

RxJava is a Java VM implementation of [ReactiveX \(Reactive Extensions\)](#): a library for composing asynchronous and event-based programs by using observable sequences.

For more information about ReactiveX, see the [Introduction to ReactiveX](#) page.

RxJava is Lightweight

RxJava tries to be very lightweight. It is implemented as a single JAR that is focused on just the Observable abstraction and related higher-order functions.

RxJava is a Polyglot Implementation

RxJava supports Java 6 or higher and JVM-based languages such as [Groovy](#), [Clojure](#), [JRuby](#), [Kotlin](#) and [Scala](#).

RxJava is meant for a more polyglot environment than just Java/Scala, and it is being designed to respect the idioms of each JVM-based language. ([This is something we're still working on.](#))

RxJava Libraries

The following external libraries can work with RxJava:

- [Hystrix](#) latency and fault tolerance bulkheading library.
- [Camel RX](#) provides an easy way to reuse any of the [Apache Camel components, protocols, transports and data formats](#) with the RxJava API
- [rxjava-http-tail](#) allows you to follow logs over HTTP, like `tail -f`
- [mod-rxvertx - Extension for VertX](#) that provides support for Reactive Extensions (RX) using the RxJava library
- [rxjava-jdbc](#) - use RxJava with jdbc connections to stream ResultSets and do functional composition of statements
- [rtree](#) - immutable in-memory R-tree and R*-tree with RxJava api including backpressure