

poc_bloc

A new Flutter project for testing [bloc package](#)

This is an exercise from the tutorial [Flutterly - Bloc from zero to hero](#)

Concepts

Cubit

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

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

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

Bloc

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

class CounterBloc extends Bloc<CounterEvent, int> {
  CounterBloc() : super(0) {
    on<CounterEvent>((event, emit) {
      if (event == CounterEvent.increment) {
        emit(state + 1);
      } else {
        emit(state - 1);
      }
    });
  }

  int _counter = 0;

  int get initialState => _counter;
}
```

StreamController

```

import 'dart:async';

enum CounterEvent { increment, decrement }

class CounterStreamController {
  StreamController<CounterEvent> eventController =
    StreamController<CounterEvent>();
  StreamController<int> stateController = StreamController<int>();

  int _state = 0;
  int get initialData => _state;
  Stream<int> get stream => stateController.stream;

  void addEvent(CounterEvent event) {
    eventController.add(event);
  }

  CounterStreamController() {
    eventController.stream.listen((event) {
      _handleEvent(event);
    });
  }

  _handleEvent(CounterEvent event) {
    switch (event) {
      case CounterEvent.increment:
        _state++;
        break;
      case CounterEvent.decrement:
        _state--;
        break;
    }

    stateController.add(_state);
  }
}

```

UI

```

StreamBuilder<int>(<
  stream: counterBloc.stream,
  initialData: 0,
  builder: (_, snapshot) {
    return Text(
      snapshot.data.toString(),
      style: Theme.of(context).textTheme.headlineMedium,
    );
  }
),

```

