* Historia, kort om vad API är, detta måste förklaras pedagogiskt. Analogier etc. I största mån copy paste.

Things are changing so fast in the world of software that it’s difficult to keep track of all the moving parts. But, when you look closely, much of that change hinges on APIs.

APIs have long been a hot topic among developers. However, the majority of people are not as thoroughly educated on their history and their use as they should be, even though they most likely interact with them every day. This trend has resulted in a widespread underestimation of APIs’ impact on Web applications—their evolution, functionality and testing.

It’s understandable that APIs are poorly understood, they were never meant to be grokked by humans in the first place. Unlike Web applications themselves, APIs are built for computer consumption rather than direct user interaction. They live behind the scenes, and have traditionally lurked in the shadows of software development since their inception – snuck into code in every industry from government to gaming without management even knowing they existed. Yes, there is something of an underground world of APIs. But that world is quickly coming into the light and, along with it, the need for a broader understanding of how APIs affect the performance and testing of our Web applications.

What exactly is an API?

An API – Application Programming Interface – is software programming written to function as a communication bridge between Web applications. An API pulls or shares data from within the firewall of a company or outwards over the Internet. In other words, it is software code written purely to be consumed by other Web applications. In a broad sense, the API of a Web application makes its best services accessible to an entire world of outside developers, so they can build those services into their own programs.

The History of APIs

Like most origin stories, how the API first emerged on the tech scene and what concept it evolved out of is up for interpretation. Some trace the earliest pre-tech spark to franchises like hotel chains attempting to utilize their services in different locations, others to the stock market machines that tickered out real-time data for media and investment firms, and still others find the original inspiration in catalogue based mail-order shopping. Either way, it was the entrepreneurial spirit of those who grappled with how to utilize a company’s underlying services in new and profitable ways that drove API development from whatever beginning.

However, Web APIs as we know them in software today have a fairly short history. Over the last decade, businesses like Salesforce, eBay, Flickr, Amazon, Facebook, and Google Maps have pioneered the modern Web API. Google Maps is probably the API we can most easily identify. Yes, we likely spend more time using Amazon's or Facebook's, but we are accustomed to seeing Google Maps adapted for site-specific purposes around the Web. Rather than just an ecommerce link or a social media impression icon, Google Maps are clearly being altered in different ways for applications. In fact, Google Maps was the first to widely demonstrate the power of “API mashups” — mashing together data from different Web applications to make a new one; like a map showing all the locations of bands playing tonight, a map that tracks my running routes, a map that highlights all the hiking trails in the nearby State Park. With this one little API, developers can harness the awesome power of Google Maps to fulfill our wildest geographical dreams. But Google Maps didn’t come up with their API idea themselves.

After the enormous popularity of Google Maps overwhelmed the Web in 2006, the descent of developers hacking their code for use on other sites demanded a response from Google. Google Maps was forced to release an API so that developers could utilize their local map services without the need to hack. That API and others of its time like Twitter’s changed the face of the Internet forever— effectively starting the trend of allowing developers to utilize the backend services of other Web applications in different ways for different purposes. It spawned a whole new field of online social and business partnerships.

In fact, external pressure to produce an API has been a major force behind many API developments. Whether it’s the public demanding freedom of information from its government, or ecommerce and social media trends threatening to leave all who won’t produce an API in economic dust, the necessity of using and producing APIs is only growing. And that’s a good thing. As one of our favorite API evangelists, Kin Lane says, “APIs bring together a unique blend of technology, business and politics into a transparent, self-service mix that can foster innovation.” Innovation indeed! Just think about the web without them. It hardly seems like a web without the underlying connective tissue of APIs. And I haven’t even mentioned cloud computing or mobile yet. If we consider all that’s happened over the last few years in API based mobile alone—with all the apps and games and so much cool stuff that it makes your head spin—

API based Web application innovation is clearly just getting started. However, we are seeing greater and greater needs for regulations and security for APIs. Not to rain on anyone’s parade but, watch out…they are coming.

The abstract concept behind an API, as a programmable access point to invoke another application, has been in existence since the early days of computer science. However, the ways and means of implementing them have gone through many transformations over the last few decades.

The history of web-based APIs, as we know today, can be traced back to the late 1990s when Salesforce launched its web-based sales automation tool. This application marked the beginning of the SaaS (Software as a Service) revolution. The World Wide Web powered the underlying infrastructure enabling this newfound way of delivering software.

Before the World Wide Web and the Internet, APIs did exist, albeit mostly in the form of proprietary protocols supporting small distributed computing networks that spanned a limited area.

Whether you look at API in the pre or post Internet era, the purpose remains the same. An API enables an API provider to expose its services, such that external systems can call the API provider to avail those services. At an Internet-scale, where developers from across the world build applications and host them over the World Wide Web, APIs are the future of distributed computing paradigm.

* Säkerhet: Ove, lägg till här angående API:er som requestar mot http gentemot https, samt authenticated API:er. Gör en bedömning i vilket omfång API:er är säkra. En basic request till ett unauthenticated API över http är sällan ett problem. Sänds ingen information värd att dölja så är det i det allra flesta fall okej. Pressa dock på att https finns av en anledning och ska användas till största möjliga mån.

Värt att kolla på är olika authentication metoder man kan använda vid API calls etc. T.ex. Basic Auth, JWT, Cookies osv.

* Värt att ta upp: Stateful och icke stateful, deklarerade responser och icke deklarerade responer.

Typ REST vs SOAP, JSON vs XML vs Protobuf, vad är skillnaden, vad är positiva samt negativa.