

[Open in app](#)[Get started](#)

Pablo Reyes

[Follow](#)Jul 6, 2021 · 3 min read · [Listen](#)

Save



Flutter: BlocBuilder vs BlocConsumer vs BlocListener

Photo by [Azazello BQ](#) on UnsplashYou can find the Spanish version here [👉](#)

[Open in app](#)[Get started](#)

Start with BLoC on a Flutter app can be awful and most of the time confusing, and this is because some concepts are not clear, like this three guys: BlocBuilder, BlocListener and BlocConsumer.

So when we should use each one? ... Let's start with this in mind, we have a restaurant, it process orders, but because we're using BLoC then we will have something like:

OrdersEvent and **OrdersState**.

- **OrdersEvent:** PlaceAnOrder, PayAnOrder, CancelOrder
- **OrdersState:** OrderRequested, OrderInProgress, OrderCompleted, OrderRefunded

BlocBuilder

This is the easy one. This is used when we want to draw a **Widget** based on what is the current **State**. In the following example a new “text” gets drawn every time the state changes.

```
1  BlocBuilder<OrdersBloc, OrdersState>(  
2    buildWhen: (context, state) {  
3      return state is OrdersState.OrderCompleted  
4    },  
5    builder: (context, state) {  
6      if (state is OrdersState.OrderCompleted) {  
7        return Container(child: Text('Order Completed!'));  
8      } else if (OrdersState.OrderInProgress) {  
9        return Container(child: Text('In Progress'));  
10     } else if (OrdersState.OrderRequested) {  
11       return Container(child: Text('A customer placed an order!'));  
12     } else {  
13       return Container(child: Text('Waiting for an order'));
```




[Open in app](#)
[Get started](#)

The builder is called only when the widget is created and when a rebuild exist and it returns true (for this example it only returns true when state = OrderCompleted)

- **buildWhen (Optional):** This is flag (true/false) indicates if the builder method should be called or not, keep in mind that this is called only during a rebuild process (explained at the bottom of the article). If this returns true then builder is called, if returns false it is not called. If buildWhen is not declared then the builder is always executed.
- **builder (Required):** This method is most important, this returns the widget that we want to draw based on the current state. i.e. The state is “OrderCompleted” then it returns “Text(“Order Completed!”)”

BlocListener

This is just a listener not a builder (like the above), that means that its job is keep listening for new changes in the state and not to return a widget. Each time the state changes to a new state this listener will receive a notification that the state has changed and then you can trigger an action (e.g. Send a notification, consume an endpoint, analytics, etc).

So what gets draw in the screen doesn't depends of what we receive in the listener, i.e. it doesn't depend of the actual state (OrdersState), it reacts depending of the state.

```
1  BlocListener<OrdersBloc, OrdersState>(  
2    listenWhen: (context, state) {  
3      return state is OrdersState.OrderCompleted;  
4    },  
5    listener: (context, state) {  
6      // Navigate to next screen  
7      Navigator.of(context).pushNamed('OrderCompletedScreen');  
8    },  
9    child: Container(child: Text('Always draw this text!')),  
10 );
```

bloc_builder_diff_med01.dart hosted with ❤ by GitHub

[view raw](#)

The listener is called only one time and this is when a rebuild process is executed



[Open in app](#)[Get started](#)

(explained at the bottom of the article). If this returns true then listener is called, if returns false it is not called. If listenWhen is not declared then the listener is always executed.

- **listener:** This method is most important, it listens for new changes in the state and execute actions based on the received state. For example: API requests, call analytics stuff, etc.

BlocConsumer

This is a mix between “BlocListener” and “BlocBuilder”. This is used when we want to draw something based on the current state and execute some actions depending on the new arriving states.

```
1  BlocConsumer<OrdersBloc, OrdersState>(  
2    listenWhen: (context, state) {  
3      return state is OrdersState.OrderCompleted ||  
4        state is OrdersState.OrderRefunded;  
5    },  
6    listener: (context, state) {  
7      if (state is OrdersState.OrderCompleted) {  
8        // Navigate to next screen  
9        Navigator.of(context).pushNamed('OrderCompletedScreen');  
10     } else if (state is OrdersState.OrderRefunded) {  
11       // Report to analytics  
12       Analytics.reportRefunded(state.orderId);  
13     }  
14   },  
15   buildWhen: (context, state) {  
16     return state is OrdersState.OrderCompleted ||  
17       state is OrdersState.OrderInProgress ||  
18       state is OrdersState.OrderRequested;  
19   },  
20   builder: (context, state) {  
21     if (state is OrdersState.OrderCompleted) {  
22       return Container(child: Text('Order Served!'));  
23     } else if (OrdersState.OrderInProgress) {
```




[Open in app](#)
[Get started](#)

```
29 );
```

bloc_builder_diff_med03_es.dart hosted with ❤️ by GitHub

[view raw](#)

The methods “buildWhen” and “listenWhen” are optional.

The “build” does the same as in BlocBuilder and “listener” does the same as in “BlocListener”.

¿What is a rebuild and when it is executed?

It is when the **BuildOwner** indicates that a widget is dirty, so the **build** method needs to be called again in order to redraw the widget. This happens only when the widget is already built, i.e. the build process was previously called.

¿How it is related with “listenWhen” and “buildWhen”? This is because these methods “listenWhen” and “buildWhen” are called only during a rebuild process. i.e these methods are **not called** the first time a widget is drawn; so the widget is drawn using the current state, then later a rebuild process is executed and these methods are called “listenWhen” and “buildWhen” in order to know if it should proceed to draw again the widget (build) or call an action (in listen).

Note: Take in mind that the rebuild process comes with a cost, that means that it affects the performance, that’s one of the reasons why there’s methods like these two “listenWhen” and “buildWhen” to avoid a redraw if is not necessary.

Conclusion: The BlocListener is different from BlocBuilder; and the BlocConsumer is merge of both. Also keep in mind to avoid the usage of BlocListener, BlocBuilder and BlocConsumer if we are not going to draw or execute and action based on an state.





Open in app

Get started

[About](#) [Help](#) [Terms](#) [Privacy](#)

Get the Medium app

