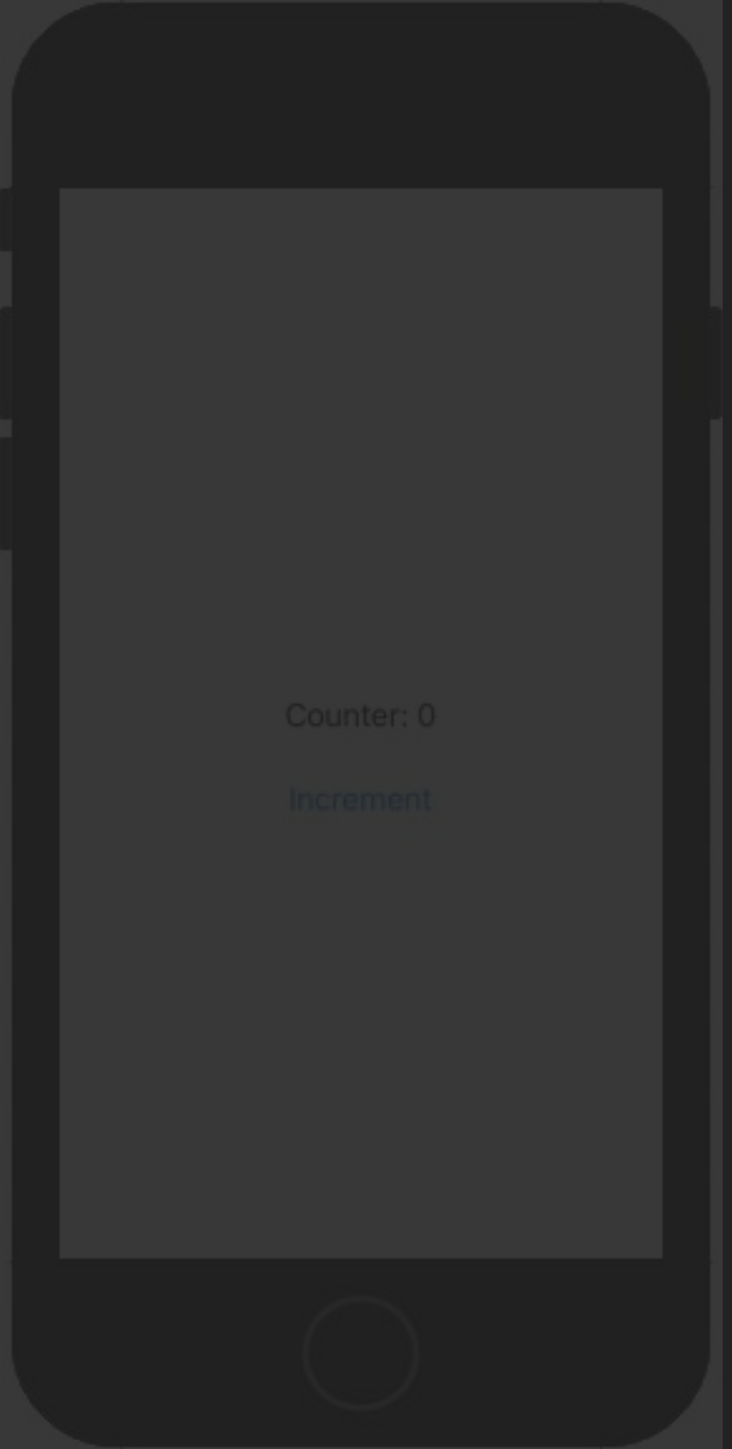
```
@State var count: Int = 0

var body: some View {
  VStack(alignment: .center,
    content: {
      Text("Counter: \(count)")
        .padding()
      Button(action: {
        self.count += 1
      }, label: {
        Text("Increment")
      })
  })
}
```

4

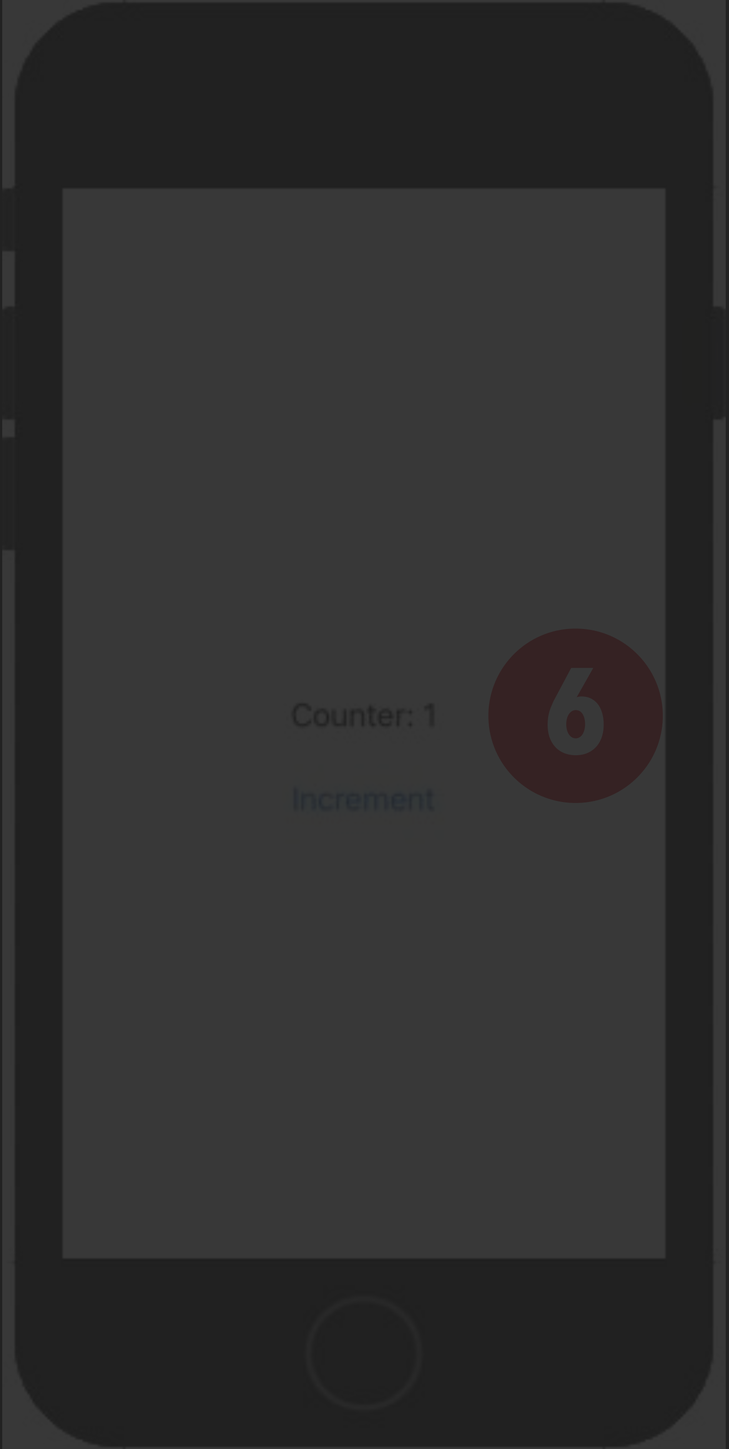


```
@State var count: Int = 0

var body: some
View {
  VStack(alignment: .center,
    content: {
      Text("Counter: \(count)")
        .padding()
      Button(action: {
        self.count += 1
      }, label: {
        Text("Increment")
      })
  })
}
```

5



```
@State var count: Int = 0

var body: some
View {
  VStack(alignment: .center,
    content: {
      Text("Counter: \(count)")
        .padding()
      Button(action: {
        self.count += 1
      }, label: {
        Text("Increment")
      })
  })
}
```



WHAT DID WE JUST SEE?

```
@State var count: Int = 0

var body: some
View {
  VStack(alignment: .center,
    content: {
      Text("Counter: \(count)")
        .padding()
      Button(action: {
        self.count += 1
      }, label: {
        Text("Increment")
      })
  })
}
```

FRONT-END AS CODE

```
@State var count: Int = 0

var body: some
View {
  VStack(alignment: .center,
    content: {
      Text("Counter: \(count)")
        .padding()
      Button(action: {
        self.count += 1
      }, label: {
        Text("Increment")
      })
  })
}
```

NEST ELEMENTS WITH BUILDER PATTERN

```
@State var count: Int = 0

var body: some
View
  VStack(alignment: .center,
    content: {
      Text("Counter: \(count)")
        .padding()
      Button(action: {
        self.count += 1
      }, label: {
        Text("Increment")
      })
  })
}
```

CHANGE STATE AFTER EVENTS, PLATFORM REDRAWS FRONT-END

FRONT-END AS CODE

NEST ELEMENTS WITH BUILDER PATTERN

CHANGES STATE AFTER EVENTS,
PLATFORM REDRAWS FRONT-END

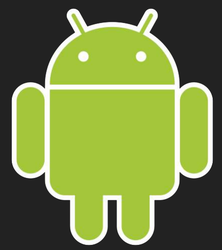
DECLARATIVE FRONT-ENDS

.NET MAUI (NOVEMBER 2021)

SWIFTUI

REACT

SWIFTUI



JETPACK COMPOSE (ALPHA)

WHAT IF CROSS-PLATFORM OPTION DOESN'T WORK OUT OR GOES AWAY?

THEN WE CAN'T TAKE OUR CODE WITH US...

...BUT OUR KNOWLEDGE STILL APPLIES!

WHY STATE OF THE ART?

WEB FRAMEWORKS: ANGULAR, REACT, VUE.JS

CROSS-PLATFORM FRAMEWORKS: FLUTTER, JAVAFX, REACT NATIVE, XAMARIN

STATE OF THE ART: DECLARATIVE FRONT-ENDS

SUMMARY: FRONT-ENDS

~~HOW CAN WE~~
~~BUILD FRONT-ENDS TODAY?~~

HOW SHOULD JAVA DEVELOPERS
BUILD FRONT-ENDS TODAY?

JAVA DEVELOPERS
WISH LIST

JAVA

WEB APPS

FAST

DECLARATIVE

MOBILE APPS

ALL NATIVE FEATURES

SAFE

DESKTOP APPS

ONE CODE BASE

POPULAR

COMPLETE

FAST BUILD, DE-PLOY & DEBUG

EASIER TO LEARN

EASIER TO BUILD, DEPLOY & DEBUG

EASIER TO HIRE

DIFFICULT TO MEASURE – INPUT WELCOME!

WHY POPULAR?

~~HOW CAN WE~~
~~BUILD FRONT-ENDS TODAY?~~

HOW SHOULD ~~JAVA DEVELOPERS~~
BUILD FRONT-ENDS TODAY?

WHAT FRONT-ENDS?

#1: JUST WEB: BUILD ONLY WEB APPLICATION, FROM SCRATCH

#2: NATIVE APPS: WEB APPLICATION EXISTS, BUILD NATIVE IOS & ANDROID APPS FROM SCRATCH

#3: DESKTOP APPS: WEB APPLICATION EXISTS

SCENARIOS

I ONLY WORKED WITH ANGULAR & FLUTTER

EVALUATIONS OF THE OTHER FRAMEWORKS BASED ON MY RESEARCH, NOT PRACTICAL EXPERIENCES

DISCLAIMER

~~HOW CAN WE~~
~~BUILD FRONT-ENDS TODAY?~~

HOW SHOULD ~~JAVA DEVELOPERS~~
BUILD ~~FRONT-ENDS~~ TODAY?

ANGULAR (GOOGLE)

REACT (FACEBOOK)

VUE.JS (PERSON: EVAN YOU)

#1: JUST WEB

POPULAR:
GOOGLE TRENDS

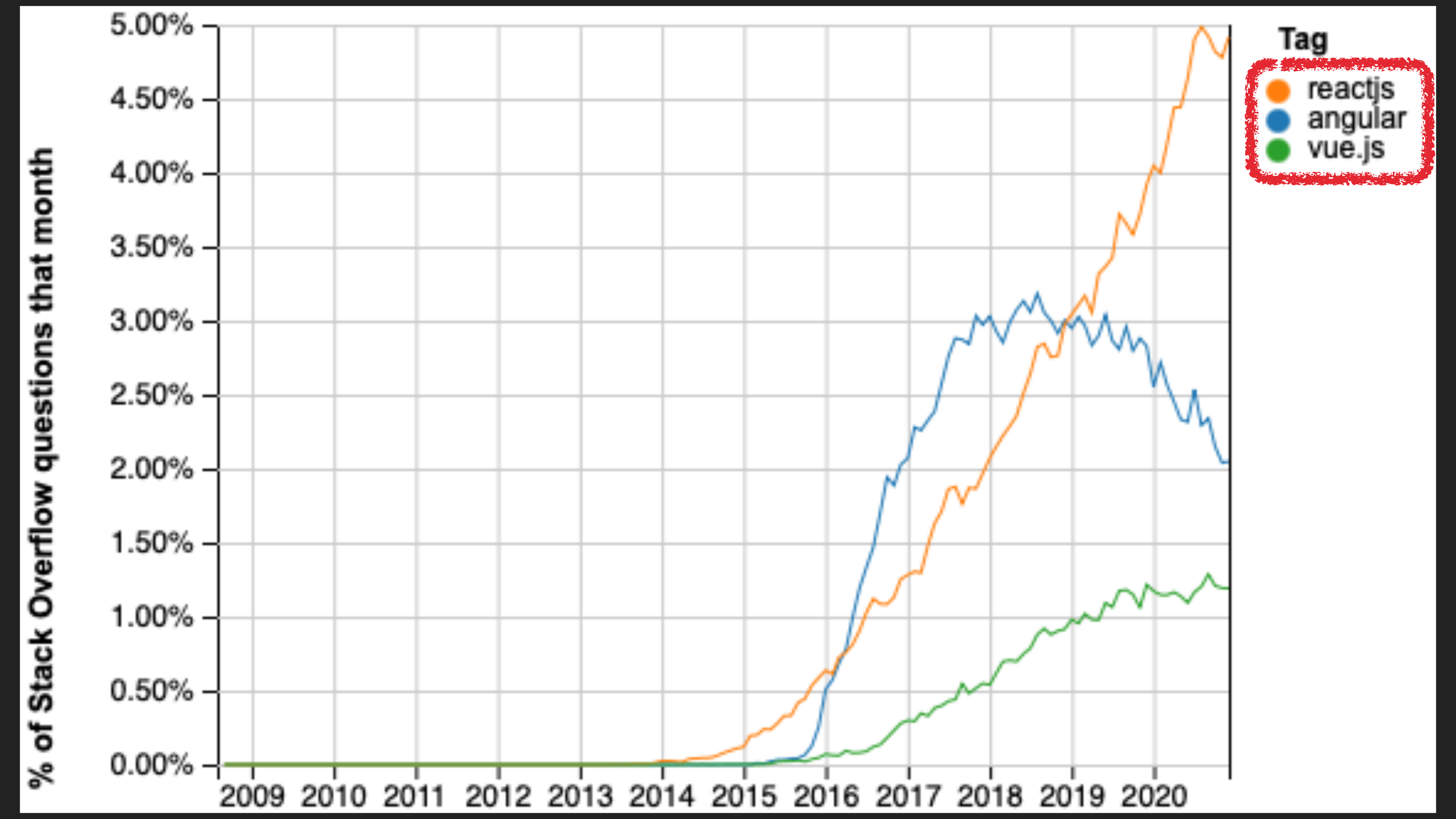
angular | react | Vue.js | + Add comparison

Worldwide | Past 5 years | All categories | Web Search

Note: This comparison contains both Search terms and Topics, which are measured differently. LEARN MORE

Interest over time

POPULAR:
STACK OVERFLOW


% of Stack Overflow questions that month
5.00%
4.50%
4.00%
3.50%
3.00%
2.50%
2.00%
1.50%
1.00%
0.50%
0.00%
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
Tag
reactjs
angular
vue.js

REACT MOST POPULAR, GROWING/FLAT

ANGULAR SECOND, SHRINKING

VUE.JS LAST, GROWING/FLAT

POPULARITY: JUST WEB

WISHLIST SCORE

JAVA

WEB APPS

FAST

DECLARATIVE

MOBILE APPS

ALL NATIVE FEATURES

SAFE

DESKTOP APPS

ONE CODE BASE

POPULAR

COMPLETE

FAST BUILD, DE-PLOY & DEBUG

JAVA ANG

FAST REA

DECLARATIVE REA

SAFE REA

POPULAR REA

COMPLETE ANG

FAST BUILD, DE-
PLOY & DEBUG

???

REACT WINS

ANGULAR SECOND

VUE.JS LAST

SCORE: JUST WEB

IF YOU ALREADY USE REACT, ANGULAR,
OR VUE.JS, THEN KEEP IT

IF YOU DON'T, THEN GO FOR REACT
FIRST, THEN ANGULAR, THEN VUE.JS

ADVICE: #1: JUST WEB

FLUTTER (GOOGLE)

JAVAFX (GLUON)

REACT NATIVE (FACEBOOK)

XAMARIN (MICROSOFT)

#2: NATIVE APPS

POPULAR:
GOOGLE TRENDS

flutter
Search term

react native
Search term

xamarin
Search term

javafx
Search term

+

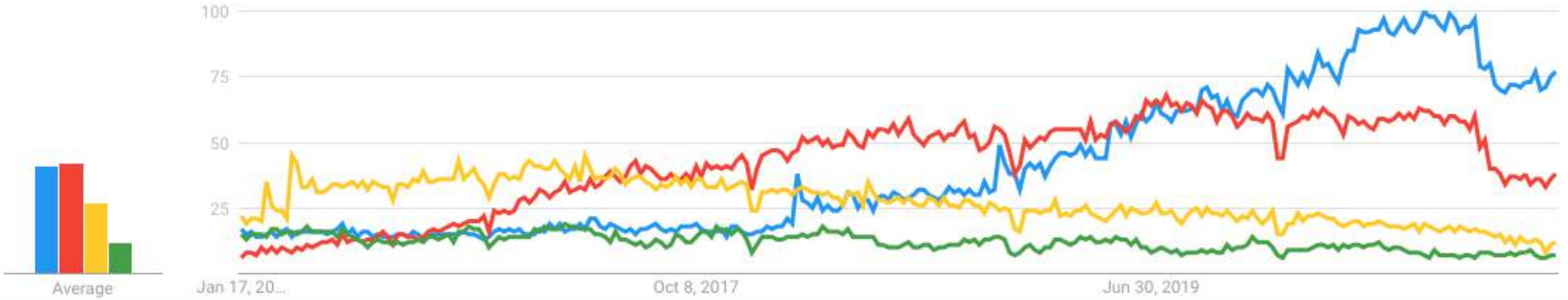
Worldwide

Past 5 years

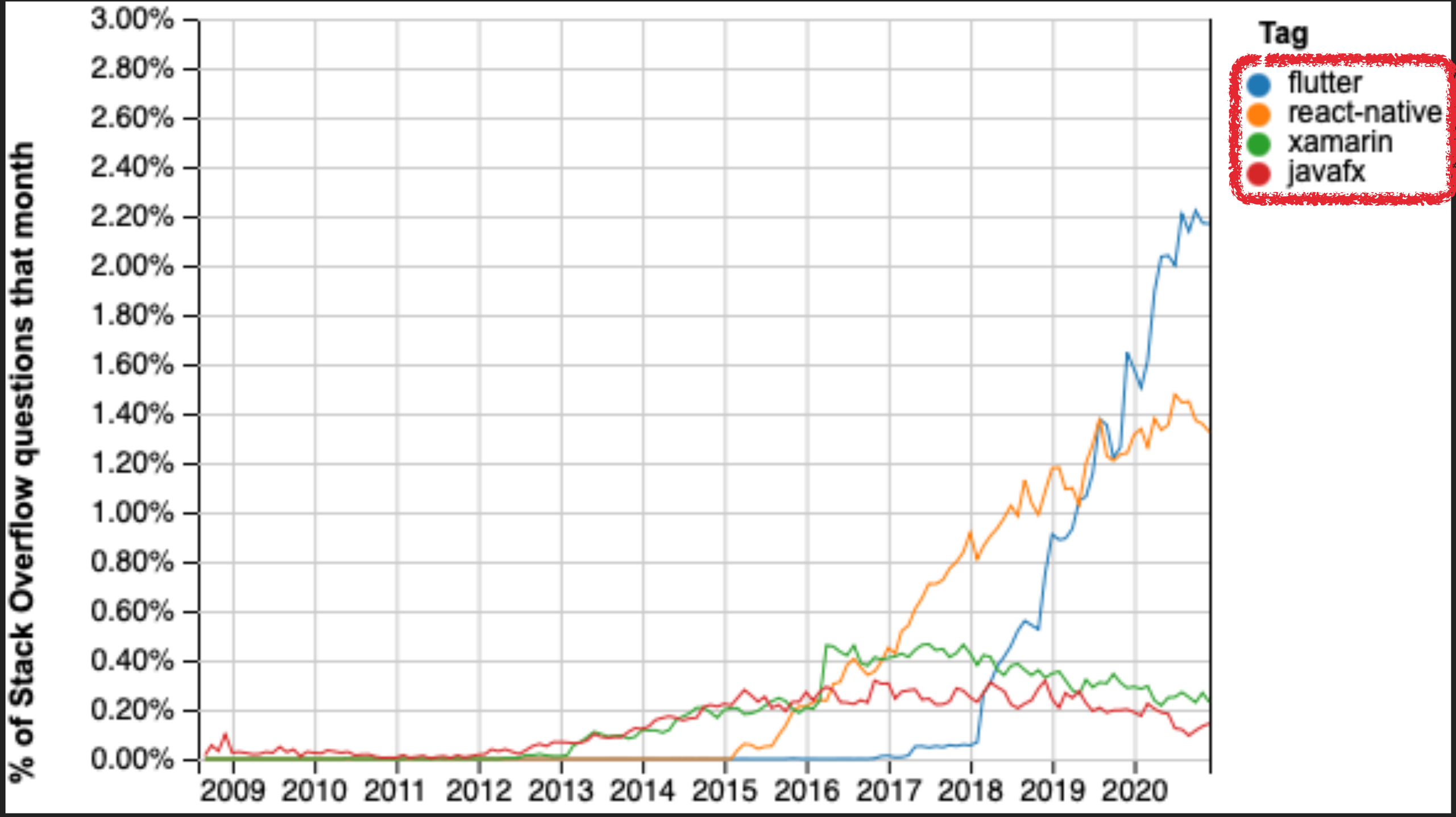
All categories

Web Search

Interest over time



POPULAR:
STACK OVERFLOW


% of Stack Overflow questions that month
3.00%
2.80%
2.60%
2.40%
2.20%
2.00%
1.80%
1.60%
1.40%
1.20%
1.00%
0.80%
0.60%
0.40%
0.20%
0.00%
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
Tag
flutter
react-native
xamarin
javafx

FLUTTER MOST POPULAR, GROWING

REACT NATIVE SECOND, FLAT

XAMARIN THIRD, SHRINKING/FLAT

JAVAFX LAST, SHRINKING/FLAT

POPULARITY: NATIVE APPS

NOT DECLARATIVE

NOT FULLY OPEN-SOURCE

WILL BE REPLACED BY .NET MAUI
(SCHEDULED FOR NOVEMBER 2021)

XAMARIN

NOT DECLARATIVE

REMOVED FROM JDK IN JAVA 11

BACKED BY GLUON

JAVAFX

XAMARIN & JAVAFX ARE OUT!

FLUTTER (GOOGLE)

JAVAFX (GLUON)

REACT NATIVE (FACEBOOK)

XAMARIN (MICROSOFT)

#2: NATIVE APPS

FLUTTER (GOOGLE)

REACT NATIVE (FACEBOOK)

#2: NATIVE APPS

HOW DO REACT NATIVE & FLUTTER WORK?

JAVASCRIPT	JAVASCRIPT VM	JAVASCRIPT BRIDGE	NATIVE UI ELEMENTS
REACT NATIVE			
FLUTTER			
DART	DEV: DART VM PROD: NATIVE CODE	SKIA REND. ENGINE	EMULATED UI ELEMENTS

- ⊖ JAVASCRIPT
- ⊖ SLOWER RENDERING
- ⊕ NATIVE UI ELEMENTS

- ⊕ DART
- ⊕ FASTER RENDERING
- ⊖ EMULATED UI ELEMENTS

REACT NATIVE

FLUTTER

I ONLY GO DEEP WITH FLUTTER IN THIS TALK!

WHAT IS FLUTTER?

"FLUTTER IS GOOGLE'S UI TOOLKIT FOR BUILDING BEAUTIFUL, NATIVELY COMPILED APPLICATIONS FOR MOBILE, WEB, AND DESKTOP FROM A SINGLE CODEBASE."

"PORTABLE UI TOOLKIT FOR ANY PLACE WHERE PIXELS ARE PAINTED"

WRITE ONCE, RUN ANYWHERE?!

REALLY BUILD MOBILE, WEB & DESKTOP FROM ONE CODEBASE?

ONLY MATERIAL DESIGN UI ELEMENTS FOR WEB & DESKTOP (GOOGLE DESIGN LANGUAGE)

WEB: BETA SINCE DEC 2019

DESKTOP: ALPHA – MAC SINCE DEC 2019, LINUX SINCE JULY 2019, WINDOWS SINCE SEP 2020

YES, BUT . . .

WHAT DOES MATERIAL DESIGN LOOK LIKE?

2:00 PM 4:00 PM

Single-line item

Single-line item

Greyhound divisive
marginally far upon

ACTION 1 ACTI

Title here

Enabled

Page title

ENABLED

ree Max

Snackbar with action ACTION

Unfilled

Lorem ipsum dolor sit amet,
consectetur adipisicing elit,
sed do eiusmod tempor

Conversion

537

Sales

345.8

Checkout form

First name

Last name

Email

Phone

Country

Country

City

Post code

Address

Additional info

Card Details

- Visa
- MasterCard
- Paypal

Name on card

Card number

CVV2

Card exp month

January

Year

SUBMIT REQUEST

IOS: NATIVE LOOK & FEEL, BUT INCOMPLETE (FORMS, TABLES)

ANDROID: MATERIAL DESIGN

PLUGINS FILL HOLES

UI ELEMENTS ON MOBILE

MY APP DEMO: IOS


07:59
radio.net
Installing...
2

WHAT DID WE JUST SEE?

NATIVE LOOK & FEEL, BUT SOME ISSUES REMAIN

PERFORMANCE OK, BUT NOT OPTIMISED, YET

USES PLATFORM (PASSWORDS, EMAIL,
PHOTO GALLERY)

SUMMARY: APP DEMO

IOS VS ANDROID

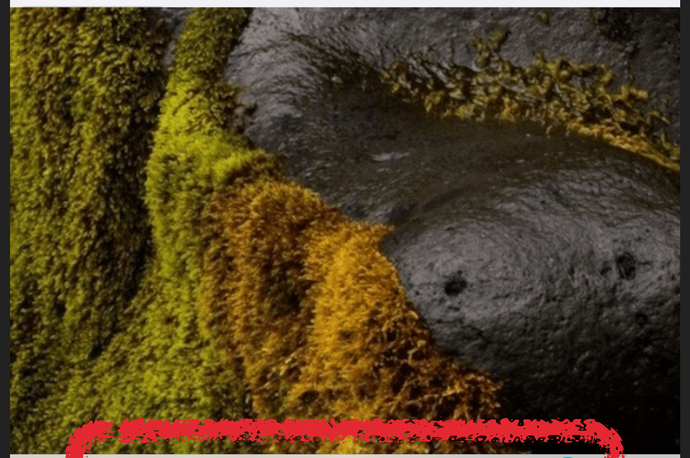
08:31

My Data

MY PROFILE

Name	Mr. Karsten Tester
Status	Active
Role	System administrator
Email	test.karsten@yourhomeingoodhands.com
Valid	19 Oct 2020 -
Fingerprint	m6tSwzrE7GhfktLEW25cYYRBxdI2
Sign-in	12 Dec 2020, 7:59

MY PICTURE



Tasks Team My Data

14:58

My Data

My profile

Mr. Karsten Tester

Status
Active

Role
System administrator

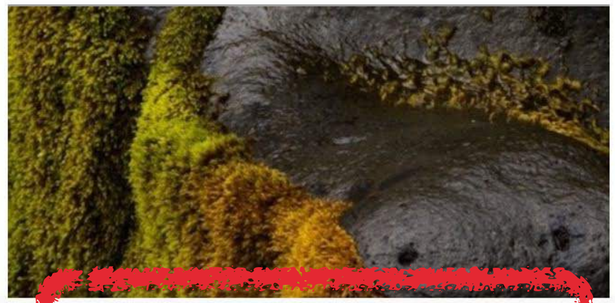
Email
test.karsten@yourhomeingoodhands.com

Valid
19 Oct 2020 -

Fingerprint
m6tSwzrE7GhfktLEW25cYYRBxdI2

Sign-in
9 Dec 2020, 9:46

My picture



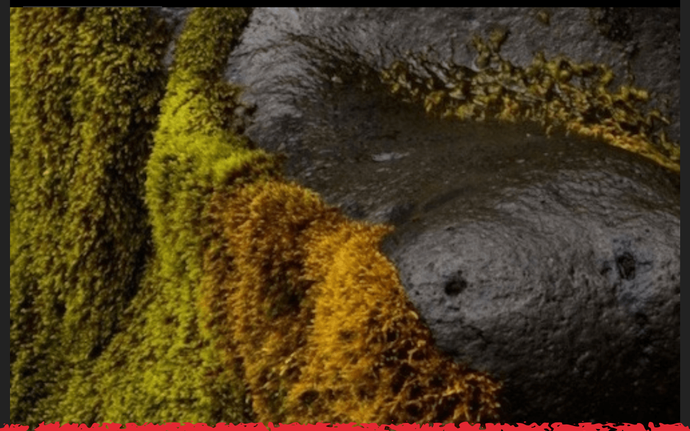
Tasks Team My Data

My Data

MY PROFILE

Name	Mr. Karsten Tester
Status	Active
Role	System administrator
Email	test.karsten@yourhomeingoodhands.com
Valid	19 Oct 2020 -
Fingerprint	m6tSwzrE7GhfktLEW25cYYRBxdI2
Sign-in	12 Dec 2020, 7:59

MY PICTURE



Tasks Team My Data

My Data

My profile

Name
Mr. Karsten Tester

Status
Active

Role
System administrator

Email
test.karsten@yourhomeingoodhands.com

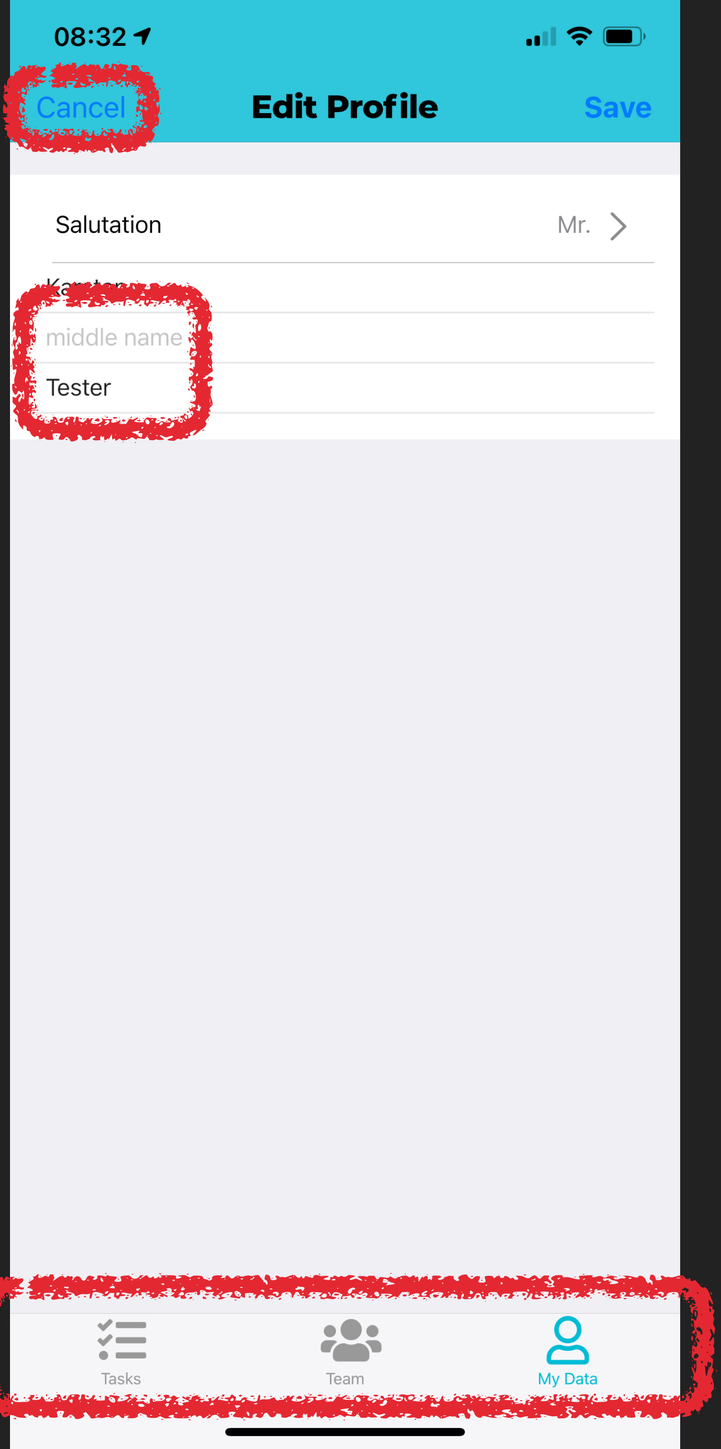
Valid
19 Oct 2020 -

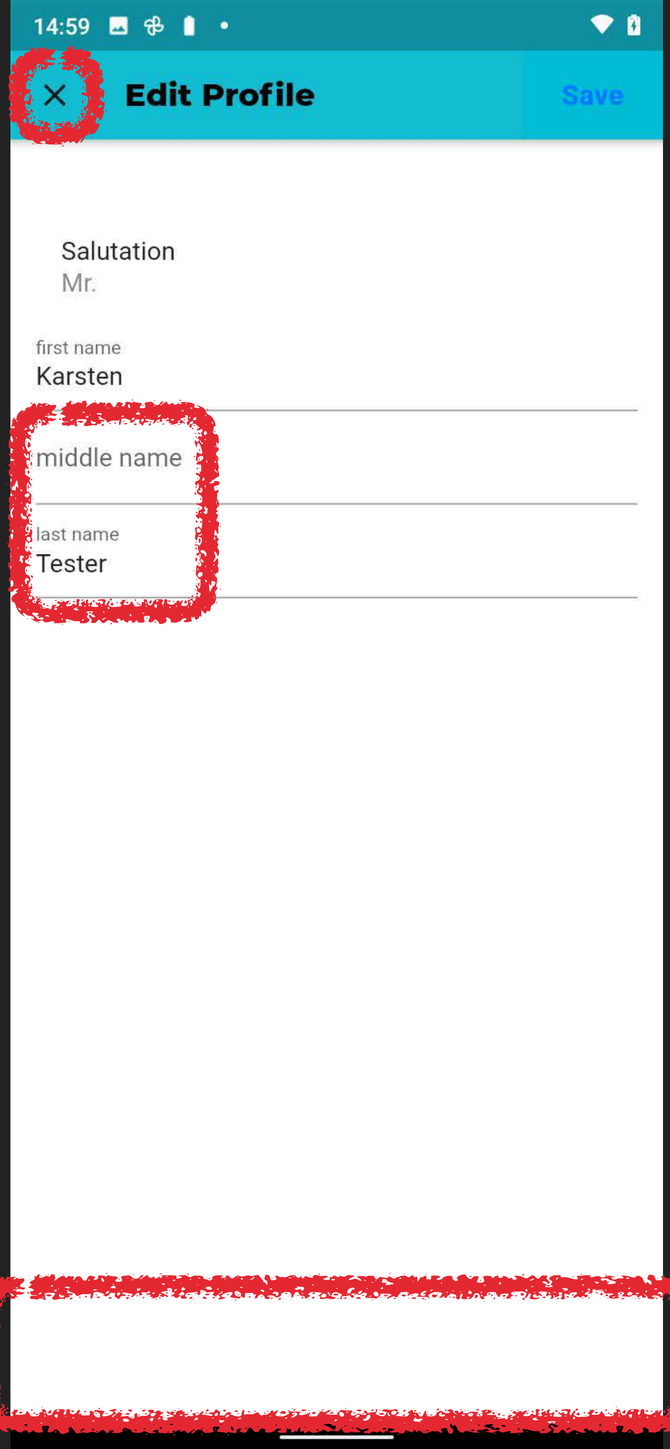
Fingerprint
m6tSwzrE7GhfktLEW25cYYRBxdI2

Sign-in
9 Dec 2020, 9:46

My picture

Tasks Team My Data


08:32
Cancel
Edit Profile
Save
Salutation
Mr.
middle name
Tester
Tasks
Team
My Data


14:59
Edit Profile
Save
Salutation
Mr.
first name
Karsten
middle name
last name
Tester

SWIFTUI VS FLUTTER

```swift
@State var count: Int = 0

var body: some View {
  VStack(alignment: .center,

    content: {
      Text("Counter: \(count)")
        .padding()
      Button(action: {
        self.count += 1
      }, label: {
        Text("Increment")
      })
  })
}
```

```dart
int _counter = 0;


return Column(
  mainAxisAlignment: MAA.center,
  children: [
    Text('Counter: $_counter'),

    TextButton(onPressed: () =>
      setState({_counter += 1}),

      child: Text('Increment'),
    ),
  ],
);
```

WHAT DID WE JUST SEE?

SYNTAX SLIGHTLY DIFFERENT
FROM SWIFTUI

STATE CHANGES ARE MARKED:
SETSTATE(() { ... });

FLUTTER IS DECLARATIVE

STATELESS WIDGETS: TEXT, BUTTON

STATEFUL WIDGETS: INPUT FIELDS

BOTH REPAINT IF PARENT CHANGES

STATEFUL REPAINT IF STATE CHANGES

EVERYTHING IS A WIDGET

STAND-ALONE APP

AS SCREENS IN NATIVE APP

HOW DOES FLUTTER RUN?

THROUGH PLUGINS (CAMERA, PICTURES, . . .)

SHOW NATIVE SCREENS IN FLUTTER

TALK TO NATIVE CODE THROUGH CHANNELS (PUBLISH & SUBSCRIBE) OR C API

NATIVE INTEGRATION?

MOBILE BUILD TIMES ARE LONG

HOT RELOAD SHOWS CHANGES IMMEDIATELY

HOT RESTART STARTS APP ANEW IN A FEW SECONDS

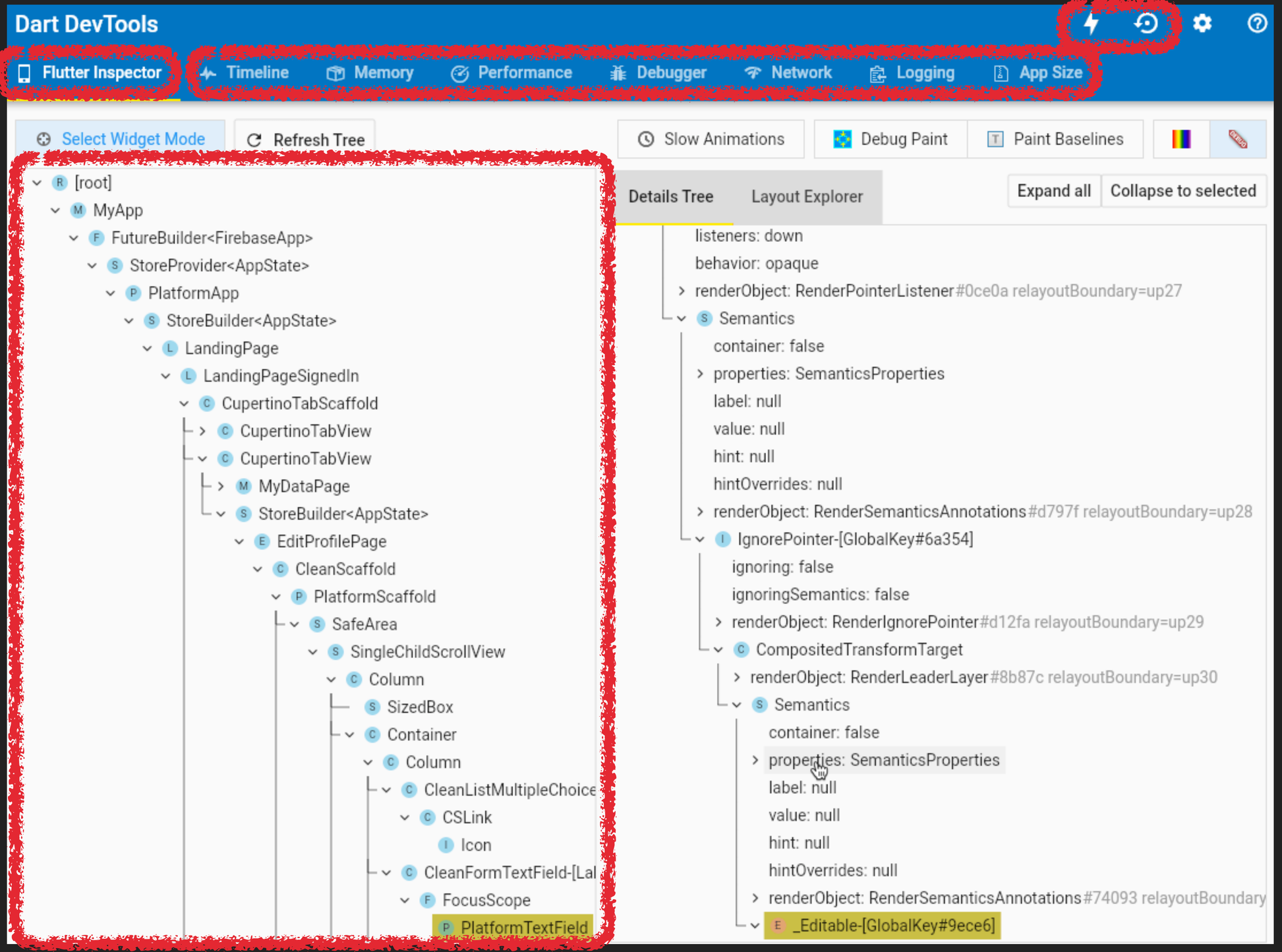
DEV TOOLS GIVE GREAT INSIGHTS

FAST BUILD & DEPLOY!

HAVE BARELY USED THEM

PROMISE TO HELP WITH TUNING THE APP

DEV TOOLS


Dart DevTools
Flutter Inspector
Timeline
Memory
Performance
Debugger
Network
Logging
App Size
Select Widget Mode
Refresh Tree
Slow Animations
Debug Paint
Paint Baselines
Details Tree
Layout Explorer
Expand all
Collapse to selected
[root]
MyApp
FutureBuilder<FirebaseApp>
StoreProvider<AppState>
PlatformApp
StoreBuilder<AppState>
LandingPage
LandingPageSignedIn
CupertinoTabScaffold
CupertinoTabView
CupertinoTabView
MyDataPage
StoreBuilder<AppState>
EditProfilePage
CleanScaffold
PlatformScaffold
SafeArea
SingleChildScrollView
Column
SizedBox
Container
Column
CleanListMultipleChoice
CSLink
Icon
CleanFormTextField-[Lal
FocusScope
PlatformTextField
listeners: down
behavior: opaque
renderObject: RenderPointerListener#0ce0a relayoutBoundary=up27
Semantics
container: false
properties: SemanticsProperties
label: null
value: null
hint: null
hintOverrides: null
renderObject: RenderSemanticsAnnotations#d797f relayoutBoundary=up28
IgnorePointer-[GlobalKey#6a354]
ignoring: false
ignoringSemantics: false
renderObject: RenderIgnorePointer#d12fa relayoutBoundary=up29
CompositedTransformTarget
renderObject: RenderLeaderLayer#8b87c relayoutBoundary=up30
Semantics
container: false
properties: SemanticsProperties
label: null
value: null
hint: null
hintOverrides: null
renderObject: RenderSemanticsAnnotations#74093 relayoutBoundary
_Editable-[GlobalKey#9ece6]

NOW ONTO DART!

WHY CAN'T WE JUST USE JAVA INSTEAD OF DART?

```
class MyClass extends AnotherClass {
  String myString;
  int myInt;
  List<String> myList = List<String>();

  String sayHello(String name) {
    var feedback = "Hello, " + name;
    return feedback;
  }
}
```

THIS IS DART!

"SIMPLIFIED JAVA": STATICALLY TYPED

GOOGLE BUILT IT ORIGINALLY FOR BROWSER

USEFUL CHANGES (ASYNC/AWAIT, PARAMETERS, . . .)

SYNTACTIC SUGAR FOR BUILDING FRONT-ENDS

WHAT IS DART?

MY EXPERIENCE WITH FLUTTER

FLUTTER PROTOTYPE SUMMER 2019

PROGRESSIVE WEB APP PROTOTYPE
END OF 2019

FLUTTER PROJECT FOR IOS & ANDROID
SINCE SEPTEMBER 2020

MY PROJECTS

60% FUN

35% WORK

5% FRUSTRATION

MY SUMMARY

FLUTTER: THE GOOD

SOURCE CODE & ISSUES PUBLIC AT GITHUB

INSPECT & DEBUG FLUTTER PLATFORM & PLUGINS

OPEN-SOURCE

SIMILAR ENOUGH TO JAVA TO BE FAMILIAR

LANGUAGE ENHANCEMENTS SAVE TIME

DART

HOT RELOAD & HOT RESTART
SAVE A TON OF TIME

PREVENT FLOW-BREAKING PAUSES

RELOAD & RESTART ROCK!

GREAT PLUGIN PORTAL

ADD NATIVE FUNCTIONALITY (CAMERA, LOCATION, . . .)

PLUG HOLES IN FLUTTER (LAYOUT, WIDGETS, . . .)

MOST OPEN-SOURCE WITH PERMISSIVE LICENSE

PLUGINS

flutter_platform_widgets 0.72.0

Published Dec 3, 2020 • stryder.dev

FLUTTER | ANDROID | IOS | WEB

281

Readme | Changelog | Example | Installing | Versions | Scores

Flutter Platform Widgets

pub v0.72.0 | license MIT

This project is an attempt to see if it is possible to create widgets that are platform aware. Currently in order to render targeted Material or Cupertino device specific styles, you need to either conditionaly check the platform or create a set of widgets to render differently depending on the running platform.

This package supports the **Stable** release as a full released version.

Beta or Dev channels might be supported when there is a pre-release version. Please check the CHANGELOG for version compatibility version.

Due to Master being in rapid development this package is unable to support Master. If this support is required then it is best to fork the repo and locally reference the forked version where changes can be made appropriately.

Installation

> pub.dev: https://pub.dev/packages/flutter_platform_widgets

How it works

The flutter `ThemeData` object used with the `Theme` widget has a `platform` property. This defaults to `TargetPlatform.android` on Android and `TargetPlatform.ios` on iOS (also for macos etc), but

281	100	96%
LIKES	PUB POINTS	POPULARITY

Publisher

stryder.dev

Metadata

Simplifying the use of both Material and Cupertino widgets with a single widget

Repository (GitHub)

View/report issues

Documentation

API reference

License

MIT (LICENSE)

Dependencies

flutter

More

Packages that depend on flutter_platform_widgets

[0.72.0] - December 03, 2020

(Flutter version support: v1.22.0 - v1.22.4)

- Adds `materialBuilder` and `cupertinoBuilder` as alternatives to `material` and `cupertino` on `PlatformTabScaffold` so that the widgets can change properties on tab index change. see example/lib/tabbed/dynamicTabbedPage.dart.
- Adds new polatform icons, thumb up and down and fix the star icon (thanks DFelten)

[0.71.0] - November 08, 2020

(Flutter version support: v1.22.0 - v1.22.3)

- Added `PlatformApp.router` to support Router 2.0

[0.70.2] - October 08, 2020

(Flutter version support: v1.22.0)

- fix export of MaterialModalSheetData and CupertinoModalSheetData (thanks in74mz)

example/lib/main.dart

```dart
import 'package:flutter/cupertino.dart'
    show
        CupertinoActionSheet,
        CupertinoActionSheetAction,
        CupertinoDynamicColor,
        CupertinoIcons,
        CupertinoThemeData,
        DefaultCupertinoLocalizations;
import 'package:flutter/material.dart'
    show
        Colors,
        DefaultMaterialLocalizations,
        Icons,
        Theme,
        ThemeData,
        ThemeMode;
import 'package:flutter/services.dart';
import 'package:flutter/widgets.dart';
import 'package:flutter_platform_widgets/flutter_platform_widgets.dart';

import 'iconPage.dart';
import 'listViewHeaderPage.dart';
import 'listViewPage.dart';
import 'tabbed/basicTabbedPage.dart';
import 'tabbed/dynamicTabbedPage.dart';
import 'tabbed/originalTabbedPage.dart';
import 'tabbed/sliverTabbedPage.dart';
```

Readme Changelog Example Installing Versions Scores

Stable versions of flutter_platform_widgets

Version	Uploaded
0.72.0	Dec 3, 2020
0.71.0	Nov 12, 2020
0.70.2	Oct 8, 2020
0.70.1	Oct 5, 2020
0.70.0	Oct 5, 2020
0.60.2	Aug 22, 2020
0.60.1	Aug 5, 2020

Readme Changelog Example Installing Versions Scores

281
LIKES

100 / 110
PUB POINTS

96%
POPULARITY

We analyzed this package on Dec 10, 2020, and awarded it 100 pub points (of a possible 110):

✓	Follow Dart file conventions	20/20
×	Provide documentation	10/20
✓	Support multiple platforms	20/20
✓	Pass static analysis	30/30
✓	Support up-to-date dependencies	20/20

WHAT IF PLUGIN HAS BUGS OR NOT MAINTAINED?

1. FORK IT ON GITHUB

2. MAKE CHANGES IN FORK

3. USE FORK GIT URL IN FLUTTER APP BUILD FILE

FIXING PLUGINS

FLUTTER RECENTLY OUTSOURCED DEVELOPMENT OF SOME FIRST-PARTY PLUGINS TO KEEP UP

QUARTERLY SURVEY FOR DEVELOPERS

FLUTTER TEAM LISTENS

FLUTTER: THE BAD

WILL GOOGLE KILL FLUTTER?

Killed by Google

Search

All (223)

All (223)
Apps (35)
Services (169)
Hardware (19)

Join a bunch of others and follow @killedbygoogle on Twitter.

January 2022

Android Things

Fading into darkness in 12 mon
Android-based embedded ope
system (originally named Brillo) aimed to run on Internet of Things (IoT) devices It will be over 6 years old.

mber 2021

AngularJS

Vanishing in 11 months, AngularJS was a JavaScript open-source front-end web framework based on MVC pattern using a dependency injection technique. It will be about 11 years old.

June 2021

Google Chrome Apps

Running out of juice in 5 months, Google Chrome Apps were hosted or packaged web applications that ran on the Google Chrome browser. It will be over 10 years old.

June 2021

Google Hangouts

Sentenced to death in 5 months, Google Hangouts was a communication platform which included messages, video chat, and VOIP features. Execution scheduled "first half 2021". It will be over 8 years old.

June 2021

Expeditions

Getting unplugged in 5 months, Expeditions is a program for providing virtual reality experiences to school classrooms through Google Cardboard viewers, allowing educators to take their students on virtual field trips. It will be almost 6 years old.

BAD: COMPETING EFFORTS AT GOOGLE – ANGULAR & JETPACK COMPOSE

GOOD: IN NOV 2019, FLUTTER #3 FOR CONTRIBUTORS AT GITHUB (13K) – POPULAR OUTSIDE OF GOOGLE

GOOD: FLUTTER PROVIDES UI IN PROJECT FUCHSIA

NOBODY KNOWS!

WHY? KEEPING UP WITH MOBILE WHILE BUILDING WEB & DESKTOP

ANNOYING ISSUES LINGER TOO LONG (BUT NO SHOWSTOPPER, YET)

SEEMS LIKE A LOT OF OPEN ISSUES

TEAM STRETCHED TOO THIN

APP THAT'S NATIVE ON IOS AND MATERIAL ON ANDROID

ALL THE PARTS ARE THERE (UI ELEMENTES, COLORS, DARK MODE, TEXT SIZE CHANGES)...

...BUT TOO MUCH MANUAL ASSEMBLY REQUIRED!

"2X NATIVE" TOO HARD

WISHLIST SCORE

JAVA FLU

DECLARATIVE –

SAFE REA

POPULAR FLU

WEB APPS REA

MOBILE APPS REA

DESKTOP APPS FLU

COMPLETE FLU

FAST FLU

ALL NATIVE FEATURES ???

ONE CODE BASE FLU

FAST BUILD, DE-PLOY & DEBUG ???

FLUTTER WINS

REACT NATIVE SECOND

JAVAFX AND XAMARIN OUT

SCORE: NATIVE

IF YOU ALREADY USE REACT NATIVE
OR FLUTTER, THEN KEEP IT

IF YOU DON'T, THEN GO FOR FLUTTER
FIRST, THEN REACT NATIVE

ADVICE: #2: NATIVE APPS

#3: DESKTOP

THERE'S NO GOOD, PRODUCTION-READY CROSS-PLATFORM OPTION FOR DESKTOP

WEB HAS LESS RESTRICTIONS ON DESKTOP THAN ON MOBILE

THE WAITING GAME

KEEP YOUR EXISTING WEB APPLICATION FOR THE DESKTOP

IF YOU CAN USE MATERIAL DESIGN ON THE DESKTOP AND YOU USE FLUTTER, THEN EVALUATE FLUTTER FOR DESKTOP IN 2022

ADVICE: #3: DESKTOP

~~HOW CAN WE~~
~~BUILD FRONT-ENDS TODAY?~~

~~HOW SHOULD~~ JAVA DEVELOPERS
~~BUILD FRONT-ENDS TODAY?~~

SUMMARY

#1: JUST WEB: BUILD ONLY WEB APPLICATION, FROM SCRATCH

#2: NATIVE APPS: WEB APPLICATION EXISTS, BUILD NATIVE IOS & ANDROID APPS FROM SCRATCH

#3: DESKTOP APPS: WEB APPLICATION EXISTS

SCENARIOS

IF YOU ALREADY USE REACT, ANGULAR,
OR VUE.JS, THEN KEEP IT

IF YOU DON'T, THEN GO FOR REACT
FIRST, THEN ANGULAR, THEN VUE.JS

ADVICE: #1: JUST WEB

IF YOU ALREADY USE REACT NATIVE OR FLUTTER, THEN KEEP IT

IF YOU DON'T, THEN GO FOR FLUTTER FIRST, THEN REACT NATIVE

ADVICE: #2: NATIVE APPS

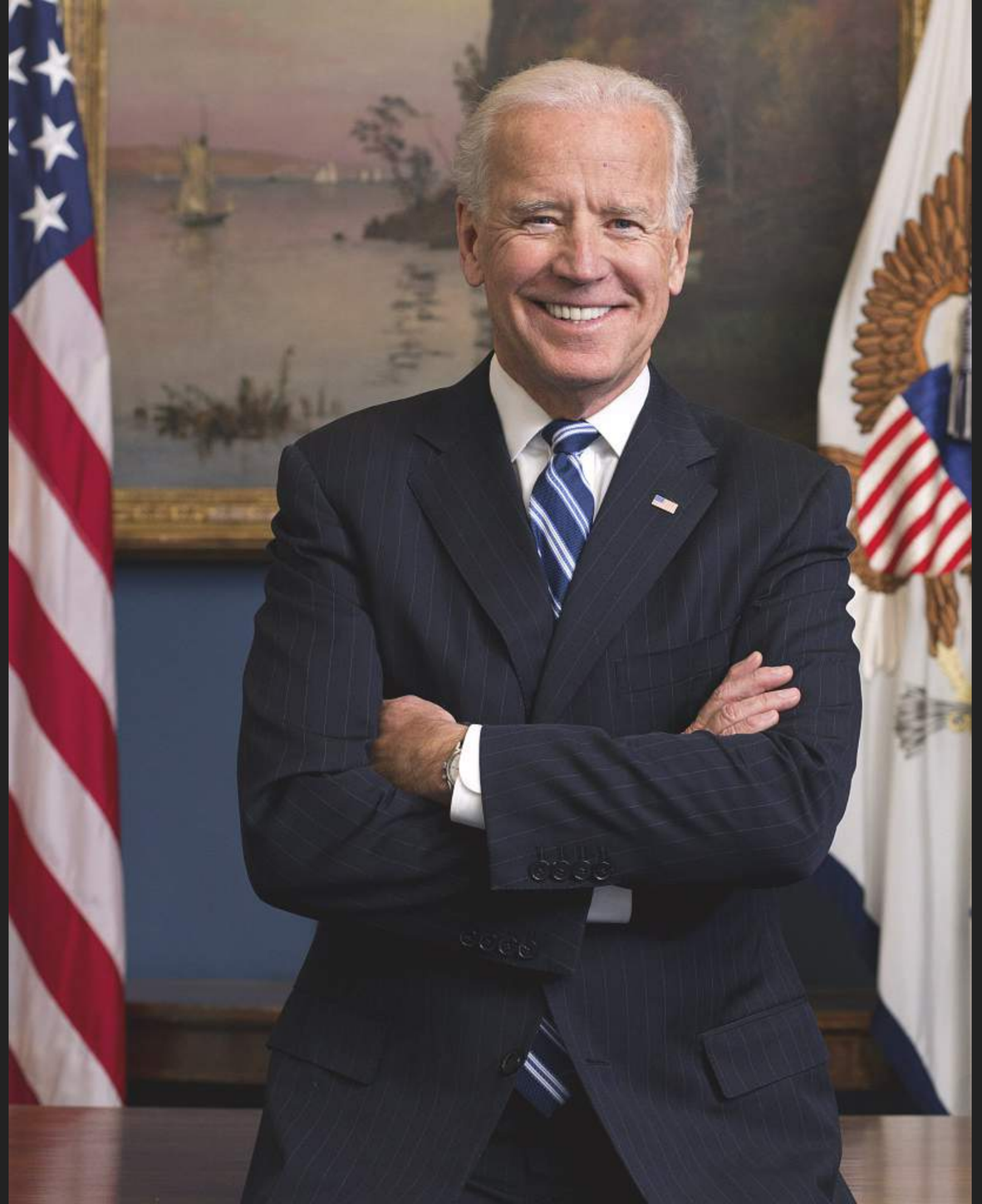
KEEP YOUR EXISTING WEB APPLICATION FOR THE DESKTOP

IF YOU CAN USE MATERIAL DESIGN ON THE DESKTOP AND YOU USE FLUTTER, THEN EVALUATE FLUTTER FOR DESKTOP IN 2022

ADVICE: #3: DESKTOP

LAST, BUT NOT LEAST:

YOU HAVE A
NEW PRESIDENT!



SLIDES & VIDEOS

GET STARTED

HOW TO BUILD JAVA APPS TODAY?

BPF.LI/PAT