Main

import 'package:flutter/material.dart';

import 'package:flutter/services.dart';

import 'package:flutter\_bloc/flutter\_bloc.dart';

import 'package:hello/for\_git\_ac/cubit\_bloc.dart';

import 'package:hello/for\_git\_ac/observer.dart';

import 'package:hello/for\_git\_ac/setstate\_home.dart';

void main() {

  Bloc.observer = ObserverPage();

  WidgetsFlutterBinding.ensureInitialized();

  SystemChrome.setEnabledSystemUIMode(SystemUiMode.immersiveSticky);

  runApp(MaterialApp(

      debugShowCheckedModeBanner: false,

      home: BlocProvider<BlocPage>(

        create: (context) {

          return BlocPage(DefaultState(

              isLoading: false, color: Colors.black, message: "Start"));

        },

        child: BlocHome(),

      )));

}

Bloc Home

import 'package:flutter/material.dart';

import 'package:hello/for\_git\_ac/cubit\_bloc.dart';

import 'package:flutter\_bloc/flutter\_bloc.dart';

class BlocHome extends StatefulWidget {

  BlocHomeState createState() {

    return BlocHomeState();

  }

}

class BlocHomeState extends State<BlocHome> {

  final numberController = TextEditingController();

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: const Text("setState method"),

        centerTitle: true,

      ),

      body: ListView(

        padding: const EdgeInsets.all(10),

        children: [

          BlocBuilder<BlocPage, StateData>(

            builder: (context, state) {

              return Container(

                padding: const EdgeInsets.only(bottom: 10),

                height: 200,

                width: double.infinity,

                color: state.color,

                alignment: Alignment.center,

                child: state is LoadingState

                    ? const CircularProgressIndicator()

                    : Text(

                        state.message,

                        style: const TextStyle(

                            color: Colors.white,

                            fontWeight: FontWeight.bold,

                            fontSize: 30),

                      ),

              );

            },

          ),

          const SizedBox(

            height: 10,

          ),

          TextField(

            controller: numberController,

            keyboardType: TextInputType.number,

            decoration: const InputDecoration(

                border: OutlineInputBorder(), hintText: "Number"),

          ),

          const SizedBox(

            height: 10,

          ),

          ElevatedButton(

            onPressed: () {

              context

                  .read<BlocPage>()

                  .add(StartButtonEvent(numberController.text));

            },

            child: const Text("Start"),

          ),

        ],

      ),

    );

  }

}

Bloc page

import 'dart:math';

import 'package:flutter/material.dart';

import 'package:flutter\_bloc/flutter\_bloc.dart';

class EventData {

  //main

}

class StartButtonEvent extends EventData {

  final String numberController;

  StartButtonEvent(this.numberController);

}

class StateData {

  //main

  final bool isLoading;

  final Color color;

  final String message;

  StateData(

      {required this.isLoading, required this.color, required this.message});

}

class DefaultState extends StateData {

  DefaultState(

      {required super.isLoading, required super.color, required super.message});

}

class LoadingState extends StateData {

  LoadingState(

      {required super.isLoading, required super.color, required super.message});

}

class BadState extends StateData {

  BadState(

      {required super.isLoading, required super.color, required super.message});

}

class NotBadState extends StateData {

  NotBadState(

      {required super.isLoading, required super.color, required super.message});

}

class GoodState extends StateData {

  GoodState(

      {required super.isLoading, required super.color, required super.message});

}

class ErrorState extends StateData {

  ErrorState(

      {required super.isLoading, required super.color, required super.message});

}

class BlocPage extends Bloc<EventData, StateData> {

  BlocPage(super.initialState) {

    on<StartButtonEvent>((event, emit) async {

      try {

        int Count = 0;

        emit(LoadingState(

            isLoading: true, color: Colors.grey, message: "Loading"));

        for (var i = 0; i < 100; i++) {

          await Future.delayed(const Duration(microseconds: 500));

          if (int.parse(event.numberController) == Random.secure().nextInt(9)) {

            Count++;

          }

        }

        if (Count < 10) {

          emit(BadState(isLoading: false, color: Colors.red, message: "Bad"));

          return;

        }

        if (Count >= 50 && Count <= 70) {

          emit(NotBadState(

              isLoading: false, color: Colors.green, message: "Not Bad"));

          return;

        }

        emit(GoodState(isLoading: false, color: Colors.amber, message: "Good"));

      } catch (error) {

        emit(

            ErrorState(isLoading: false, color: Colors.pink, message: "Error"));

        addError(error);

      }

    });

  }

}

Observer page

import 'package:flutter\_bloc/flutter\_bloc.dart';

class ObserverPage extends BlocObserver {

  @override

  void onChange(BlocBase bloc, Change change) {

    // TODO: implement onChange

    super.onChange(bloc, change);

  }

  @override

  void onClose(BlocBase bloc) {

    // TODO: implement onClose

    super.onClose(bloc);

  }

  @override

  void onCreate(BlocBase bloc) {

    // TODO: implement onCreate

    super.onCreate(bloc);

  }

  @override

  void onError(BlocBase bloc, Object error, StackTrace stackTrace) {

    // TODO: implement onError

    super.onError(bloc, error, stackTrace);

  }

}