Getting Binaries

You can find binaries and dependency information for Maven, Ivy, Gradle, SBT, and others at http://search.maven.org.

Example for Maven:

```
<dependency>
    <groupId>io.reactivex.rxjava3</groupId>
    <artifactId>rxjava</artifactId>
    <version>3.0.4</version>
</dependency>
```

and for Ivy:

```
<dependency org="io.reactivex.rxjava3" name="rxjava" rev="3.0.4" />
```

and for SBT:

```
libraryDependencies += "io.reactivex" %% "rxscala" % "0.26.5"

libraryDependencies += "io.reactivex.rxjava3" % "rxjava" % "3.0.4"
```

and for Gradle:

```
implementation 'io.reactivex.rxjava3:rxjava:3.0.4'
```

If you need to download the jars instead of using a build system, create a Maven `pom` file like this with the desired version:

```
<?xml version="1.0"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
      <modelVersion>4.0.0</modelVersion>
      <groupId>io.reactivex.rxjava3</groupId>
      <artifactId>rxjava</artifactId>
      <version>3.0.4</version>
      <name>RxJava</name>
      <description>Reactive Extensions for Java</description>
      <url>https://github.com/ReactiveX/RxJava</url>
      <dependencies>
          <dependency>
              <groupId>io.reactivex.rxjava3</groupId>
              <artifactId>rxjava</artifactId>
              <version>3.0.4</version>
          </dependency>
      </dependencies>
</project>
```

Then execute:

```
$ mvn -f download-rxjava-pom.xml dependency:copy-dependencies
```

That command downloads `rxjava-*.jar` and its dependencies into `./target/dependency/`.

You need Java 6 or later.

Snapshots

Snapshots are available via JFrog:

```
repositories {
    maven { url 'https://oss.jfrog.org/libs-snapshot' }
}

dependencies {
    compile 'io.reactivex.rxjava3:rxjava:3.0.4'
}
```

Building

To check out and build the RxJava source, issue the following commands:

```
$ git clone git@github.com:ReactiveX/RxJava.git
$ cd RxJava/
$ ./gradlew build
```

To do a clean build, issue the following command:

```
$ ./gradlew clean build
```

A build should look similar to this:

```
$ ./gradlew build
:rxjava:compileJava
:rxjava:processResources UP-TO-DATE
:rxjava:classes
:rxjava:jar
:rxjava:sourcesJar
:rxjava:signArchives SKIPPED
:rxjava:assemble
:rxjava:licenseMain UP-TO-DATE
:rxjava:licenseTest UP-TO-DATE
:rxjava:compileTestJava
:rxjava:processTestResources UP-TO-DATE
:rxjava:testClasses
:rxjava:test
:rxjava:check
:rxjava:build
```

```
BUILD SUCCESSFUL

Total time: 30.758 secs
```

On a clean build you will see the unit tests run. They will look something like this:

```
> Building > :rxjava:test > 91 tests completed
```