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<CubitBloc>(

        create: (context) {

          return CubitBloc(RequiredData(

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

        },

        child: CubitBlocHome(),

      )));

}

Cubit Home

import 'package:flutter/material.dart';

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

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

class CubitBlocHome extends StatefulWidget {

  CubitBlocHomeState createState() {

    return CubitBlocHomeState();

  }

}

class CubitBlocHomeState extends State<CubitBlocHome> {

  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<CubitBloc, RequiredData>(

            builder: (context, state) {

              return Container(

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

                height: 200,

                width: double.infinity,

                color: state.color,

                alignment: Alignment.center,

                child: state.isLoading

                    ? 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<CubitBloc>().Checker(numberController.text);

            },

            child: const Text("Start"),

          ),

        ],

      ),

    );

  }

}

CubitBlocPage

import 'dart:math';

import 'package:flutter/material.dart';

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

class RequiredData {

  final bool isLoading;

  final Color color;

  final String message;

  RequiredData(

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

}

class CubitBloc extends Cubit<RequiredData> {

  CubitBloc(super.initialState);

  Future<void> Checker(String textFieldController) async {

    try {

      int Count = 0;

      emit(RequiredData(

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

      for (var i = 0; i < 100; i++) {

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

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

          Count++;

        }

      }

      if (Count < 10) {

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

        return;

      }

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

        emit(RequiredData(

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

        return;

      }

      emit(

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

    } catch (error) {

      emit(

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

      addError(error);

    }

  }

}

ObserverPage

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);

  }

}