What If Your API Program Had a Fresh Start?

Janet Wagner — Read time: 6 minutes

Today we thought we'd engage in an intriguing thought experiment:

Imagine your API program had a fresh start — what would you do differently?

You could approach this premise in several ways, starting your API program over again:

- Under the same circumstances.
- Under different or better circumstances.
- At a different starting point.

We're taking the approach where you start your API program over again under the same or better circumstances, changing your strategy based on *what you learned the first time around*.

We reached out to several API industry experts, asking them to provide their thoughts on our premise:

- Lorinda Brandon, VP of Software Engineering at BetterCloud - @lindybrandon
- Jason Harmon, CTO at Stoplight - @jharmn

- James Higginbotham, Executive API Consultant at LaunchAny - @launchany
- Sophie Rutard, Head of Doc Mgmt, Identity and Access Mgmt, API DevRel at Euler Hermes - LinkedIn
- Erik Wilde, Catalyst at Axway - @dret

They provided many great insights, which we used to create this list of tips on building an API program from scratch.

Find the Right Partners

Many companies focus on the technical implementation of their API program and don’t pay enough attention to the business side of things. You need to bring together people across your company who understand the business value of your API program and can help you move forward with it.

In this podcast, Tanya Vlahovic offers advice for those starting their API program from scratch. Tanya is a former architect at eBay and currently an architect at Salesforce.

She recommends that you partner with the architects from your organization, from various teams — although that depends on your organization’s structure. You also need to partner with business stakeholders to understand the vision for the program. There is a technical vision and a business vision that you need to realize, so you need to bring these different stakeholders together at the beginning.

Erik Wilde told us that “next time, I would be less tech-deterministic and look more at the business and organizational side of it. The sooner we involve those who think about where and /how value gets created and the structures to make things actually work, the better."

Improve Communication Across the Board

You must communicate effectively with stakeholders so that everyone gets on the same page regarding the goals and vision for your API program. If you collaborate around design documents, you can help bridge the communication gap between technical and non-technical stakeholders by visualizing API designs and using an API specification like OpenAPI as a focal point.

Sophie Rutard told us that if she had a fresh start for an API program, she would:

"Invest significantly more in communications — communications to internal key stakeholders (CIO, Market Mgmt, Sales) regarding the business value and operational imperatives (API governance, design, etc.) to the program's success." She would do this investment "to win stakeholders over as promoters much earlier in the process."

She would also "invest more into communications to API consumers — be they internal or external — and try to build upon their feedback." Additionally, she would "improve things based on their critical feedback and promote the API program based on their success stories and positive feedback."

Treat APIs as Products

You might be surprised at how many companies have an API program but don’t treat APIs as products. You need to treat your APIs as products viewing them as critical business assets and not technical projects you release and then forget. You should regularly talk with your customers and partners about what they want from your APIs. Release new versions of your APIs over time with new features and improvements based on customer feedback.

Lorinda Brandon told us that BetterCloud’s API program IS getting a fresh start, and one of the things they’re doing is treating their API as a product.

“We hired a Product Manager, and we're talking to customers and partners about what they want from our API,” Lorinda said.

James Higginbotham shares some tips on treating APIs as products in this API Intersection podcast episode.

Take an API Design-First Approach

An API design-first approach means that you describe your API designs in an iterative way that both humans and computers can understand — before you write any code. With this approach, every team speaks the same language, and every tool they use leverages the same API design. Lorinda told us that two other things they’re doing differently with their API program are:

- “Researching the whole API landscape and choosing the API formats and infrastructure that work for our specific use cases.”
- “Designing our APIs first, before we write any implementation code, and getting feedback from consumers.”

One of the advantages of an API design-first approach is that teams can work in parallel. They don't have to wait for other teams to finish different stages of the API.

Provide Teams with the Right Tools

The success of your API program depends on the quality of your APIs. You can't build high-quality APIs without consistency. The only way to ensure consistency across all your APIs is to make sure developers follow the same design rules, best practices, and standards. And to do that, developers need the right tools.

Erik Wilde from Axway explains that: "Tooling matters. Next time I'd spend more time thinking about how to help our teams and scale that help. So, for example, when we have guidelines, I'd make sure that we automate checking these as soon as possible so that it's easier and less painful for teams to comply."

Teams can quickly edit and create API style guides with an automated API linting tool like Spectral. They can also use Spectral to check API descriptions automatically, making sure they follow specific rules. Erik has a video on his YouTube channel where Chris Wood, another Catalyst at Axway, explains how to use Spectral for various use cases.

Focus on Capability-Driven APIs

You shouldn't go all-in on one platform or centralize the API design and development process when creating APIs.

“Let's not bet on one platform,” Erik told us. “Let's make sure we are planning on an evolving and decentralized way of how our APIs are created. Centralization only works for so long, and it really isn't the way you set up a modern organization.”

When it comes to API reviews, you have two approaches to choose from: centralized and decentralized. In this post about getting started building APIs, Jason Harmon, CTO at Stoplight, recommends taking a decentralized approach, where the API team “focuses on education, capacity-building within maker teams, and curating and easily enforcing standards for the broader organization.” He says that the team plays more of a consulting and evangelism role and is less of a top-down organization. He also says that, “a decentralized API team focuses on building an API program that works and is user-friendly, so people will choose to use it because it makes the most sense.”

In this podcast, Jason chats with Tanya Vlahovic about building an API program — which includes API governance. Additionally, Erik talks with Marcelo Araujo about API governance at scale in this video to gain a better understanding of good governance tips. Marcelo is formerly an API Center for Enablement Lead at Bosch USA and currently a Sr. API architect at Edward Jones.

Focus on Capability-Driven APIs

James told us that many of LaunchAny's clients that already have an API program have expressed their regret for focusing on data-centered APIs ahead of capability-driven APIs. “With data-centered APIs, they only send data and therefore force the API consumer to re-implement their core capabilities outside of the API,” James explained.

"As we work with these organizations, they transition to the Align-Define-Design-Refine (ADDR) approach to their API design to ensure that their APIs represent the digital capabilities that they need to offer to their internal developers and partners. The result is a shift from data-centric APIs that strictly focus on moving data to capability-driven APIs that achieve the desired outcomes of their users and developers. This approach also changes the ownership of APIs to a value-based model."

James explains the ADDR process in his book, "Principles of Web API Design: Delivering Value with APIs and Microservices."

## Stoplight Helps You Create a Successful API Program

We provide a wide range of tools and resources to help you create a successful API program. For example, with Stoplight Studio, you can collaborate on API designs, taking an API design-first approach and implementing effective API governance. You can improve the quality of your API descriptions with our open-source tool, Spectral, to ensure consistency across your APIs. Additionally, with our open-source tool, Prism, you can create mock servers from OpenAPI documents, validating and iterating on your APIs before writing any code.

So, if you could start your API program over again, what would you do differently? Share your thoughts with us on Twitter or check out the API Intersection podcast, where we ask every guest at the end of each episode this very question..

Interested in trying out Stoplight?
You can check it out for free by creating a free account.