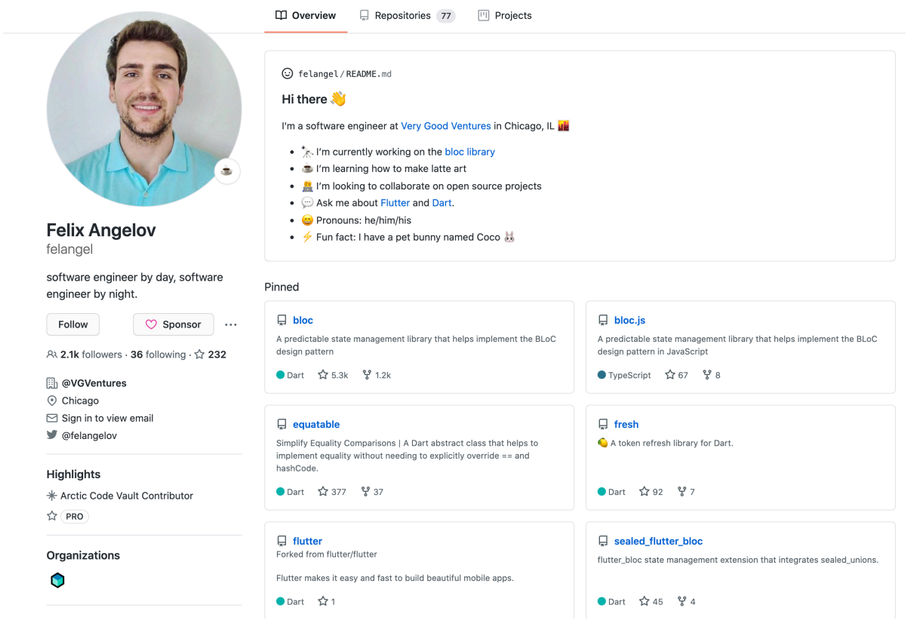
Cubit and Beyond 🚀

Still the new kid on the bloc

Felix Angelov @ Very Good Ventures

Very Good Ventures, Chicago 🕶️🦄



https://verygood.ventures

Agenda

- What's new in v6.1.0
- Meet Cubit
- Cubit vs. Bloc
- Live Coding

What's New?

context.watch

```
@override
Widget build(BuildContext context) {
  final state = context.watch<MyBloc>().state;
  return Text('$state');
}
```

- Rebuilds when the state of `MyBloc` changes.
- Only allowed directly in `build`.
- Equivalent to top level BlocBuilder.

context.watch basics

```
// 😊
@override
Widget build(BuildContext context) {
  return BlocBuilder<MyBloc, MyState>(
    builder: (context, state) => Text('$state'),
  );
}

// 😎
@override
Widget build(BuildContext context) {
  final state = context.watch<MyBloc>().state;
  return Text('$state');
}
```

context.watch gotchas

```
// 😬
@override
Widget build(BuildContext context) {
  final state = context.watch<MyBloc>().state;
  return Scaffold(
    appBar: ...
    body: Text('$state'),
    floatingActionButton: ...
  );
}
```

```
// 😎
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: ...
    body: Builder(
      builder: (context, state) {
        final state = context.watch<MyBloc>().state;
        return Text('$state');
      },
    ),
    floatingActionButton: ...
  );
}
```

context.watch recap

```dart
// BEFORE (bad) 😬
@override
Widget build(BuildContext context) {
  final state = context.bloc<MyBloc>().state;
  return Text('$state');
}
```

```dart
// NOW (good) 😊
@override
Widget build(BuildContext context) {
  return BlocBuilder<MyBloc, MyState>(
    builder: (context, state) => Text('$state'),
  );
}
```

```dart
// NOW (good) 😎
@override
Widget build(BuildContext context) {
  final state = context.watch<MyBloc>().state;
  return Text('$state');
}
```

Closed **malithkuruppu** opened this issue on Sep 13 · 1 comment

malithkuruppu commented on Sep 13

Are there any plans to implement `MultiBlocBuilder` and `MultiBl`

[Feature Request] MultiBlocBuilder #538

Closed **pczn0327** opened this issue on Sep 25, 2019 · 67 comments

pczn0327 commented on Sep 25, 2019

Sometimes I would need to access different blocs in the same widget, and the code goes pretty messy. I am thinking maybe we can have something like `MultiBlocBuilder` which serve the similar purpose of `MultiRepositoryProvider` and `MultiBlocProvider`.

13 4 1

ne commented on Jun 27

`iBlocBuilder` like `MultiBlocProvider`.

MultiBlocBuilder #42

Closed **amreniouinnove**

BlocBuilder nest hell #714

Closed **ffpetrovic** opened this issue on Dec 9, 2019 · 6 comment

ffpetrovic commented on Dec 9, 2019 • edited

Is your feature request related to a prob

bloclibrary.dev shows that we are supp
app. Consider blocs `Bloc1` `Bloc2` `Bloc3`
pattern is to have a `...Loading` `...Loaded`
want to check whether ANY of these blocs are it,
even 4) for every Bloc (because, honestly, I feel like
use case for me), would be really appreciated. Sorry if

Describe alternatives you've considered

Nesting BlocBuilders in the root of the build method.

amreniouinnovent c

Is your feature request

I want better readability fo

MultiBlocBuilder? #841

Closed **edsonbonfim** opened this issue on Feb 2 · 6 comments

edsonbonfim commented on Feb 2

A case of use when this feature is helpful is when we have a widget that responses at two or more independents external events. In this case, the widget is wrapped with two or more bloc builder that managed this events. Nested blocs can't be a really problem, but is expressive to write, like that as create another bloc which has dependencies on the two blocs.

A example where this feature can be helpful.

```
return BlocBuilder<BlocA, bool>(
  builder: (context, stateA) {
    return BlocBuilder<BlocB, int>(
      builder: (context, stateB) {
        if (stateA || stateB == 1) {
          // do something
        }
        // do another thing
      }
    );
  },
);
```

@felangelov - Flutter Vikings 2020

Some help?

Nested bloc builders in flutter

Closed **francipvb** opened this issue on Aug 20 · 2 comments

francipvb commented on Aug 20

Hello,

Can I construct a nested bloc builder widget tree? What I have to be conscerned about?

Furter explanation

I have a list loaded in a bloc state, but I want to track the changes for the items individually and avoid any unnecessary rebuild of the parent bloc builder. I want it to be rebuilt only if the length of the list changes. At the deeper level, I want to rebuild the list item when the individual item changes it's vaue.

Is it possible to do without major isues?

Thanks,

MultiBlocBuilder?

MultiBlocBuilder and MultiBlocConsumer #1724

Closed malithkuruppu opened this issue on Sep 13 · 1 comment

malithkuruppu commented on Sep 13

Are there any plans to implement `MultiBlocBuilder` and `MultiBl`

Adding multiBlocBuilder interface #1347

mohammadne opened this issue on Jun 27 · 1 comment

[Feature Request] MultiBlocBuilder #538

Closed pczn0327 opened this issue on Sep 25, 2019 · 67 comments

pczn0327 commented on Sep 25, 2019

Sometimes I would need to access different blocs in the same widget, and the code goes pretty messy. I am thinking maybe we can have something like `MultiBlocBuilder` which can serve the similar purpose of `MultiRepositoryProvider` and

`iBlocBuilder` like `MultiBlocProvider`.

MultiBlocBuilder #42

Closed amreniouinnove

BlocBuilder nest hell #714

Closed ffpetrovic opened this issue on Dec 9, 2019 · 6 comment

ffpetrovic commented on Dec 9, 2019 · edited

Is your feature request related to a pro

blocilibrary.dev shows that we are su

app. Consider blocs `Bloc1` `Bloc2` `Bloc`

pattern is to have a `...Loading` `...Loaded`

want to check whether ANY of these blocs are in

even 4) for every Bloc (because, honestly, I feel like

use case for me), would be really appreciated. Sorry if

Describe alternatives you've considered

Nesting BlocBuilders in the root of the build method.

amreniouinnovent c

Is your feature request

I want better readability fo

MultiBlocBuilder? #841

Closed edsonbonfim opened this issue on Feb 2 · 6 comments

edsonbonfim commented on Feb 2

A case of use when this feature is helpful is when we have a widget that responses at two or more independents external events. In this case, the widget is wrapped with two or more bloc builder that managed this events. Nested blocs can't be a really problem, but is expressive to write, like that as create another bloc which has dependencies on the two blocs.

A example where this feature can be helpful.

```
return BlocBuilder<BlocA, bool>(
  builder: (context, stateA) {
    return BlocBuilder<BlocB, int>(
      builder: (context, stateB) {
        if (stateA || stateB == 1) {
          // do something
        }
        // do another thing
      }
    );
  },
);
```

Thanks for this unbelievable lib!!!

Ne flutter

Closed francipvb opened this issue on Aug 20 · 2 comments

francipvb commented on Aug 20

Hello,

Can I construct a nested bloc builder widget tree? What I have to be conscerned about?

Furter explanation

I have a list loaded in a bloc state, but I want to track the changes for the items individually and avoid any unnecessary rebuild of the parent bloc builder. I want it to be rebuilt only if the length of the list changes. At the deeper level, I want to rebuild the list item when the individual item changes it's vaue.

Is it possible to do without major isues?

Thanks,

MultiBlocBuilder 🎉 (sort of)

```
Builder(
  builder: (context) {
    final stateA = context.watch<BlocA>().state;
    final stateB = context.watch<BlocB>().state;
    final stateC = context.watch<BlocC>().state;

    // return a Widget which depends on the state of BlocA, BlocB, and BlocC
  }
);
```

context.read

```
@override
Widget build(BuildContext context) {
  return RaisedButton(
    onPressed: () {
      context.read<MyBloc>().add(MyEvent());
    },
    ...
  )
}
```

- Does not rebuild when the state of `MyBloc` changes.
- Not allowed directly in `build`.

context.read recap

```
// BEFORE (bad) 😬
@override
Widget build(BuildContext context) {
  final bloc = context.bloc<MyBloc>();
  return RaisedButton(
    onPressed: () {
      bloc.add(MyEvent());
    },
    ...
  )
}
```

```
// NOW 😎
@override
Widget build(BuildContext context) {
  return RaisedButton(
    onPressed: () {
      context.read<MyBloc>().add(MyEvent());
    },
    ...
  )
}
```

context.select

```
@override
Widget build(BuildContext context) {
  final name = context.select(
    (UserBloc bloc) => bloc.state.user.name,
  );
  return Text('Hello $name!');
}
```

- Rebuild in response to part of a bloc's state.

Bloc Test Seeding State

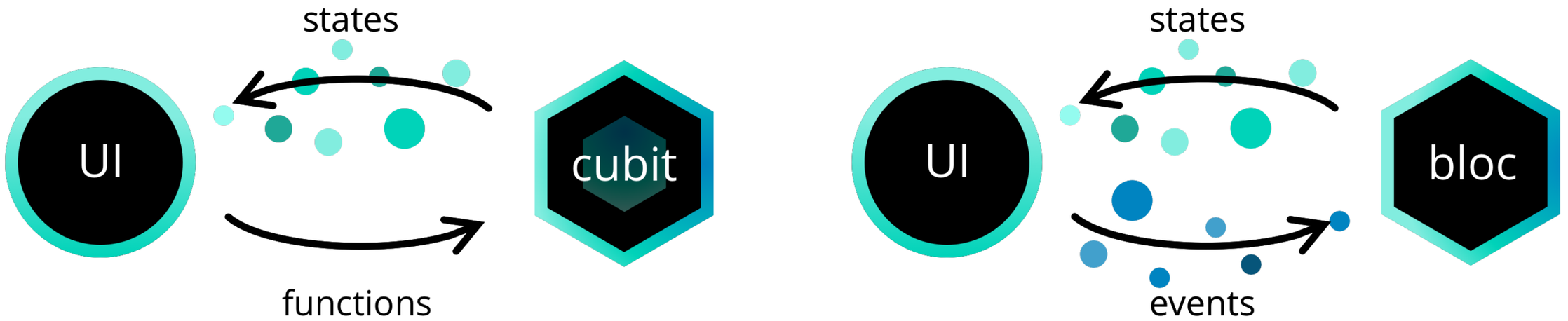
```
// BEFORE 😬
blocTest<CounterBloc, int>(
  'emits [1, 2] when increment is called and state is 1',
  build: () => CounterBloc(),
  act: (bloc) {
    bloc
      ..add(CounterEvent.increment)
      ..add(CounterEvent.increment);
  },
  expect: const <int>[1, 2],
);

// NOW 😎
blocTest<CounterBloc, int>(
  'emits [2] when increment is called and state is 1',
  build: () => CounterBloc(),
  seed: 1,
  act: (bloc) => bloc.add(CounterEvent.increment),
  expect: const <int>[2],
);
```

Complete migration guide

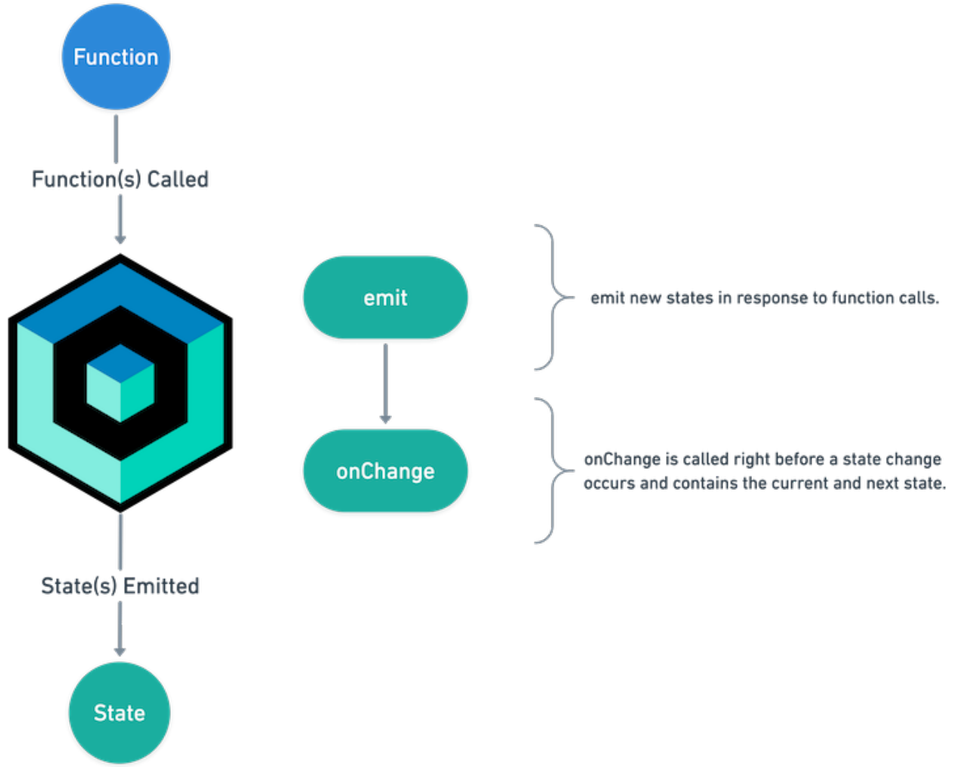
https://bloclibrary.dev/#/migration

Meet Cubit

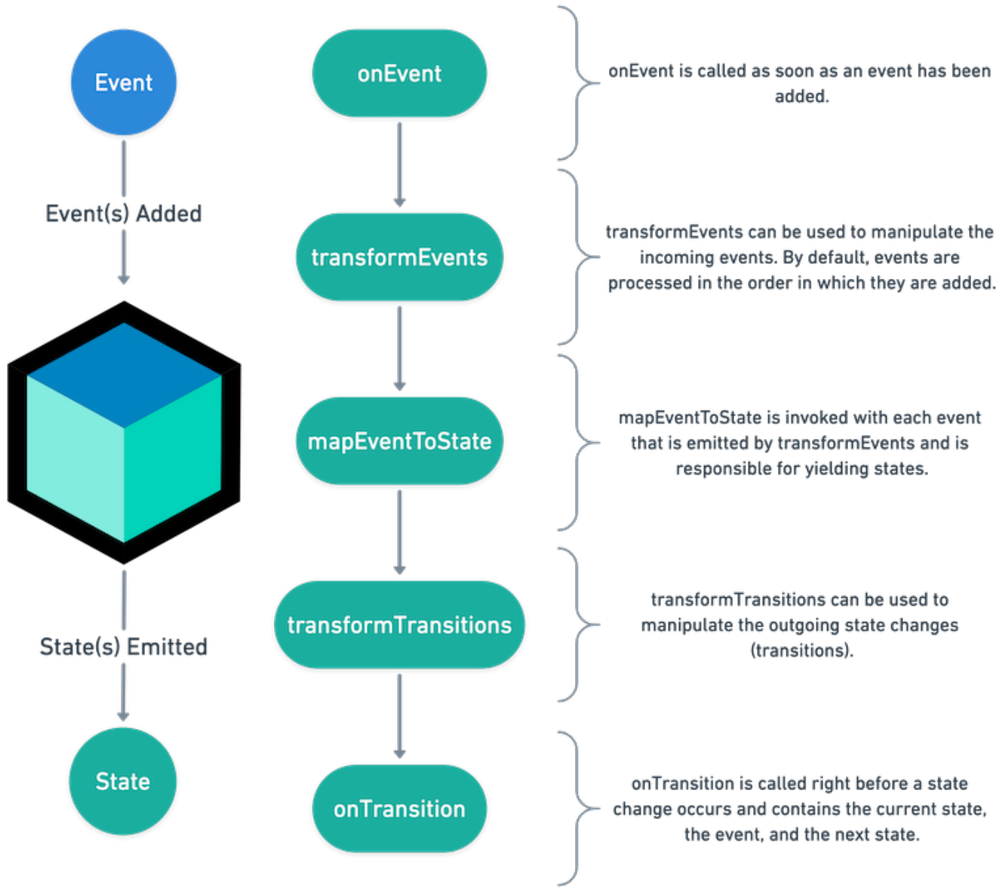
Cubit = Bloc - Events



Cubit


Function
Function(s) Called
State(s) Emitted
State
emit
onChange
emit new states in response to function calls.
onChange is called right before a state change occurs and contains the current and next state.

Bloc


Event
Event(s) Added
State(s) Emitted
State
onEvent
transformEvents
mapEventToState
transformTransitions
onTransition
onEvent is called as soon as an event has been added.
transformEvents can be used to manipulate the incoming events. By default, events are processed in the order in which they are added.
mapEventToState is invoked with each event that is emitted by transformEvents and is responsible for yielding states.
transformTransitions can be used to manipulate the outgoing state changes (transitions).
onTransition is called right before a state change occurs and contains the current state, the event, and the next state.

CounterCubit

```dart
import 'package:bloc/bloc.dart';

class CounterCubit extends Cubit<int> {
  CounterCubit() : super(0);

  void increment() => emit(state + 1);
}
```

Cubit vs. Bloc

Cubit

```
class CounterCubit extends Cubit<int> {
  CounterCubit() : super(0);

  void increment() => emit(state + 1);
}
```

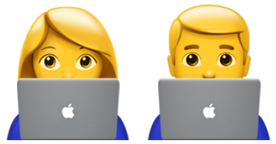
Bloc

```
enum CounterEvent { increment }

class CounterBloc extends Bloc<CounterEvent, int> {
  CounterBloc() : super(0);

  @override
  Stream<int> mapEventToState(CounterEvent event) async* {
    switch (event) {
      case CounterEvent.increment:
        yield state + 1;
        break;
    }
  }
}
```

Cubit vs. Bloc

	Cubit	Bloc
Simple	✅	
Concise	✅	
Traceable		✅
ReactiveX Operators		✅
Testable	✅	✅
Scalable	✅	✅
Tooling Support	✅	✅

Live Coding



https://github.com/felangel/bloc/blob/master/packages/flutter_bloc/example

Thanks! 🙏

Twitter @felangelov

Github @felangel