THE API OBSESSION

(and why it matters)

James Higginbotham
LAUNCHANY.COM

INTRODUCTION

There has been a good deal of discussion, both positive and negative, about the recent obsession of APIs. Some say that it is ushering the next wave of product companies, while others are asking, “What’s the big deal?”

To the non-technical, APIs might be a bit of mystery and magic – build an API, sell your company for lots of money. To the technical, it offers a way to build and share your work with other developers, making money along the way.

APIs themselves are nothing special. That’s right, I said it…and I’m not taking it back.

SO, WHAT IS AN API, ANYWAY?

An application programming interface (API) specifies how software components should interact with one another.

Traditionally APIs were used to build a standalone application. Now web APIs enable applications to talk to other applications across the Internet using the standards of the web.

If you have done any of the following, you have experienced an application that used APIs without you even knowing it:

- Published a tweet using a Twitter web or mobile interface
- Viewed the weather on your mobile phone or tablet
- Shared files between your desktop and a mobile device (e.g. Box, Dropbox, etc)

A LITTLE API HISTORY

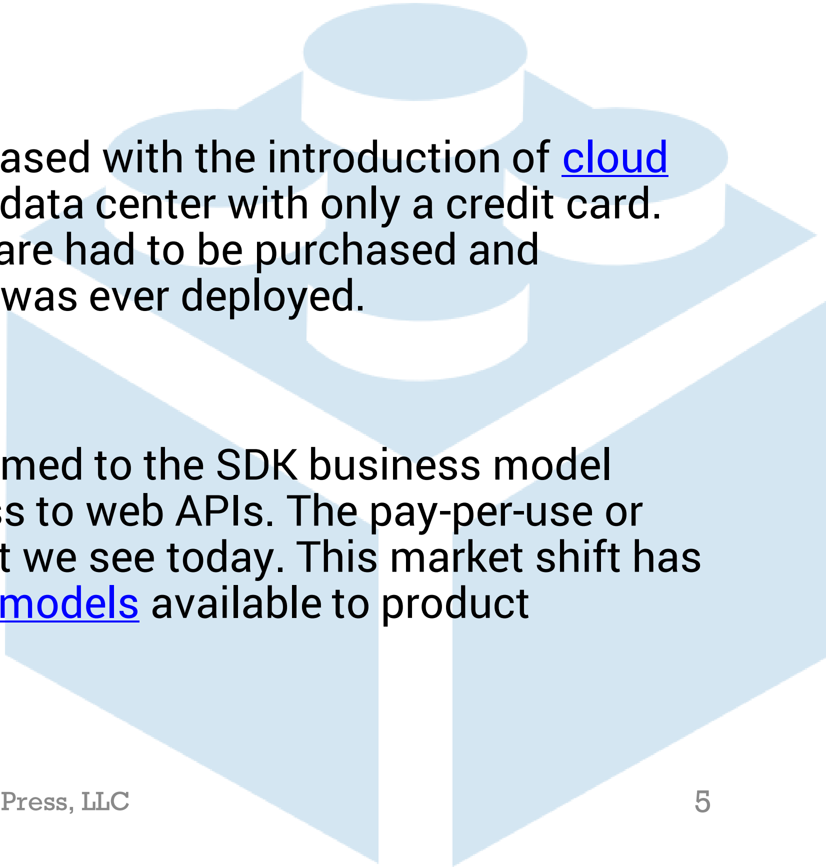
You may not have realized it, but APIs have been around for a long time.

Before the current incarnation, the most popular way to distribute APIs were through SDKs. They were commonly used to integrate new functionality into applications, or to construct an entire application (e.g. the Windows Mobile or Android SDKs).

SDKs define APIs that developers build their code against for a specific programming language (e.g. Java, Ruby, or C/C++). They do this by providing a contract with the developer on what they will do, how to call the code, and what the result will be. The developer then builds code on top of the SDK to get the functionality they desire.

After the SDK came the initial attempt at web services. Many of these early web services used technologies such as SOAP, WSDL, and a number of other acronyms. These technologies required a deep understanding of their specifications before the web services could be used. Additionally, the high cost to develop and deploy these services prevented many companies from offering them in the early stages of the company.

WHAT CHANGED?



Simplicity

The newest web service APIs speak a simple language: HTTP, the language of the web. Combined with other common usage patterns such as REST and HATEOAS, web APIs promote easy integration from any programming language – no SDK required!

Lower Cost

In addition, the cost of infrastructure has decreased with the introduction of cloud computing, allowing for rapid provisioning of a data center with only a credit card. Compare that to just 10 years ago when hardware had to be purchased and installed in a co-location facility before the API was ever deployed.

New Business Models

Finally, the market that was previously accustomed to the SDK business model started to accept new ways of paying for access to web APIs. The pay-per-use or subscription models are the most common that we see today. This market shift has even opened an amazing 20 new API business models available to product companies today.

APIS, EVERYWHERE

The API market has been hot for several years and doesn't show signs of slowing down. In fact, more than 85% of Enterprises will have an API program by 2018 (Layer 7 survey). Here are just a few companies using APIs to their advantage:

Best Buy is using APIs to make their product data and reviews available to mobile applications. This helps their customers price compare and make more informed decisions at purchase time.

Amazon Web Services allows anyone with a credit card to design and provision their own data center. AWS APIs offer servers, databases, and data analytics engines to anyone that needs them.

Netflix gained significant market adoption by enabling any device that has Internet access to stream movies and shows. Now, anything that connects to a TV likely offers Netflix streaming.

Traditional API Development

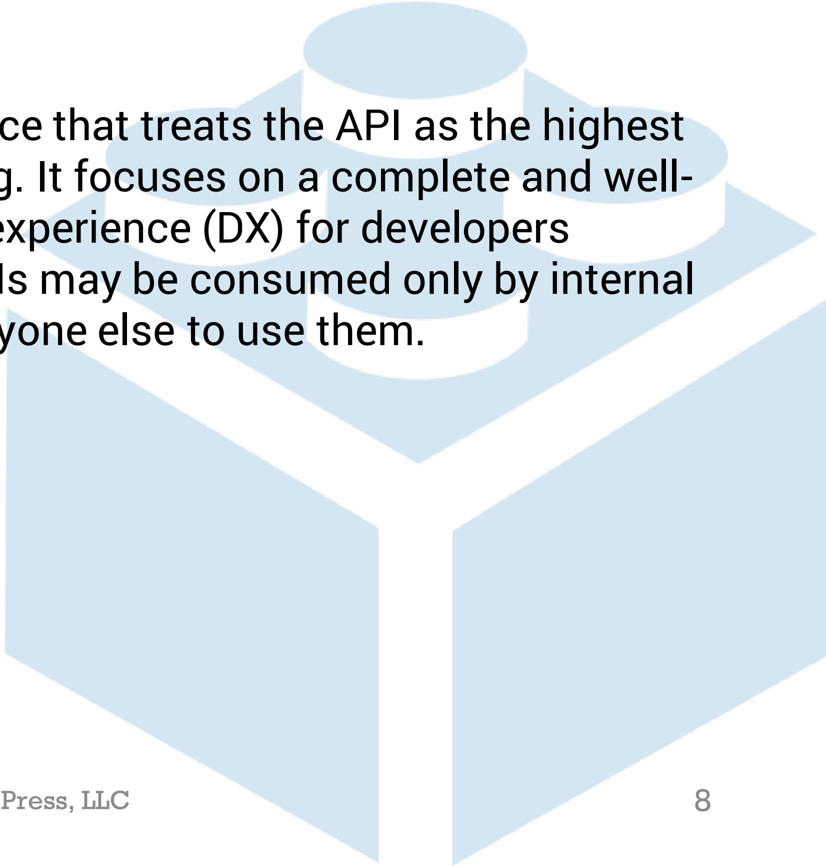
Historically, API design comes only after a company develops a data-rich application. Along the way, the need to build an API emerges to provide access to data or complex processing logic.

However, today we live in a multi-screen, multi-device world. Mobile phones, tablets, televisions, and even our cars have an Internet connection and can have custom applications installed. Each device offers a specific experience based on its purpose and what we are doing at the time. This “Internet of Things” (IoT) trend is leading ISVs and enterprises to reconsider how they build applications. Rather than adding an API on after an application has been built, APIs are now becoming the first thing developed.

MOVING TO AN “API-FIRST” APPROACH

I have been building web APIs for over a decade. From those terrible standards that made my head hurt every day, to the most recent approaches that are clean and simple. I now build web APIs for mobile backends, rich browser applications, and for internal cloud infrastructure management. As a result, I started recommending an “API-first” approach.

An API-first architecture is an architecture choice that treats the API as the highest priority within an application or product offering. It focuses on a complete and well-documented API that offers a great developer experience (DX) for developers building web or mobile applications. These APIs may be consumed only by internal developers or made available for vendors or anyone else to use them.



REASONS FOR GOING API-FIRST

There are five reasons why I recommend considering an API-first approach:

Reason 1: APIs improve the mobile application experience
APIs enable mobile applications to reach beyond their local data to enable collaboration between mobile devices. They also allow for rapidly changing mobile application behavior without the need to resubmit to app stores after every change.

Reason 2: APIs enable more sophisticated web applications
APIs enable more sophisticated web applications that mimic desktop applications. This is currently accomplished through Javascript MVC frameworks such as Backbone and Ember, making web applications just “another device” alongside mobile applications.

Reason 3: APIs create community
Your software product likely reaches out to a specific community and solves a real problem for them. APIs enable developers to rally around these communities, integrating their users with your offering. This enables your product to reach out to your existing community in new ways. It also allows for reaching into new, untapped communities as well.

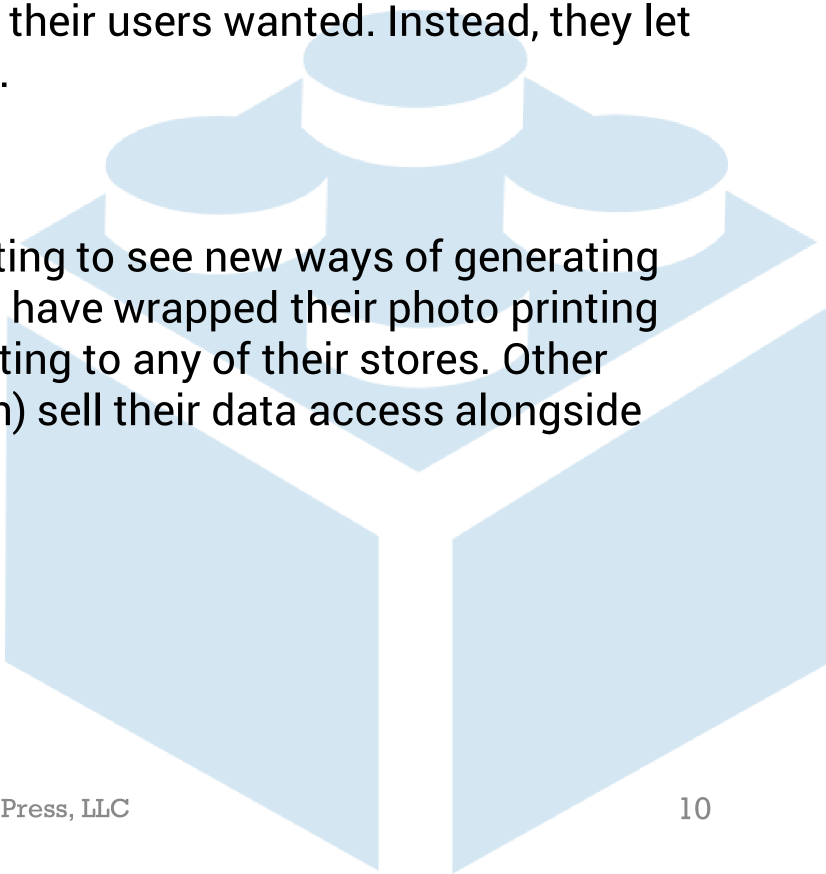
REASONS FOR GOING API FIRST (CON'T)

Reason 4: APIs capture passion

Companies that release APIs to popular websites enable passionate developers to build mobile applications or add-on products. Twitter become popular because developers built custom Twitter clients using their API, reducing the burden on Twitter product managers trying to guess what their users wanted. Instead, they let those passionate about the platform build on it.

Reason 5: APIs generate revenue

Walgreens is one example of how they are starting to see new ways of generating revenue from their existing infrastructure. They have wrapped their photo printing service with an API that enable easy photo printing to any of their stores. Other companies, such as IMDB (acquired by Amazon) sell their data access alongside their premium membership services.



The API Obsession (and why it matters)

NEXT STEPS FOR YOUR API STRATEGY

Now that you understand a little more about what an API is, why it is important, and how the API-first approach can benefit your company, it is time to plan your next steps. I recommend seeking the answers to the following questions to get started:

1. Will my API be internal only, free, commercial, or freemium?
2. What are my KPIs for the API to determine it is successful?
3. What is my API marketing strategy?
4. What are the first set of APIs that I plan to build?
5. Does my team have the expertise to design and build highly adopted APIs?

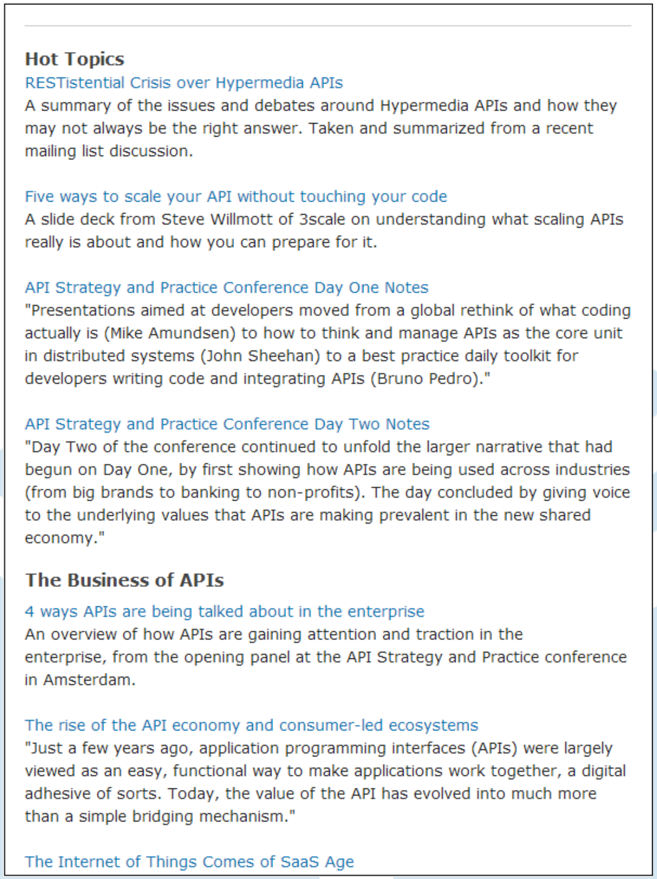
Once you begin to answer these questions, your API strategy will start to take shape.

WANT TO STAY UPDATED ON THE LATEST API NEWS?

Stay updated with the latest API articles and strategies by subscribing to the API Developer Weekly newsletter. It is hand-curated, free, and your email is not shared with anyone else.

Sign up now and be ready for the next issue:

http://bit.ly/api-dev-weekly-obs



Hot Topics

RESTistential Crisis over Hypermedia APIs

A summary of the issues and debates around Hypermedia APIs and how they may not always be the right answer. Taken and summarized from a recent mailing list discussion.

Five ways to scale your API without touching your code

A slide deck from Steve Willmott of 3scale on understanding what scaling APIs really is about and how you can prepare for it.

API Strategy and Practice Conference Day One Notes

"Presentations aimed at developers moved from a global rethink of what coding actually is (Mike Amundsen) to how to think and manage APIs as the core unit in distributed systems (John Sheehan) to a best practice daily toolkit for developers writing code and integrating APIs (Bruno Pedro)."

API Strategy and Practice Conference Day Two Notes

"Day Two of the conference continued to unfold the larger narrative that had begun on Day One, by first showing how APIs are being used across industries (from big brands to banking to non-profits). The day concluded by giving voice to the underlying values that APIs are making prevalent in the new shared economy."

The Business of APIs

4 ways APIs are being talked about in the enterprise

An overview of how APIs are gaining attention and traction in the enterprise, from the opening panel at the API Strategy and Practice conference in Amsterdam.

The rise of the API economy and consumer-led ecosystems

"Just a few years ago, application programming interfaces (APIs) were largely viewed as an easy, functional way to make applications work together, a digital adhesive of sorts. Today, the value of the API has evolved into much more than a simple bridging mechanism."

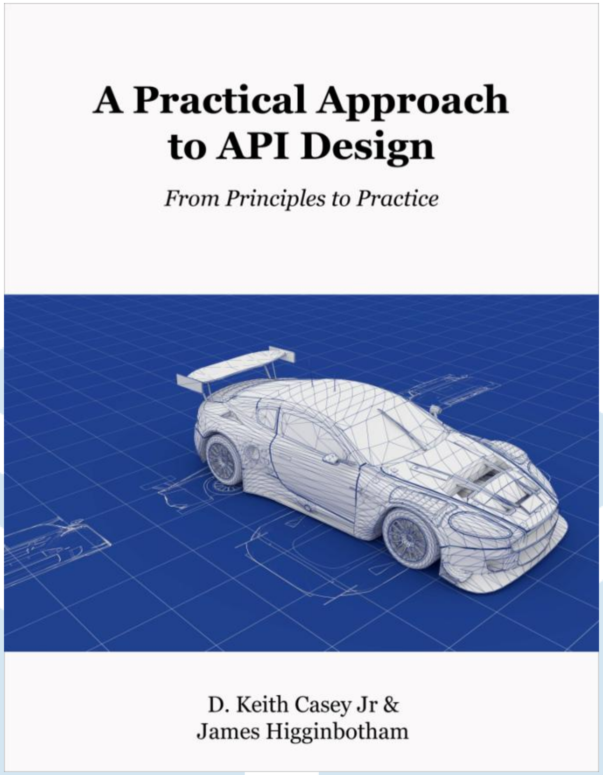
The Internet of Things Comes of SaaS Age

NEED BETTER INSIGHT INTO API DESIGN?

Start your development team off right by learning more about API strategy, design, and development with my recent book, A Practical Approach to API Design, to help you through the process:

- Principles of Pragmatic API Design
- Moving Beyond CRUD APIs
- Tips for modeling and validating your APIs
- API design patterns
- Data architecture
- API product management
- Community and support

Get a free sample chapter: http://bit.ly/1eeyDr1



A Practical Approach to API Design

From Principles to Practice

D. Keith Casey Jr &
James Higginbotham

ARE YOU READY?

While not without its challenges, today's opportunities are endless for both providing and integrating with APIs. You can now build APIs that are ready to integrate and accessible to anyone with a credit card.

Are you API curious? Not sure how to get started? Perhaps you are ready to build out your API but you aren't sure if you have designed the API for a great developer experience.

Contact us to find out more about how to design, architect, and launch your API.

Telephone: +1.512.537.8493
Email: info@launchany.com
Web: launchany.com
Twitter: @launchany

ABOUT THE AUTHOR

James Higginbotham is the founder of LaunchAny, an Austin-based firm that specializes in launching software products. James has launched over 50 products in his career. He enjoys working with companies on their technology and product strategy. James has experience in architecting, developing, and deploying SaaS and APIs to the cloud.

James serves on the advisory board of a non-profit organization that focuses on helping people become advocates and activists for their charities and causes. When not writing or consulting, he enjoys landscape photography.

James can be reached at james@launchany.com