Kotlin Flow API ~ Android cheat sheet

Type	Supertype	Description	Features	Implementation	Code example	RxJava equivalent	Android usecase
Flow	-	An asynchronous data stream that sequentially emits values and completes normally or with an exception (Unicast broadcaster).	• Flow starts separately for each collector • Various intermediate & terminal operators • Automatic backpressure management	• General go to type • Can be converted to SharedFlow / StateFlow with .shareIn and .stateIn operators.	Example A Example B	Flowable (Observable with BPM)	General data type for multi shot asynchronous data streams like the many callbacks inside the View.java class or remote server data.
SharedFlow	Flow	A Flow shared between multiple collectors (aka subscribers), so that only one flow is effectively run (Multicast broadcaster).	• All subscribes receive all emited values • *n* most recent values are saved in replay cache • New subscribers get the replay cache & new values	• Can be configured with sharing strategy from the SharingStarted interface. • Eagerly, Lazily and WhileSubscribed	Example A Example B	PublishSubject (Starts with no value)	Useful for broadcasting expensive events to subscribers that can come and go. Like sharing remote gps data between multipe activities
MutableSharedFlow	SharedFlow & FlowCollector	A mutable SharedFlow that provides functions to emit values to the flow.	• emit() function to update value • tryEmit() function for non-suspending updates	Same as SharedFlow	See SharedFlow	PublishSubject (Starts with no value)	Same as SharedFlow
StateFlow	SharedFlow	A specialized and limited version of SharedFlow that requires an initial value and emits a read only single data value to its subscribers.	• Always has an initial value • Fixed replaysize of 1 • No buffering • Read access to current value without collecting	• Can be configured with sharing strategy from the SharingStarted interface. • Eagerly, Lazily and WhileSubscribed	Example A Example B	BehaviorSubject (Always emits something)	Similar to LiveData but with far more operators and not limited to mainthread. Recommended for KMM projects. Does require manual lifecycle management.
MutableStateFlow	StateFlow & MutableSharedFlow	A mutable StateFlow that provides a setter for value.	• Read/write access to current value without collecting • Setting the same value as before does nothing (distinct until changed built in)	Same as StateFlow	See StateFlow	BehaviorSubject (Always emits something)	MutablaLiveData equivalent, same up- and downsides from StateFlow apply
callbackFlow (function, returns Flow)	-	Creates a Flow and allows values to be emitted from a different CoroutineContext. Uses a hot SendChannel internally.	• Conceptually very similar to a blocking queue • Default capacity of 64 elements • Buffer can be configured	• Useful when you need to run computations in different CoroutineContexts • Can convert cluncky callback API's to nicer Flows	Example A Example B	-	Converting multi shot Android (Java) callback API's to a Flow like the onLocationResult() and the onTextChanged() listeners.

Version 1.1 ~ March 2021
by Remy Benza