First, we try to separate the events and states:

In my case, I have two even: the increment; and the decrement.

We need to create a bloc file (just to be much more organize)

Now let’s declare an abstract class:



(this is the base class that the two events above will inherit)

Now let’s create a class (our counter class)



We need to have a state controller

Input steam (SINK)

This will be the sink. The sink takes any incoming data. (Take note we only take Int as the stream only have one-expecting- data type)



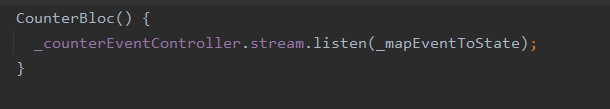
Output stream

This will be the stream, it outputs the data

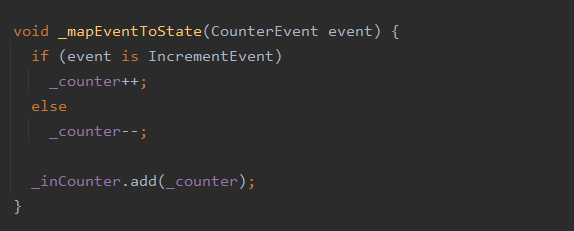


Now let's create another a stream controller (this will be the sink). So it will be from the UI to the sink then to the counter value above



Now we need to have a listener(stream listener)

Then we have to declare what will trigger during the event. By creating a function.



The counter Stream is the one that outputs

Now we need to implement the things from above to our main dart.

We need to instantiate the counterbloc class



Now we have to use “StreamBuilder()”

Inside it we are going to declare out bloc counter that we have created earlier

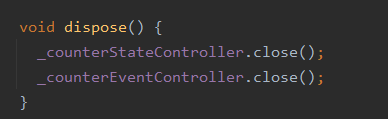
We had instantiated our class so we just going to access one of its variables, the counter.

Now we have to connect the event listener to the bloc

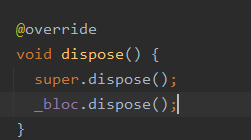


So whenever the button triggers it access the counterBloc class and access the function that we did earlier. It’s either increment or decrement.

Last. We need to have a closing stream. Streams should always be closed when they’re no longer needed. Otherwise, we may get a memory leak in our app.



Now we need to call the above function from the Main class



Notes:

Basically, we need to do be able to determine which events should we setup a bloc. We want to determine the events that might cause a state change.

It is similar to a getter setter.