This section explains the `BlockingObservable` subclass. A Blocking Observable extends the ordinary Observable class by providing a set of operators on the items emitted by the Observable that block.

To transform an `Observable` into a `BlockingObservable`, use the [Observable.toBlocking()](#) method or the [BlockingObservable.from()](#) method.

- [forEach()](#) — invoke a function on each item emitted by the Observable; block until the Observable completes
- [first()](#) — block until the Observable emits an item, then return the first item emitted by the Observable
- [firstOrDefault()](#) — block until the Observable emits an item or completes, then return the first item emitted by the Observable or a default item if the Observable did not emit an item
- [last()](#) — block until the Observable completes, then return the last item emitted by the Observable
- [lastOrDefault()](#) — block until the Observable completes, then return the last item emitted by the Observable or a default item if there is no last item
- [mostRecent()](#) — returns an iterable that always returns the item most recently emitted by the Observable
- [next()](#) — returns an iterable that blocks until the Observable emits another item, then returns that item
- [latest()](#) — returns an iterable that blocks until or unless the Observable emits an item that has not been returned by the iterable, then returns that item
- [single()](#) — if the Observable completes after emitting a single item, return that item, otherwise throw an exception
- [singleOrDefault()](#) — if the Observable completes after emitting a single item, return that item, otherwise return a default item
- [toFuture()](#) — convert the Observable into a Future
- [toIterable()](#) — convert the sequence emitted by the Observable into an Iterable
- [getIterator()](#) — convert the sequence emitted by the Observable into an Iterator

> *This documentation accompanies its explanations with a modified form of "marble diagrams." Here is how these marble diagrams represent Blocking Observables:*
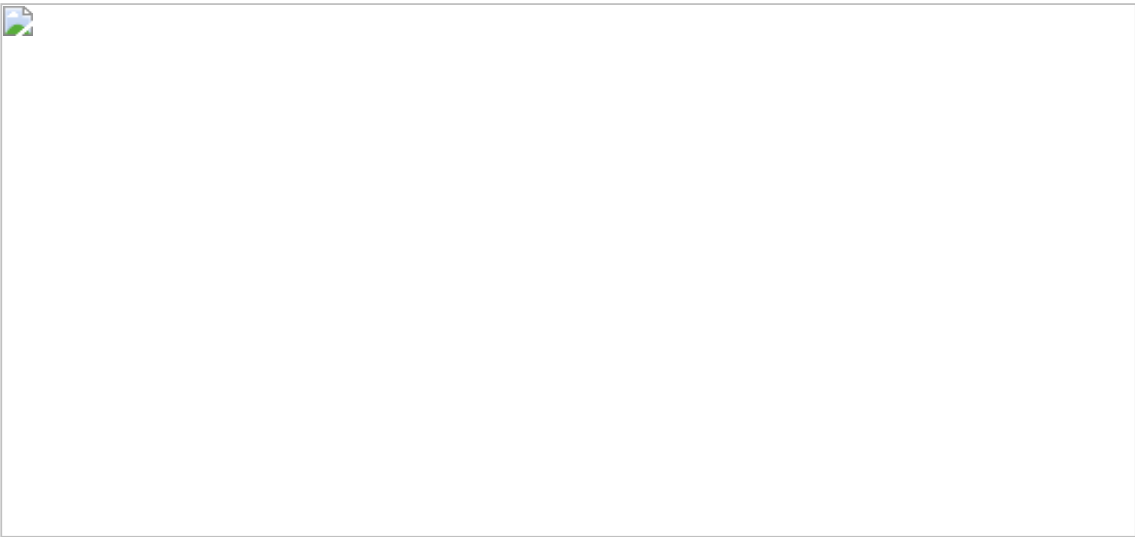
see also:

- javadoc: [BlockingObservable](#)
- javadoc: [toBlocking()](#)
- javadoc: [BlockingObservable.from()](#)

Appendix: similar blocking and non-blocking operators

operator	result when it acts on			equivalent in Rx.NET
	Observable that emits	Observable that emits	Observable that emits no items	

	multiple items	one item		
Observable.first	the first item	the single item	*NoSuchElement*	firstAsync
BlockingObservable.first	the first item	the single item	*NoSuchElement*	first
Observable.firstOrDefault	the first item	the single item	the default item	firstOrDefaultAsync
BlockingObservable.firstOrDefault	the first item	the single item	the default item	firstOrDefault
Observable.last	the last item	the single item	*NoSuchElement*	lastAsync
BlockingObservable.last	the last item	the single item	*NoSuchElement*	last
Observable.lastOrDefault	the last item	the single item	the default item	lastOrDefaultAsync
BlockingObservable.lastOrDefault	the last item	the single item	the default item	lastOrDefault
Observable.single	*Illegal Argument*	the single item	*NoSuchElement*	singleAsync
BlockingObservable.single	*Illegal Argument*	the single item	*NoSuchElement*	single
Observable.singleOrDefault	*Illegal Argument*	the single item	the default item	singleOrDefaultAsync
BlockingObservable.singleOrDefault	*Illegal Argument*	the single item	the default item	singleOrDefault