What is an API?

An application-programming interface (API) is a set of programming instructions and standards for accessing a Web-based software application or **Web tool**. A software company releases its API to the public so that other software developers can design products that are powered by its service.

For example, Amazon.com released its API so that Web site developers could more easily access Amazon's product information. Using the Amazon API, a third party Web site can post direct links to Amazon products with updated prices and an option to "buy now."

An API is a software-to-software interface, not a user interface. With APIs, applications talk to each other without any user knowledge or intervention. When you buy movie tickets online and enter your credit card information, the movie ticket Web site uses an API to send your credit card information to a remote application that verifies whether your information is correct. Once payment is confirmed, the remote application sends a response back to the movie ticket Web site saying it's OK to issue the tickets.

As a user, you only see one interface -- the movie ticket Web site -- but behind the scenes, many applications are working together using APIs. This type of integration is called **seamless**, since the user never notices when software functions are handed from one application to another [source: TConsult, Inc.]

An API resembles Software as a Service (SaaS), since software developers don't have to start from scratch every time they write a program. Instead of building one core application that tries to do everything -- e-mail, billing, tracking, etcetera -- the same application can contract out certain responsibilities to remote software that does it better.

Let's use the same example of Web conferencing from before. Web conferencing is SaaS since it can be accessed on-demand using nothing but a Web site. With a conferencing API, that same on-demand service can be integrated into another Web-based software application, like an instant messaging program or a Web calendar.

The user can schedule a Web conference in his Web calendar program and then click a link within the same program to launch the conference. The calendar program doesn't host or run the conference itself. It uses a conferencing API to communicate behind the scenes with the remote Web conferencing service and seamlessly delivers that functionality to the user.

Now we'll explain some of the technology that makes conferencing APIs work.

API and protocols

An API can also be an implementation of a protocol.

In general the difference between an API and a protocol is that the protocol defines a standard way to exchange requests and responses based on a common transport and agreeing on a data/message exchange format, while an API (not implementing a protocol) is usually implemented as a library to be used directly: hence there can be no *transport* involved (no information physically transferred from/to some

remote machine), but rather only simple information exchange via *function calls* (local to the machine where the elaboration takes place) and data is exchanged in formats expressed in a specific language.[4]

When an API implements a protocol it can be based on proxy methods for remote invocations that underneath rely on the communication protocol. The role of the API can be exactly to hide the detail of the transport protocol. E.g.: RMI is an API that implements the JRMP protocol or the IIOP as RMI-IIOP.

Protocols are usually shared between different technologies (system based on given computer programming languages in a given operating system) and usually allow the different technologies to exchange information, acting as an abstraction/mediation level between the two *worlds*. While APIs can be specific to a given technology: hence the APIs of a given language cannot be used in other languages, unless the function calls are wrapped with specific adaptation libraries.

Web APIs

Main article: web service

When used in the context of web development, an API is typically defined as a set of Hypertext Transfer Protocol (HTTP) request messages, along with a definition of the structure of response messages, which is usually in an Extensible Markup Language (XML) or JavaScript Object Notation (JSON) format. While "Web API" is virtually a synonym for web service, the recent trend (so-called Web 2.0) has been moving away from Simple Object Access Protocol (SOAP) based services towards more direct Representational State Transfer (REST) style communications.[5] Web APIs allow the combination of multiple services into new applications known as mashups.[6]

Web use to share content

The practice of publishing APIs has allowed web communities to create an open architecture for sharing content and data between communities and applications. In this way, content that is created in one place can be dynamically posted and updated in multiple locations on the web.

1. Photos can be shared from sites like Flickr and Photobucket to social network sites like Facebook and MySpace.
2. Content can be embedded, e.g. embedding a presentation from SlideShare on a LinkedIn profile.
3. Content can be dynamically posted. Sharing live comments made on Twitter with a Facebook account, for example, is enabled by their APIs.
4. Video content can be embedded on sites which are served by another host.
5. User information can be shared from web communities to outside applications, delivering new functionality to the web community that shares its user data via an open API. One of the best examples of this is the Facebook Application platform. Another is the Open Social platform.[7]