Introduction to Flutter

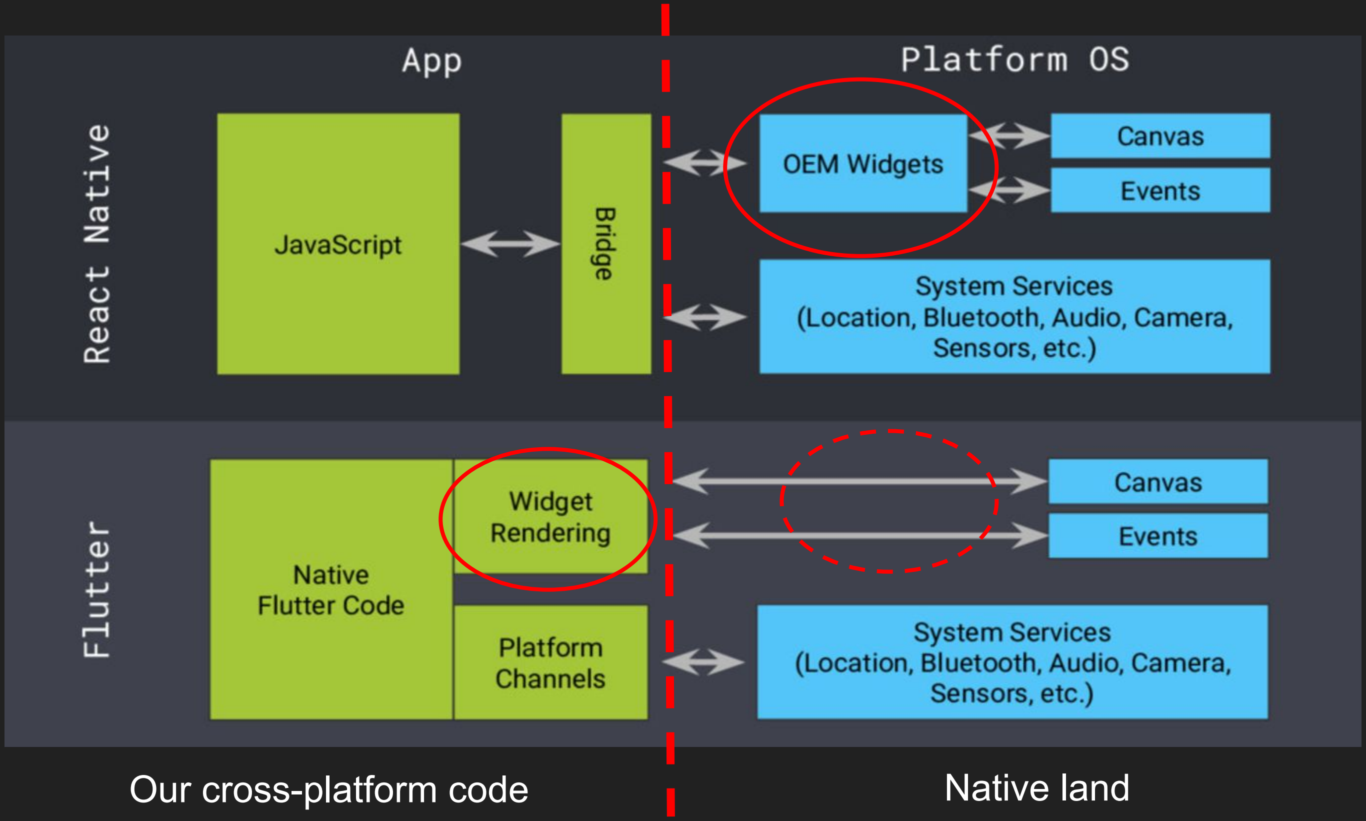
Slides for an internal talk we had at Codemate Ltd.
14th of September 2017

What is Flutter?

- SDK for building cross-platform mobile apps, built by Google
- Targets Android, iOS and Fuchsia
- Consistent UIs across devices and manufacturers
- Superb performance

Why Flutter?

- Strongly typed, modern language (Dart with Strong Mode on)
- Same codebase, two platforms: Android & iOS
- AOT compilation -> no Javascript or any other runtime / VM
- No WebViews, no native Views
    - Why? We’ll see.
- Once your UI works, it just works. And keeps working.
    - Manufacturers / OS versions / different devices can’t break it
- Especially Android APIs require a lot of ceremony for simple things
    - Flutter was able to start from scratch and avoid previously made pitfalls



Our cross-platform code

- Everything in our control
- Things we do here have fantastic performance and are cheap
- We should stay here as much as possible

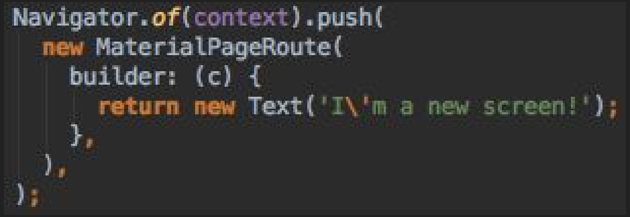
Native land

- Also fast performance here, however:
  - Expensive to travel to
- We can't afford to go here too often, just like we can't afford beach vacations every week

Base image from: https://speakerdeck.com/passsy/flutter-60-fps-ui-of-the-future

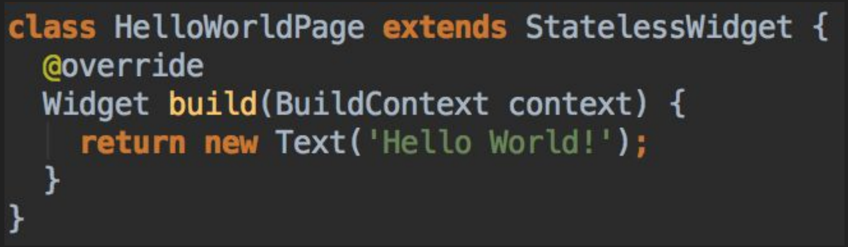
Widgets

- To build UIs, we have Widgets -> the only UI building block in Flutter
- The whole app is a Widget. A screen is a Widget that contains Widgets. Widgets are made by composing basic Widgets into more advanced Widgets.
  - Yo dawg?
- There’s a huge amount of different Widgets
- Can represent a:
  - UI element, such as Text, Button, BottomNavigationBar, TextField, etc.
  - Layout element, such as Padding, Center, Stack, Column, etc.
  - Completely new screen (Activity/ViewController equivalent), for example:



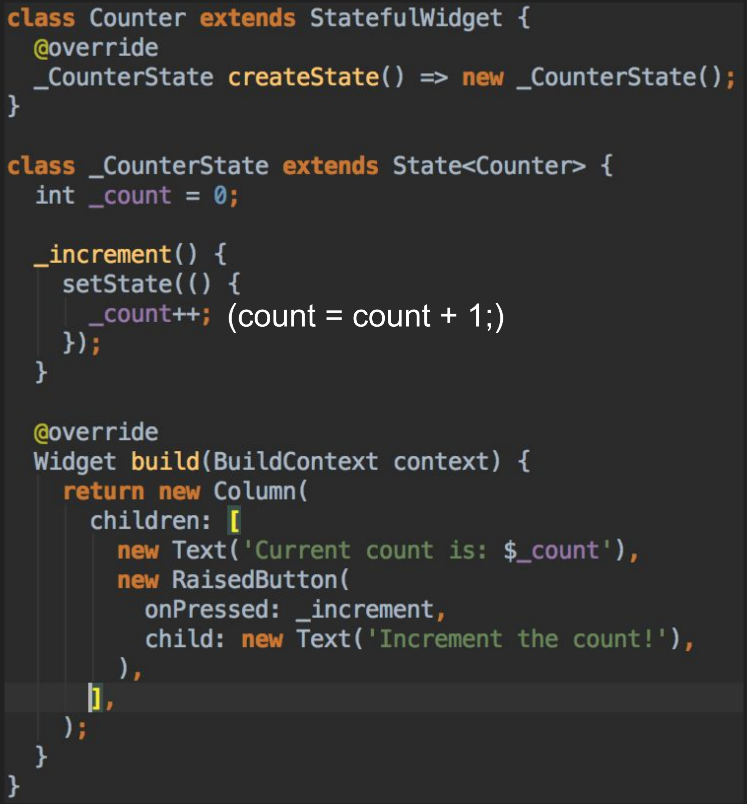
```
Navigator.of(context).push(
  new MaterialPageRoute(
    builder: (c) {
      return new Text('I\'m a new screen!');
    },
  ),
);
```

Stateless Widgets



```
class HelloWorldPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new Text('Hello World!');
  }
}
```

- Have no state (duh)
- Immutable -> all instance fields are final
- For displaying something that doesn’t change once it has been initialized

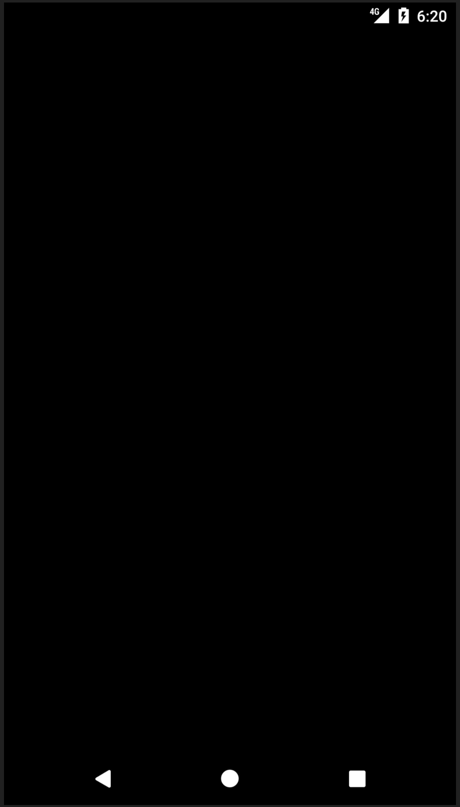


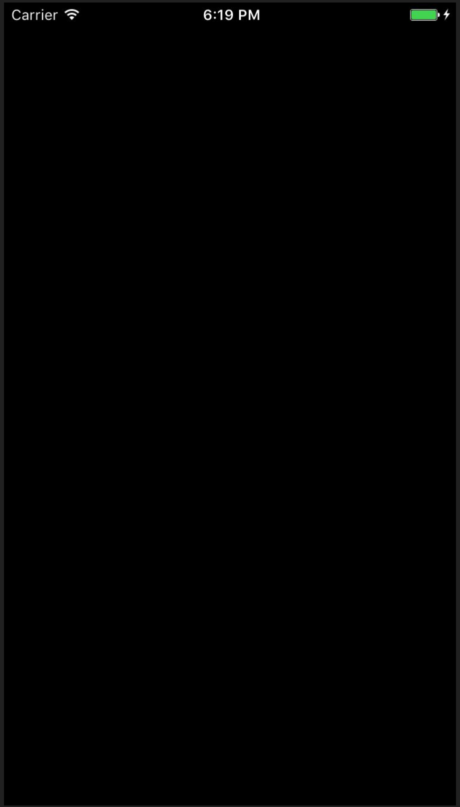
```
class Counter extends StatefulWidget {
  @override
  _CounterState createState() => new _CounterState();
}

class _CounterState extends State<Counter> {
  int _count = 0;

  _increment() {
    setState(() {
      _count++; (count = count + 1;)
    });
  }

  @override
  Widget build(BuildContext context) {
    return new Column(
      children: [
        new Text('Current count is: $_count'),
        new RaisedButton(
          onPressed: _increment,
          child: new Text('Increment the count!'),
        ),
      ],
    );
  }
}
```

Stateful Widgets

- Have a state (duh)
- The state has mutable instance fields that can be read synchronously
- Call setState() method for updating the state
- Framework handles UI Widget updates intelligently and efficiently when necessary
- So basically, really similar to React concepts

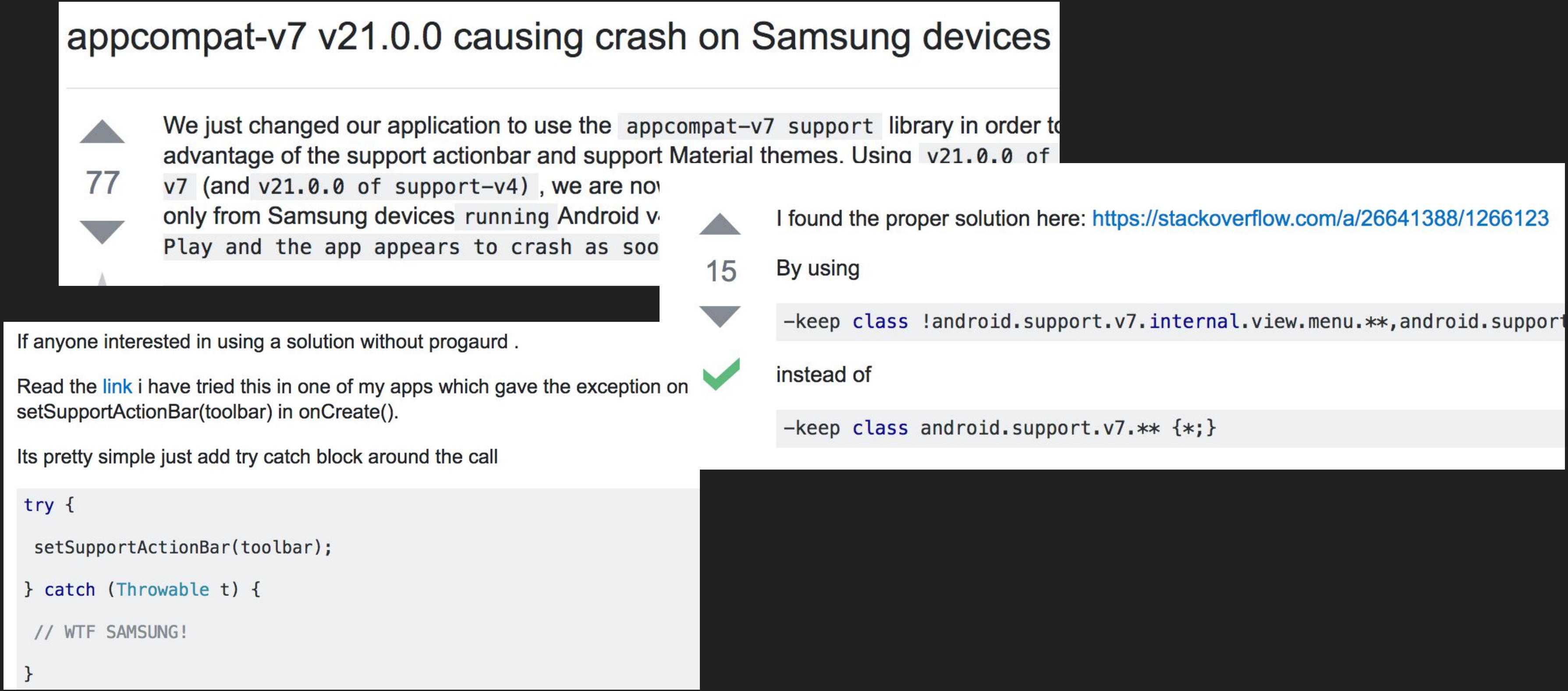
Widget rendering





- No native Views or WebViews
- Instead, a completely blank Canvas as seen on left
- The Material & Cupertino widgets are made by composing more basic Widgets
- Widgets are made of low-level rendering layer objects
  - In the end, Skia, C++ graphics library, renders them directly to these Canvases
- We have direct access to Canvas
  - pretty much any UI, even a wilder one, is doable

What’s the value in custom rendering anyway?



appcompat-v7 v21.0.0 causing crash on Samsung devices

77

We just changed our application to use the `appcompat-v7 support` library in order to
advantage of the support actionbar and support Material themes. Using `v21.0.0 of`
`v7` (and `v21.0.0 of support-v4)` , we are no
only from Samsung devices `running` Android v
`Play and the app appears to crash as soo`

15

I found the proper solution here: https://stackoverflow.com/a/26641388/1266123

By using

```
-keep class !android.support.v7.internal.view.menu.**,android.support
```

instead of

```
-keep class android.support.v7.** {*;}
```

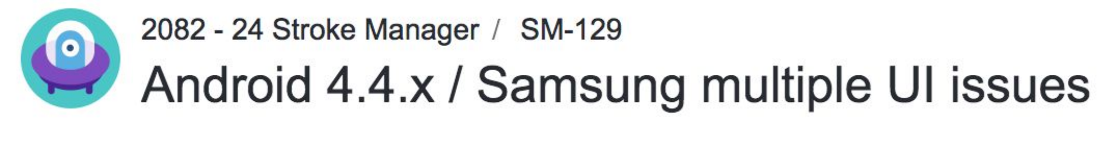
If anyone interested in using a solution without progaurd .

Read the link i have tried this in one of my apps which gave the exception on setSupportActionBar(toolbar) in onCreate().

Its pretty simple just add try catch block around the call

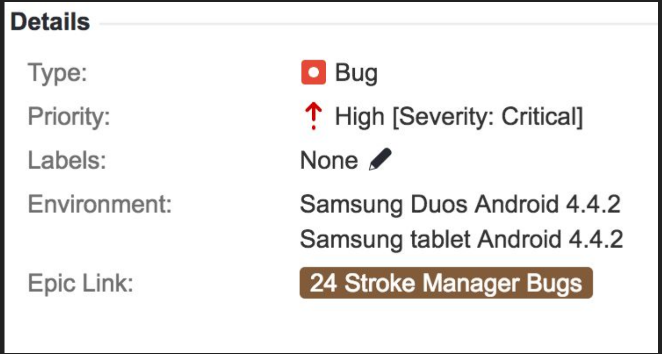
```
try {

 setSupportActionBar(toolbar);

} catch (Throwable t) {

 // WTF SAMSUNG!

}
```

An actual ticket from our QA



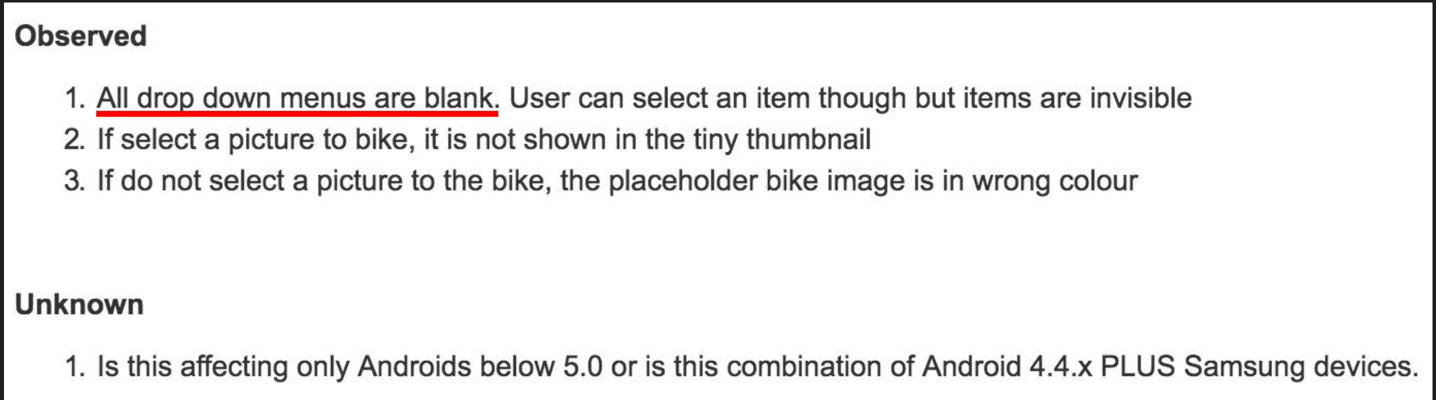
2082 - 24 Stroke Manager / SM-129

Android 4.4.x / Samsung multiple UI issues



Details

Type:	Bug
Priority:	↑ High [Severity: Critical]
Labels:	None
Environment:	Samsung Duos Android 4.4.2 Samsung tablet Android 4.4.2
Epic Link:	24 Stroke Manager Bugs



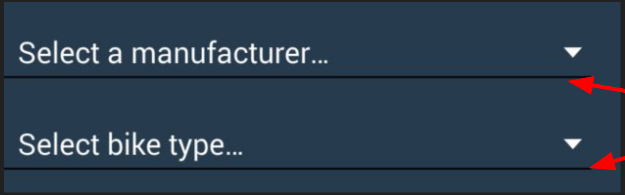
Observed

1. All drop down menus are blank. User can select an item though but items are invisible
2. If select a picture to bike, it is not shown in the tiny thumbnail
3. If do not select a picture to the bike, the placeholder bike image is in wrong colour

Unknown

1. Is this affecting only Androids below 5.0 or is this combination of Android 4.4.x PLUS Samsung devices.

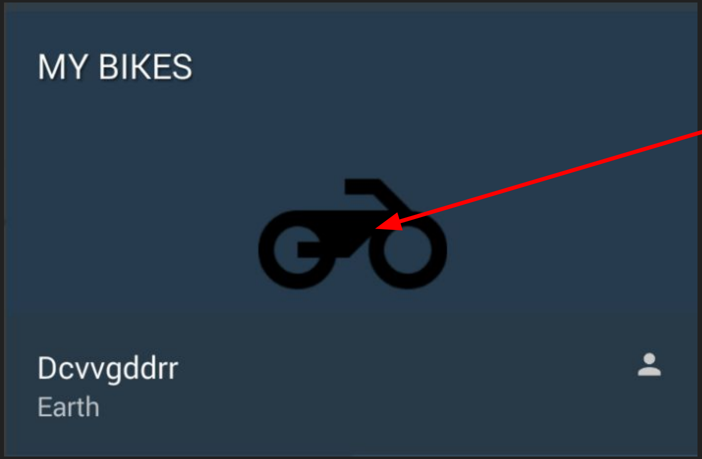
Here’s how it looked like



Select a manufacturer…

Select bike type…

weird black lines

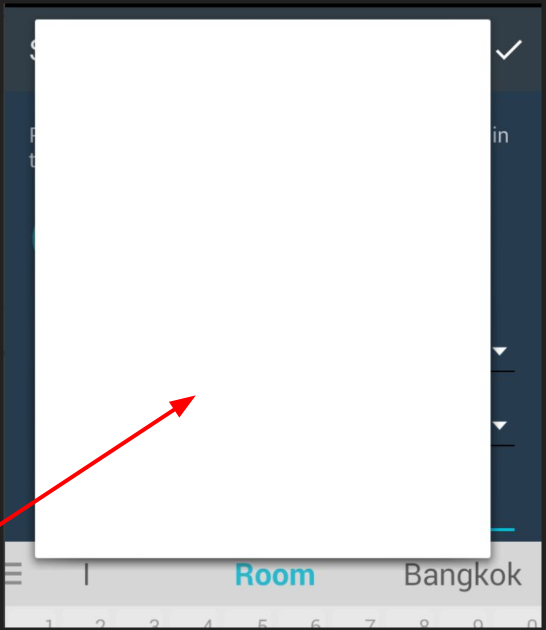


MY BIKES

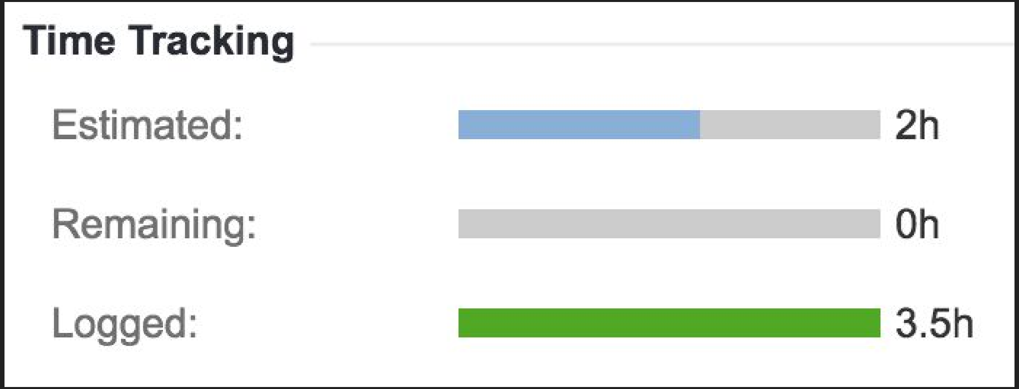
Dcvvgddrr

Earth

Supposed to be white



Dropdown items that are supposed to be, you know, visible?



Time Tracking

Estimated: 2h

Remaining: 0h

Logged: 3.5h

Half a day for fixing bugs on two specific devices
on one specific OS version.

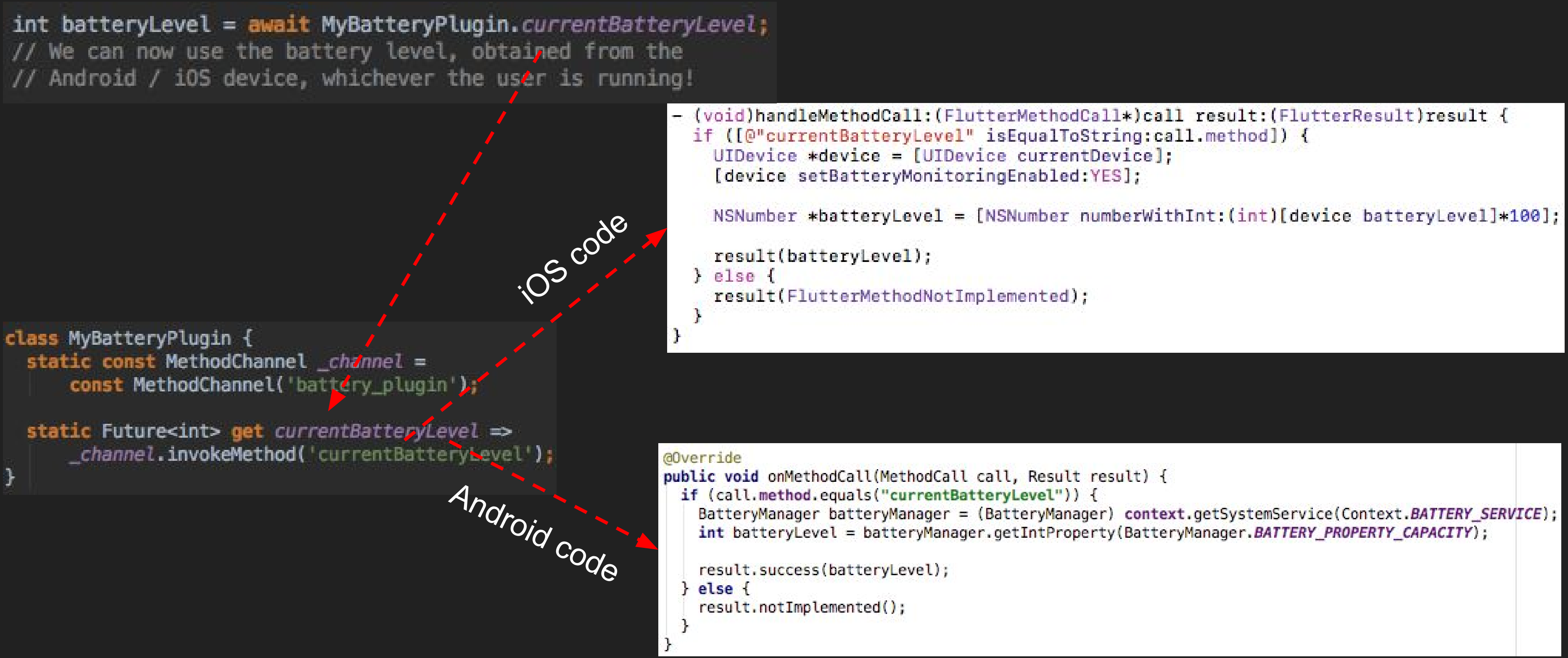
Dependency management

- Ships with Pub, a modern dependency manager for Dart
- Official package repository hosted at pub.dartlang.org
- The whole existing ecosystem of Dart libraries available
  - Excludes lots of web-related libraries
- Also supports packages from Git, if you’re feeling lucky

Native Plugins

- Allow access to every native platform API
  - Bluetooth, geolocation, sensors, fingerprint, camera, etc.
- Both official and community-driven plugins available
- Some plugins missing or in early stages
  - There’s a community-driven geolocation plugin with really limited API
  - There’s a community-driven Bluetooth plugin that doesn’t work with iOS just yet
- If a plugin for your use case doesn’t exist, you’ll have to make it yourself
- This is where other frameworks like React Native & Xamarin currently shine and Flutter takes the loss
  - Situation expected to be solved with time

Plugin sample: get current battery level



```
int batteryLevel = await MyBatteryPlugin.currentBatteryLevel;
// We can now use the battery level, obtained from the
// Android / iOS device, whichever the user is running!
```

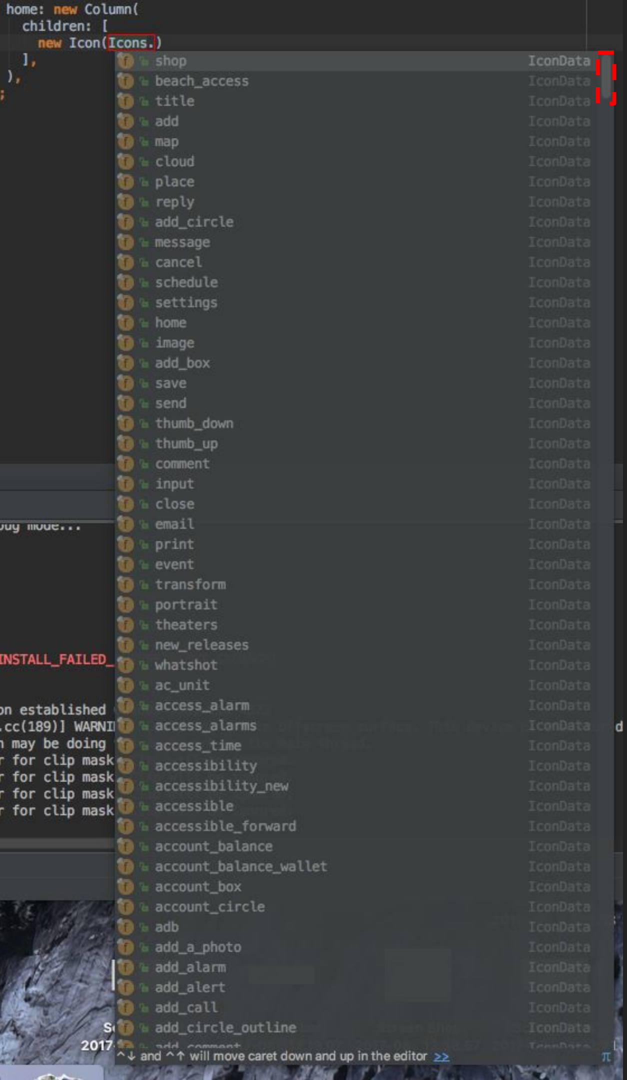
```
class MyBatteryPlugin {
  static const MethodChannel _channel =
      const MethodChannel('battery_plugin');

  static Future<int> get currentBatteryLevel =>
      _channel.invokeMethod('currentBatteryLevel');
}
```

iOS code

```
- (void)handleMethodCall:(FlutterMethodCall*)call result:(FlutterResult)result {
  if ([@"currentBatteryLevel" isEqualToString:call.method]) {
    UIDevice *device = [UIDevice currentDevice];
    [device setBatteryMonitoringEnabled:YES];

    NSNumber *batteryLevel = [NSNumber numberWithInt:(int)[device batteryLevel]*100];

    result(batteryLevel);
  } else {
    result(FlutterMethodNotImplemented);
  }
}
```

Android code

```
@Override
public void onMethodCall(MethodCall call, Result result) {
  if (call.method.equals("currentBatteryLevel")) {
    BatteryManager batteryManager = (BatteryManager) context.getSystemService(Context.BATTERY_SERVICE);
    int batteryLevel = batteryManager.getIntProperty(BatteryManager.BATTERY_PROPERTY_CAPACITY);

    result.success(batteryLevel);
  } else {
    result.notImplemented();
  }
}
```



```
home: new Column(
  children: [
    new Icon(Icons.)
  ],
),
;
```

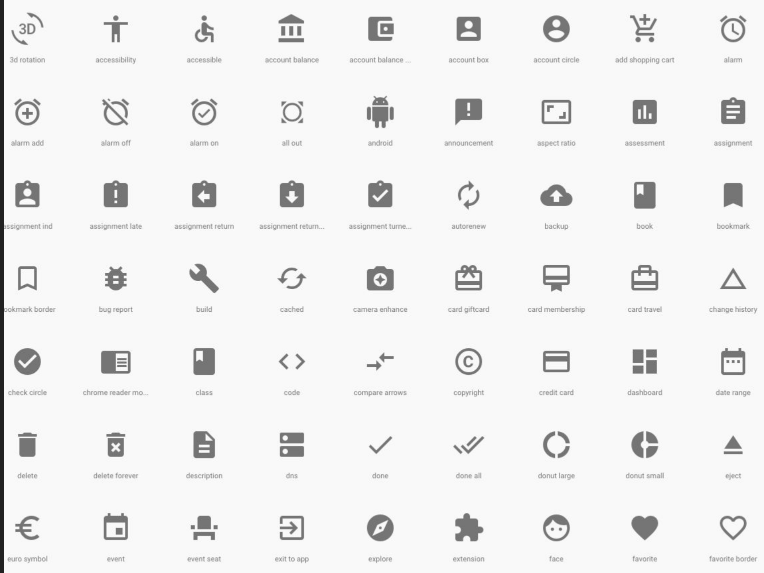
```
shop                      IconData
beach_access              IconData
title                     IconData
add                       IconData
map                       IconData
cloud                     IconData
place                     IconData
reply                     IconData
add_circle                IconData
message                   IconData
cancel                    IconData
schedule                  IconData
settings                  IconData
home                      IconData
image                     IconData
add_box                   IconData
save                      IconData
send                      IconData
thumb_down                IconData
thumb_up                  IconData
comment                   IconData
input                     IconData
close                     IconData
email                     IconData
print                     IconData
event                     IconData
transform                 IconData
portrait                  IconData
theaters                  IconData
new_releases              IconData
whatshot                  IconData
ac_unit                   IconData
access_alarm              IconData
access_alarms             IconData
access_time               IconData
accessibility             IconData
accessibility_new         IconData
accessible                IconData
accessible_forward        IconData
account_balance           IconData
account_balance_wallet    IconData
account_box               IconData
account_circle            IconData
adb                       IconData
add_a_photo               IconData
add_alarm                 IconData
add_alert                 IconData
add_call                  IconData
add_circle_outline        IconData
```

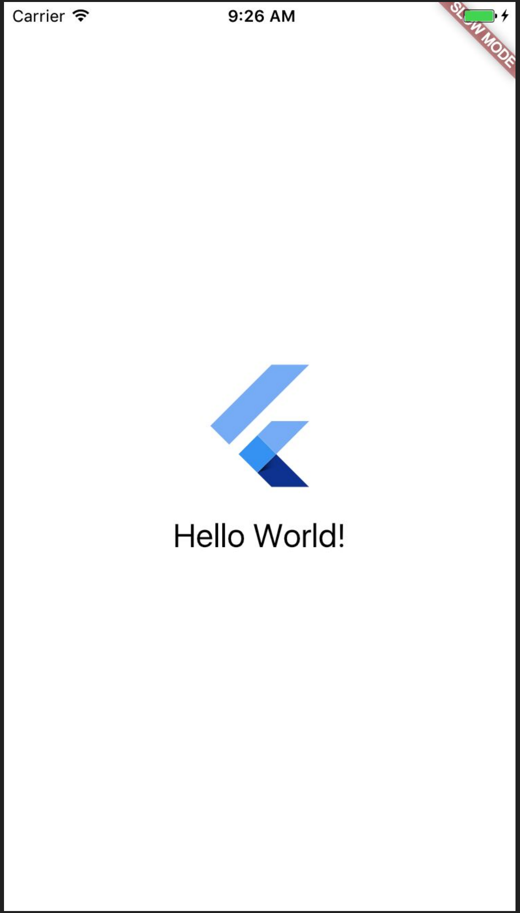
```
bug mode...
INSTALL_FAILED_
on established
.cc(189)] WARNI
n may be doing
r for clip mask
r for clip mask
r for clip mask
r for clip mask
```

^↓ and ^↑ will move caret down and up in the editor >>

Icons

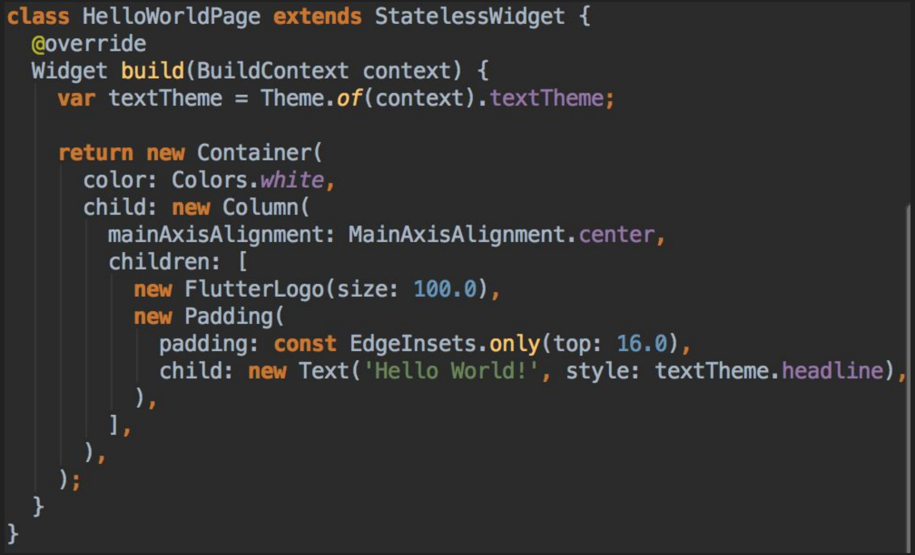
- Ships with a whole lot of premade, quality vector icons
- Just say “new Icon(Icons.add_call);”
- You can also import your own icons & icon fonts if you want





Hello World!

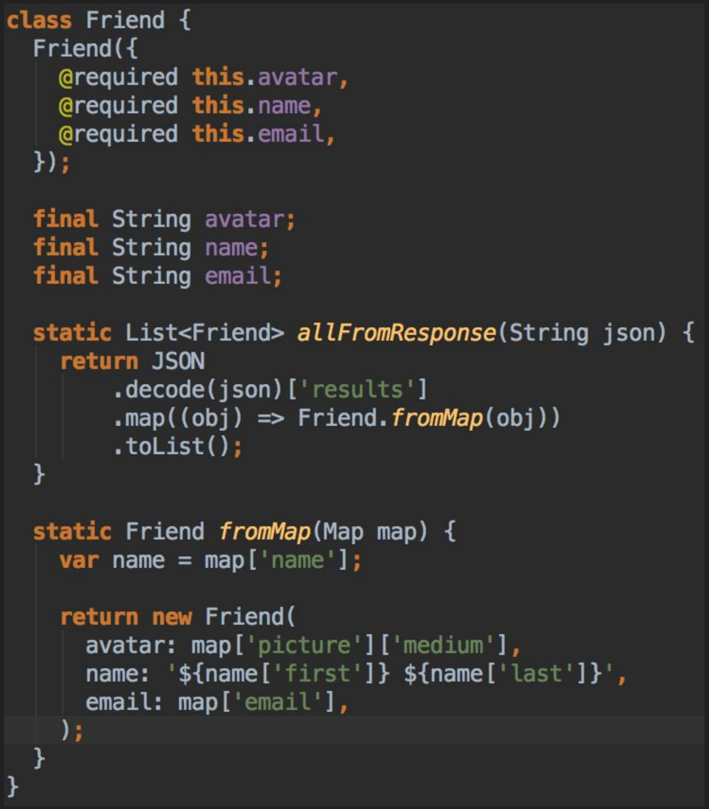
- Column is a vertical stack of children.
  - Opposite of Row, which stacks its children horizontally
- We give padding to the Text widget by wrapping it inside the Padding widget
- Styles come from the app level Theme object, so the whole theme of the app can be easily changed
  - You have the freedom to define your own styles inline too



```
class HelloWorldPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var textTheme = Theme.of(context).textTheme;

    return new Container(
      color: Colors.white,
      child: new Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          new FlutterLogo(size: 100.0),
          new Padding(
            padding: const EdgeInsets.only(top: 16.0),
            child: new Text('Hello World!', style: textTheme.headline),
          ),
        ],
      ),
    );
  }
}
```

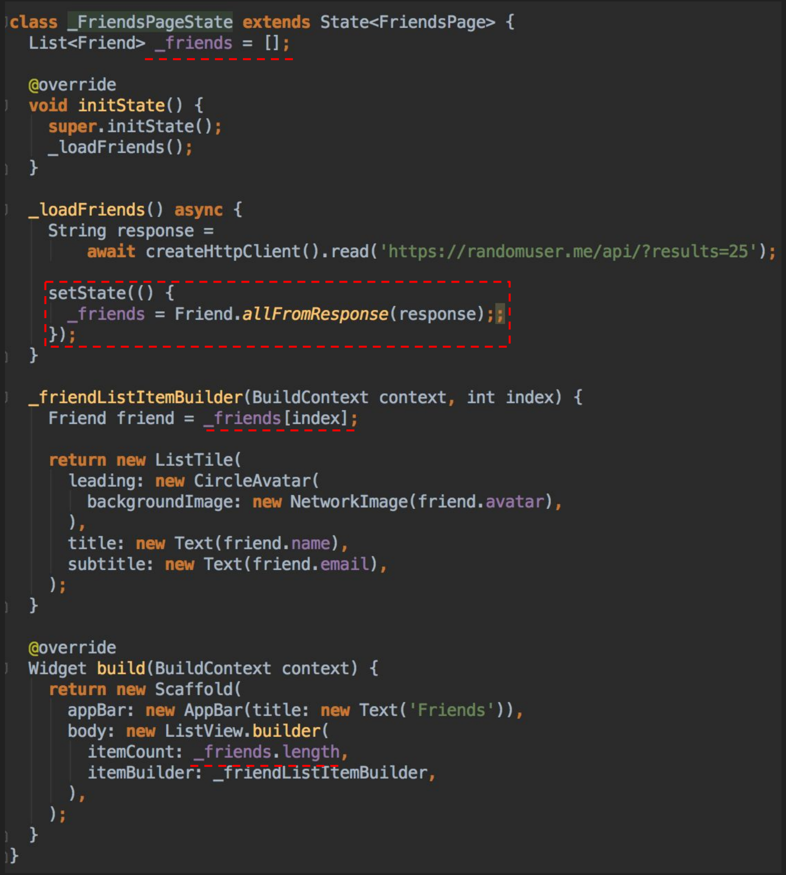
Let’s make some friends

friend.dart

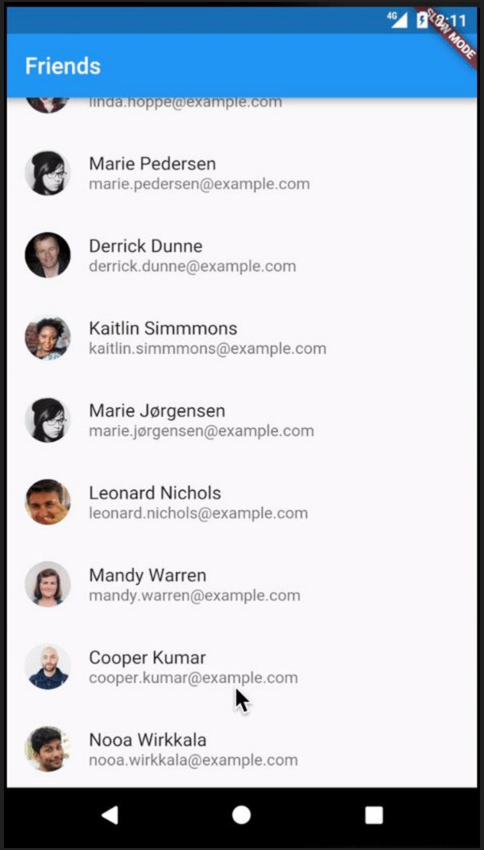


```
class Friend {
  Friend({
    @required this.avatar,
    @required this.name,
    @required this.email,
  });

  final String avatar;
  final String name;
  final String email;

  static List<Friend> allFromResponse(String json) {
    return JSON
        .decode(json)['results']
        .map((obj) => Friend.fromMap(obj))
        .toList();
  }

  static Friend fromMap(Map map) {
    var name = map['name'];

    return new Friend(
      avatar: map['picture']['medium'],
      name: '${name['first']} ${name['last']}',
      email: map['email'],
    );
  }
}
```

friend_page.dart



```
class _FriendsPageState extends State<FriendsPage> {
  List<Friend> _friends = [];

  @override
  void initState() {
    super.initState();
    _loadFriends();
  }

  _loadFriends() async {
    String response =
        await createHttpClient().read('https://randomuser.me/api/?results=25');

    setState(() {
      _friends = Friend.allFromResponse(response);
    });
  }

  _friendListItemBuilder(BuildContext context, int index) {
    Friend friend = _friends[index];

    return new ListTile(
      leading: new CircleAvatar(
        backgroundImage: new NetworkImage(friend.avatar),
      ),
      title: new Text(friend.name),
      subtitle: new Text(friend.email),
    );
  }

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: new AppBar(title: new Text('Friends')),
      body: new ListView.builder(
        itemCount: _friends.length,
        itemBuilder: _friendListItemBuilder,
      ),
    );
  }
}
```



Friends

linda.hoppe@example.com

Marie Pedersen
marie.pedersen@example.com

Derrick Dunne
derrick.dunne@example.com

Kaitlin Simmmons
kaitlin.simmmons@example.com

Marie Jørgensen
marie.jørgensen@example.com

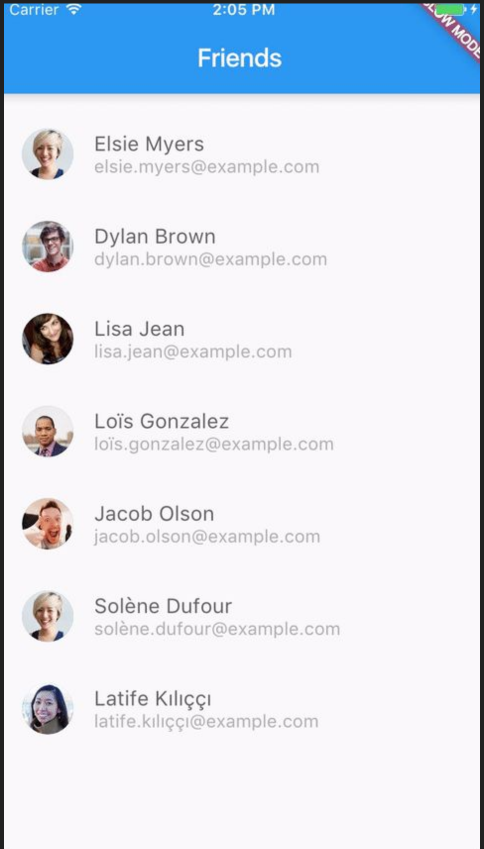
Leonard Nichols
leonard.nichols@example.com

Mandy Warren
mandy.warren@example.com

Cooper Kumar
cooper.kumar@example.com

Nooa Wirkkala
nooa.wirkkala@example.com

Android



Friends

Elsie Myers
elsie.myers@example.com

Dylan Brown
dylan.brown@example.com

Lisa Jean
lisa.jean@example.com

Loïs Gonzalez
loïs.gonzalez@example.com

Jacob Olson
jacob.olson@example.com

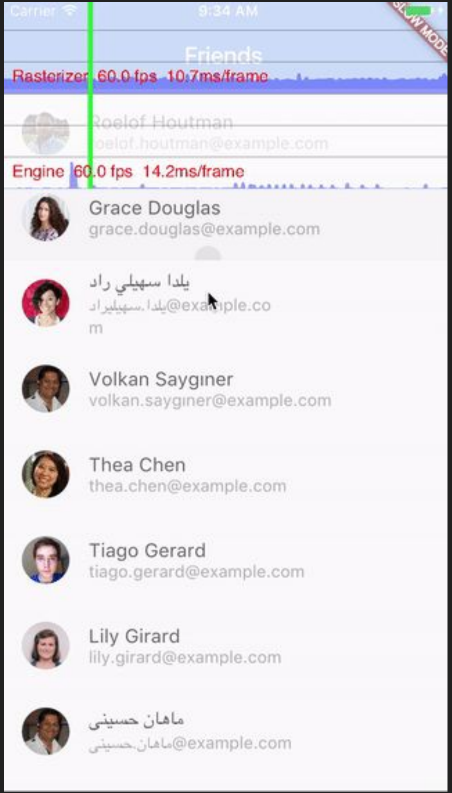
Solène Dufour
solène.dufour@example.com

Latife Kılıççı
latife.kılıççı@example.com

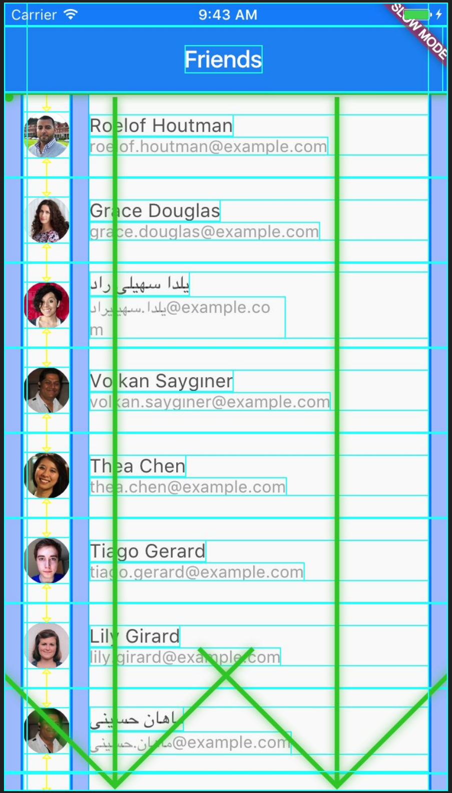
iOS

Debug tools

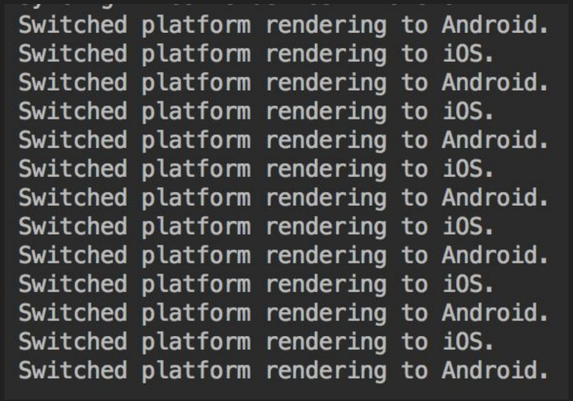
Performance overlay (GIF taken on slow debug mode)



Debug Paint



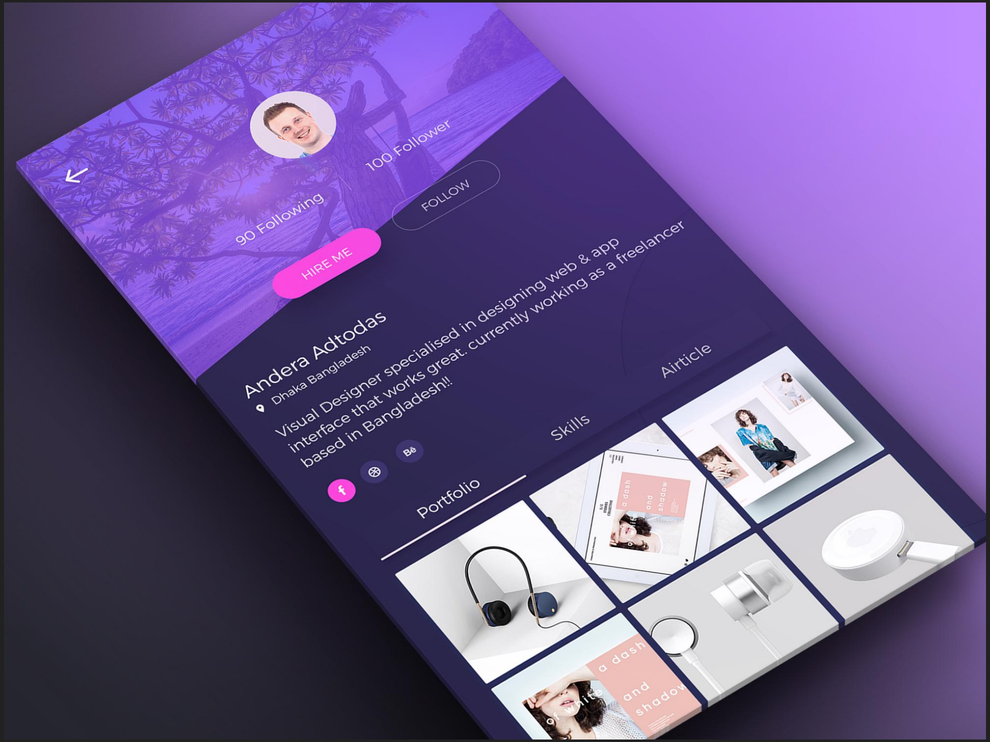
Quickly switch between Android / iOS UI on the same device



```
Switched platform rendering to Android.
Switched platform rendering to iOS.
Switched platform rendering to Android.
Switched platform rendering to iOS.
Switched platform rendering to Android.
Switched platform rendering to iOS.
Switched platform rendering to Android.
Switched platform rendering to iOS.
Switched platform rendering to Android.
Switched platform rendering to iOS.
Switched platform rendering to Android.
Switched platform rendering to iOS.
Switched platform rendering to Android.
```

Also debugger, logs and animation debugging.

Getting creative



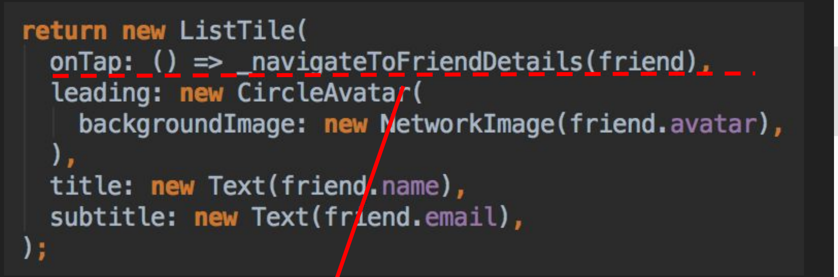
Andera Adtodas

Dhaka Bangladesh

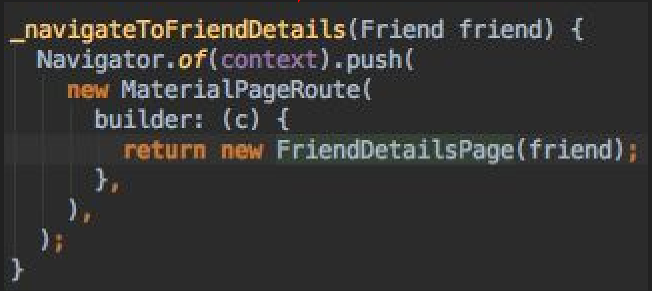
Visual Designer specialised in designing web & app interface that works great. currently working as a freelancer based in Bangladesh!!

https://www.uplabs.com/posts/profile-ui-exploration

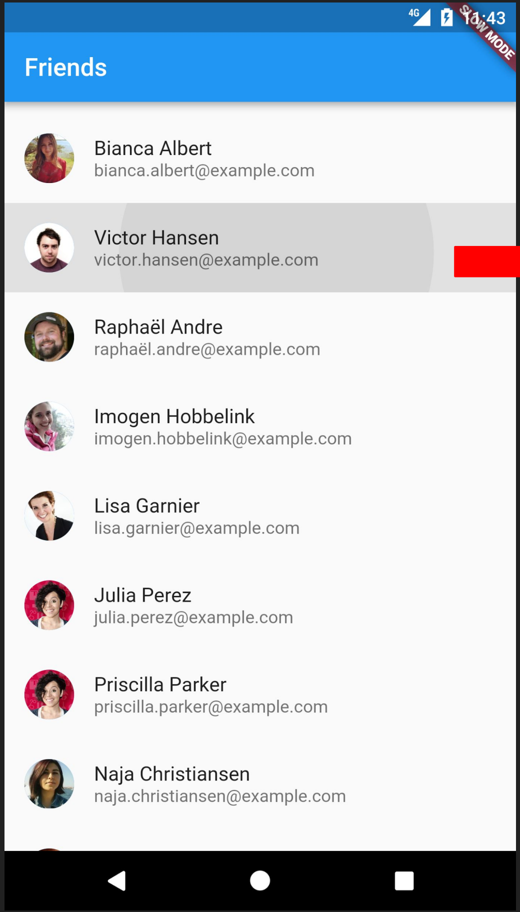
Navigation



```
return new ListTile(
  onTap: () => _navigateToFriendDetails(friend),
  leading: new CircleAvatar(
    backgroundImage: new NetworkImage(friend.avatar),
  ),
  title: new Text(friend.name),
  subtitle: new Text(friend.email),
);
```



```
_navigateToFriendDetails(Friend friend) {
  Navigator.of(context).push(
    new MaterialPageRoute(
      builder: (c) {
        return new FriendDetailsPage(friend);
      },
    ),
  );
}
```



Friends

Bianca Albert
bianca.albert@example.com

Victor Hansen
victor.hansen@example.com

Raphaël Andre
raphaël.andre@example.com

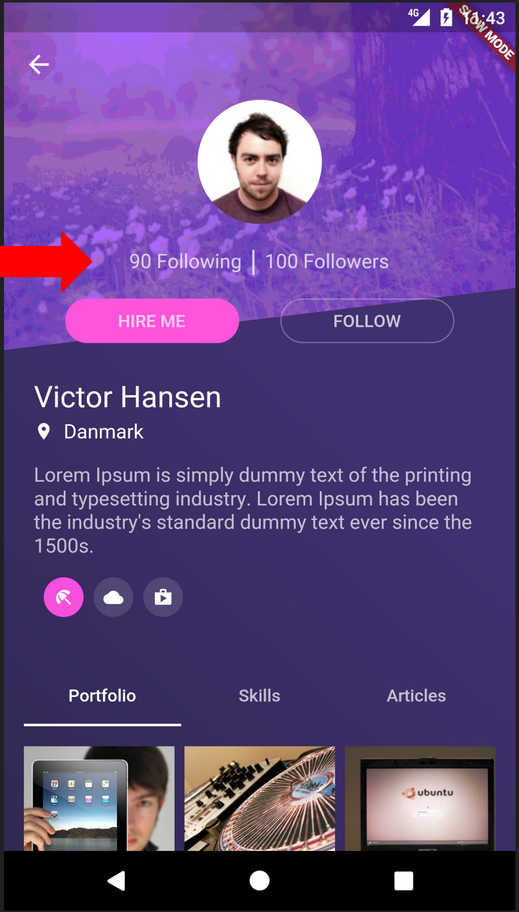
Imogen Hobbelink
imogen.hobbelink@example.com

Lisa Garnier
lisa.garnier@example.com

Julia Perez
julia.perez@example.com

Priscilla Parker
priscilla.parker@example.com

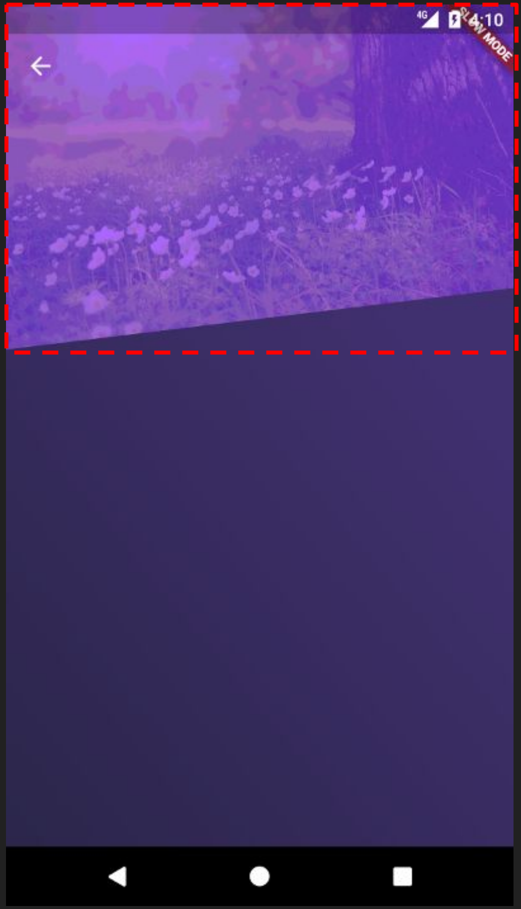
Naja Christiansen
naja.christiansen@example.com

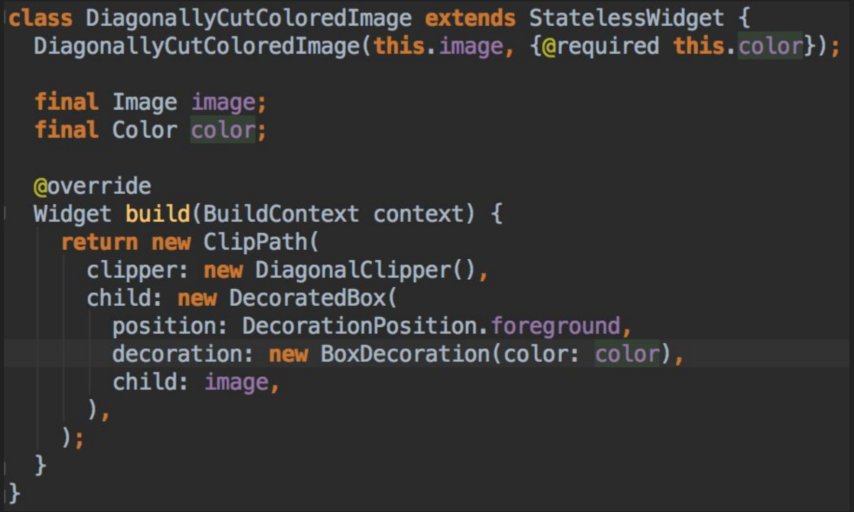


Victor Hansen

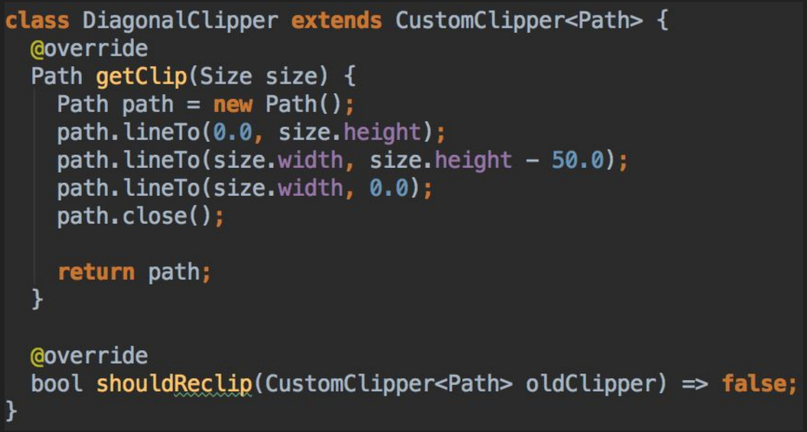
Danmark

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.

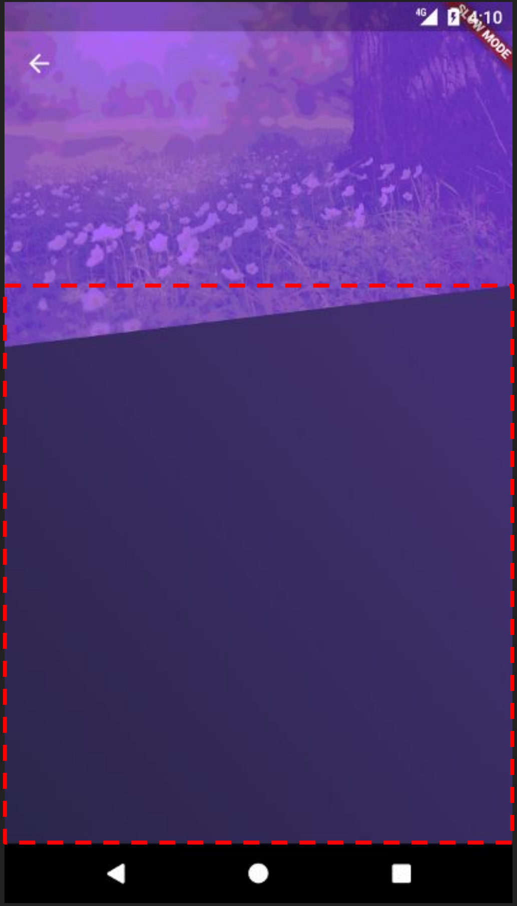


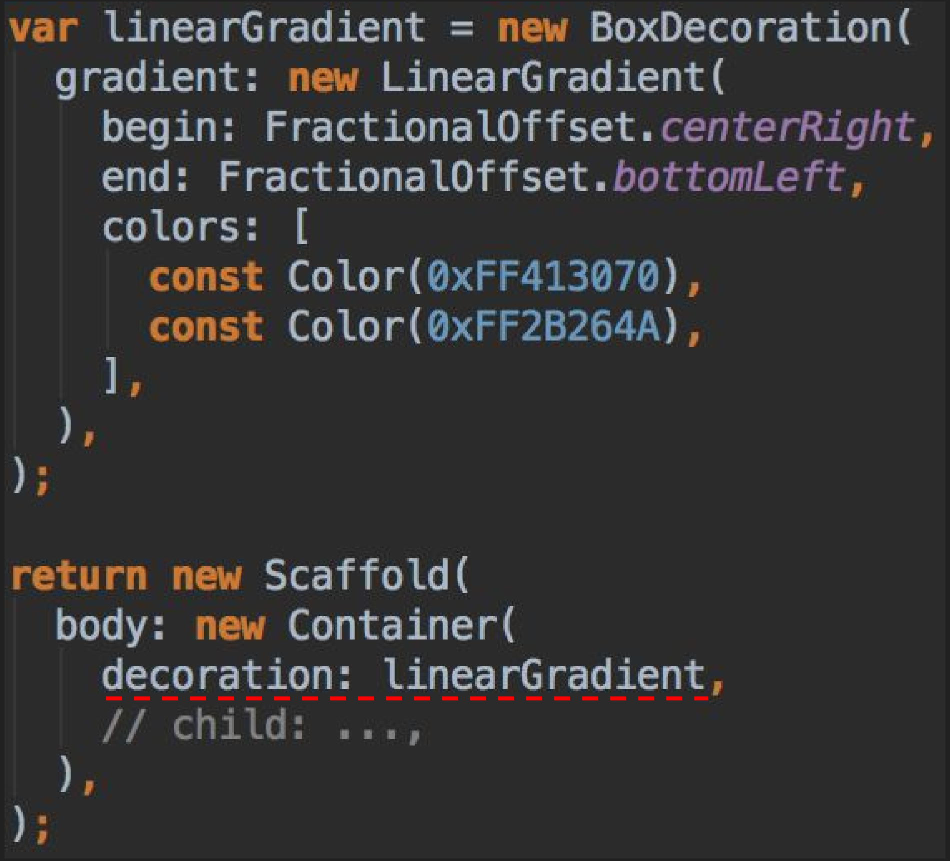


```
class DiagonallyCutColoredImage extends StatelessWidget {
  DiagonallyCutColoredImage(this.image, {@required this.color});

  final Image image;
  final Color color;

  @override
  Widget build(BuildContext context) {
    return new ClipPath(
      clipper: new DiagonalClipper(),
      child: new DecoratedBox(
        position: DecorationPosition.foreground,
        decoration: new BoxDecoration(color: color),
        child: image,
      ),
    );
  }
}
```

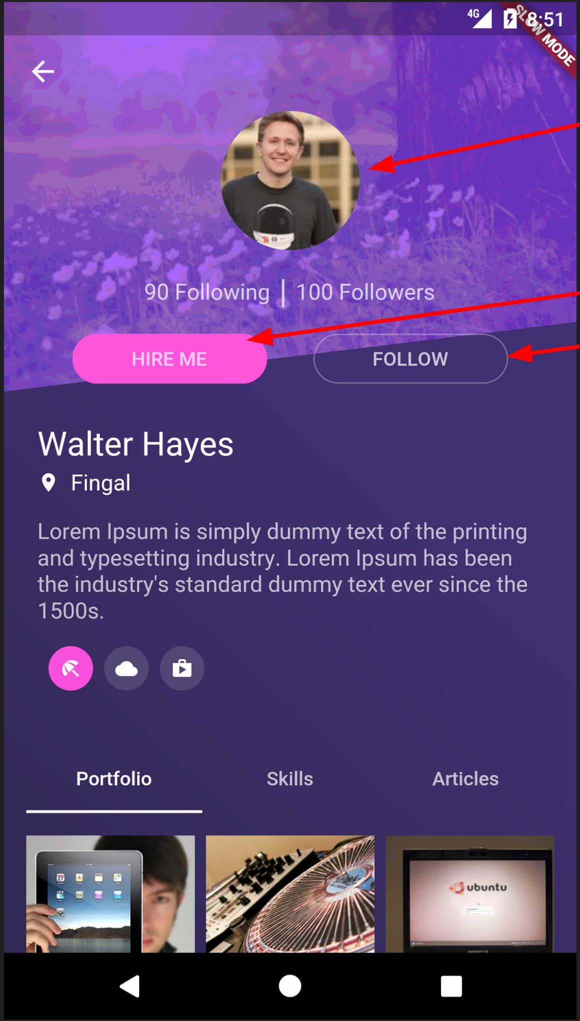


```
class DiagonalClipper extends CustomClipper<Path> {
  @override
  Path getClip(Size size) {
    Path path = new Path();
    path.lineTo(0.0, size.height);
    path.lineTo(size.width, size.height - 50.0);
    path.lineTo(size.width, 0.0);
    path.close();

    return path;
  }

  @override
  bool shouldReclip(CustomClipper<Path> oldClipper) => false;
}
```





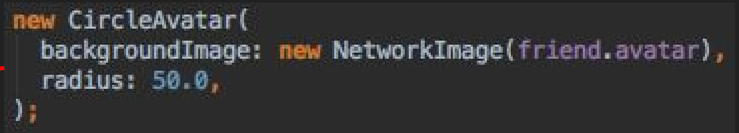
```
var linearGradient = new BoxDecoration(
  gradient: new LinearGradient(
    begin: FractionalOffset.centerRight,
    end: FractionalOffset.bottomLeft,
    colors: [
      const Color(0xFF413070),
      const Color(0xFF2B264A),
    ],
  ),
);

return new Scaffold(
  body: new Container(
    decoration: linearGradient,
    // child: ...,
  ),
);
```



Walter Hayes

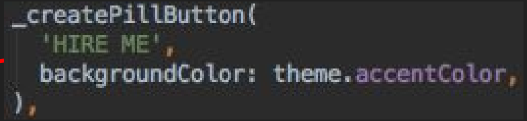
Fingal

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.

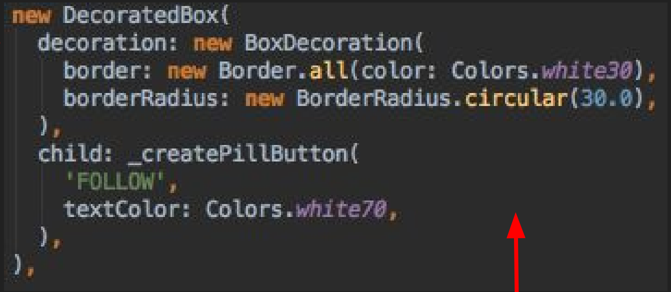


```
new CircleAvatar(
  backgroundImage: new NetworkImage(friend.avatar),
  radius: 50.0,
);
```

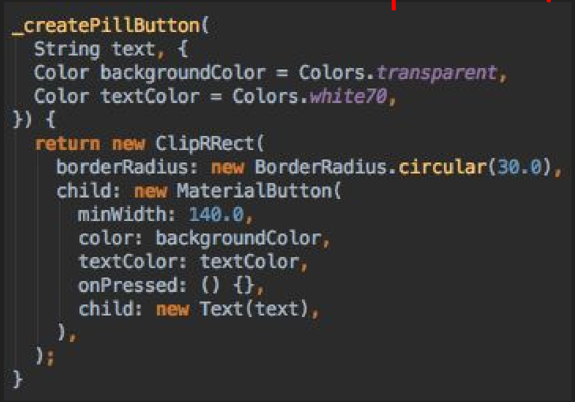
= avatarUrl



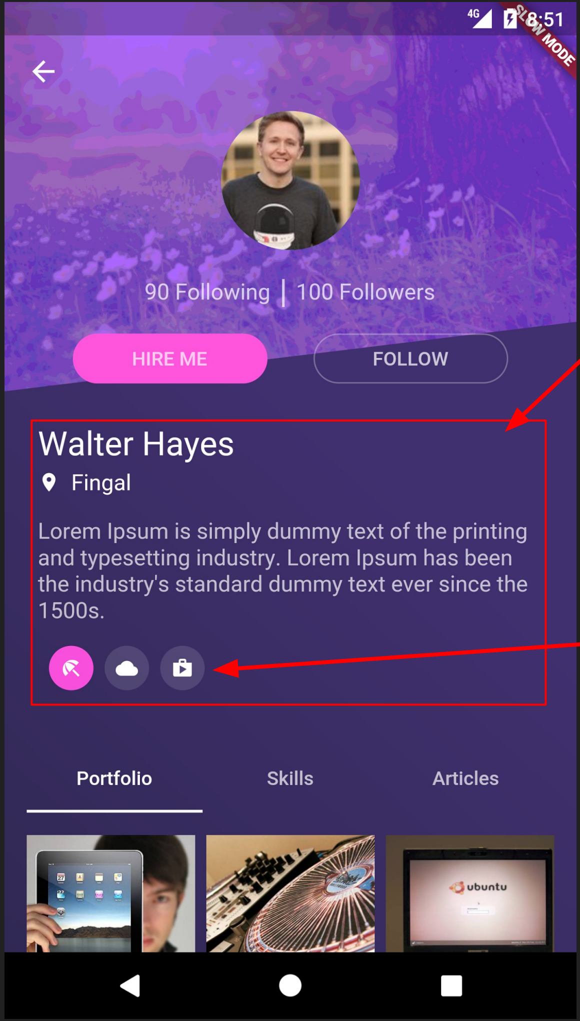
```
_createPillButton(
  'HIRE ME',
  backgroundColor: theme.accentColor,
),
```



```
new DecoratedBox(
  decoration: new BoxDecoration(
    border: new Border.all(color: Colors.white30),
    borderRadius: new BorderRadius.circular(30.0),
  ),
  child: _createPillButton(
    'FOLLOW',
    textColor: Colors.white70,
  ),
),
```



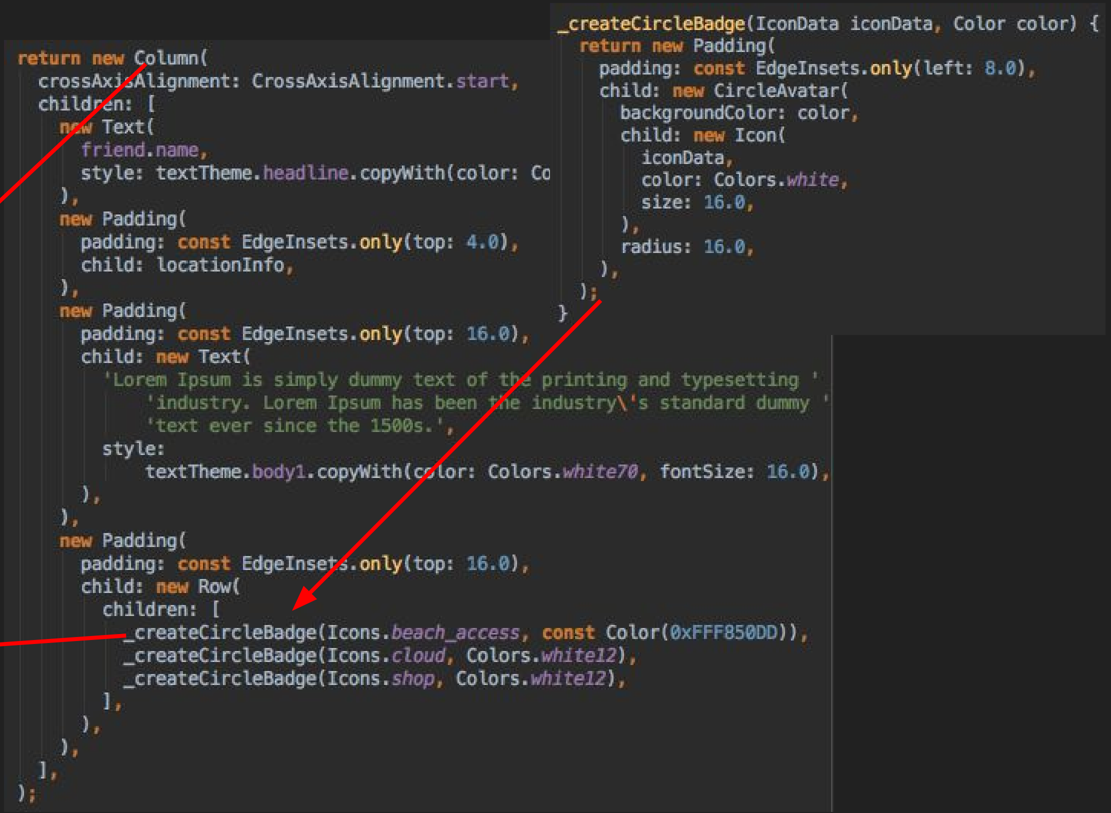
```
_createPillButton(
  String text, {
  Color backgroundColor = Colors.transparent,
  Color textColor = Colors.white70,
}) {
  return new ClipRRect(
    borderRadius: new BorderRadius.circular(30.0),
    child: new MaterialButton(
      minWidth: 140.0,
      color: backgroundColor,
      textColor: textColor,
      onPressed: () {},
      child: new Text(text),
    ),
  );
}
```



Walter Hayes

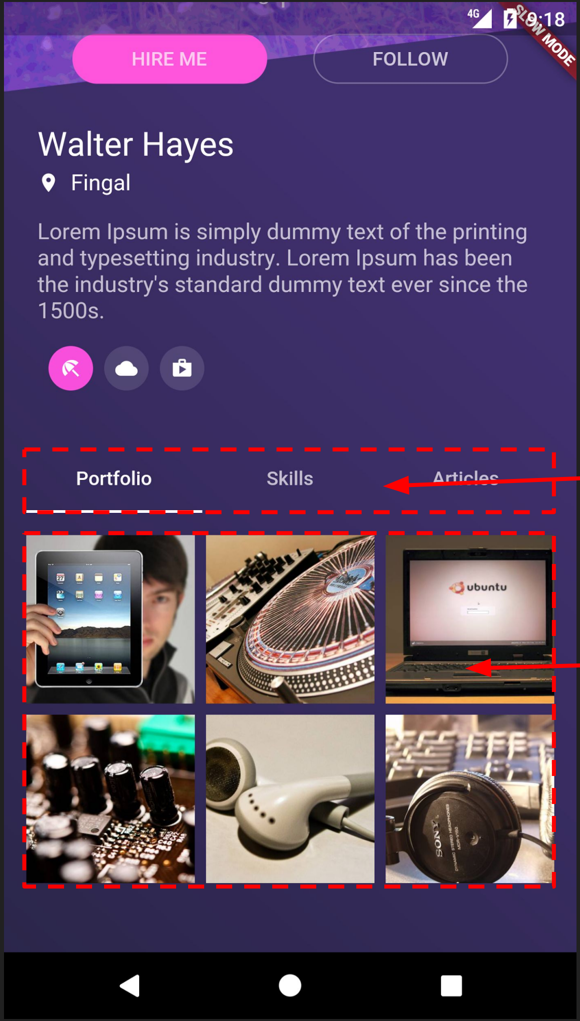
Fingal

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.



```
return new Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    new Text(
      friend.name,
      style: textTheme.headline.copyWith(color: Co
    ),
    new Padding(
      padding: const EdgeInsets.only(top: 4.0),
      child: locationInfo,
    ),
    new Padding(
      padding: const EdgeInsets.only(top: 16.0),
      child: new Text(
        'Lorem Ipsum is simply dummy text of the printing and typesetting '
            'industry. Lorem Ipsum has been the industry\'s standard dummy '
            'text ever since the 1500s.',
        style:
            textTheme.body1.copyWith(color: Colors.white70, fontSize: 16.0),
      ),
    ),
    new Padding(
      padding: const EdgeInsets.only(top: 16.0),
      child: new Row(
        children: [
          _createCircleBadge(Icons.beach_access, const Color(0xFFF850DD)),
          _createCircleBadge(Icons.cloud, Colors.white12),
          _createCircleBadge(Icons.shop, Colors.white12),
        ],
      ),
    ),
  ],
);
```

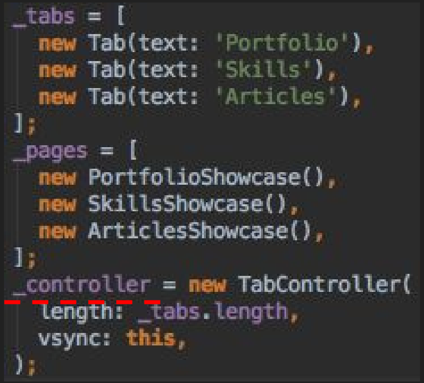
```
_createCircleBadge(IconData iconData, Color color) {
  return new Padding(
    padding: const EdgeInsets.only(left: 8.0),
    child: new CircleAvatar(
      backgroundColor: color,
      child: new Icon(
        iconData,
        color: Colors.white,
        size: 16.0,
      ),
      radius: 16.0,
    ),
  );
}
```



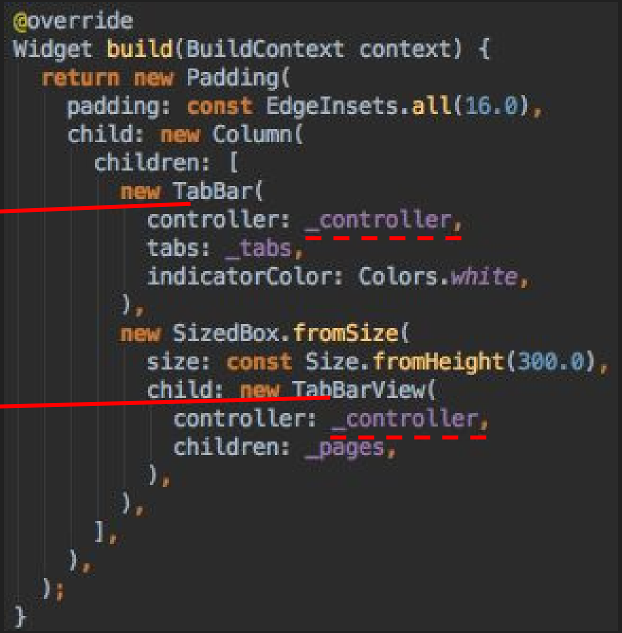
Walter Hayes

Fingal

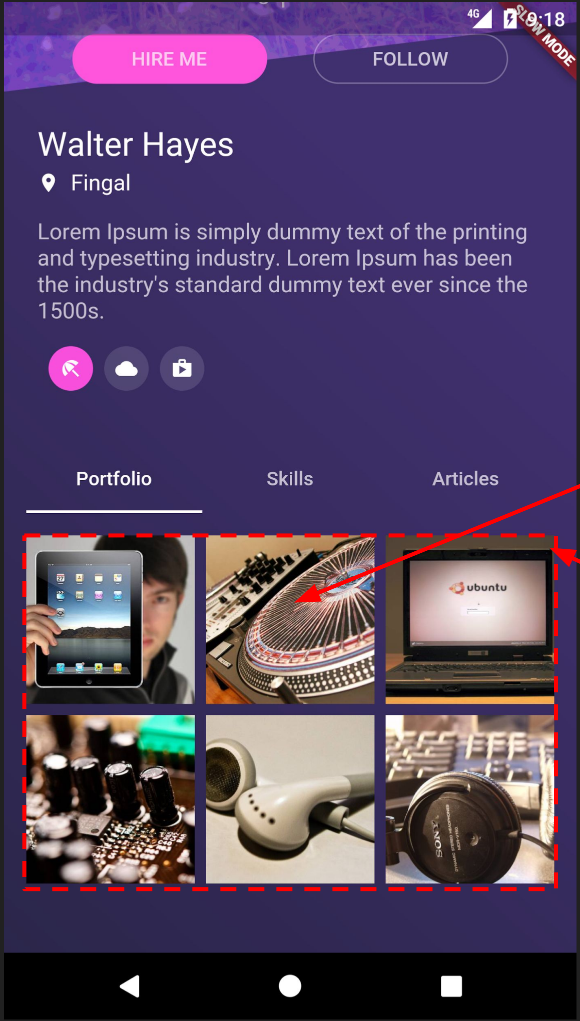
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.



```
_tabs = [
  new Tab(text: 'Portfolio'),
  new Tab(text: 'Skills'),
  new Tab(text: 'Articles'),
];
_pages = [
  new PortfolioShowcase(),
  new SkillsShowcase(),
  new ArticlesShowcase(),
];
_controller = new TabController(
  length: _tabs.length,
  vsync: this,
);
```



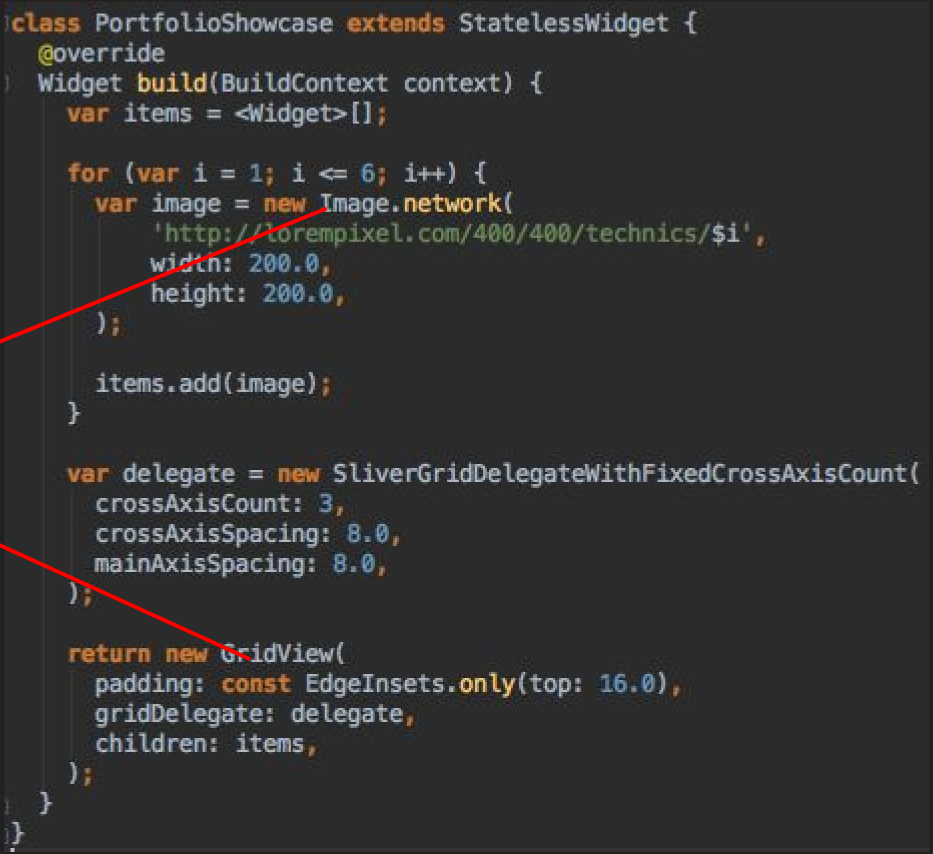
```
@override
Widget build(BuildContext context) {
  return new Padding(
    padding: const EdgeInsets.all(16.0),
    child: new Column(
      children: [
        new TabBar(
          controller: _controller,
          tabs: _tabs,
          indicatorColor: Colors.white,
        ),
        new SizedBox.fromSize(
          size: const Size.fromHeight(300.0),
          child: new TabBarView(
            controller: _controller,
            children: _pages,
          ),
        ),
      ],
    ),
  );
}
```



Walter Hayes

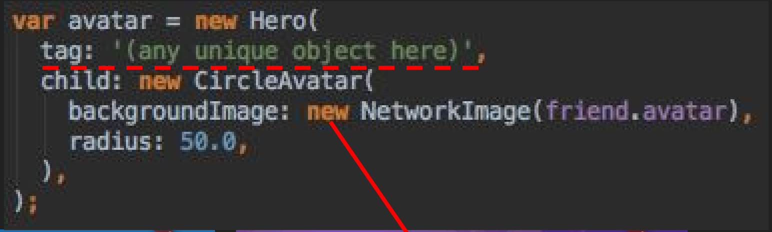
Fingal

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.

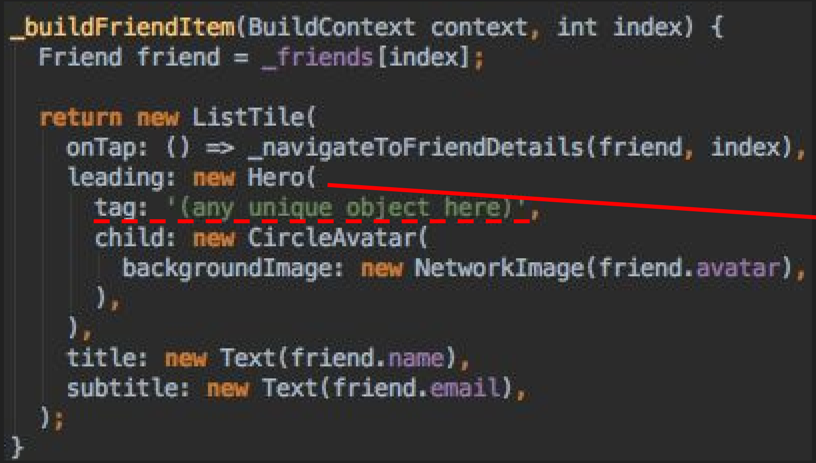


```
class PortfolioShowcase extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var items = <Widget>[];

    for (var i = 1; i <= 6; i++) {
      var image = new Image.network(
          'http://lorempixel.com/400/400/technics/$i',
          width: 200.0,
          height: 200.0,
      );

      items.add(image);
    }

    var delegate = new SliverGridDelegateWithFixedCrossAxisCount(
      crossAxisCount: 3,
      crossAxisSpacing: 8.0,
      mainAxisSpacing: 8.0,
    );

    return new GridView(
      padding: const EdgeInsets.only(top: 16.0),
      gridDelegate: delegate,
      children: items,
    );
  }
}
```

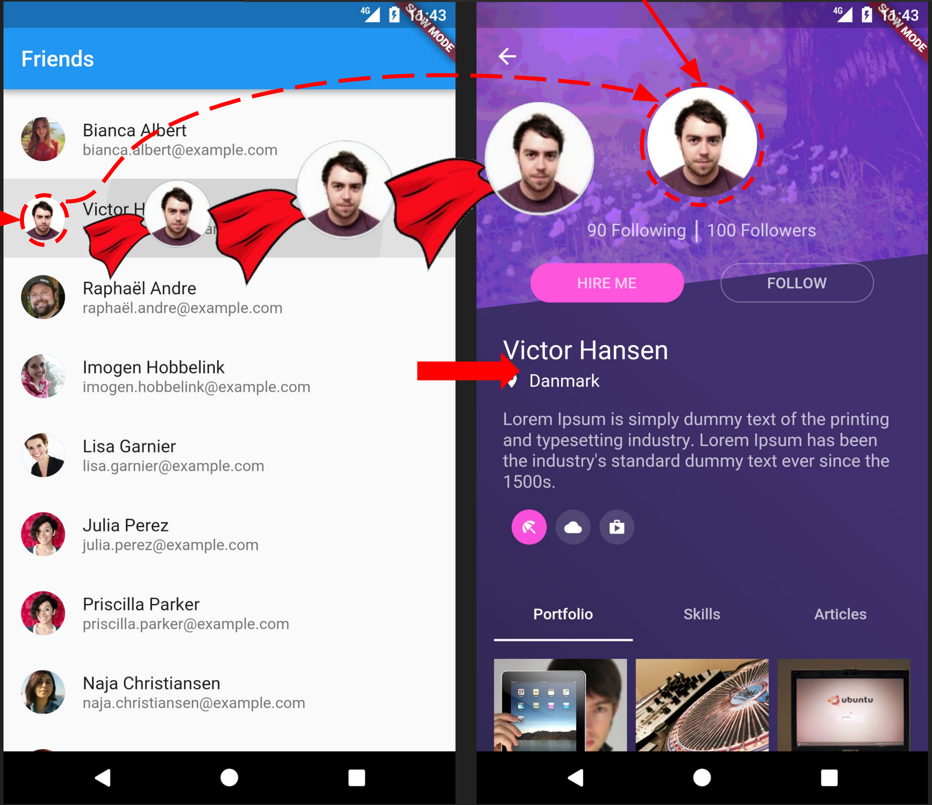
The Hero Widget



```
var avatar = new Hero(
  tag: '(any unique object here)',
  child: new CircleAvatar(
    backgroundImage: new NetworkImage(friend.avatar),
    radius: 50.0,
  ),
);
```



```
_buildFriendItem(BuildContext context, int index) {
  Friend friend = _friends[index];

  return new ListTile(
    onTap: () => _navigateToFriendDetails(friend, index),
    leading: new Hero(
      tag: '(any unique object here)',
      child: new CircleAvatar(
        backgroundImage: new NetworkImage(friend.avatar),
      ),
    ),
    title: new Text(friend.name),
    subtitle: new Text(friend.email),
  );
}
```



Friends

Bianca Albert
bianca.albert@example.com

Victor H

Raphaël Andre
raphaël.andre@example.com

Imogen Hobbelink
imogen.hobbelink@example.com

Lisa Garnier
lisa.garnier@example.com

Julia Perez
julia.perez@example.com

Priscilla Parker
priscilla.parker@example.com

Naja Christiansen
naja.christiansen@example.com

90 Following | 100 Followers

Victor Hansen
Danmark

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.

Demo & source

https://github.com/CodemateLtd/FlutterMates

Drawbacks

- UI markup & layout system learning curve
  - UI code can look quite ugly
- Not a lot of “hold your hand guides” available
  - Documentation is amazing
- Google’s product loyalty
- Inline maps & video, etc. can’t be done (at least yet)
  - Possible on full screen though

However...

- Hot reload
- AOT compilation & direct canvas rendering for widgets -> amazing performance
- UI will work the same on different devices & manufacturers
  - Native look and feel on Android & iOS
  - Nobody can break our UIs
- You can create as complex and customized UIs as you want

Thanks!

Questions?

Inspiration & references

- Flutter - A new hope: https://www.youtube.com/watch?v=0ijVuVtu6a4
- Flutter - 60 FPS UI of the Future: https://speakerdeck.com/albrecht87/flutter-60-fps-ui-of-the-future-droidcon-2017
- The Official Flutter homepage: https://flutter.io/