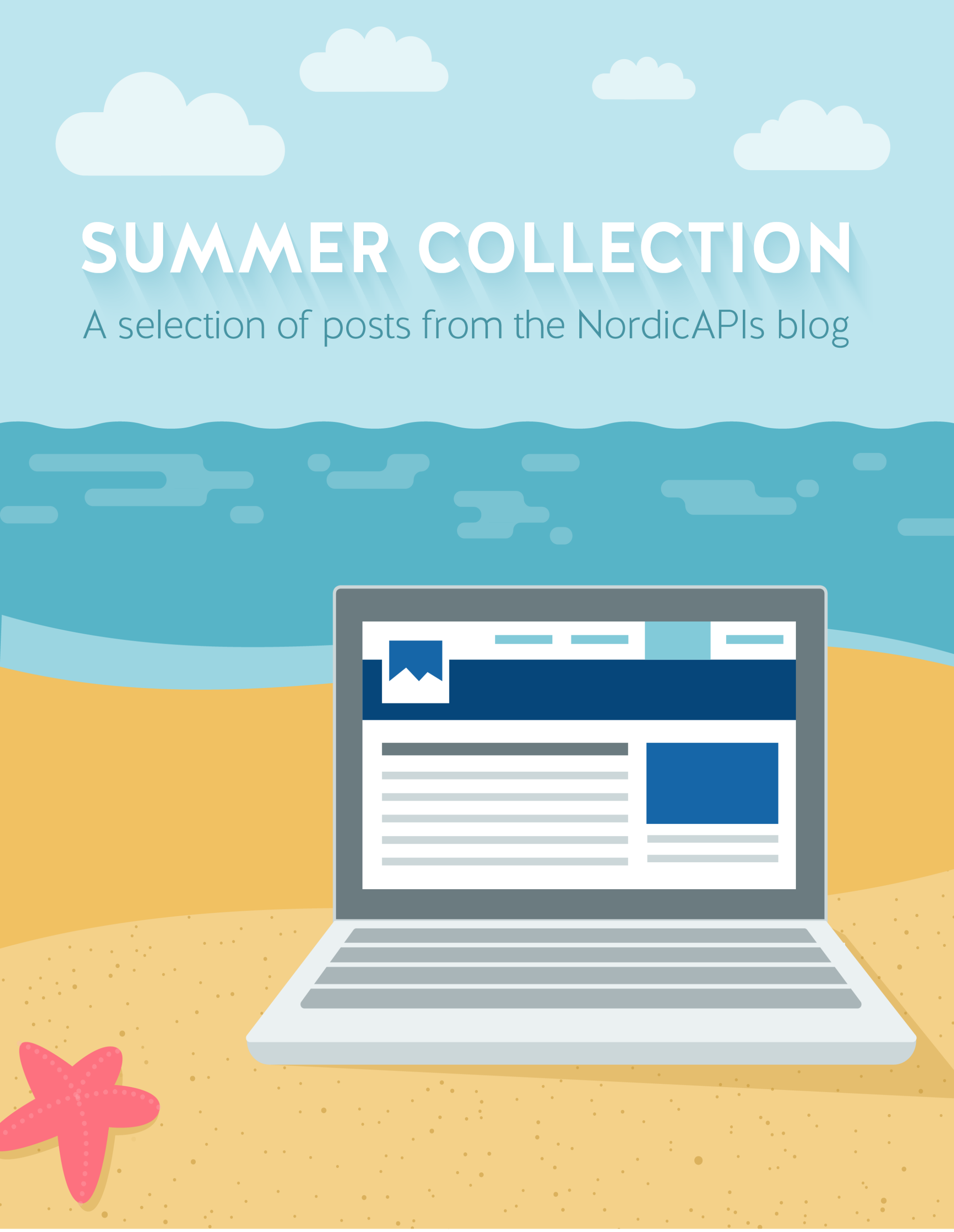
# SUMMER COLLECTION

A selection of posts from the NordicAPIs blog



**AUTHORS**

Bruno Pedro
Mark Boyd
Rhys Fisher
Tim Mares
Travis Spencer
Andreas Krohn
Staffan Sölve

NORDIC **APIS**
nordicapis.com

# Summer Collection

## A SELECTION OF POSTS FROM THE NORDIC APIS BLOG

Nordic APIs

This book is for sale at
http://leanpub.com/nordicapis-summercollection

This version was published on 2014-10-20

# Tweet This Book!

Please help Nordic APIs by spreading the word about this book on Twitter!

The suggested tweet for this book is:

I just downloaded @NordicAPIs’ e-book, ”Summer Collection.”

The suggested hashtag for this book is #NordicAPIs.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

https://twitter.com/search?q=#NordicAPIs

# Contents

# Preface

What an amazing summer! Such fantastic weather, barbecues, boating, camping, and APIs! Since the Nordic tour back in April, we have organized an after-work meetup in Gothenburg, which was really fun. We have also been working hard to connect with more of you online. To do this, we have grown our team substantially, bringing on many new bloggers. Every single week, all summer long, they were blowing my mind. Post after post, one thing kept coming up again and again: **PLATFORM**

We all had it on our minds and were seeing it all around us: **how do you become an API platform**? We talked a lot about it together, and typed up those thoughts on our blog. We defined what exactly we meant by that, we found some good examples like Apple and Podio, and we wrote about how to transform your business into such a platform. In this e-book, we have collected some of these together, but we're not done yet!

We are just starting to unpack this topic of becoming an API platform. We are hoping that this e-book will inspire you to share your story about how you have become an API platform, what struggles you have faced, and how you have overcome them. We are hoping that you will share this with the community both on- and offline. Many people, including CA Technoloiges, Microsoft, Mulesoft, SmartBear Software, and others will be on hand in Stockholm this fall to give their testimonies first-hand. I really hope you will join us this October 20 - 22 in Sweden for our biggest event ever where we will be focusing on this notion of being an API platform. I also hope that you will join us online and become even more engaged. We are discussing these topics on Twitter, Facebook, and our blog every single day. Connect with us there and share your thoughts.

Please enjoy our first ever e-book and let us know how we can make the next one better.

– Thanks, Travis Spencer, Co-founder, Nordic APIs

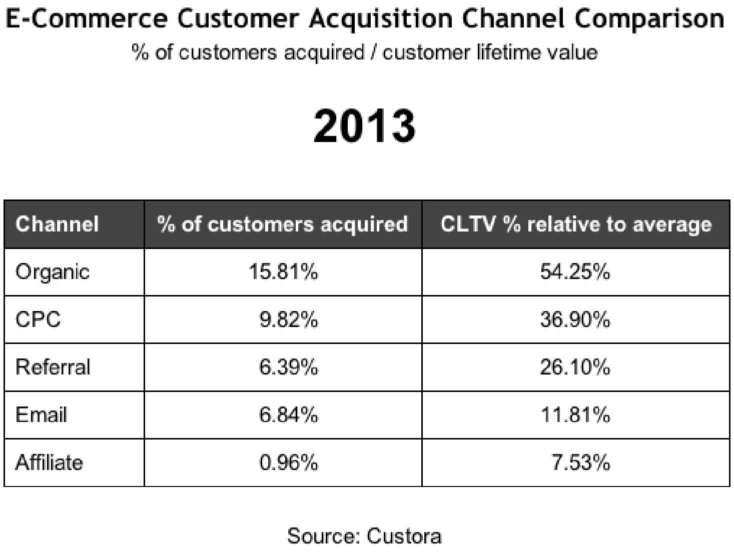
# 5 Ways APIs will Increase your Revenue

*By Bruno Pedro*

In this article, I will show you five ways to increase your business revenue by offering an API that third-party developers can use to build apps and consume your information. These are not the typical API business models. Instead, these are ways to make money with or without an API. What I will show you though is that these models deliver more value if executed with an API. Let me start by showing how you can increase your business revenue by acquiring more customers using an API as an inbound funnel.

## 1. As a Customer Acquisition Channel

APIs are not traditionally seen as acquisition channels because there is not enough data available. As API adoption continues to grow, this will change. Till then, we can draw some comparisons. Similarly to the way Web sites acquire new customers through referral traffic and affiliates, APIs need apps to refer more patrons. The more developers use your API, the more third-party applications will be built and consumed. Even if those other applications use your API to provide a single feature, more users will be consuming your API in the end. By looking at the usual channels and considering this correlation to referral marketing, we can infer that APIs have the potential to provide around 26% more Customer Lifetime Value (CLV or CLTV) compared to the average customer acquisition sources.



**E-Commerce Customer Acquisition Channel Comparison**

% of customers acquired / customer lifetime value

**2013**

| Channel | % of customers acquired | CLTV % relative to average |
|---|---|---|
| Organic | 15.81% | 54.25% |
| CPC | 9.82% | 36.90% |
| Referral | 6.39% | 26.10% |
| Email | 6.84% | 11.81% |
| Affiliate | 0.96% | 7.53% |

Source: Custora

As a proof of this, look at Spotify. They have built a robust ecosystem that delivers great value to their customers. They are using APIs to funnel in new customers in two ways. Originally, Spotify used their API to make third-party widgets available inside their desktop application. This created an application marketplace. Since their API relaunch in June of this year, Spotify is now also promoting and encouraging the creation of stand-alone applications that make use of their API.

From this example, we can see that there are two ways to use an API as an inbound channel:

1. As the means to build extensions that are available to end users through your app
2. As the platform on which numerous apps can be created and marketed independently

The first requires people to already be consumers of your application, and does not take full advantage of the cross-marketing potential of the app economy. The second does. Like the music publisher, you can encourage third-parties to build entirely new applications that use your API in completely new ways. While the former technique will result in a better user experience due to a tighter integration and control of the ecosystem through an application marketplace, the latter will generate the biggest impact as an inbound acquisition channel. By employing the second tactic, it is clear that Spotify has the ambition to become an API platform. If this is your intention as well, make it possible for devs to use your API in new and creative ways divorced from your own apps.

Using either of these methods, your goal must be to create a growing ecosystem of third-party developers and encourage them to innovate with your API. In so doing, you multiply your chances of acquiring more customers while also reducing your overall risk. To increase the CLTV of these customers, more app usage of your API, especially outside of your own app, will result in more referral traffic (e.g., like backlinking for your Web site). These strategies will allow you to simultaneously benefit from somebody else's efforts.

Sounds exciting, right?! And this is just the first way an API maximizes the revenue of a business model that you can execute without one. Let's look at another.

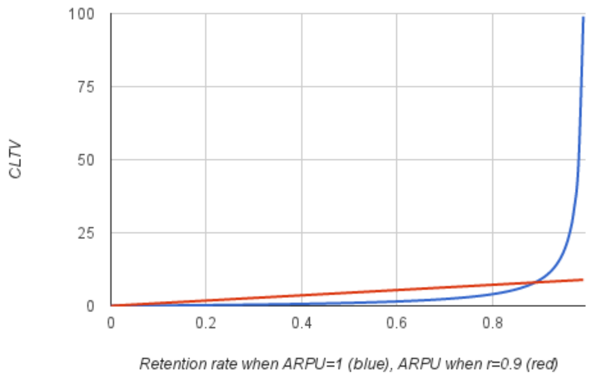
## 2. As a Way to Increase Retention

After acquisition, CLTV is the most important factor in determining the success of a business. It is a way to calculate how much profit each customer will generate over time. Retention is the number one way of increasing this key performance indicator. According to Elie Ofek, Professor of Marketing at Harvest Business School, CLTV can be calculated by using the following formula:

$$CLTV = \frac{ARPU \times r}{1 + d - r}$$

where *ARPU* is the average revenue earned per user or customer, *r* is the retention rate, and *d* is the yearly discount rate (which is often the interest rate as a proxy).

By analysing the formula, one can see that, if nothing else changes, linearly increasing the retention rate will generate an exponential growth in CLTV. On the other hand, increasing the ARPU will only increase CLTV linearly.



The more people that use apps that rely on your API, the more market share you will have for that type of API. After achieving critical mass, it will be hard for competitors to unseat you, or for those third-party app developers to migrate to another API provider. Consequently, it will be harder for users to move to the competition, thus increasing your retention rate.

Evernote is successfully following this strategy. By showcasing third-party applications on their App Center, they are promoting work done by external developers and encouraging users to try those products. This increases user engagement with Evernote. Another API platform that is showcasing integrations made by partners is Podio, a Danish cloud software company now owned by Citrix. They are highlighting the way in which third-party developers are allowing end users to get their work done more easily using their platform. In this way and others, they are acquiring and retaining customers through a balance of User Experience (UX) and Developer Experience (DX).

Though the effects are not as impactful, increasing ARPU also increases CLTV. The easiest way to increase the ARPU without increasing overall prices is to sell more to existing customers. An API can also help you do that, as you will see next.

## 3. As an Upsell Driver

Interesting API integrations present opportunities to upsell access to certain features that enable this possibility. You can make certain API features only available to those who have purchased access. A classic example of this is Web Single Sign-on (SSO). For those customers that want to allow seamless access to your API's OAuth server using identities of their employees or customers via SAML, you can upsell them on the Web SSO feature. Salesforce does this, for example. In fact, Salesforce customers below the "Enterprise" edition do not even have access to the API. Alternatively, you can charge customers after they exceed some particular rate limit. This is a good technique to attract customers to use your API more, as it removes any cost-related barrier of adoption. As they use your API more, they will see the value of it and pay for their increased usage.

To employ this strategy, your business must have a large user base and be well known in the market. If this is not the case, you may try this strategy for very valuable API features. The danger is that users might migrate to a competitor that offers the same features for free. New market entrants may even try to break in by differentiating themselves from incumbents by offering the same features for free.

## 4. As an Affiliate Channel

To encourage more developers to integrate their apps with your API, you can also turn them into affiliates of your primary offering. This will drive additional demand and, thus, revenue for your company. As shown in the table above, affiliate marketing can increase CLTV by nearly 8% compared to average acquisition methods. Correlating this to an API affiliate program like I did above for referrals, you can see that using APIs to drive demand for other products has significant potential. Imagine if just one third-party app is successful in this. You will see an increase in customers, similarly to what you would see on your Web site from a devoted affiliate.

You can also turn this around and become an affiliate yourself. Like Evernote and Podio, you will want to showcase the integrations that others have done with your API. From your showcase pages, you can drive traffic to your partners and gain a finder's fee for yourself. In this way, third-parties help you with app development and co-marketing and even pay you to drive new customers to them. Sound far fetched? A decade ago maybe, but certainly not today! Look at Apple, Google, and Facebook. It's not just the big guys either. Look at any company providing a marketplace like

Basecamp and Freshbooks. This is what being a platform offers! As an API platform, your service is integrated by co-creators who must market their inventions and cut you in on the sales transacted in your marketplace.

## 5. As a Distribution Channel

If your business depends on the number of people that get access to your content and interact with it, you will want to maximize its distribution. You can do this without an API, but you'll be able to maximize it by using one. Companies like Expedia and Booking follow this strategy and are having great success. When you are searching for a hotel on your favorite Web site, you are often served up results obtained from a bigger vendor's API, e.g., from Booking.com's. These large providers take advantage of the fact that smaller companies do not have the resources to manage big data sets. They distribute their content via an API in a revenue sharing model.

Another good example of a company that is increasing distribuiton using APIs is Fyndiq, a Swedish online retailer. After releasing their public API in March 2011, they saw a dramatic increase in sales. A big part of this growth was related to an integration they made with a third-party platform that exposed Fyndiq's products to all of their shops. Smaller companies can easily jump start their business by offering search capabilities on large catalogs. Because these catalogs are available through APIs, there are clear contracts and the information is easily obtained programmatically. Guillaume Ballas, 3scale CMO, touches this topic during a presentation he gave at the 2013 Nordic APIs conference in Stockholm.

## Conclusion

As you can see, an API can definitely help you increase your revenue. All of the models described in this post can be done with or without an API, but hopefully you can see that using an API in one or more of these ways will maximize your chances of pushing your business' topline to the next level.

As you think about how to apply these, remember that APIs can be used:

1. *As a new customer acquisition channel:* With an API, you can open a new channel that might be able to provide around 26% more CLTV than other traditional channels.
2. *To increase customer retention rates:* CLTV is the most important metric after acquisition and the best way to maximize it is to increase the customer retention. By offering an API that is integrated with third-party apps, you will be able to guarantee maximum retention of your customers.
3. *For upselling:* You can create demand for higher-priced features by making them available through an API. If customers use your app features through an integration with a third-party, they will find value in specific features that you can upsell.
4. *In affiliate marketing:* you can either turn third-party developers into affiliates of your app, become an affiliate yourself, or both. In the end, you will always make more money.
5. *To maximize distribution:* By enabling third-party apps to obtain your content or products, you are making it very easy for others to make money with you. If others make money, you make money.

Do you have more ideas or examples of ways an API can increase revenue? If so, share it in a comment here or let us know on Twitter or Facebook!

# Accelerating Growth: A Case For Opening up your Core

*By Tim Mares*

Have you ever wondered why companies like Salesforce open up their organization by providing third-parties with tools such as APIs? I know I did! To understand this, I dug really deep into accelerated growth strategies, and wanted to share some of the insights I have gained. What I learned is that many businesses are struggling to keep up with the pace of innovation. In the face of disruption, many organisations simply ignore it, or try to partner up instead of transforming their business model to adapt.

Through this investigation, I discovered various elements necessary to adapt to disruption. Understanding these elements and instruments is key when you run a platform business. An important part of platformizing your organization is the way you create and capture value. The approach of out doing competitors by keeping knowledge, tools and resources private has become old fashioned. Today's fastest growing platforms work differently. They open up their core, publish APIs, and start up developer programs. Understanding the mechanics of becoming an API platform is key. In this post I will show you the transformation of one pioneering API platform businesses, and the massive impact of open value creation.

## Transformation begins with Understanding

The transformation into an API platform requires a concerted strategy. The facets of this plan will depend on how you characterize your business model. Traditionally, businesses have either been product or services companies. More and more, product companies have morphed into services providers. This metamorphosis is happening again; this time, however, from service-based to platform-based.As we said in our definition of an API platform, a fundamental characteristic of such a business is that it allows third-parties to leverage its core capabilities and/or resources. This usage may bring either direct or indirect revenue to the platform provider; it will also provide some commercial benefit to the software vendor building on the platform. In all cases, ecosystem members will symbolically coexist for the mutual benefit of all parties. By exposing your core value as an API, you can change your position in the market from being just another fish in the sea to to being the Barrier Reef!

If you are one of those I mentioned that is struggling to keep up with innovation, you may dismiss this claim as exaggerated marketing. Don't! It's happening all around you. Dismissal will result in obslence. You have been warned!

## Punctuate your Business Model with an API

As a service, you focus on user growth and optimization of your offering. This tuning is natural for any company. You can not be a service provider or platform without being a profitable business. . While tuning their offering, many businesses end up in the standardization loop. Don't get stuck here! You will be relegated to little more than a service, and disrupters must grow beyond this into platforms. The standardization loop can be seen as a combination of increasing efficiency, cutting expenses and making acquisitions. Although this might influence growth, it does not *rapidly* accelerate it. After proving your business model, business platformization accelerates growth.

Business platformization is central to the success of some of today's most successful companies, e.g., Android and Salesforce. They have grown exponentially to the place they are now, and exemplify how businesses should open up their core value proposition. It begins with a vision of forming an ecosystem, and growing into the "reef" through a solid, standard offering. Let's take Salesforce as an example of this.

## Salesforce: An API Platform Pioneer to Follow

They are the leader in cloud computing for enterprises in many respects, and have been for over a decade. The standardization phase took them many years. During the first several years, Salesforce launched multiple customer relation management applications. As they did this, they focussed on improving and creating their value proposition. After having fully standarized all services and growing their userbase, Salesforce launched the Appexchange and Force.com in 2006 and 2007, respectively. These enabled developers to use

Saleforce's tools and APIs for commercial purposes. They opened up their core value proposition and quickly accelerated their growth by running a viable marketplace on the platform that they had bootstrapped.

Platformization begins with a vision, followed by product standardization, and then assumption of the central position with the ecosystem that's been formed.

## Control the Tools that Define the Market

Two years after launching the Appexchange, almost 300 thousand test drives took place and almost 800 applications were available. The Force.com platform counted over 100 thousand developers after being avaialble for one year. Salesforce's platforms was growing quickly while they kept controlling and developing tools that enabled developers to create value. As with Salesforce, your strategy to become a platform should progress from standardization to enabling developers with tools for value creation. A platform should create an infrastructure of co-creation that helps developers increase the value of their offerings. Helping developers understand how the infrastructure works is essential to optimize the common value proposition that they and the platform can create. For example, Salesforce offers a technical library that informs developers how to use their APIs and commercialize and manage their Force.com applications.

Part of successfully managing an open value creation strategy is understanding the role of your platform in the market. It is to facilitate the mutual benefit of all members of the ecosystem, including end users, application developers, and the platform provider. To obtain and maintain continuous growth, platforms should not capture all of the value that is created by third-parties for themselves. Neither should they seek to completely control the outcomes of what developers create using the API. A platform needs to provide the ability for others to co-create using its facilities. Benevolently managing this infrastructure will lead to an increase in value creation, and, thus, growth. Being a platform will enable you to keep up with the rapid pace of innovation; the faster the better, you'll find.

## Open up or Lose Out

Salesforce is the world's leading online CRM software vendor. How did they get to this position? Salesforce allowed its 1.4 million registered developers to create and commercialize their applications. No wonder almost 2.5 million Salesforce customers have used their marketplace to procure additional apps. Salesforce exemplifies the power of enabling third-parties with tools and knowledge.

If Salesforce had kept all these resources private, it would not have succeeded in building a "workforce" of millions of developers that were willing to risk their time and effort (at no cost to Salesforce) to try and create and commercialize their applications on the Salesforce platform. Salesforce envisioned the possibility to grow well beyond what they could do through in-house creation. By opening up their core, the CRM provider in tandem with API developers, produced more value for the entire ecosystem. Salesforce enabled developers to build apps that then could be sold to customers of Salesforce. The API was the doorway into this new channel.

Salesforce's strategic decision to enable developers to market their products in its own ecosystem resulted in them becoming the number one marketplace for business applications. Imagine if Salesforce would not have opened up their core and supported developers with knowledge and tools. Their growth would have been limited by their own internal resources. Salesforce would never have grown to the level they have without:

1. Envisioning the day that they would be a platform
2. Standardize and solidify their offering to pick up the cash necessary to bootstrap the ecosystem
3. Grow beyond a service provide into a platform business by opening up their core as an API platform
4. Maintaining and nurturing the ecosystem that formed around them

## Grow like Salesforce into a Platform

You can grow without becoming an API platform. However, accelerating growth, as Salesforce has done, is much more likely if you make a platform move. Technology is rapidly changing and has presented all of us with the opportunity to make such a move. To do this, create a shared vision within your organization to become an API platform. When momentum grows internally, don't stop there. Instead, couple the sales of a standardized offering with the introduction of more and more APIs and API-related resources. This will result in your obtainment of the platform role within your ecosystem, springboarding your growth to new levels.

Please share your thoughts here in a comment, on Twitter, or on Facebook.

# API Platform Defined: When an API Provider is a Platform

*By Travis Spencer, Andreas Krohn, Rhys Fisher, and Mark Boyd*

These types of organizations are API platforms, but what is an API platform? We talk about API platforms so much here on the Nordic APIs blog and at our events that we wanted to define it clearly.

## The Definition of an API Platform

An API provider is an organization that exposes data and/or capabilities through a programmatically consumable service or an Application Programming Interface (API). This organization is more than a provider and has become a platform when its API:

- Enables access to the organization's core value proposition;
- Is technically and non-technically scalable;
- Enables consumers to create shared value;
- Is instrumental in securing the organizationâ€™s position as a market leader; and
- Is seen by top management as business critical.

## Elaborating on this Statement

An organization is an API platform if it exposes its resources and assets in a machine readable format to other members of the ecosystem. This allows those third-parties to build on the API platformâ€™s core value and create new value for their customers.

An API platform is an organization which brings together two or more distinct, but interdependent, groups of consumers using APIs. This creates a foundation for automated transactions between different networks. The platform unlocks hidden value within the organization by exposing its core as an API. In this way, it acts as a centerpiece for open innovation, co-creation, and collaborative development. This facilitates an ecosystem effect where the platform becomes the basis on which others conduct their business.

Simply put, an API platform is:

- A catalysts for growth
- A hub for innovation
- A co-creating ecosystem
- An organization which has automated strategic partnering
- An efficient system that adapts to its surrounding environment

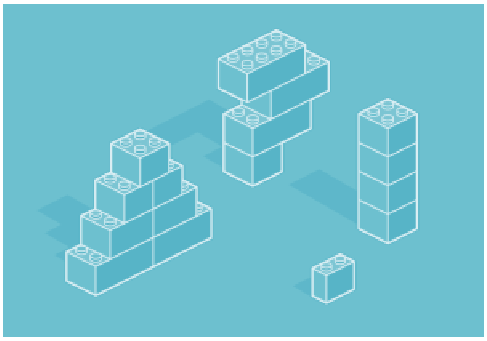
## Why be an API Platform?

To become an API provider can be challenging enough. Why would an organization strive to be more? Why invest in the people and technology necessary to become a platform? Business leaders are seeking to transform their organizations into API platforms because of their potential to disrupt entire markets. An API platform is necessary to pull off a platform business model (e.g., like Amazon's and the others' we mentioned).

What do you think? Did we correctly define what an API platform is? Let us know in the comments, on Twitter, or on Facebook.

# 2 Changes that will Transform your Business into a Platform for Growth

*By Travis Spencer*



Many of the stories we have heard at past Nordic APIs events and have recounted on our blog are ones of transformation. Organizations providing APIs are reinventing themselves as platforms that are ideally suited for modern-day business. These stories have been told by PlanMill, Bisnode, Fyndiq, and others of various sizes operating in different industries. I presented on this last year at Internetdagarna, but feel that we should talk more about it given the prevailing trend toward platformification.

Becoming a platform will require changes in two main areas of your business:

1. **People**: The roles, personnel and processes of your organization will have to undergo major reorientation if they are

to function as a platform on which third-parties can conduct their business.

2. **Technology**: Existing technology must be bridged from the old ways of doing business to the new ones. If the gap is insurmountable or if legacy systems are unable to adjust, new technology investments will be required. The first is the hardest, but don't be overwhelmed. I have shared the following advice with many would-be platform companies who have followed it and achieved great results.

## It's a People Thing

Moving toward platform-based business operations is a team effort. There is *no* chance of doing this by yourself. Sorry, Kemosabe. You have to find friends and allies. Building a platform is as much about building relationships as anything else. This will be challenging at first because many potential allies will not grasp the concept or benefits. Platform-oriented business based on APIs is a paradigm shift. Pivoting others' mindsets can be an uphill battle. To overcome these challenges, I suggest that you employ the following tactics:

- Demonstrate value early
- Refer to examples of API pioneers
- Strive for small, successive wins
- Be an intrapreneur
- See yourself as an internal API evangelist
- Get top-level buy-in Your goal is the last point. Only with agreement from the C-level (and perhaps even the board) are you going to succeed. This is going to be a long and expensive process that will not be possible without upper management's help. How you get that though will require a course of action that includes the other methods.

I will not go into all of these; it would make this post too long. (Keep sending us your thoughts though, and I will make sure they are posted on our blog.) I will dig into one that you can begin with right away at no cost.

## Tell the Story of API Pioneers

APIs are not new. They are *new-ish*, but many people are already doing amazing things with them. To gain mindshare and win allies within your organization, tell your colleagues about what these pathfinders have done. Find cases of organizations in the same industry, of similar size, and operating in the same market. If you can find a very close match, it is likely that they are a direct competitor. If you want to motivate your company to quickly make a platform move, explain that the competition has already launched an API. Tell your coworkers that only by becoming a platform can you outflank the competition. I have seen cases where this argument was all that was needed to *quickly* start companies down the API platform route.

To help you with this, here are a few resources about platform-based and API-driven businesses that you can refer to:

| Organization | Industry | Size | Market |
|---|---|---|---|
| Amazon | E-commerce | Large | Global |
| Fyndiq | E-commerce | Small | Sweden |
| Pearson | Education | Large | Global |
| PlanMill | Technology | Small | Europe |
| Salesforce | Technology | Large | Global |
| SoundCloud | Entertainment | Small | Global |
| Spotify | Entertainment | Medium | Global |
| Twilio | Telecommunications | Medium | Global |

To hear directly from SoundCloud, join us in October when Erik Michaels-Ober, the company's Developer Evangelist, will present firsthand.

## Internal Evangelism & Intrapreneurship

Entrepreneurship and API evangelism go hand in hand. The qualities of an API evangelist are very similar to a technology founder, says Brandon West, API evangelist for SendGrid. The external type of evangelism Brandon is talking about though comes at a point when the company needs to market its APIs to developers. By then, APIs are an important part of the company's product strategy. Before that, internal API evangelism is going to be needed. The work will be somewhat similar though.

You will need to network like your life depends on it! Mingle with engineering. Attend company parties with project managers. Go jogging with salespeople. Each lunch in the canteen with the support folks. Take product managers out for sushi when you get sick of the cafeteria food. Tell them all about the massive shift the world is undergoing, and how APIs are making organizations more programmable. Use examples of other API platform companies like the ones above to explain how APIs are disrupting entire markets.

### Create an Internal Newsletter & Blog

Everyone has to eat, but lunching will not scale. To reach more people, publish articles on your business' internal blog. Compose a letter and send it to coworkers every month or so – an internal newsletter of sorts. Try something like this to get started:

#### Initial Mail

Hi All,

I have been reading and thinking a lot about how our company needs to become more of a platform on which others can conduct their business. By focusing on our core and exposing our key capabilities and data, others will be enabled to create higher-level value than we alone can provide. I have talked with some of you about this in the past, but thought it would be great to send out an email every month or so to keep the conversation going.

As mobile, cloud, and social make way for the Internet of Things (IoT), wearables, quantified self, and other new tech-centric possibilities, our company has an immense opportunity. By extending our use of APIs, we can become the basis on which partners and even the general public can create new innovations that we cannot dream of. This will give us a central position within a growing ecosystem which will ensure on-going growth. To achieve this, I believe we must:

- Follow the example of early adopters like Twilio
- Learn more about APIs
- Form a working group that can discuss this topic on a regular basis
- Attend the upcoming Nordic APIs Platform Summit

What else should we do? Please reply with your thoughts.

Next month, I will address the security issues that a platform move could impose, and talk about how we can overcome those.

Till then.

## Mail about Security

Hi Again All,

Last month, I emailed you about the importance in growing our API program and transforming our organization into a platform. Nordic APIs has a helpful blog post with more on that topic, and we had some great internal conversations as well. This month though, I wanted to write about one of the hardest parts of building an API platform: **security**.

There are a few very important security aspects of any API platform. In general, we must ensure that our platform is secured:

- With open standards that have broad industry adoption
- Using common algorithms that have been scrutinized by expert cryptographers
- Using pure-play products rather than home-grown software or heavyweight alternatives

It is important as we grow our API platform that we leverage existing systems to implement the desired functionality. This will require us to broker the security of our API to the models used by our existing systems. I have read that this should be done in the API Security Service. This gateway will stand in front of our existing (and new) APIs, and will transform the incoming security model to our legacy ones. The model we expose outwardly should be OAuth 2 and OpenID Connect, suggest API security specialists.

To get this setup, we should talk this month about:

- What security models do our existing services use? How can we transform OAuth into that?
- What are the requirements of any API Security Service software that we may need to procure? Do we have any software already that does this or something like it?
- Do we need change the network topography to allow for secure access to our APIs? How will the API Security Service fit into our deployment?
- Do we have expertise in OAuth 2 and OpenID Connect? Will we need help from outside experts? If so, who?

To lean more about this before we meet, check out these resources:

- Nordic APIs security positings
- How security gateways help protect APIs
- API Security Services in context of a broader solution

Next month, I will share some thoughts on how becoming an API platform will grow our company at a tremendous rate.

Till then.

## Mail about Growth

Hi All,

Over the last few months, we have been talking about the need to transform our company into a platform. I touched on the security challenges of this in my last mail, and we have discussed others in our meetings. As more and more people have joined the discussion, I wanted to write this month about how a platform can enable hyper growth of our organization, and get all of your feedback.

What is the difference between providing an API and being an API platform? One distinction that occurs to me is that a platform enables others within the value chain to create higher-value products and services using the base capabilities of the API provider. We are already doing this in other aspects of our business and with our APIs (to a degree), but we must become a platform to scale this way up. If we expose our *core* capabilities and data through our API platform, others will combine our *unique* value proposition with their own. If we allow them to do this in an automated fashion, the resulting combination will be more valuable to a larger market than

> we are able to service on our own. Self-service capabilities will ensure that our costs for this remain low as consumption increases. By providing a critical component of these higher-level offerings, we will position ourselves at the center of numerous other organizations' product strategies. This will give us a larger and larger market reach, resulting in more and more growth for us. If we can achieve a critical mass, our API will provide us with tremendous growth.
>
> How we scale our API into a platform is a question that we need to explore more as a group. Please reply with your thoughts, and let's find some time this month to meet and discuss. In the meantime, have a look at these resources:
>
> - Transforming your Business into a Platform for Growth
> - The Dangers of Ignoring the API Revolution
> - An OAuth-protected API Platform for Private, Partner & Public Use
> - The story of Fyndiq's API
>
> Till next month.

Check out these useful API resources for ideas of what to include in future mailings.

In everything you do, aim to motivate, inspire, encourage, and lead.

## The Technology of API Platforms

Building a platform requires new technologies to be deployed. What exactly these are will very depending on many factors unique to your organization. In general, however, the following advice should be considered:

- Reuse existing IT investments, systems, and processes

- Build on open, international standards that have broad industry adoption
- Avoid vendor lock-in
- Follow best practices and industry norms

## Reuse IT Wherever Possible

The only way to become a platform is to extend and reuse the technologies that you already have. These are working and helping your company earn its precious revenue. To abandon them is folly, but they are most likely ill suited for direct usage by platform consumers. Instead, these will have to be abstracted and adapted to alternative access methods. The general pattern is to obscure them with an API Integration Service. Such integration software will orchestrate and transform existing services that were designed for another era into virtual APIs that can be consumed by modern, mobile and cloud-based consumers. By forming an integration layer that transforms incoming content, an API Integration Service pivots the data format from old world to new world, from modern to legacy.

Transforming content is only part of what must be done to repurpose legacy IT systems. The other is the identity of the consumer. You will almost certainly have to adapt the security model exposed to new third-party API consumers to the ones employed by your legacy systems. This is usually done by a different entity called an API Security Service. This gateway brokers the security mechanims exposed on your API to the methods in use by the systems you are abstracting. Practically speaking, this typically means transitioning OAuth and OpenID Connect, which are exposed to API consumers, to WS-Security, certificates, and/or symetric keys that legacy systems support.

The combination of the API Integration Service and the API Security Service form what is called the API Managment System. This is one of three core pillar of any API platform. I will cover these in a future post. Subscribe to our newsletter to be notified of new posts, and catch that as soon as it is out.

## Insist on Standards

The lifetime of any successful platform will be long – 7 years minimum. It is not a short-term project. It is a stratic initiative. To make certain that it is future proof, the technologies must be standards-based. This will ensure longevity.

I cannot speak authoritatively about standards for API design; however, I can tell you with full assurance that the security of your API platform should be based on the Neo-security Stack which is comprised of these open, international specifications:

- **OAuth 2** for delegated access
- **OpenID Connect** for federation
- **JSON Identity Suite** for identities and tokens
- **SCIM** for provisioning
- **ALFA / XACML** for authorization

## Avoid Vendor Lock-in

As I emphasised when I was on tour with Nordic APIs in April, it is essential that you avoid vendor lock-in. **No one sells an API platform**. If anyone tells you they do, they are lying to you! An API platform must be built. To protect your investment and to ensure optimal interoperabily with new and existing systems, you should select best-of-breed vendors that focus on a very narrow part of the API platform. Pure-play vendors always support standards because they have to connect with so many other software components; standards-based interfaces are the cheapest way for them to achieve this. These suppliers are also more innovative and responsive due to their smaller size, and will be quick to upgrade their offerings to meet new requirements that you introduce (if they are generally useful).

> A future-proof platform is standards-based and built from multiple vendors' products; it does **not** expose proprietary interfaces, nor is it sourced from one supplier.

Not only does no one sell an API platform, no one is going to sell you *your* APIs either. Perhaps that is obvious, but it does mean that you must have a strong competence in software development. If you outsource this development, you are exposing your core competence to someone else. Big outsourcing firms may then hold you hostage or begin competing with you directly. Also, be mindful about using consultants instead of employees. Both can go at any time, but the later are more loyal. You will need them as your platform ages. Incentivize them to stay, and invest in them as well as the platform itself. This may require a reversal of your sourcing strategy, so find allies in HR and management.

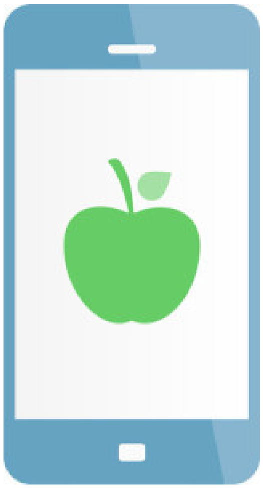
## Continuing the Conversation

As mentioned above, your internal evangelisation will require you to connect with a lot of different people. Do this externally as well. For example, generalize your internal blog posts and share what you can on your company's public blog or on your own personal blog. Share those and other ideas on your social networks. Present your ideas at conferences like our upcoming Platform Summit. For more on this topic in the meantime, check out my presentation on how to transform into a platform for innovation and agility.

Be sure to share your thoughts in a comment, on Twitter, Facebook, or Google+.

# Lessons Learned from Apple's API Strategy

*By Andreas Krohn*



Apple is one of the biggest API platforms in the world – every app running on every iPhone and iPad is built using the Silicon Valley giant's APIs. We can all develop a better API strategy by examining how one of the leading API platforms:

- Uses their own APIs within their own ecosystem;
- Handles changes to these services; and
- Prioritises end user experiences over other concerns.

During three excellent episodes of the iOS/OSX programming podcast, Debug, hosts Guy English and Rene Ritchie interviewed Apple veteran developer Nitin Ganatra. Nitin has been at Apple for many years and has worked on everything from System 7 to OS X to iOS. In the third interview, Nitin and the hosts talk about Apple's views on APIs. The three were not discussing the kind of

APIs we usually write about on Nordic APIs (i.e., web-based ones), but rather lower-level Objective-C APIs. Despite this, the interview does reveal a lot about Apple's approach that we all can learn from as we build higher-level, RESTful APIs.

## Tasty, Tasty Dog Food

In the initial release of the iPhone operating system, Apple used internal classes to handle operations like scrolling and tables in their own apps. These classes had good performance and were easy to use, but were available for Apple developers only. External developers did not have access to these classes, and instead had to use public APIs that Apple created for third-parties. These were harder to use and required tricks to get comparable performance. To make app development easier for these important stakeholders, Apple changed their apps to use the same classes as these external developers. According to Nitin this was done ".even though we knew that by doing that we were deliberately slowing down our apps."

This "dog-fooding" of the public API forced Apple to improve the usability and performance of their outward interface. This type of testing echos recommendations given by other API providers like Fyndiq and PlanMill which we have shared before. Keep this in mind when you are developing apps on your own API. If you need a special API that is not available to external developers in order to build your app, that is a clear sign that something might be wrong with your external API.

## End User Experience Matters Most

There is a lot of talk about Developer Experience (DX) in API circles. This is not surprising when you consider that developers are the consumers of APIs. When the API is the product, it is important to delight the customer with great experiences. What we cannot forget in these discussions, however, is that there is another very important stakeholder – the end users of the apps. In the Apple case that Nitin discussed, this is anyone with an iOS device. If an app stops working because the developer does not keep up with Apple's API changes, for example, it is taken very serious by Apple, as Nitin explains:

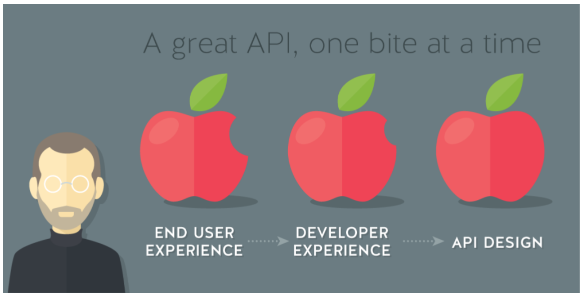
> Even in a case where a developer is knowingly not doing the right thing for one of their own customers, it's still, to that customer, going to look like an Apple problem. You can still end up with unhappy customers, and that's going to reflect poorly on Apple. We had to take that very seriously.

One of the hosts, Guy English, responded to Nitin's explanation of Apple's views by saying:

> I think what's interesting here is the way that you're describing the end user of the software. A third party [developer] may see the person using their software as their customer, but Apple is seeing that customer as

> Apple's customer. It is *their* duty to *their* customer that stops them from shipping APIs or making promises to third-party developers that they may not be able to keep. The intention is not to limit developers and the possibilities of developers, but the intention is to give the customers – the end users, the real customers – the best possible experience.

Guy's summary points out that the goal with any API is to provide value. As with Apple, this value is not created when a developer uses the API. Rather, it comes when an end user consumes the app that developers have created with the API. Do not lose this perspective as you strive to create the best API you can. If you do, the API you make will have a great DX which in turn will help developers provide a fantastic end user experience.



## Be Careful with Change

It is easier to change an app than to change an API because the stability of this interface is depended upon by many different apps which are implemented by different developers. As Guy English puts it in the podcast:

> APIs are one of the hardest things to shift because you can always update an app to fix a problem. [With] an

> API, you’re stuck with it for probably far longer than you want to be.

Apple is known for being conservative with API changes, and they have good reason to be, as Nitin expresses:

> You have to think about the next five plus years out when you’re implementing some of these things or developing these APIs, because, like [Guy] said, you’re going to be stuck with it for years. It comes from that hard-fought knowledge that every time something relatively simple has worked it’s way into an interface, or most of the times that you’ve tried to cobble over a problem with an API, or provide some new functionality by some “clever trick”, those are the things that are going to bite you in the ass hard in three years.

When an API is popular, there can be no such thing as breaking changes because such updates would impact so many developers and end users. This type of break places the focus on the wrong things. As discussed above, UX and DX trump API design. Changes will be needed, that’s for sure. When they are, however, make them very careful since their repercussions are hard to foresee and long-lasting. After just a couple breaking changes, how popular will your API be then?

## Summary

From this interview with Apple, we can see that good APIs require us to:

- Focus on the end users experience;
- Eat our own dog food; and

- Be careful with changes.

This is not novel advice, but it is powerful in its simplicity. Hearing these priorities from a company with as much API experience as Apple is very compelling and noteworthy. For more, listen to the interview or read the full transcript yourself. Thank you to Debug for a great podcast and a big thank you to Nitin Ganatra for sharing!

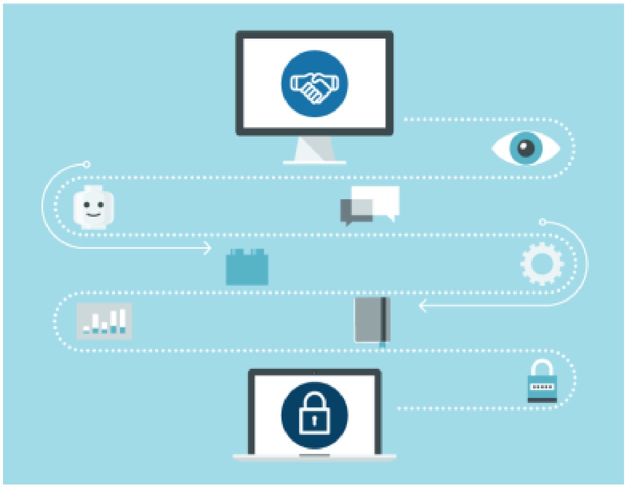
If you have learned valuable lessons in your API project like these, please share them in a comment on our blog, on Facebook, or Twitter.

# LEGO's API Strategy: Resourcing Developers & Building a Business Case

*By Mark Boyd*

APIs are like LEGO building blocks, it is often said. Many API talks start off by explaining that APIs are like these bricks. They can be combined in innovative and creative ways beyond their originally intended purpose. What are APIs like for LEGO though? For those that are planning on attending our Platform Summit in October, you will hear the Danish worldwide phenomenon tell their API story first-hand. However, we wanted to share some of their experiences with our entire community. After catching up with Dennis Bjørn Petersen, Platform Architect at LEGO, we're excited to give you the scoop on their API strategy and implementation.

The following insights from LEGO will help you execute a better API strategy that is able to meet the requirements of your customers, partners, and authorities.



## LEGO's API Strategy

LEGO's API journey is somewhat unique in that they started with a partner API and progressed to a private API from there. This flow usually goes the other way, increasing in openness not decreasing. For LEGO though, the initial point in this progression came out of discussions the toy maker had with their video-game partners, Warner Brothers and Funcom. These third-parties were creating LEGO-themed video games that would be compatible with a wide range of mobile devices and for the web. "When someone has to log in to any LEGO system, they should only log in to one system, and that is our authorization. So, that's why we started collaborating," Dennis explains.

> In that way, we ensure we know what sort of information we are collecting from users... We are often working with children under 13 and being clear about what information is collected is crucial for us," the platform architect continues. Around the world, many

> countries have strict regulations about what data can be collected from children using the internet, and compliance with these is a duty that LEGO takes very seriously. It is not a responsibility that LEGO would consider passing on to a business partner. Dennis says this is not a reflection on any of their business relationships, but that "it is in our mutual interest that this is covered by us.

As work on the API design progressed, the company was also looking to upgrade its identification and authentication processes used internally. Dennis' team quickly saw the benefits of using this new API that they were creating for internal authentication processes as well. In this way, the Partner API came first and led to it also being used as a private or internal API by the company. "We decided we may as well create an API for both external and internal developers. It is used in games, on our message board, in our rewards program. It's used everywhere you need to log in with a Lego ID," Dennis told us. This system thinking undoubtedly led to a much higher return on the investment that LEGO made to integrate with these two partners.

> To be a platform is to in-source your core. Platforms cannot outsource critical aspects of their business to others. ## Tools & Processes

Though LEGO is a global market leader with large amounts of resources at its disposal, they have adopted a lean approach to managing their API. They have assembled a series of lightweight tools, processes, and techniques to help them meet the business objectives of their API. Dennis shares some of the simple strategies they use in their partnership and internal work.

## Documentation

As we have blogged about before, API consumers need information if they are to use your service to solve their problem. This includes API documentation. LEGO knew this, and prioritized the creation of such resources that would aid API consumers in the use of the service. As Dennis told us, they built a test site with examples and snippets of JavaScript code that consumers could add to their applications to quickly and easily begin using the API. Providing API documentation is a best practice that Ben Nunney, European Marketing Manager at Twilio, shared last fall at our business-focused event.

## API Metrics

If you don't count it, don't do it, the adage goes. Tracking key indicators of your API's usage and performance is really important. The level of sophistication of your tracking system should increase with your API consumption. Otherwise, you may implement a large analytics system for an API that does not end up being adopted. Take a lean, just-in-time approach instead. This is what LEGO did. Dennis told us that a manual process was sufficient from the beginning. This is similar to the techniques that Västtrafik has shared with the Nordic APIs community in the past. This DIY approach is a good place to start as you dog food your API, but should be replaced by purpose-built tools over time. This advice is aligned with LEGO's API rollout. As Dennis says, "We can see that usage [of our API] is increasing, so we are looking into an API management tool to take the load off our backs."

## Error Logging

Like Storebrand, a large Norwegian life insurance company which I've previously blogged about, LEGO has built an error logging system to help them discover, diagnose, and fix errors in their API.

Finding help when something goes wrong is pretty easy within LEGO and amongst the smaller partner ecosystem that is currently using it. As consumption grows though, the importance of this system will grow. "We do like to keep an eye on how our API is being used and if it is being used correctly," says Dennis. Following these practitioners examples is a good idea as you roll out your API to internal and partner consumers. Starting with a Minimal Viable Product (MVP), which may include rudimentary error logging, is another best practice that Ben of Twilio encourages. As with metrics though, the sophistication of this system will likely grow with increased consumption. At that point, a DIY approach may have to give way to more comprehensive, commercial log monitoring products.

## Mentorship & Pair Programming

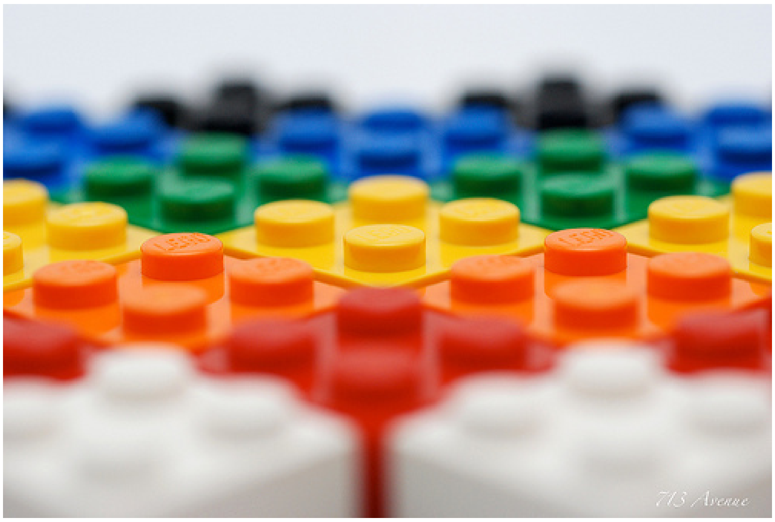
Pair programming is a technique encouraged by agile software development experts worldwide. LEGO follows this advice even when the developers being paired together are not from the same organization. Dennis says that, "[w]hen working with external games developers, we normally do a code review, and we send one of our developers to sit with our external partners and work on the first integrations." Sending a developer to another city just to perform the integration might seem suboptimal. Overall, however, it is the best way in some cases. As Dennis continues, "It's not that our API is complicated…it is just easier and faster to sit with them." This technique is one that Travis Spencer, CEO of Twobo Technologies and co-founder of Nordic APIs has seen in previous partner API implementations. He explains:

> I once did a job with a large TV studio in Hollywood. We were working 1,000 miles away in Portland to develop the data access APIs for a complicated Content Management System. Every month or so, we flew

> down to Hollywood to work with the customer and their partners to integrate new drops. This work wasn't complicated when sitting in the same room, but it would have been hard over the phone or via email. Working shoulder to shoulder saved tons of time, and was cheaper in the long run.

As you implement your internal and partner APIs, consider if pair programming can reduce the Time to Value (TtV). As your API scales to the general public, however, you will need to consider other means to facilitate sharing within your developer community. At large scales, electronic, on-line methods will be required. Learn from the techniques you use when providing your API to internal and partner stakeholders and iterate over them as usage grows.

## Building One Brick at a Time



Like any of LEGO's construction sets, the company is assembling the API one "brick" at a time. What started with the LEGO ID API that is now being used across the company, others may follow as well. Dennis says that they "met with Disney a couple of years ago and they have some great ideas about what you can do with an API." This type of market-driven development is probably the best approach for larger enterprises. In so doing, the platform opportunities to create value using APIs will be apparent to all parts of the business (e.g., sales, marketing, finance, top management, etc.). This people-oriented approach is one of the two things that will transform your business into a platform for growth.

Comment here, on Twitter, or on Facebook with your thoughts on tools, processes, and best practices you would recommend to other API practitioners in the community.

# API Strategy Must Balance Developer and User Experience

*By Mark Boyd*

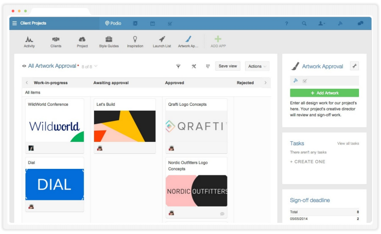
We recently talked about how people and software must change if you are to grow your API. We also looked at a case study of Apple which highlighted the importance of a UX-focused API strategy. The Apple example is interesting because it shows that you must strive to delight your API developers' consumers as if they were your own. To win them though, you must first attract the devs. This requires an API program that also prioritizes Developer Experience (DX). To become a successful API platform, you and your team have to get both DX and UX right.

We have discussed DX before on our blog and at our events, but not from a platform-oriented perspective. With an aim toward a broader view, we wanted to explore this topic further by examining another API platform called Podio.

## Becoming a Platform, One Developer at a Time

Podio, a Danish cloud software company now owned by Citrix, provides composable widgets that can be used interchangeably to create business-specific workflows and processes. The LEGO-like ability of Podio Apps, as they're called, to be connected together by non-technical workers enables the creation of collaborative spaces where people are able get their jobs done. Podio's self-service tools allow end users to manage their work in a way that suites them best. Using custom-assembled dashboards and heads-up displays, users are able to oversee all stages of their projects. In this way, the apps running on Podio's work platform create additional value for the end customer, a hallmark of any API platform.

Podio



Podio's clients range from startups to independent contractors to enterprise customers, all looking for flexible project management software that is pliable enough to match their specific needs. Looking out over this customer base, Podio identified two main groups of creators:

1. Systems integrations
2. Mobile app developers

These two groups can use the API to build Podio Apps that integrate platform services with other apps and business systems. These constituencies of third-party developers can launch new integrations and products that they can commercialize through Podio's marketplace. By making it easier for members of the ecosystem to co-create, Podio is fostering the growth of their own business and the businesses of those who are adding value to their API. In other words, **their API platform is successful because of a balanced prioritization between DX and UX**.

To make it even easier for developers, Podio has identified additional ways of reducing the friction that these two groups feel when attempting to consume their API. These include:

1. Introduction of a mobile SDK
2. Showcasing of successful API usage

## SDKs can Speed up Developer On-boarding

This optimization strategy enables developers to create mobile apps that *seamlessly* connect with their API. This use of an SDK to reduce the Time to First Use (TTFU) is a best practice recommended by API expert, Holger Reinhardt of Layer 7. Like any SDK, Podio's new offering is a helpful add-on but no substitute for a well-designed API. Like Holger, Gustav Jonsson, Product Manager at Podio, says SDKs are an important entry point to help developers get up and running quickly. "It's really about creating a seamless and easy way for developers to get started with the API," Gustav says.

## DX Starts with a Well-designed API

At Podio, the need to build a mobile SDK was born out of dev requests. "We saw questions coming from developers in our community around utilizing Podio in native apps, and we wanted to help people build stuff faster on mobile," Gustav says. In this way, the SDK enhanced the DX, as it was exactly what developers wanted. Podio's API roadmap did not start with the SDK, but it is becoming an important part of a platform strategy to make it easier to onboard more developers. As Gustav says: "We already have an active and dedicated developer community. SDKs are an important part of that puzzle." This path follows the route suggested by Ben Nunney, European Marketing at Twilio, who says that you should not begin with an SDK. Instead, you should build an API that caters to the desires of the developers in your communities. This DX-lead strategy is also advocated by Ronnie Mitra, another API expert at Layer 7.

## Success of an API Platform Equates to Success of its Users

Podio has also developed a section of its website that markets developers' products and solutions that are build on the Podio API. These showcase pages strengthen the ecosystem of third-party developers. For example, Oval Business Solutions offers free apps like the Timeline for Podio. While this integration is free, the Podio App refers users to Oval's site where they are offered additional services. Danish development shops like BendixKiel and Phases are building freemium and paid apps that use the Podio API. These are marketed to other Podio customers via the platform's third-party developer showcase.

As in these three cases, Gustav explains that "there are multiple ways that extension developers are charging for their products. Giving this freedom to the developers is a good thing, since they can be flexible in the ways that they'd like to charge. We want to share data with the developers on what model is most efficient." This innovative idea to provide feedback on API consumers' business models is a telltale sign that Podio aims to create a mutually beneficial ecosystem with their platform.

By highlighting the success of others, providing a marketplace, and delivering feedback on successful API-based business models, Podio is leveraging ambitious, outside talent to position itself in the center of more and more people's work day. This is the power of a platform play.

## Summary

The priorities and development of Podio reflect a desire to grow as a platform-based API business. As such, Podio recognizes that it can reach into new markets by being part of a larger ecosystem. By fostering a vibrant community of product creators and specialist service providers, Podio will allow others to add value atop their offerings by making use of the various APIs, tools, and software they provide.

By looking at the Apple case and this one, we can infer some important points about an API platform strategy:

- Design an API with the end user in mind (UX) that also appeals to your developer community (DX)
- Know who is using your API (e.g., through analytics, customer interviews, etc.)
- Give developers the resources needed to quickly begin using your API (e.g., with an SDK)
- Create opportunities for partners to earn revenue by using your platform
- Establish a channel and marketplace that connects your API partners' customers to you and you to them
- Strive for a balance between UX and DX

What do you think? How does DX differ between an organization that is providing an API and one that is an API platform? What are some of the challenges of balancing DX and UX? We would love to hear your thoughts. Share them in a comment here, on Twitter, or on Facebook.

# Marketing your Public API

*By Rhys Fisher*

So you've decided to publish a public API to enhance your business. That's great news; however, it is just the first step if you're serious about reaping the full benefits. The API needs to be consumed in order to create value for its publisher, which means finding API consumers; a lot easier said than done. This job requires some marketing, and just like any demand generating initiative, there's the possibility of wasting money, which we know all you smart API practitioners don't want to do. **In this post, we will dive into the do's and don'ts of API marketing**.

## Recognize your API is a Product

### Don't Expect Them To Come Running

Whatever you do, realize that the "Build it and they will come" philosophy is a myth. If you don't tell people about your API, then they aren't going to come running. They have better things to do. If you don't care enough about your API to spend time marketing it, you will have a hard time inspiring others to want to use it.

### Do Encourage Them To Come Walking

Treat your API as a product and market it as such. This includes building awareness, interest, and desire; enough of it to spark action. Get others to spread the word by being involved in your community as early as possible. Word of mouth is the most effective form of marketing on the planet, but you need to get them talking.

## Involve Outsiders

Either an organisation has an API and the decision is made to publish it to the world, or, no API exists, but IT decides to develop one because it is what the competition and the cool kids are doing. IT publishes the API. End of story! Both these scenarios are common, and they both usually end in zero adoptions and minimal gains for the publisher.

### Don't Develop an API in Isolation

If you do, it probably won't fit the requirements that the potential new API users might have, or your data structure simply won't make sense to them. If you do not already have the luxury of having your own developer community, try to find existing communities. Ask on Twitter, find relevant Facebook groups, be active on Stack-Overflow, ask on Quora, and come to API conferences to meet people in person.

### Do Involve Marketing Talent

If your marketing department does not know about the API (or about what an API is for that matter), how can they then assist in getting the word out? Successful APIs are not IT only projects; they are company projects.

## Keep The User in Mind

Ignore this advice and users will ignore you just as you ignore products in ugly boxes when you go shopping.

### Don’t Ignore Your Users’ Questions

If you’re not sure what type of support material you need to create, don’t worry. Check out this great example from Twilio, and pay close attention to the questions you’re receiving from the API users. Use these queries to guide your API support material.

### Do Provide Your Users With Solutions

That includes everything from the documentation to the support, everything that you need in addition to the actual technical API. This stuff takes time to produce, and it is hard to do well, so commonly it is completely or partly ignored. Don’t be common, be a good example and get noticed for it.

## Build a Community

### Don’t Believe You Know Everything

Save yourself some heartache by listening to lessons learnt from the world of entrepreneurship: you don’t know what’s going to work! Getting things right is an iterative process. Make it easier for yourself by building a community from which you can draw ideas. Community building is a long term effort, but at the very least they need a place to gather, and engage with your brand. This could take the shape of a blog, forum, and social media.

### Do Learn from Your Users

Having a community, and ignoring its conversation is like reading a few pages of a book while dwindling on how the outcome of your last meeting, and then needing to re-read a page or two. It's an easy error to make, but can completely reverse the benefit of the action. Start a dialogue with the community as soon as possible, and at the very least, record the pains and questions that pop up frequently. Use this resource to help guide your support material.

## Getting Outside the Building

Marketing is not always as fun as producing a viral video, or as simple as Microsoft's Hotmail viral growth case study, but it doesn't need to be rocket science. It does, however, involve making an effort.

### Don't Avoid the Hustle

As Uber entrepreneur, Steve Blank, once said, there are no facts inside the building. If you're building a public API with the idea of it getting used, you are going to want to get in front those end users, and figure out how they are going to use your product (What's the job to be done?). Consuming others APIs can highlight which best practises you should be implementing, as well as those practises you should be avoiding.

## Do Get Your Hands Dirty

Developers, entrepreneurs, and enterprise architects are not your usual target audience, but they are people just like you and me, so meet them. Hackathons, meetups and conferences attract this demographic, and are great for meeting potential early adopters. It's about getting your API a necessary dose of reality.

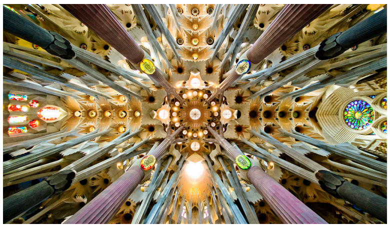
## Summary

By now you're probably realizing that publishing a public API is no free lunch and that you will need to keep your users close if you're planning on getting your API used. You will also need to channel your "front line learning" into quality support material, which is the much needed packaging of your API; something that every quality product needs, especially if branding is at all important to you. Let's not forget how important it is to encourage API evangelism by creating meaningful relationships with your community. It's hard work, we know. But doing anything worthwhile tends to have an element of challenge. When you start benefitting from the ecosystem advantages from becoming a platform, that's when you'll realize that all the work was well worth it.

If you wish to learn more about API marketing and how to become an API platform, join the conversations that is taking place on Twitter](http://twitter.com/nordicapis), Facebook, and Google+.

# 12 Killer Resources for API Practitioners

*By Rhys Fisher and Andreas Krohn*



A hammer, chisel, file, rasp – all humble tools. In the hand of a master like Antoni Gaudí though, they are the instruments used to create a world wonder.

Just like these crafting instruments, the API is the tool providing developers with the opportunity to reshape our digital world. It is the tool that is forming online wonders that will bring about the next phase of the digital economy – the great connection. It is an exciting time to be working with APIs.

Just as for the master Spanish sculpture, continuous learning is essential. It is critical that your API skills and knowledge do not become outdated, an easy outcome given the fast moving environment we find ourselves in.

This is why I felt it would be useful to bring together some of the best online API learning resources. Don't hesitate to add your own favorites in a comment or on our Facebook page.

## Zapier's Introduction to APIs

Zapier has done a great job putting together a course introducing APIs. It is perfect for those just starting their API journey and covers protocols, data formats, API design, authentication, real-time communication, and implementation.

## Entrepreneurship and APIs

If you are looking for an easy read about how important APIs are in entrepreneurial endeavors, I wrote an explanation on the topic a few weeks back. It briefly introduces some important business benefits of APIs, and why upcoming business builders should be learning and using this technology.

## API Academy

API Academy provides free online lessons and in-person consulting services that cover essential API techniques. This useful resource provides business managers, interface designers and enterprise architects with some important knowledge. The growing repository is one you will want to revisit every so often.

## API Evangelist's Whitepapers

Kin Lane's white papers on the basics of APIs, their history, and how to deploy and manage them are a useful collection of resources. He has also just released one on API design that is worth checking out.

## News from ProgrammableWeb

If you want to keep up to date on API news and information, subscribe to the ProgrammableWeb.com news feed. They have daily updates about the latest API scoops. They also have the world's largest API directory and search engine. If you want to get the word out about your API, getting it listed there is a smart idea.

## API Industrialization by Accenture

This informative white paper demonstrates how to deliver differentiated APIs. It explains how to use them to scale and capture new opportunities. If your organization is transforming itself into a platform, Connecting the Digital Ecosystem will provide useful insights.

## Traffic and Weather API Podcast

This semi-regular podcast is hosted by John Sheehan of Runscope and Steve Marx from Dropbox. They talk about the latest API- and cloud-related tools, news and technologies. If you enjoy listening to your content, catch their broadcasts.

## API Testing DOJO

API testing is an important part of the development process, and it is essential to hone this skill. Just as Gaudí had multiple tools, so will every API practitioner. Testing is an important one. To help you improve these abilities, check out the API Testing Dojo. It provides API developers with resources that help determine if their services work as required.

## APIUX

For an interesting point of view, check out APIUX. This blog examines APIs from the usability perspective of API developers. As Ronnie Mitra has explained during past Nordic APIs events, this focus is essential for successful API adoption. The articles, product reviews, and guidelines from APIUX are a great resource to learn more about developer experience.

## API Economist

The API Economist is a place for developers, and enterprise architects that are seeking a conversation about the business, politics, and technology of APIs. It is a great blog that is worth subscribing to. Readers will find thought-provoking posts from some of the community's pioneers who are working hard to advance the API economy.

## API Magazine

Manfred Bortenschlager curates a collection of ideas and articles that revolve around APIs, developer evangelism, mobile app strategies, and ecosystem dynamics. His assembly of various gems provides a useful anthology of API info.

## Nordic APIs

Smarter Tech Decisions Using APIs
High impact blog posts on API business models, and tech advice
Connect with market leading platform creators at our events
Join a helpful community of API practitioners

Like these fine folks, we are striving to provide useful info about APIs. We are doing this through impactful and useful blog posts, and by hosting regular API events. Through on- and off-line community outlets, our mission is to help API "doers" to succeed. Sign up for our newsletter to be notified of new blog posts, to get access to VIP content, and to receive special discounts to our events.

What are some of your favorite sources? Did we miss one of your most useful resources? Share a link to sites, videos, presentations, and other learning materials in a comment here, on Twitter, or on Facebook.

# How to use APIs to Spot the International Space Station

By Staffan Sölve

Some of the objects you can spot on a clear, starry night move a bit faster across the sky than stars or planets. The space above us is sprinkled with airplanes, satellites and other man-made objects. Every now and then, from where you are, you can spot the International Space Station (ISS), soaring like a bright shining bullet across the night sky. Depending on where you are on the planet, it may be several days between sightings.

Spot The Station. Here, you can sign up for an alert email, or use the RSS feed, for thousands of places worldwide. It tells you the exact time of appearance, and where to look for the ISS on the night sky.

## Houston: We Have a Problem

The alert from NASA works perfectly, but it does not take into account the weather conditions. No point waking up in the early morning hours to find the sky is too cloudy for an ISS sighting, is there? But if we can match the ISS alerts against a weather forecast, we could filter out only those alerts which are likely to appear on a clear, starry night.

For example, we can use the weather forecast API from yr.no, which is available in XML format. The RSS feed from Spot The Station provides ISS sightings. Both services should be free to use for experimental purposes like this, and require no API keys. For complete terms of use, please see om.yr.no/verdata/free-weather-data and spotthestation.nasa.gov respectively.

## The Essentials of the API Provided by yr.no

It is easy access the weather forecast from yr.no in XML format. Search for your location on yr.no,. When you see the forecast web page, just add `"forecast.xml"` to the URL. The XML output starts with a header, containing for example your location:

```
<location>
    <name>Lund</name>
    <type>Administration centre</type>
    <country>Sweden</country>
    <timezone id="Europe/Stockholm"
            utcoffsetMinutes="120"/>
    <location altitude="51" latitude="55.70584"
            longitude="13.19321" geobase="geonames"
            geobaseid="2693678"/>
</location>`
```

The actual weather forecasts begin after the tags `<forecast>` `<tabular>`, and each time period is enclosed by the tags `<time from=...` and `</time>`. The code symbol number="1" below is the essential information, that this particular time period will have a clear sky:

```
<forecast>
<tabular>
<time from="2014-05-23T09:00:00"
    to="2014-05-23T12:00:00" period="1">
        <symbol number="1" numberEx="1" name="Clear sky"
                var="01d"/>
        <precipitation value="0"/>
        <windDirection deg="133.8" code="SE" name="Southeast"/>
        <windSpeed mps="5.5" name="Moderate breeze"/>
        <temperature unit="celsius" value="21"/>
        <pressure unit="hPa" value="1008.0"/>
</time>
...
```

## The NASA RSS Feed for ISS Sightings

Go to spotthestation.nasa.gov and select Location lookup. When you found your location, there is a button for the RSS feed. If you save the file and look in the header, it includes the link to your location:

```
<link>http://spotthestation.nasa.gov/sightings/view.cfm\
?country=United_States&amp;region=Alabama&amp;city=Abbe\
ville</link>
```

Following the header, you have a list of sightings ahead, each enclosed by the tag `<item>`:

```
<item>
        <title>2014-05-11 ISS Sighting</title>
        <pubDate>06 May 2014 08:34:00 GMT</pubDate>
        <description>
                Date: Sunday May 11, 2014 <br/>
                Time: 5:04 AM <br/>
                Duration: 3 minutes <br/>
                Maximum Elevation: 16 ° <br/>
                Approach: 10 ° above SSE <br/>
                Departure: 14 ° above ESE <br/>
        </description>
        <guid>http://spotthestation.nasa.gov/sightings/view.cf\
m?country=United_States&amp;region=Alabama&amp;city=Abb\
eville&amp;ss=26AE328A-A321-777C-A06E3DAFC6069155</guid\
>
</item>
```

The tag `<title>` includes the dates, which we can match against the dates in the weather forecast from YR. The tag `<description>` includes all the details of the sighting, which are useful in an email alert. Note that NASA only lists sightings which will occur during dark hours, in local time for your location.

## Putting it All Together

Now, these data sources provide the sources we need to create our own alert function. It filters out only those ISS sightings which are likely to occur on a clear, starry night sky. I have put together a working PHP script, which you can use to experiment further: github.com/staffansolve/Hello-Sky

Happy star-gazing!

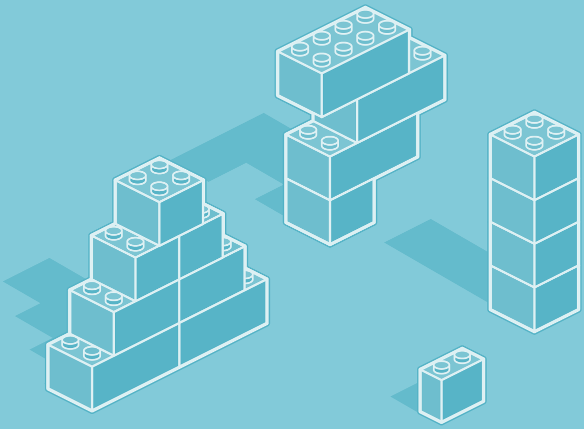
# Endnotes

- Gaudi photo by SBA73
- LEGO photo by 713 Avenue
- Telescope photo by Ryan Wick
- Thanks to Kin Lane and Mike Amundsen for initial feedback and critique of our API platform definition.

NORDIC **APIS**

## PLATFORM SUMMIT

October 20-22, Stockholm



Learn how to become an API business platform from authorities such as SoundCloud, LEGO & Kin Lane in our **2nd annual conference**.

REGISTER AT NORDICAPIS.COM