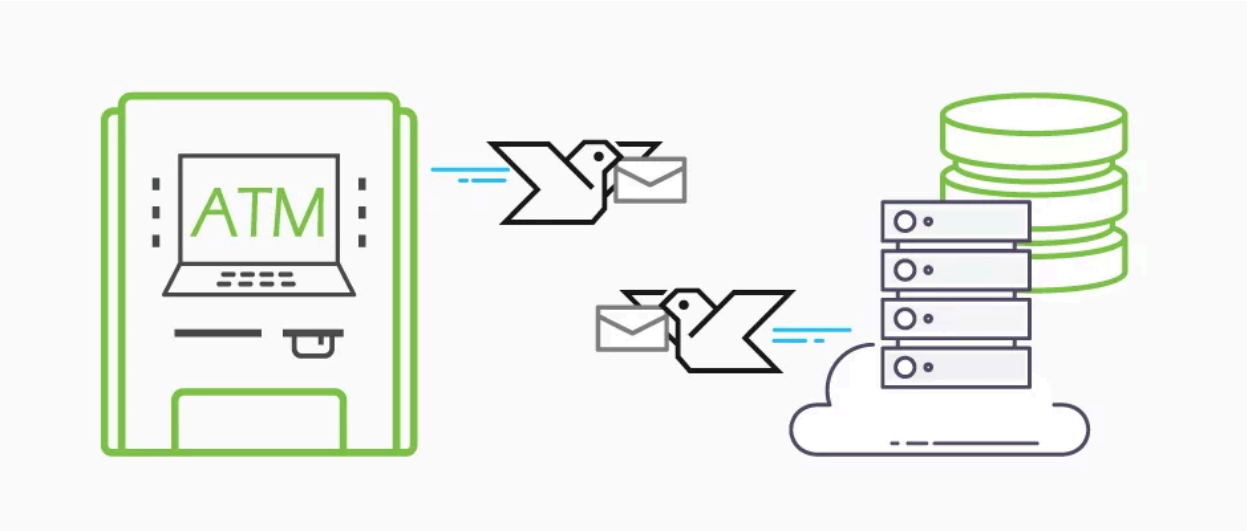
Intro to APIs: What Are They & What Do They Do?

adapted from Upwork, July 25, 2016

https://www.upwork.com/hiring/development/intro-to-apis-what-is-an-api/



Application programming interfaces (API) have been around since the 1960s. So if they aren't a new concept, why are they such a hot topic lately?

In short, without APIs, the digital experiences that we expect every day as consumers wouldn't be possible. They're doing everything from driving information-rich marketing campaigns and connecting mobile apps to streamlining internal operations. Businesses now know that investing in an API strategy can pay significant dividends.Here's a brief introduction to what APIs are and what they can do.

First, let's look at an example of an API.

APIs do a lot of heavy lifting, both in mobile and on the web. They're responsible for nearly everything we do—and with just a few taps or clicks, let you do things like order a pizza, book a hotel, rate a song, or download software. APIs work quietly in the background, making the interactivity we expect—and rely upon—possible.

Here's how it may look in everyday life: You're searching for a hotel room from an online travel booking site. Using the site's online form, you select the city you want to stay in, check-in and checkout dates, number of guests, and number of rooms. Then you click "search."

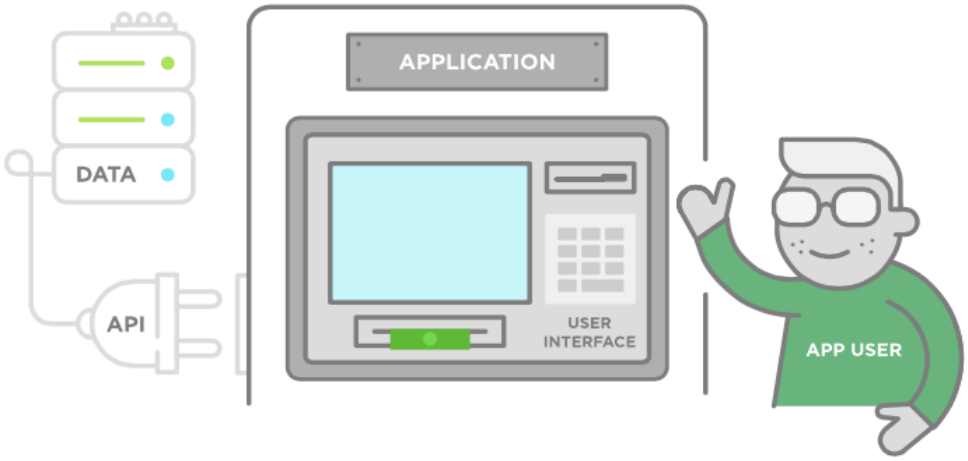
As you may know, the travel site aggregates information from many different hotels. When you click "search," the site then interacts with each hotel's API, which delivers results for available rooms that meet your criteria. This can all happen within seconds because of an API, which acts like a messenger that runs back and forth between applications, databases, and devices.

What is an API? Breaking down the acronym

Let's start with the help of an analogy: withdrawing and depositing cash from an automated teller machine (ATM).

> **Application:** Think of an application like an ATM. When you walk up to an ATM, you expect it will allow you to access your account and complete a transaction like withdrawing cash. Like an ATM, an app provides a function, but it's not doing this all by itself—it needs to communicate both with the user, and with the "bank" it's accessing. An app deals in inputs and outputs, too. A web, mobile, or back-end application is like a machine that solves a specific problem. The software may be a customer-facing app like a travel booking site, or a back-end app like server software that funnels requests to a database.
>
> **Programming:** APIs allow the ATM to communicate with your bank. The programming is the engineering part of the app's software that translates input into output. In other words, it translates your request for cash to the bank's database, verifies there's enough cash in your account to withdraw the requested amount, the bank grants permission, then the ATM communicates back to the bank how much you withdrew so that the bank can update your balance.
>
> **Interface:** A user interface (UI) is how we interact with an application. In the case of the ATM, it's the screen, keypad, and cash slot—where the input and output occurs. We enter our pin number, punch in how much cash we'd like to withdraw, then take the cash that's spit out. Interfaces are how we communicate with a machine. With APIs, it's much the same, only we're replacing users with software.

In a nutshell, that's an API: an interface that software uses to access whatever currency it needs: data, server software, or other applications. In the case of the ATM, the machine is the end user of an API, not the customer pressing the buttons. It's the same in the digital world.



What Do APIs Do?

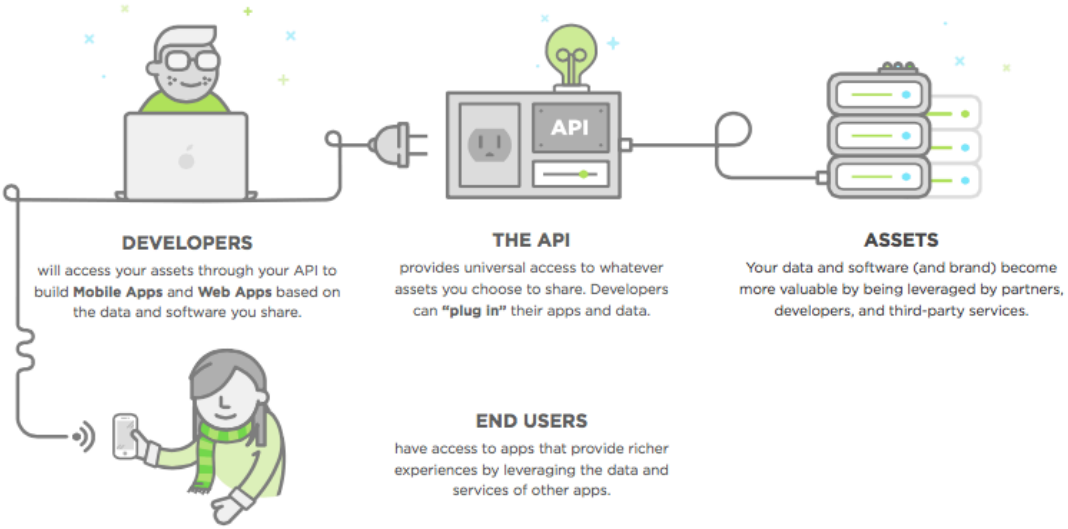
A website uses a URL address to make a call to a server and pull up a webpage in a browser. APIs also facilitate calls to a server, but they execute it more simply. They connect the web, allowing developers, applications, and sites to tap into databases and services (or, assets)—much like open-source software. APIs do this by acting like a universal converter plug offering a standard set of instructions.

The Components of APIs: What You're Sharing and Who You're Sharing It With

1. **All APIs begin with shared assets—they're the currency of an API.**They can be anything a company wants to share—whether that's internally between teams, or externally with other developers: data points, pieces of code, software, or services that a company owns and sees value in sharing.
2. **Next is the API, which acts like a gateway to the server.** It provides a point of entry for your audience—developers who will use those assets to build their own software—but it also acts like a filter for those assets. You never want to open up your entire server and all of its contents to the outside world. APIs only reveal what you want them to reveal.
3. **The immediate audience of an API is rarely an end user of an app; it's typically developers creating software or an app around those assets.** This is where assets take flight, yielding creative, new ways to implement data that previously, may or may not have had any real business value to its owner. It also lets developers use reusable software components so they're not repeating work that's already been done.
4. **All of this results in apps that are connected to data and services, allowing these apps to provide richer, more intelligent experiences for users.** API-powered apps are also compatible with more devices and operating systems, providing more seamless experiences.
5. **In the end, the beneficiaries of these apps are the end users themselves.** The apps enable end users tremendous flexibility to access multiple apps seamlessly between devices, use social profiles to interact with third-party apps, and more.

How do APIs work?

APIs have endless business opportunities. So how can you leverage the functionality explained above? Here are a few ways to think about APIs and how they can work for you.



- **APIs act as a doorway that people with the right key can get through.** Want to give specific people—but not everyone—access to your assets? An API acts like a doorway to your server and database that those with an API key (or a paid subscription) can use to access whatever assets you choose to reveal. A key could give a user read access, write access, or both—it's up to you.
- **APIs let applications (and devices) seamlessly connect and communicate.** An API can create a seamless flow of data between apps and devices in real time. This not only lets developers create apps for any format—a mobile app, a wearable, or a website—it allows apps to "talk to" one another. This is the heart of how APIs create rich user experiences in apps.
- **APIs let you build one app off another app.** Entire businesses and popular web applications like Hootsuite, Zapier, and IFTT (If This Then That) have been built solely on creative ways to leverage APIs. APIs allow you to write applications that use other applications as part of their core functionality. Not only can developers get access to reusable code and technology, they can leverage other technology for their own apps.
- **APIs act like a "universal plug."** What if all of those people with keys to your door speak different languages? With an API, it doesn't matter—everyone, no matter what machine, operating system, or mobile device they're using—gets the same access. Think about those universal outlet plugs that let you use an appliance in any country's socket. An API is a lot like that; it standardizes access.
- **APIs act as a filter.** Security is a big concern with APIs—after all, you're giving outsiders access to your servers and all they contain—which is why they have to be carefully constructed. APIs should give controlled access to assets, with permissions and other measures that keep too much traffic—or malicious traffic—from bringing down your server. This is a very important API design consideration for industries that are heavily regulated, like healthcare and finance.

Artificial intelligence, APIs and the transformation of computer science

Adapted from Computer World, September 21, 2016 by Kris Hammond
https://www.computerworld.com/article/3120773/application-development/artificiai-intelligence-apis-and-the-transformation-of-computer-science.html

These are exciting times in the world of artificial intelligence as advanced capabilities escape the labs and become central to mainstream products -- think Amazon Echo -- and increasingly accessible to everyday software developers.

Recently, we've seen a run of large companies open-sourcing their A.I. platform APIs. Google has open-sourced TensorFlow, its machine learning library. Not to be outdone, Amazon has released its TensorFlow competitor DSSTNE (or Deep Scalable Sparse Tensor Network) and Facebook has done the same with fastText, its machine learning engine for text classification and language recognition. Those big-company A.I. platforms join -- and in some sense compete with -- notable open source deep learning libraries like Caffe, Theano and Torch.

In many ways, open sourcing access to A.I. represents the culmination of the "API economy," where software developers set aside their not-invented-here pride of creation and instead access the work done by others -- and in the process accelerate their own development work -- via API hooks. If software development-by-API made sense in the past, it may be the only practical path forward when it comes to fully leveraging new capabilities in the artificial intelligence field.

Indeed, the impact of these moves can't be overstated. On a business level, they significantly lower the barrier for entrepreneurs looking to start up the next wave of artificial intelligence-driven companies by democratizing access to machine learning resources. That's hugely significant. But I'd argue for an even more fundamental change: Open source A.I. APIs represent the beginning of a transformation of computer science (CS) into a field focused on impact.

Innovators, Integrators and Designers

At its most basic, CS can be described as people working on new algorithms or improvements to existing ones, with new “inventions” being built on the back of what came before it. In CS, there are two standard ways of doing this: creating libraries of functionality that developers can build into their own applications; or providing access to functionality in one fell swoop via API.

Software development via API is hardly new. Microsoft became a behemoth on the strength of its platform APIs and the swarms of developers that consumed them. More recently, SaaS companies like Salesforce and Google have offered developers the ability to extend their platforms via API. And in the latest trend, new programming concepts like microservices and bots are centered on assembling applications via modular, distributed processes accessed via open APIs.

The latest surge in A.I. APIs opens up a whole new world of capabilities to software developers. Need an engine that can understand spoken language, recognize a single face from a crowd of thousands, pull meaning from a mountain of social media sentiment or act as the "brain" behind an industrial robot or self-driving car? Yesterday’s programmatic algorithms won’t cut it; you need machine learning and A.I.

Like yesterday’s code libraries, you could try to build A.I. platforms yourself -- if you had a few years and a dozen data scientists to throw at the problem. Or you can access A.I. engines like IBM’s Watson or Google’s TensorFlow “as-a-service,” taking advantage of the planet’s most advanced, fundamental CS work via an API call.

When one looks at the world of software in this way, the choice for most companies today is straightforward: spend years of effort and millions of dollars in expense duplicating extremely important -- but ultimately commodity, especially once it’s open-sourced -- computer science work, or instead focus on leveraging that work to develop and improve their own products and intellectual property. For most businesses, the choice is simple.

In this new future, the choice for developers also becomes clear. They can be software innovators, integrators or designers. These roles will slightly overlap in some ways given an individual’s skillset and company needs but by simplifying the roles, its easier to communicate how I envision our way forward.

- Innovators work out on the edge of research and development, creating new algorithms that have not yet been built and extending the capabilities of those that already exist. Fundamental CS work is always required and absolutely crucial but in today’s open source-driven software environment also increasingly a commodity.
- Designers focus on crafting new products, solutions and applications, using powerful technologies like machine learning and A.I. -- accessed via open APIs -- to identify new opportunities, create new solutions and solve problems in new ways altogether.

- Integrators take existing capabilities and build them into new products. Their role is to deal with the nuts and bolts of implementing the designs that bring innovations together. They do not design the algorithms or products; rather, they make them work.

It’s absolutely crucial for academia, businesses and even developers themselves to understand the distinctions between these three very different types of software developers. They represent three different skill sets and approaches to the business of software development.

For companies, hire wrong and the results can be devastating. Bring on an innovator to do an integration job and nine months later they’ll still be tinkering under the hood. Need some exploratory data work done? Hire an innovator, yes, but don’t expect them to build your industry’s next breakout product. For individual developers, mismatched skills and ambitions can be just as damaging, not only to someone's career path but day-to-day enjoyment of the job and task at hand.

More than ever in a computer science landscape defined by systems in artificial intelligence and big data manipulation and analytics, not all data scientists and programmers are created equal. That’s not a value judgement, i.e., good vs. bad. It’s about a difference in kind. Innovators, designers and integrators.

Tomorrow’s worlds of business -- and software development -- needs them all, and more importantly needs to be able to tell them apart and pick the right tool for the right job. The future of computer science depends on it.

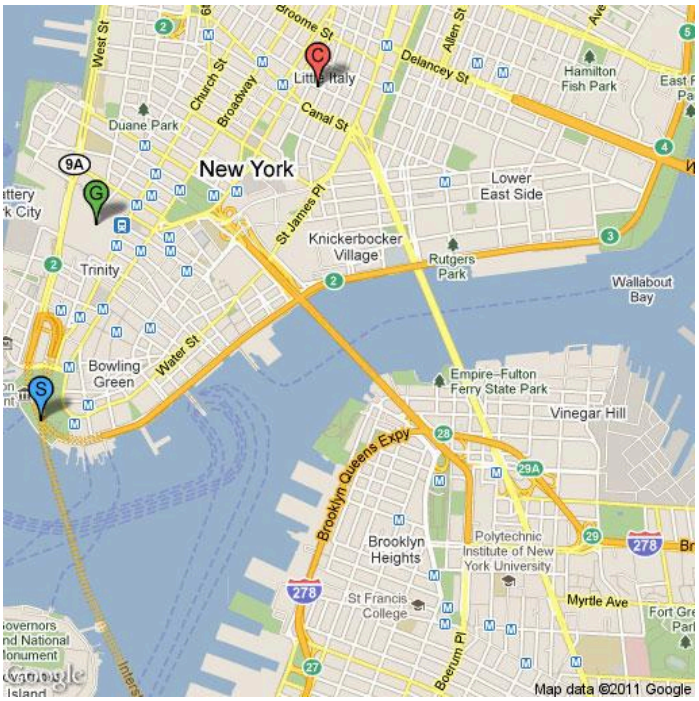
40 useful APIs for web designers and developers

Adapted from WebDesignerDepot by Cameron Chapman
https://www.webdesignerdepot.com/2011/07/40-useful-apis-for-web-designers-and-developers/

An application programming interface (API) is a set of rules and specifications that software programs can follow to communicate or 'interface' with each other. As developers are well aware, there are hundreds of APIs out there for doing almost anything you could imagine online. Some are better than others, and some are definitely more useful than others.

Below are forty of the most useful APIs out there. The included APIs will let you do everything from shortening a URL to displaying a book preview on your site to interacting with your Twitter account, and everything in between.

The Google APIs



Google offers dozens of APIs for web designers and developers. Some are specifically related to popular Google products, like Gmail and Analytics, while others are more specialized and aren't part of public programs. All are free to use, of course. You can view all of Google's APIs and code tools on their site directory.

- Feed API – The Google Feed API lets you download any public feed (including RSS, Media RSS, and Atom) and then combine them into mashups. It simplifies the mashup process by using JavaScript rather than more complex server-side coding.
- Places API – Google Places is a large directory of local businesses and attractions all around the world. The Places API lets you access that information and display it on your website, as well as display check-ins by users.
- Geocoding API – The Geocoding API lets you convert any address into geographic coordinates, which can then be used to place markers on a map.
- Tasks API – The Tasks API offers endpoints for reading, searching, and updating Google Tasks content and metadata.
- Analytics Management API – The Analytics Management API gives improved access to your Analytics data, and lets you fine-tune your requests to just pull the information and reports you need for your application.

- Blogger Data API – The Blogger Data API lets your application create and post new blog posts, edit or delete existing posts, and search for posts based on specific criteria.
- Books API – The Google Books API lets you integrate book searches into your application, and embed book previews on your site.
- Calendar API – The Calendar API gives access to many of the standard web interface tools and operations to your web app. Public calendar events can be searched and viewed without authentication, while authenticated sessions can access private calendars, as well as edit, create, or delete those calendars.
- Moderator API – Google Moderator is a tool for collecting ideas, questions, and recommendations from any size audience. The API allows your website or application to do the same.
- Prediction API – The Prediction API helps you make smarter apps that can analyze historic data and predict future outcomes. It can be used for things like recommendation systems, spam detection, upsell opportunity analysis, and more.
- Picasa Web Albums Data API – The PWA Data API can be used to create albums and upload, retrieve, or comment on photos, among other features. It's been used for everything from powering digital photo frames to full-featured mobile clients and more.
- Static Maps API – You don't always want an interactive map on your site. Sometimes a static map is just what you need. The Static Map API lets you embed static Google Maps onto your site, including custom styled maps.
- Directions API – The Directions API lets your users get directions from one point to another using a variety of travel modes from within your site or app, and doesn't require a Google Maps API Key.
- YouTube APIs – YouTube has two APIs available: Player APIs and Data API. The Player APIs allow you to have an embedded player, or a chromeless player that you can then customize within HTML or Flash. The Data API lets your app perform a lot of the operations available on YouTube, including uploading videos and modifying user playlists.
- Webmaster Tools API – The Webmaster Tools API lets your client application use a variety of Webmaster Tools functions, including viewing sites, adding and removing sites, verifying site ownership, and submitting and deleting Sitemaps.
- Google Web Fonts API – The Web Fonts API makes it easy to add free web fonts to your website or application. Their collection of fonts grows on a continuous basis and already includes a huge variety.
- OpenSocial – OpenSocial can be used for building social applications, creating social app platforms, and sharing and accessing social data.

The Yahoo! APIs

Like Google, Yahoo! offers a number of APIs useful for developers. All are free to use and can help you integrate a variety of Yahoo!-owned web services into your app, including Flickr and Delicious.

- Answers API – The Answers API lets you access the collective knowledge contained within Yahoo! Answers. You can search Answers based on a variety of criteria (including specific user, category, and more), set your app to watch for new questions in the categories you choose, and track new answers from specific users.
- Contacts API – The Contacts API lets you access relationships in your Yahoo! address book. It reads a user's Contacts information while respecting user privacy and permission settings.
- Delicious API – The Delicious API gives read/write access to Delicious bookmarks and tags.
- Fire Eagle Developer API – The Fire Eagle API helps you create location-aware websites and applications.
- Flickr API – With the Flickr API you can view, search, and manipulate photo tags, display photos from a specific user or group, and more.
- Local API – The Local API lets you access location-based information and user-contributed content.
- Maps APIs – Yahoo! offers a number of APIs for their Maps services, including an Ajax API, a REST API, and a No Coding API.
- Meme API – Meme is a multimedia light-blogging platform. The API lets you create apps that can read, post, and repost content through Meme.
- PlaceFinder – The PlaceFinder API, similar to Google's GeoCoding API, and lets you convert a street address into geographic coordinates.

More APIs

Yahoo! and Google aren't the only ones offering powerful APIs for designers and developers. A number of social media sites and others have their own API(s), including Twitter, Facebook, Yelp, Bit.ly, and many more.

- Twitter API – Twitter has a host of developer tools surrounding their API that let you create apps that interact with virtually any of Twitter's functions.
- Facebook APIs – Facebook offers APIs for working with Credits, Ads, Chat, and more, including a couple of legacy APIs that are no longer actively supported. Also found here is the Graph API, which is the backbone of the Facebook Platform, and enables your app to read and write data to Facebook.
- Awe.sm – Awe.sm offers a number of developer APIs for integrating their social media campaign tracking tools into your app or website.
- Foursquare APIv2 – The Foursquare API not only allows you to create apps that interact with the Foursquare service, but also to use Foursquare's place-related database as a standalone service.

- Ning API – Ning offers a set of APIs for developing desktop and mobile apps, custom network features, profile apps, and data importers.
- Soundcloud API – Soundcloud's API includes tools for sharing, streaming, and customizing the Soundcloud player for your website.
- Klout API – The Klout API makes a variety of data available to developers, including Klout Scores, Network Influence, Amplification Probability, True Reach, and more.
- Social Mention API – The Social Mention API provides a stream of real-time search data from a number of social media services for integration into other applications. It's free for personal and non-commercial use.
- Opus Social Media API – The Opus Social Media API can serve as a basis for developing a social networking and digital media site or app.
- Digg API – Digg offers an API that lets you access their newsfeeds for your own sites and applications.
- Yelp API – The Yelp API lets you access business listing info, business ratings, and review excerpts from Yelp in your application or website.
- Zillow Neighborhood Information APIs – Real estate site Zillow offers APIs that give access to neighborhood information that can be integrated into other applications. (They also offer a number of other APIs, including postings, property details, home valuations, and more.)
- Tropo – The Tropo API adds Twitter, IM, voice and SMS functionality to a variety of common programming languages. Development is free, though sending messages varies in price (with Twitter and IM messages currently free).
- Bit.ly API – Bit.ly offers an API for integrating URL shortening into your app or site.