

In RxJava an object that implements the *Observer* interface *subscribes* to an object of the *Observable* class. Then that subscriber reacts to whatever item or sequence of items the Observable object *emits*. This pattern facilitates concurrent operations because it does not need to block while waiting for the Observable to emit objects, but instead it creates a sentry in the form of a subscriber that stands ready to react appropriately at whatever future time the Observable does so.

For information about the Observable class, see [the Observable documentation page at ReactiveX.io](#).