WIKIPEDIA

List of Java APIs

There are two types of Java programming language application programming interfaces (APIs):

- The official core Java API, contained in the Android (Google), SE (OpenJDK and Oracle), MicroEJ. These packages (java.* packages) are the core Java language packages, meaning that programmers using the Java language had to use them in order to make any worthwhile use of the Java language.
- Optional APIs that can be downloaded separately. The specification of these APIs are defined according to many different organizations in the world (Alljoyn, OSGi, Eclipse, JCP, E-S-R, etc.).

The following is a partial list of application programming interfaces (APIs) for Java.

Contents

APIs

Name	Acronym	Description and Version History	Available from
Java Advanced Imaging	JAI	A set of interfaces that support a high-level programming model allowing to manipulate images easily.	
Association for the standardization of embedded platforms	E-S-R consortium	here (http://www.e-s-r.net)	
Java Data Objects	JDO	A specification of Java object persistence.	
Android API	Google	here (https://developer.android.com)	
JavaHelp		A full-featured, extensible help system that enables you to incorporate online help in applets, components, applications, operating systems, and devices.	available here (https://javaee.github.io/javahelp/)
Java Media Framework	JMF	An API that enables audio, video and other time-based media to be added to Java applications and applets.	
Java Naming and Directory Interface	JNDI	An API for directory services.	
Java Persistence API	JPA	A specification for object-relational mapping.	JSR 338 (https://jcp.org/en/jsr/detail?id=338)
Java Speech API	JSAPI	This API allows for speech synthesis and speech recognition.	
Java 3D	J3D	A scene graph-based 3D API.	available here (https://web.archive.org/web/20070611104000/https://java3d.dev.java.net/)
Java OpenGL	JOGL	A wrapper library for OpenGL.	available here (http://jogamp.org/)
Java USB for Windows	(none)	A USB communication of Java applications	available here (https://sourceforge.net/projects/javausbapiforwi/)
Facebook4j	(none)	Facebook API wrapper in Java.	available here (https://facebook4j.github.io/en/index.html)
Twitter4j	(none)	Java library for the Twitter API	available here (http://twitter4j.org/en/index.html)

Name	Acronym	Java package(s) that contain the API
JavaBeans Activation Framework	JAF	javax.activation
JavaMail	(none)	javax.mail
Java Message Service	JMS	javax.jms
JavaServer Faces	JSF	javax.faces

Name	Acronym	Available from
Java API for XML-Based RPC	JAX-RPC	available here (http://java.sun.com/xml/downloads/jaxrpc.html)
XQuery API for Java	XQJ	here (http://xqj.net/) and here (http://jcp.org/en/jsr/detail?id=225)

Name	Acronym	Available from
Connected Limited Device Configuration	CLDC	Reference implementation is available here (http://java.sun.com/products/cldc/)
Java Telephony API	JTAPI	available here (http://www.oracle.com/technetwork/java/jtapi-136088.html)
STM32 Java technology	STM32Java	available here (http://www.stm32java.com)
MicroEJ embedded platform	MicroEJ	available here (https://www.microej.com/)

Following is a very incomplete list, as the number of APIs available for the Java platform is overwhelming.

Rich Client platforms

- Eclipse Rich Client Platform (RCP)
- NetBeans Platform

Office_compliant libraries

- Apache POI
- Aspose
- JXL - for Microsoft Excel
- JExcel - for Microsoft Excel

Compression

- LZMA SDK, the Java implementation of the SDK used by the popular 7-Zip file archive software (available here (http://www.7-zip.org/sdk.html))

JSON

- Jackson (API)

Game engines

- Slick
- jMonkey Engine
- JPCT Engine
- LWJGL

Real-time libraries

Real time Java is a catch-all term for a combination of technologies that allows programmers to write programs that meet the demands of real-time systems in the Java programming language.

Java's sophisticated memory management, native support for threading and concurrency, type safety, and relative simplicity have created a demand for its use in many domains. Its capabilities have been enhanced to support real time computational needs:

- Java supports a strict priority based threading model.

- Because Java threads support priorities, Java locking mechanisms support priority inversion avoidance techniques, such as priority inheritance or the priority ceiling protocol.

To overcome typical real time difficulties, the Java Community introduced a specification for real-time Java, JSR001. A number of implementations of the resulting *Real-Time Specification for Java* (RTSJ) have emerged, including a reference implementation from Timesys, IBM's WebSphere Real Time, Sun Microsystems's Java SE Real-Time Systems,[1] Aonix PERC or JamaicaVM from aicas.

The RTSJ addressed the critical issues by mandating a minimum (only two) specification for the threading model (and allowing other models to be plugged into the VM) and by providing for areas of memory that are not subject to garbage collection, along with threads that are not preemptable by the garbage collector. These areas are instead managed using region-based memory management.

Real-Time Specification for Java

The *Real-Time Specification for Java* (RTSJ) is a set of interfaces and behavioral refinements that enable real-time computer programming in the Java programming language. RTSJ 1.0 was developed as JSR 1 under the Java Community Process, which approved the new standard in November, 2001. RTSJ 2.0 is being developed under JSR 282. A draft version is available at JSR 282 JCP Page. More information can be found at RTSJ 2.0

- Javolution

Windowing libraries

The windowing library is a set of classes available in the Streams Processing Language (SPL) Runtime C++ API and the SPL Java™ Operator API. The library is used to implement primitive operators that need windows following the SPL window semantics. Using the windowing library provides a consistent window policy semantic across operators, and simplifies the operator implementation.

SPL offers both tumbling and sliding windows. Both types of windows keep all the incoming data in memory until its tuple eviction policy triggers. Use the SPL support for windows when the functionality required by the primitive operator can be built using the semantics provided by SPL window constructs.

One example operator from the SPL Standard Toolkit that uses the windowing library and syntax is the `Aggregate` operator. An example operator that buffers recently received tuples but that does not use the windowing library and syntax is the `DeDuplicate` operator. This action occurs because this operator has different eviction and trigger semantics than the ones provided by SPL. This operator needs to maintain only unique tuples and discards all repeated tuples that are received within a time window.

With the windowing library, developers can specify different eviction and trigger policies but can implement the event handling actions independently of the window policy details. There are a few differences when implementing primitive operators in C++ and Java that take advantage of the SPL window clause.

In the C++ implementation, developers have no obligation to use the window library (the preferred practice is to use library). They are free to use the SPL Operator Code Generation API just to get the specified window policy for a given operator instance. Then, they can generate code using other containers. In addition, developers can check for valid window configurations during code generation time.

In the Java implementation, developers must use the window library. This action can be achieved by registering a class that implements `com.ibm.streams.operator.window.StreamWindowListener<T>` that handles events generated by a window (`com.ibm.streams.operator.window.StreamWindow<T>`). This use is required because Java operators are not based on code generation. As a result, windows are automatically managed by the runtime to guarantee SPL window semantics. Developers can check for valid window configurations during runtime by checking the window policy associated to a given input port.

- Standard Widget Toolkit (SWT)

Physics libraries

- JBox2D
- JBullet
- dyn4j

See also

- Java Platform
- Application programming interface
- Java ConcurrentMap
- List of Java Frameworks

Notes

External links

- APISonar - Search Java API examples (http://apisonar.com)