

API

An application programming interface (API) is a connection between [computers](#) or between [computer programs](#). It is a type of software [interface](#), offering a service to other pieces of [software](#).^[1] A document or standard that describes how to build such a connection or interface is called an API specification. A computer system that meets this standard is said to implement or expose an API. The term API may refer either to the specification or to the implementation

In contrast to a [user interface](#), which connects a computer to a person, an application programming interface connects computers or pieces of software to each other. It is not intended to be used directly by a person (the [end user](#)) other than a [computer programmer](#) who is incorporating it into software. An API is often made up of different parts which act as tools or services that are available to the programmer. A program or a programmer that uses one of these parts is said to call that portion of the API. The calls that make up the API are also known as [subroutines](#), methods, requests, or [endpoints](#). An API specification defines these calls, meaning that it explains how to use or implement them. One purpose of APIs is to [hide the internal details](#) of how a system works, exposing only those parts a programmer will find useful and keeping them consistent even if the internal details later change. An API may be custom-built for a particular pair of systems, or it

may be a shared standard allowing [interoperability](#) among many systems.

The term API is often used to refer to [web APIs](#),^[2] which allow communication between computers that are joined by the [internet](#). There are also APIs for [programming languages](#), [software libraries](#), computer [operating systems](#), and [computer hardware](#). APIs originated in the 1940s, though the term did not emerge until the 1960s and 70s

Key Components of APIs:

1. Endpoint: A URL where the API service can be accessed.
2. Request: Data sent to the API (via HTTP methods like GET, POST, PUT, DELETE).
3. Response: Data returned by the API, often in formats like JSON or XML.
4. Authentication: Security measures like API keys, OAuth, or tokens to restrict access.
5. Rate Limiting: Restrictions on the number of API calls a client can make in a given period.

Libraries and frameworks

The interface to a [software library](#) is one type of API. The API describes and prescribes the "expected behavior" (a specification) while the library is an "actual implementation" of this set of rules.

A single API can have multiple implementations (or none, being abstract) in the form of different libraries that share the same programming interface.

The separation of the API from its implementation can allow programs written in one language to use a library written in another. For example, because Scala and Java compile to compatible bytecode, Scala developers can take advantage of any Java API.

API use can vary depending on the type of programming language involved. An API for a procedural language such as Lua could consist primarily of basic routines to execute code, manipulate data or handle errors while an API for an object-oriented language, such as Java, would provide a specification of classes and its class methods. Hyrum's law states that "With a sufficient number of users of an API, it does not matter what you promise in the contract: all observable behaviors of your system will be depended on by somebody." Meanwhile, several studies show that most applications that use an API tend to use a small part of the API.

Language bindings are also APIs. By mapping the features and capabilities of one language to an interface implemented in another language, a language binding allows a library or service written in one language to be used when developing in another language.^[24] Tools such as SWIG and F2PY, a Fortran-to-Python interface generator, facilitate the creation of such interfaces.

An API can also be related to a software framework: a framework can be based on several libraries implementing several APIs, but unlike the normal use of an API, the access to the behavior built into the framework is mediated by extending its content with new classes plugged into the framework itself.

Moreover, the overall program flow of control can be out of the control of the caller and in the framework's hands by inversion of control or a similar mechanism.

Operating systems^[edit]

An API can specify the interface between an application and the operating system.^[28] POSIX, for example, specifies a set of common APIs that aim to enable an application written for a POSIX conformant operating system to be compiled for another POSIX conformant operating system.

Linux and Berkeley Software Distribution are examples of operating systems that implement the POSIX APIs.

Microsoft has shown a strong commitment to a backward-compatible API, particularly within its Windows API (Win32) library, so older applications may run on newer versions of Windows using an executable-specific setting called "Compatibility Mode".

An API differs from an application binary interface (ABI) in that an API is source code based while an ABI is binary based. For instance, POSIX provides APIs while the Linux Standard Base provides an ABI.

Remote APIs[[edit](#)]

Remote APIs allow developers to manipulate remote resources through protocols, specific standards for communication that allow different technologies to work together, regardless of language or platform. For example, the Java Database Connectivity API allows developers to query many different types of databases with the same set of functions, while the Java remote method invocation API uses the Java Remote Method Protocol to allow invocation of functions that operate remotely, but appear local to the developer.

Therefore, remote APIs are useful in maintaining the object abstraction in object-oriented programming; a method call, executed locally on a proxy object, invokes the corresponding method on the remote object, using the remoting protocol, and acquires the result to be used locally as a return value.

A modification of the proxy object will also result in a corresponding modification of the remote object.