APIs and their business relevance

Table of Contents

last modified: 2023-05-15

1. Definition of API

API: acronym for **Application Programming Interface**. An API is the way to make software programs “easy to plug and share” with other programs. An API is simply a group of rules (you can also call it a convention, or an agreement…) which programmers follow when writing the part of their code which is in charge of communicating with other software. These rules are then published (on a webpage for example), so that anyone who needs to connect to the program can learn what rules to follow.

Is an API simply a way to write code to interface with other programs? Yes. Why the fuss then? Having conventions on how to write a software so that somebody can plug it to its own software is one thing. APIs as an aspect of software design is a classic topic in computer science, but we are not

concerned with this here. APIs we are going to discuss are about communication between distant computers, in a business context. To understand better their business relevance, it is useful to recall a brief history of APIs:

2. The origin of APIs

Companies which need to exchange data is nothing new. Manufacturers, retailers, banks, … they need to exchange information at regular interval. Sending invoices, receiving receipts for merchandise, and many other administrative records generated in the course of business. These receipts, invoices… can be printed and mailed (this solution still exists of course). With informatics developing in the 1970s and 1980s, a new system emerged: the exchange of information via computers: Electronic Data Interchange.

a. EDI: Electronic Data Interchange

EDI is not an exchange of file attachments in emails or via a file transfer on a website, because emails and websites did not exist yet! (emails and the Web were adopted by firms in the late 1990s). Instead, exchanging data via EDI consisted in using complex electronic tools (like the fax but even more complicated) because:

- each industry has its own protocol to exchange data (one protocol for logistics, one for payments, one for this or that retailer chain, etc.)
- you need a dedicated device or software for each EDI protocol, and these are not given for free
- EDI protocols can vary from one country to another
- EDI protocols are controlled by industry associations which do not adopt innovation quickly
- EDI protocols created "closed systems": a company A can connect to company B via an EDI only if the two have a pre-agreement to use this EDI.

To summarize: EDIs are fragmented, complicated to implement, slow to evolve, expensive and restricts the communication to a "club" of partners who agreed to use it. EDIs still exist, especially in large B2B industries like transportation, but it lost in popularity in the wider economy because… APIs have arrived.

b. The emergence of web APIs

In the late 1990s and early 2000s, Internet and the World Wide Web expanded dramatically. More and more servers in different parts of the world needed to exchange data with each other, and this required to use interfaces more convenient than EDIs. It became increasingly convenient to define simple and universal conventions that everyone could learn and follow to standardize these exchanges, for free and easily. That is what **web APIs** do. They are also often called:

- **API** for short
- **web services**
- **REST** API (see below for this last one).

A **web API** extends the logic of the APIs we have seen in the beginning of this document, to software communicating via the web. To recall, an API is a convention followed when writing a software, making this software available to other software.

Example: the API of Microsoft PowerPoint enables the import of Excel tables in pptx documents, because the API of Powerpoint plugs to the API of Excel. In this example Excel and Powerpoint are supposed to be installed on the same computer of course!

A Web API is an API which enables two pieces of software to communicate, via Internet. **They do not need to be installed on the same computer.**

c. The benefits of a web API compared to an EDI

Unlike an **EDI**, a web API drops any industry-specific concern. Web APIs are just a convention to send and receive data over the Internet, without any saying on the content of the data. The data sent and received can be invoices, webpages, train schedules, audio, video… whatever. Contrary to an EDI, a company creating a web API can choose to leave its access open (remember that EDIs need the two parties to have a pre-established agreement). So that a potential client interested in using the web API of a company can set it up in a couple of clicks, instead of waiting weeks or months before a contract is signed and the EDI is setup.

Saying that APIs are open does not mean an absence of security: communication through APIs can easily be identified and encrypted, as needed.

d. REST API?

Two popular web API conventions emerged in the 1990s and competed for popularity:

- SOAP: Simple Object Access Protocol
- (REST: Representational State Transfer

REST APIs became ultimately the most widely adopted, because it uses the same simple principles that webpages use to be transferred over the Internet (the "http" protocol that you see in web page addresses). This is why APIs are often called "REST APIs, presented in this pedagogical video". In 2000-2010, it became increasingly easy and natural to adopt the REST convention to make one's software and data available to another computer. This simple evolution to ease interoperability had **immense effects**:

3. Business consequences of APIs

a. APIs opened software to the world

An API transforms a closed software into something that can be plugged to anything other computer or object, as long as it is connected to the Internet. For instance, APIs were a key factor of success for SalesForce in the early 2000s. SalesForce, created in 1999, has a revenue of US$8.39

billion in 2017: - SalesForce developed a CRM as a SaaS where features of the CRM were **exposed as APIs** (meaning, these features could be plugged to external apps via the REST protocol). - SalesForce created a PaaS to host apps that could plug to the SalesForce CRM via the APIs developed by SalesForce. This platform is called Force.com and external developers can put their apps there, as long as they are compatible with the SalesForce API. SalesForce takes a commission on the sales made by these third party apps hosted on Force.com, but more importantly, the platform creates an **ecosystem** of apps and developers around the SalesForce products which makes it hard for a customer company to switch to a different product.

b. APIs accelerated software innovation

Thanks to API it is now easier to add software blocks together and create new apps, even if these software blocks originate from different countries, industries, big and small. As an extreme example: the Australian Victoria Police deployed a project for the recognition of stolen vehicles through the video recognition of licence plates on cars passing in the street (stolen vehicles get their license plates immediately recognized). This is a $86,000,000 project. An individual actually replicated this project with just 57 lines of code and a dashcam. How so? Just because he could use existing software for licence plate recognition, available as an API, instead of re-developing this by himself.

Another example: Pieter Levels demonstrated the potential of APIs by building a "Luggage delivery service" just with existing apps connected together via their APIs. Without a line of code:

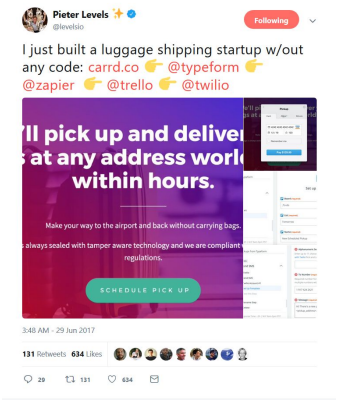

Figure 1. Building a world wide luggage delivery service without code

This service is designed by organizing several sub-services, which coordinate by communicating via their APIs. How does communication work? Who "orchestrates" these services? Pieter uses Zappier, a service whose role is to make these APIs communicate with each other. Beyond these striking examples, the lessons to be learned are: - more and more services are available via API. Do not reinvent the wheel, just use the APIs. - coordinating multiple APIs allows you to create entirely new services (not just: "manage my emails by API") - services like Zappier allow coordination / communication between APIs, but it also favors **automation**.

c. APIs opened data

Companies and public organization own many datasets of great business interest. The use of these

datasets can be free (for small projects and NGOs) or monetized if the user is an enterprise. Without APIs, datasets can be made publicly available as docs (eg, Excel spreadsheets) to download but this is not practical (try downloading something like `all_train_schedules_2000_to_2017.xls` !). Let's take the example of a transportation company like French SNCF which finds it interesting to publish station names, train schedules, real time information on train traffic, etc. because it could be used by other companies to build new services : how can it do it?

- The data is on a server of SNCF
- SNCF adds an API and its documentation, making the data available to developers able to connect to APIs, which is a basic skill in software development.
- Entrepreneurs and programmers in general will be able to access the data via the API and use it, creating new services based on this train information. **Open data** designates this movement to make datasets available to a broad audience, and web APIs have been a key technological ingredient in this movement.

4. Does your company need to open an API to share its data?

A company can create its own APIs to "project" its services farther and stronger than just the "web page" interface. When is it a good idea to do that? Here we must brainstorm, with two benchmarks to keep in mind. These examples are taken from Pieter Levels, a Dutch entrepreneur who specializes in creating web platforms for remote workers. He created:

- Nomadlist, which presents the cities of the world and how they can be attractive to live and work
- RemoteOK, which is a job board for ads exclusively for remote jobs.

Pieter Levels has implemented APIs so that users can access ("consume") these two services, **with very different results**:

- The access of NomadList by API, without control, has resulted in siphoning its data by competitors who used it to develop copycats. (read the tweets announcing the opening of the API and its closing).
- Access to RemoteOK ads by an API allows third-party platforms to integrate the RemoteOK ad catalog with their own: it amounts to free and large-scale referencing, which increases the likelihood of an ad finding a candidate (read again the tweet that announces the opening of the API).

The screenshot below summarizes the contrast between the two situations: although in both cases it involves opening its data via an API. Between NomadList and RemoteOK, the results are opposite:

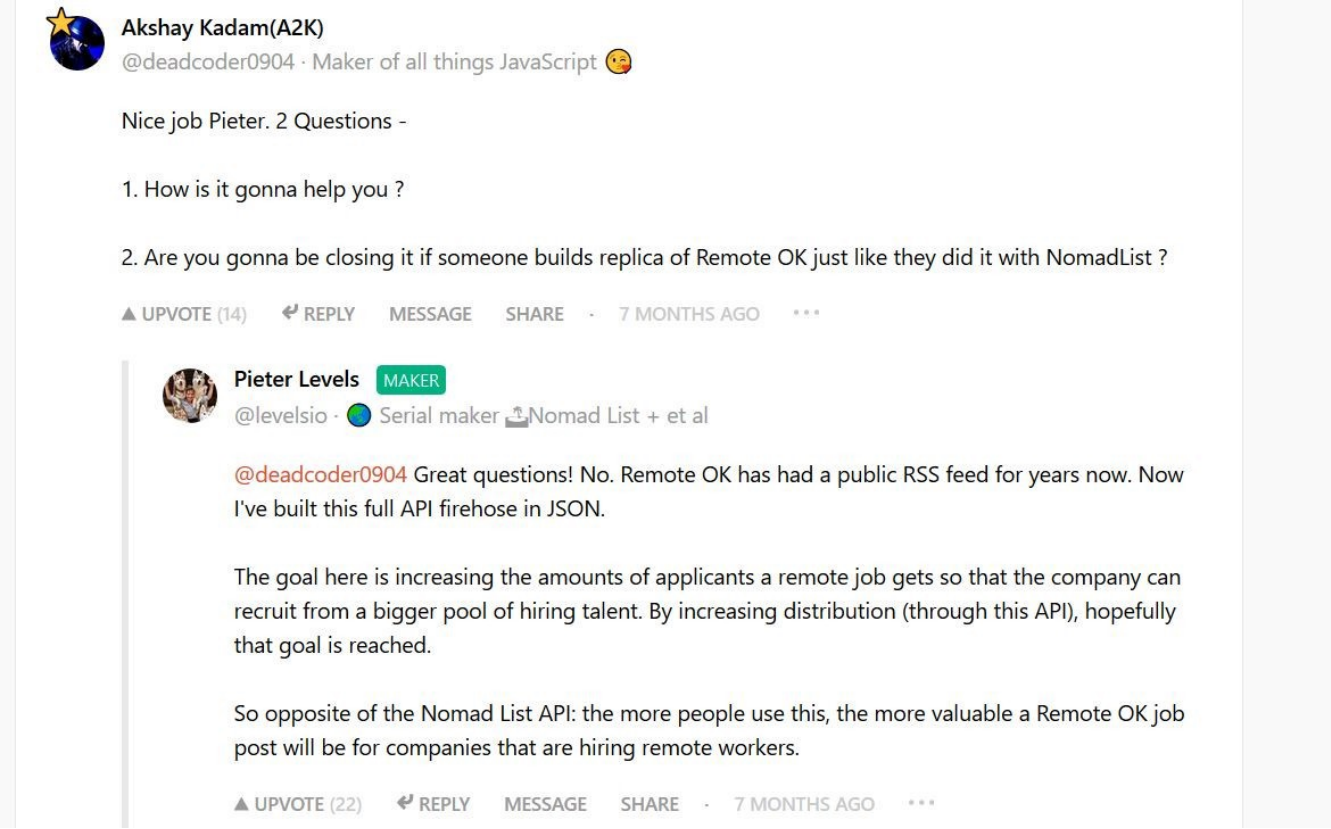

Figure 2. Pieter Levels explains his reasons for opening or closing an API

Product managers must have a fine appreciation of the uses that the opening of an API will stimulate. One of the most virtuous effects is that of a "sound box": by reusing our content for their own purposes, the users of the API will unwittingly promote our content, which reinforces the efficiency of our product.

5. The ecosystem of APIs

a. A wealth of APIs

To discover public APIs, or to make your public APIs easier to discover, you can visit https://apilist.fun/, https://apislist.com/ or https://publicapis.io. Searching on these websites, you will find APIs providing business services, or APIs of a fun or odd sort.

Still, many APIs are not listed on this website, and a google search for "info I need + API" is also a good way to find if the API you need exists. Interested in bird watching? There is an API for that.

b. APIs: a business world of its own

APIs have become central to the economy. As a result, a large number of services associated to APIs have developed to cater for all the needs of companies that use them:

- how to create an API
- how to manage the documentation of a large number of APIs

- how to connect a wide variety of APIs
- how to control and audit the security of APIs
- how to monetize an API…

Mehdi Medjaoui from APIscene keeps an updated, searchable landscape of the main companies active in the API industry:

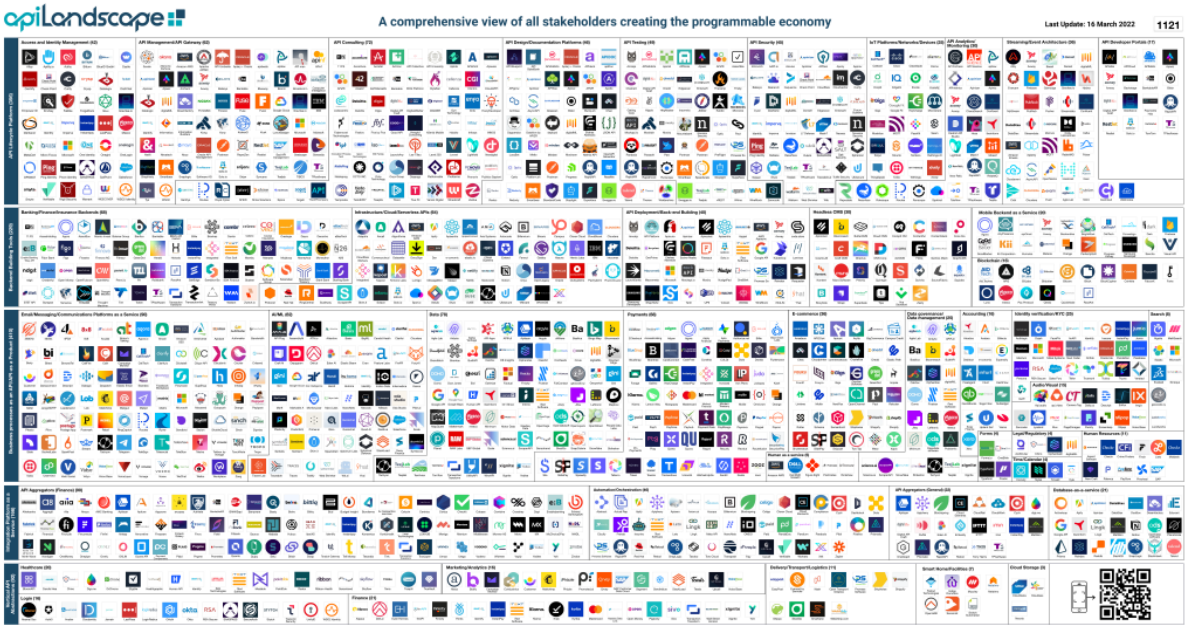

Figure 3. The API ecosystem in 2023 by Mehdi Medjaoui

(or download the full size version of this poster here)

To go further

- my memocard on web APIs in pdf.
- API Business Models: 20 models in 20 minutes (□ 20 min read).
- An introduction to APIs: a complete course for real beginners. A short online course, just a bit technical, on APIs. (□ 4 hours).
- Best practices for API design

Find references for this lesson, and other lessons, here. image::round_portrait_mini_150.png[align="center", role="right"]

Discover my other courses in data / tech for business or get in touch via Twitter: seinecle