

API Evangelist History of APIs

This is a walk through on the history of web APIs, highlighting some of the pioneers who have made the API space what it is.

- The History of APIs
- Commerce
- SalesForce
- Ebay
- Social
- del.icio.us
- Flickr
- Facebook
- Twitter
- Business and Marketing
- ProgrammableWeb
- Mashery
- Mapping
- Google Maps
- Cloud Computing
- Amazon S3
- Amazon EC2
- A Mobile World
- Foursquare
- Instagram
- Twilio
- Understanding Our API History



The History of APIs

History is everything. Understanding where we have come from is critical to knowing where we are going. While pushing forward with the latest technology, it is always healthy to pause and take a look at that past--so let's take a look at the history of APIs.

"The concept of an API pre-dates even the advent of personal computing, let alone the Web, by a very long time! The principal of a well documented set of publicly addressable "entry points" that allow an application to interact with another system has been an essential part of software development since the earliest days of utility data processing. However, the advent of distributed systems, and then the web itself, has seen the importance and utility of these same basic concepts increase dramatically." - Martin Bartlett

When I talking about APIs, I'm focused one version that was born out of the enterprise, during the Service Oriented Architecture (SOA) movement, which sometime around 2000, a portion of the SOA experiment left the enterprise and found a more fertile environment in the world of start-ups.

Where we stand in 2016, there are some obvious technical reasons of why web APIs are finding success in companies of all shapes and sizes, and even within government; but not all the reasons for this success is technical. There are many other, less obvious aspects of web APIs that have significantly contributed to their success, things we can only learn by closely studying the past and looking at why some of the pioneers of web APIs were successful.

In 2016, we need to make sure we emulate the best practices that have been established over the last 13 years, and not ignore some of the critical variables that contribute to the success of early API providers like Amazon, Salesforce, Ebay and Twitter--much of what is still working in 2016.

When we look back, we can begin see some clear patterns emerge that have defined the industry. Patterns that need to be emulated, and some that should be avoided.

Commerce

As soon as the first .COM bubble burst, platforms were looking for innovative ways to syndicate products across e-commerce web sites, and web APIs, built on the backs of existing HTTP infrastructure proved to be the right tool for the job.

With this in mind, a handful of tech pioneers stepped up to define the earliest uses of APIs for sales and commerce management, kicking-off a ten year evolution that we can now call the early history of web APIs.

However, even with the early success of APIs, they wouldn't reach maturity without several other critical ingredients that would prove as important as essential commerce variables like products, payments and sales.

SalesForce

February 7th, 2000 Salesforce.com officially launched at the IDG Demo 2000 conference.

Salesforce.com launched its enterprise-class, web-based, sales force automation as a "Internet as a service". XML APIs were part of Salesforce.com from day one. Salesforce.com identified that customers needed to share data across their different business applications, and APIs were the way to do this.

A screenshot of the original Salesforce.com login page. It features a green header with the text "salesforce.com" and "Secure Customer Login". Below the header is a form with fields for "User Name" and "Password", and a "Go!" button. To the right of the form are links for "Home", "Accounts", "Contacts", "Opportunities", and "Forecasts". At the bottom right, there is a red banner with the text "Just Sign On!" and "Online Sales Force Automation". Below the banner, smaller text reads "Exploit the power of the Internet to access, manage and...".

Marc R. Benioff, chairman and founder of salesforce.com stated, "Salesforce.com is the first solution that truly leverages the Internet to offer the functionality of enterprise-class software at a mere fraction of the cost."

Salesforce.com was the first cloud provider to take an enterprise class web application and API and deliver what we know today as Software-as-a-Service.

Even with SalesForce being the first mover in the world of web APIs, they are still a powerhouse in 2016. SalesForce continues to lead when it comes to real-time APIs, testing, deployment and most recently taking a lead when it comes to mobile application development and backend as a service (BaaS).

Ebay

On November 20, 2000, eBay launched the eBay Application Program Interface (API), along with the eBay Developers Program.

The eBay API was originally rolled out to only a select number of licensed eBay partners and developers.

As eBay stated:

"Our new API has tremendous potential to revolutionize the way people do business on eBay and increase the amount of business transacted on the site, by openly providing the tools that developers need to create applications based on eBay technology, we believe eBay will eventually be tightly woven into many existing sites as well as future e-commerce ventures."

The launch of the eBay API was a response to the growing number of applications that were already relying on its site either legitimately or illegitimately. The API aimed to standardize how applications integrated with eBay, and make it easier for partners and developers to build a business around the eBay ecosystem. eBay is considered the leading pioneer in the current era of web-based APIs and web services and still leads with one of the most successful developer ecosystem today.

Social

As API driven commerce platforms were still getting their footing, and understanding the best way to use APIs, a new breed of technology platforms emerged in the area content and messaging on the web in a way that was very user centric and socially enabling.

Publishing user generated content, and the sharing of web links, photos and other media via APIs emerged with this birth of new social platforms between 2003 and 2006. This was an entirely new era for APIs, one that wasn't about money, it was about connections.

These new API driven, social platforms would take technology to new global heights, and ensure that applications from here forward, would all always contain essential social features, that were defined via their platform APIs.

Social, was an essential ingredient the API industry was missing.

del.icio.us

del.icio.us is a social bookmarking service for storing, sharing and discovering web bookmarks to web pages, that was founded by Jousha Schachter in 2003.

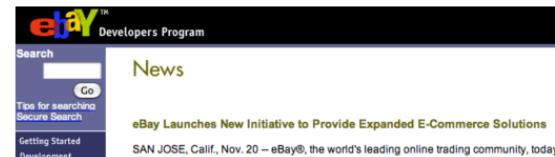
Del.icio.us implemented a very simple tagging system which allowed users to easily tag their web bookmarks in a meaningful way, but also established a kind of folksonomy across all users of the platform. Which proved to be a pretty powerful way for cataloging and sharing web links.

The innovative tagging methodology used by del.icio.us allowed you to pull a list of your tags, or public web bookmarks by using the URL [http://del.icio.us/tag/\[tag name\]](http://del.icio.us/tag/[tag name]). So if I was searching for bookmarks on airplanes, I could <http://del.icio.us/tag/airplane> and I would GET a list of all bookmarks that have been tagged airplane. It was simple.

When it came to the programmatic del.icio.us interface, the API was built into the site, creating a seamless experience—if you wanted the airplane tags via HTML you entered <http://del.icio.us/tag/airplane>, if you wanted RSS of the tags you entered <http://del.icio.us/rss/tag/airplane>, and if you wanted XML returned you used <http://del.icio.us/api/tag/airplane>. This has changed with the modern version of Delicious API.

del.icio.us was the first, concrete example of how the web could deliver HTML content, alongside machine readable like RSS and XML, using a URL structure that was simple and human readable. This approach to sharing bookmarks would set the stage for future APIs, in making APIs easy to understand for developers and even non-developers alike. Any slightly technical user could easily parse the XML or RSS, and develop or reverse engineer widgets and apps around del.icio.us content.

del.icio.us has been sold twice since its early popularity, which included to Yahoo! in 2005 and AVOS Systems on April, 2011. However del.icio.us was one of the pillar platforms that ushered in the social era of the API movement, establishing sharing via APIs as critical to the API economy, but also showing that simple rules when it comes to API design.

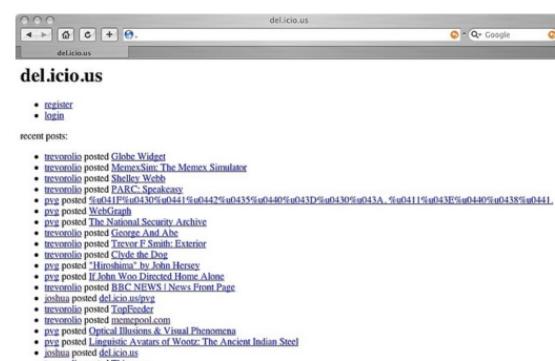


eBay Launches New Initiative to Provide Expanded E-Commerce Solutions
SAN JOSE, Calif., Nov. 20 — eBay®, the world's leading online trading community, today launched a new initiative to provide expanded e-commerce solutions for developers. The groundbreaking initiative will initially be rolled out to the eBay API.

"Our new API has tremendous potential to revolutionize the way people do business on eBay," said eBay President of Technology and Research, Steve Jobs. "We're providing the tools that developers need to create applications based on eBay technology."

The API will provide three basic benefits. First, it will allow eBay to be fully integrated into money from the start-up process. For example, a site selling musical instruments through 19 million registered users.

The API will provide three basic benefits. First, it will allow eBay to be fully integrated into money from the start-up process. For example, a site selling musical instruments through 19 million registered users.



Flickr

In February 2004 the popular photo sharing site Flickr launched. Six months later they launched their now infamous API, and six months after that, they were acquired by Yahoo.

Flickr was originally created as an online game, but quickly evolved into a social photo sharing sensation.

The launch of the RESTful API helped Flickr quickly become the image platform of choice for the early blogging and social media movement by allowing users to easily embed their Flickr photos into their blogs and social network streams.

The Flickr API is the driving inspiration behind the concept of BizDev 2.0, a term coined by Flickr co-founder Caterina Fake. Flickr couldn't keep up with the demand for its services, and established the API as a self-service way to deal with business development.

The core concepts established by Flickr using its API would transcend the company and its acquisition by Yahoo. Business development using APIs is embedded in the philosophy of the business of APIs pushing APIs to something beyond technical.

APIs became something that any company could use to actually conduct business with its partners and the public, but we still had a ways to go before APIs would grow up.



Share your photos instantly!



- No downloads
- Drag and drop
- Easy to use

Facebook

On August 15th 2006, Facebook launched its long-awaited development platform and API. Version 1.0 of the Facebook Development Platform allowed developers access to Facebook friends, photos, events, and profile information for Facebook.

The API used REST, and responses were available in an XML format, following common approaches by other social API providers of the time.

Almost immediately, developers began to build social applications, games, and mashups with the new development tools.

The Facebook Development Platform gave Facebook an edge over then popular competitor MySpace, and established itself as the top social gaming platform with games like Farmville.

While the Facebook API and platform is considered by many developers to be unstable, it continues to play a significant role in the evolution of the entire platform with applications and partnerships that drive new features and experiences on Facebook.

facebook developers

Developer Home
Documentation
Developer Account
Discussion Group
Product Directory
Open Source FAQ

Welcome to Facebook Developers (beta), where you can create your own projects using our API (Application Programming Interface).

The Facebook Development Platform allows you to create applications utilizing the following data from Facebook:



Developers: Get Started!

- Sign up for an account key.
- Learn about the documentation.
- Join the discussion group.
- Show off what you made in our directory.

Your First Visit?

- Learn what the Facebook Development Platform is all about.
- Check out some interesting applications in our directory.

facebook.com/login

Featured Product



MoochSpot
By Steve and Co.
Forgot how much you owe from that last group dinner? This helps you and your friends easily track expenses.

Latest News

New batch of updates
1:46am Tuesday, Sep 26

We've received lots of great feedback from all of you and made some changes to the API that we think you'll like. This round of changes includes:

- New fields for users.getInfo, status, wall_count, notes_count, and religion.
- Since we added wall_count to user.getInfo, we are deprecating the wall.getCount function. We think you'll like this new variation better. To that end, we're setting the wall_count of a bunch of users at once with a single call. We'll leave it in for a couple more days to give you guys time to switch over.
- To make it easier for your users to message each other, you can now link users to message pages with pre-filled messages, as well as the form to post new wall posts, as described here.
- We added 'type' attributes to some of our XML responses. Initially this will help some of you with your XML parsing.
- In response to some of you asked for it, we added session ping to our list of functions - mainly just useful for desktop applications.

Keep the feedback coming, and enjoy!

[Read Older Stories](#)

a Mark Zuckerberg production
Facebook © 2006

Twitter

On September 20, 2006 Twitter introduced the Twitter API to the world.

Much like the release of the eBay API, Twitter's API release was in response to the growing usage of Twitter by those scraping the site or creating rogue APIs.

Twitter exposed the Twitter API via a REST interface using JSON and XML.

In the beginning, Twitter used Basic Auth for API authentication, resulting in the now infamous Twitter OAuth Apocalypse almost four years later, when Twitter forced all those using the API to switch to OAuth.

In four short years Twitter's API had become the center of countless desktop clients, mobile applications, web apps, and businesses -- even by Twitter itself, in its iPhone, iPad, Android apps via its public website.

Twitter is one of the most important API platforms available, showing what is possible when a dead simple platform does one thing well, then opens up access via an API and lets an open API ecosystem build the rest.

Twitter is also one of the most cautionary tales, of how your API ecosystem can also begin to work against you, unless you properly address the political considerations of an API ecosystem as it grows.

Introducing the Twitter API

Wednesday, September 20, 2006

Some smart folks out there on the Internet have been putting together interesting projects like this map and this page without any help from us so we thought it was high time to release an API.

THE INTERNET
A SERIES OF TUBES
nX-1 nX-2 nX-3

Business and Marketing

As APIs evolved from commerce, through social, it was clear that the industry was going to need some standardizing, by introducing some common business practices. The industry needed to standardize how APIs were deployed as well as provide marketing to help get the word out about the potential of APIs and common business practices.

The establishment of common business and marketing practices for the API space took a lot of grassroots outreach as well as storytelling of the behalf of APIs, companies and the industries they rose out of. There were two separate API pioneers that stepped up to help define the API industry we know today, between the years of 2005 and 2012.

ProgrammableWeb

While writing about the history of APIs, it is easy to be so focused on just APIs, that you overlook the single most important player in the entire history of the web API—ProgrammableWeb.

In July 2005, John Musser started ProgrammableWeb. According to his original about page:

ProgrammableWeb is a web-as-platform reference site and blog delivering news, information and resources for developing applications using the Web 2.0 APIs.

I started this site because I couldn't find what I was looking for: a technology focused starting point for web platform development. (For a bit more see my initial post.) Although no guarantees, the last time I started a reference site it somehow became Google's highest rated link on the topic. Given that this site will be a collaborative effort with community input as well, this can be what we make it.

I hope you find the site useful.

John Musser - Seattle, August 2005

John's original blog post on why he started ProgrammableWeb, says it all: Why? Because going From Web Page to Web Platform is a big deal.

Web APIs are a big deal! Whether its social networking, government, healthcare or education—having a programmable platform to make data and resources available will be a critical part of how commerce and society operates from here on forward.

John made a early decision to showcase open and RESTful approaches to deploying APIs vs.

parallel attempts of Service Oriented Architecture (SOA) and Web Services, and focused on telling stories about open APIs—way before it was the thing to do in Silicon Valley.

When I started API Evangelist in July 2010 (5 years after PW), and started talking about the business of APIs, the technology of web APIs was already widely accepted in Silicon Valley, because of the stories that have told on ProgrammableWeb.

As we progress through 2016, a year in which I think we can confidently say APIs are moving mainstream, and I feel we owe much of the success to ProgrammableWeb. The stories John, Adam and other writers have been telling on ProgrammableWeb have been crucial to quantifying and defining the API industry—allowing us all to build, iterate and move things forward.

Without stories around the technical, business and politics of APIs, these virtual interfaces would not have been able to find a place in our real life worlds.

Mashery

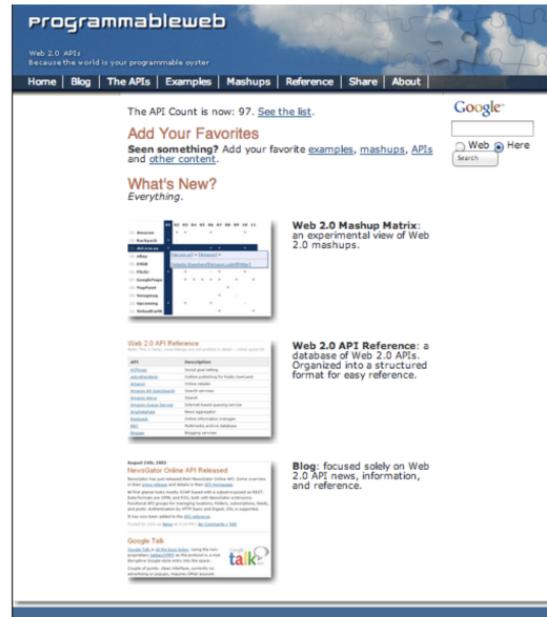
In November 2006, API the first API service provider Mashery came out of "stealth mode" to offer documentation support, community management and access control for companies wishing to offer public or private APIs—from a blog post in TechCrunch titled API Management Service is Open for Business.

At this point in time, in 2006, we were moving from the social period of APIs into the cloud computing phase with the introduction of Amazon Web Services. It was clear that the world of web APIs was getting real, and there was opportunity for companies to offer API management as a service.

While there were tools for deploying APIs, there was no standard approach to managing your API deployment. Mashery was the first to bring a standard set of services to API providers, that would help set the stage for the future growth of the API industry.

It would take almost six more years before the API industry would come of age, in which Mashery significantly helped contribute. The space we all know today was defined by early API commerce pioneers like SalesForce and Amazon, social pioneers like Flickr and Delicious, and from Mashery who helped define what is now known as the business of APIs.

In 2016, Mashery was acquired by Intel, helping recently validate that the API industry truly has come of age.



Mapping

One early API pioneer saw the need for provider web developers with a simple JavaScript map that could assist users in navigating online, localized content or even navigate in the real world.

Web developers quickly saw the potential of embeddable maps, and found ways to hack these mapping sources to innovate and build the web properties users desired, focused on solving the local problems we all face daily.

This early use of APIs in providing mapping tools and services for developers laid the groundwork for much of the early mobile developer talent that would drive the coming mobile API period.

The world is your JavaScript-enabled oyster

June 29, 2005

Posted by Bret Taylor, Google Maps Product Manager

If you like [Google Maps](#), but think you could do something better, now's your chance. Check out the [Google Maps API](#), which lets web developers put Google Maps on their own sites, just like [housingmaps.com](#) and [chicagocrime.org](#). You can also reach out to other API developers and the Google Maps team in the [API discussion group](#).

Google Maps

On June 29th, 2006, Google launched Google Maps API allowing developers to put Google Maps on their own sites using JavaScript.

The API launch was just shy of 6 months after the release of Google Maps as an application, and was in direct response to the number of rogue applications developed that were hacking the application. Google Maps was immediately so popular that developers hacked the JavaScript interface and developed application such as [housingmaps.com](#) and [chicagocrime.org](#). The demand was so great for information on hacking Google Maps it spawned books such as *Mapping Hacks* and *Google Hacks* from O'Reilly.

Google Maps API started a trend of API mashups with its valuable location based data, with over 2000 mashups to date.

The API demonstrates the incredible value of geographic data and mapping APIs, as well as the power users can have in influencing the direction an application or API takes. Lars Rasmussen, the original developer of Google Maps commented how much he learned from the developer community by watching how they hacked the application in real-time, and they took what they learned and applied it to the API we know today.

Few other companies have the resources to tackle a problem like mapping the world's resources and delivering a reusable, API driven resource, like Google did. Google has played many roles in moving forward the API space, but Google Maps played a pivotal role in the history of APIs.



Cloud Computing

As APIs were generating social buzz across the Internet, Amazon saw the potential of an RESTful approach to business, internalized and saw APIs in a way that nobody had seen them before—giving birth to an approach to using APIs that was much more than just e-commerce, it would re-invent the way we compute.

Amazon transformed the way we think about building applications on the web, delivering one of the essential ingredients we needed for APIs to work, by putting APIs to work. What we now know as cloud computing changed everything, and made the mobile, tablet, sensor and other API driven realms possible.

Cloud computing, was an essential ingredient the API industry was missing.

Amazon S3

In March, 2006 Amazon launched a new web service, something completely different from the Amazon bookseller and e-commerce site we've come to know. This was a new endeavor for Amazon: a storage web service called Amazon S3.

Amazon S3 provides a simple interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. It gives developers access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of websites.

Amazon S3 or Simple Storage Service was initially just an API. There was no Web interface or mobile app. It was just a RESTful API allowing PUT and GET requests with objects or files.

Developers using the Amazon S3 API were charged \$0.15 a gigabyte per month for storing files in the cloud.

With this new type of API and billing model, Amazon ushered in a new type of computing we now know as cloud computing.

This also meant that APIs were no longer just for data or simple functionality. Now they could be used to deliver computing infrastructure.

AWS Blog Amazon S3

by Jeff Barr | on 14 MAR 2006 | in Announcements | Permalink | [Share](#)

Earlier today we rolled out [Amazon S3](#), our reliable, highly scalable, low-latency data storage service.

Using SOAP and REST interfaces, developers can easily store any number of blocks of data in S3. Each block can be up to 5 GB in length, and is associated with a user-defined key and additional key:value metadata pairs. Further, each block is protected by an ACL (Access Control List) allowing the developer to keep the data private, share it for reading, or share it for reading and writing, as desired.

The system was designed to provide a data availability factor of 99.99%; all data is transparently stored in multiple locations. S3 is a very cost-effective data storage solution. Using S3's economical pay-as-you-go model, storing 1 GB of data for 1 month costs just 15 cents. Transferring data in and out of the system costs 20 cents per GB. As Colin Faulkingham notes, there's also a BitTorrent interface.

I've got to catch a plane to Silicon Valley in a few minutes, or I'd write a lot more. You can read more at [TechCrunch](#), [Business Week](#), and at [Scripting News](#). We are looking forward to seeing developers create all sorts of cool applications with this new system.

— Jeff;

Amazon EC2

In August 2006, shortly after Amazon launched its new cloud storage service Amazon S3, the company followed with a new cloud computing service dubbed Amazon EC2 or Elastic Compute Cloud.

Amazon EC2 provides re-sizeable compute capacity in the cloud, by allowing developers to launch different sizes of virtual servers within Amazon data centers.

Just like its predecessor Amazon S3, Amazon EC2 was just a RESTful API. Amazon wouldn't launch a web interface for another three years.

Using the Amazon EC2 API developers can launch small, large and extra large servers and pay for every hour that the server is running.

Amazon EC2, combined with Amazon S3 has provided the platform for the next generation of computing with APIs at the core.

Announcing Amazon Elastic Compute Cloud (Amazon EC2)

- beta

Posted On: Aug 24, 2006

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. Just as Amazon Simple Storage Service (Amazon S3) enables storage in the cloud, Amazon EC2 enables "compute" in the cloud. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use.

A Mobile World

With the introduction of iPhone and Android smart phones, APIs had evolved from powering e-commerce ,social and the cloud to delivering valuable resources to the mobile phones in our pockets, that are quickly becoming commonplace around the globe.

APIs make valuable resources modular, portable and distributed, making them a perfect channel for developing mobile and table applications of all shapes and sizes.

A small group of API driven technology platforms have helped defined the space and won over the hearts and minds of both developers and the end users of the applications they develop.

Foursquare

In March 2009 Foursquare launched at the SXSW interactive festival in Austin, TX.

Foursquare is a location-based mobile platform that makes cities more interesting to explore, by checking in via a smartphone app or SMS, users share their location with friends while collecting points and virtual badges.

In November 2009 after a round of angel funding from several investors including Union Square Ventures and O'Reilly AlphaTech Ventures, Foursquare launch their API.

At the time of API launch, Foursquare had an impressive set of applications developed by a closed group of partners including anAndroid application and augmented reality with Layar.

Even with growing competition from early mover Gowalla and major players like Facebook and Google, Foursquare has emerged as the dominant mobile location-sharing and check-in platforms.

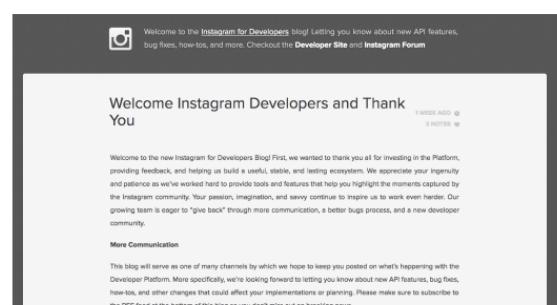


Instagram

On October 6, 2010 Instagram launched its photo-sharing iPhone application.

Less than three months later, it had one million users. Kevin Systrom the founder of Instagram focused on delivering a powerful, but simple iPhone app that solved common problems with the quality of mobile photos and users' frustrations with sharing. Immediately many users complained about the lack of central Instagram web site or an API, with Instagram remaining firm on focusing its energy on the core iPhone application.

In December a developer name Mislav Marohni? took it upon himself to reverse engineer how the iPhone app worked, and built his own unofficial Instagram API. By January Instagram shut down the rogue API and announced it was building one of its own. Then in February of 2011, Instagram released the official API for the photo platform. Within days many photo applications, photo - sharing sites, and mashups built around the API started showing up. Instagram became a viral iPhone app sensation, but quickly needed an API to realize its full potential. Asserting the platforms place in history as one of the defining players in the mobile period of APIs.



Twilio

In 2007, a new API-as-a-product platform launched, called Twilio, which introduced a voice API allowing developers to make and receive phone calls via any cloud application. Over the last 3



years Twilio has additionally released a text messaging and SMS short code API, making itself a critical telephony resource in many developers toolbox.

Twilio is held up as a model platform to follow when evangelizing to developers. Twilio has helped define which technical and business building blocks are essential for a healthy API driven platform, set the bar for on the ground evangelism at events and hackathons, and worked hard to showcase, support and invest in its developer ecosystem.

Alongside Foursquare and Instagram, Twilio has come to define mobile application development, helping push APIs into the mainstream. While Twitter has sometime been held up as a cautionary tale when it comes to APIs, Twilio has demonstrated, that when done right, API driven ecosystems do work.

By 2011 the bar for delivering APIs, via HTTP, has been well established by early pioneers like SalesForce and Amazon, but Twilio has shown how mature the business of APIs has become with its evolution into the mobile period. However, mobile development via APIs, owes its roots to the foundation laid by the commerce, social and cloud API pioneers.



Understanding Our API History

Understanding history is critical to understanding where we are going. Calling this document the "history" of web APIs seems kind of silly. We are actually talking about a span of 13 years. But there are so many important lessons to be learned from the approach of these API pioneers, we can't ignore this history. If the technologists had their way, APIs would have purely been successful in the period of commerce, but with the radical innovation of companies like Amazon, Twitter and Twilio, we now understand that APIs needed several essential ingredients to succeed: commerce, social, cloud and mobile.

Of course, all of this has to make money, but APIs need to be scalable, while also delivering the meaningful tools, services and resources that are important end users, otherwise none of this matters. As we stand solidly in the mobile period of API evolution, looking at the evolution to a period that will be more about devices and an internet of things (IOT), we need to understand our own history and how we've gotten here, to make sure we the right decisions for what is next.

Web APIs are about delivering valuable, meaningful, scalable and distributed resources across the World Wide Web. While Silicon Valley keep pushing forward with the next generation of technology solutions, we need to make sure that we know our past.