

# What does API Monitoring Mean for API Product Managers and Growth Teams



Today, countless engineering teams have leveraged API monitoring to track infrastructure health and report when services are down or unhealthy. There are a variety of API metrics that can be tracked that are aligned with engineering goals such as uptime, average latency, requests per minute,

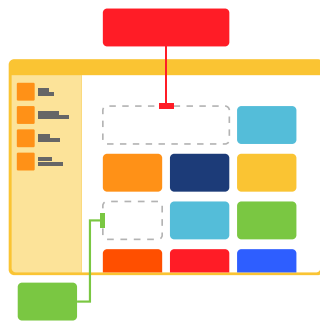
and errors per minute. However, these metrics are not aligned with the business goals of product owners and growth teams. This article goes through how to leverage API monitoring tools to further your business growth and product road map.

# Overview of business goals

The cost of building an API analytics system depends on data volume and feature complexity but can be broken down into three areas: cost can be broken down into three areas: Adoption, Engagement, Retention.

Companies focus on different areas depending on the state of the product. At the launch of an API, adoption will likely be your first priority since it's hard to optimize for engagement and retention when no one even knows about your API in the first place. Once you experimented with adoption and have a small set of users who tested your API, you may start looking for ways to boost engagement and then retention which can be done by optimizing onboarding, ensuring the right product features exist, and are valuable for the users. Once you see engagement and retention metrics at good levels (i.e. a small group of loyal users), you may then shift back to focusing on adoption since you can now double down on a product that is working.

1



## Adoption

Adoption focuses on top of the funnel growth of your API. The methods to drive adoption depends on the type of API you're creating. For an internal API, you may evangelize the API with other departments and teams at your company who can leverage it. For a public API, you may leverage a combination of paid ads and content along with direct outreach.

2



## Engagement

Once you have a group of users sign up and test your API, you want to ensure you can drive product engagement (i.e. usage) After all, APIs that generate a large amount of interest and signups, but cannot maintain high product usage are at risk of losing investment. Any project initiative is launched to create value for its users, whether those are internal teams, or external customers and partners.

3

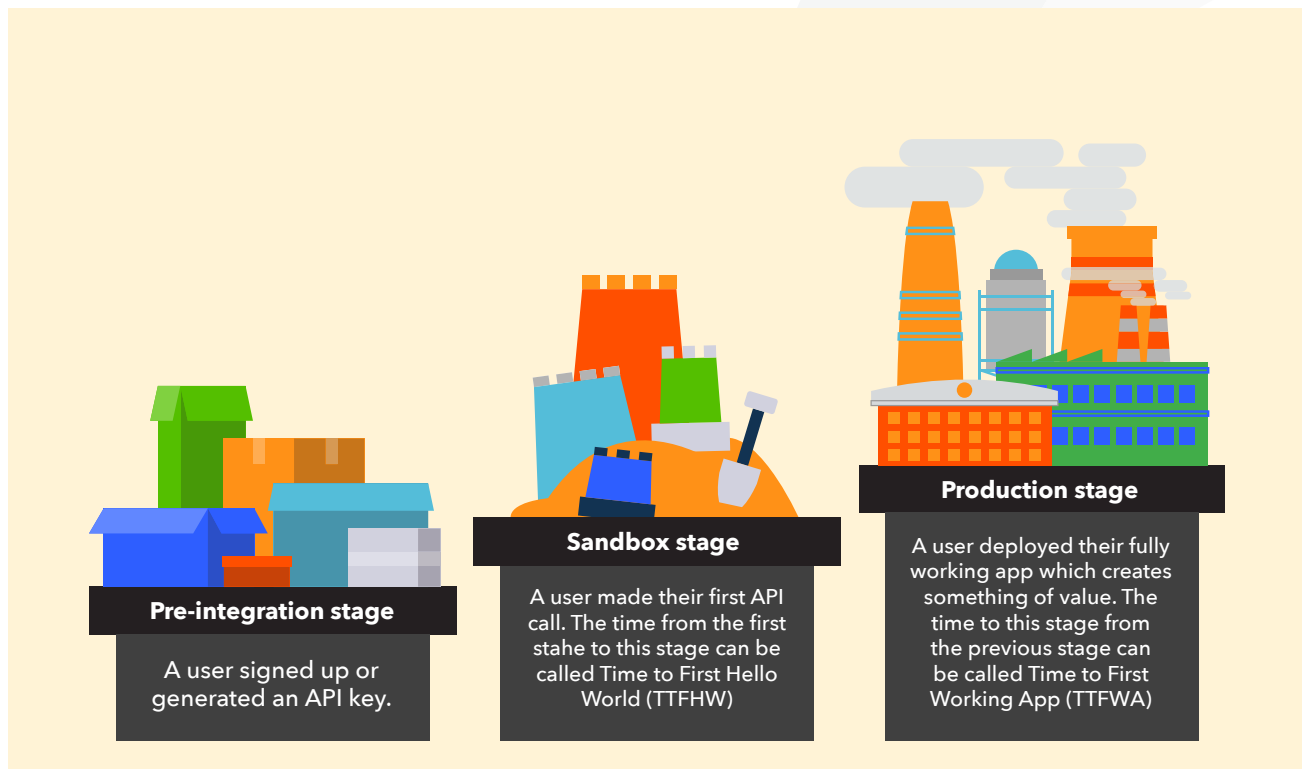
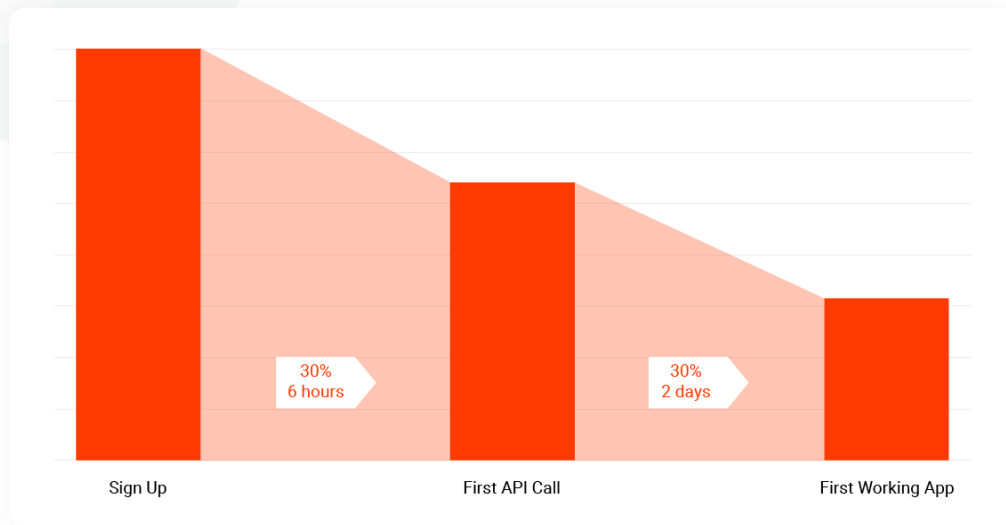


## Retention

Retention is similar to engagement except it tracks the decay function of that engagement over time for a cohort of users. Of the three product metrics, retention is one of the best proxies to product/market fit, since it excludes external factors such as high adoption pumped up by super large sales and marketing budgets. If you have a loyal and sticky product, then retention will be higher than a product that is a leaky boat.

# Monitoring API Adoption

Tracking developer adoption of API products is similar to the adoption of other products and best done with a funnel. The developer funnel tracks a user's journey from the initial sign up to shipping a working app using your API. Unlike other funnels such as for measuring mobile app acquisition, an API user could stay in a single funnel stage for days or longer. This means you should not only be tracking the conversion rate for each step, but also the time it takes to reach the next step.



Recommended steps to track in your funnel

By mapping out your adoption flow in stages, you can discover areas in the product requiring improvement.

For many API platforms, lack of docs and easy onboarding slows TTFHW. This can include:

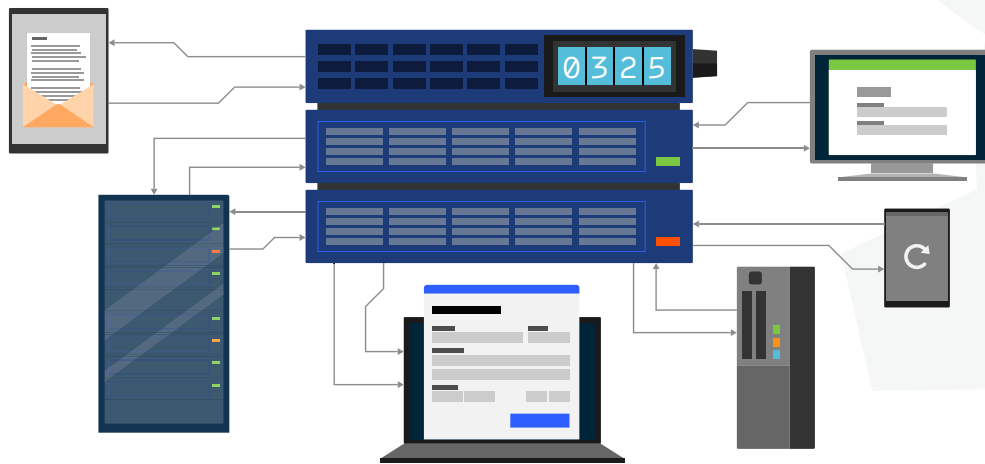
1. A lengthy onboarding process with many integration steps
2. No SDK that targets their programming language or framework requiring extra development
3. Poor or out of date documentation. Most developers love doing research into an API before integrating.

On the other hand, other internal stakeholders outside the developer's control slow down TTFWA such as:

1. Legal and compliance risk
2. Project priorities
3. Blockers on performance testing and security review much risk to the business.

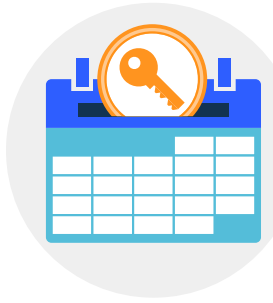
# Monitoring API Engagement

Once a user adopts your API, you want to drive high engagement and usage.



## Vanity Metrics

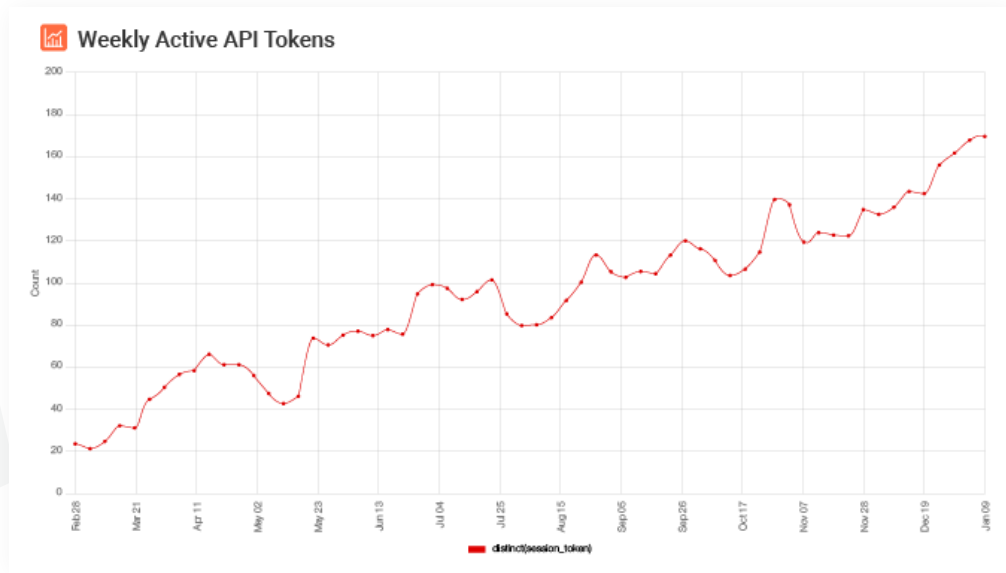
Choose your metrics wisely and don't fall in the trap of false (i.e. vanity) product metrics. While many marketing teams are accountable for page views and signups, these metrics only measure the top of the funnel and not the success of the product itself. You could even have a landing page without any product and drive high number of signups just from a large advertising budget. Similarly, there are many infrastructure metrics that are aligned with engineering goals rather than product goals. For example, a growing Requests per Minute metric could imply a successful product, yet the growth could be artificially high due to lots of health probes or bot traffic. Even average latency is not the best metric to measure product performance since a consistently slow API can be fine, unlike the experience loading a slow website.



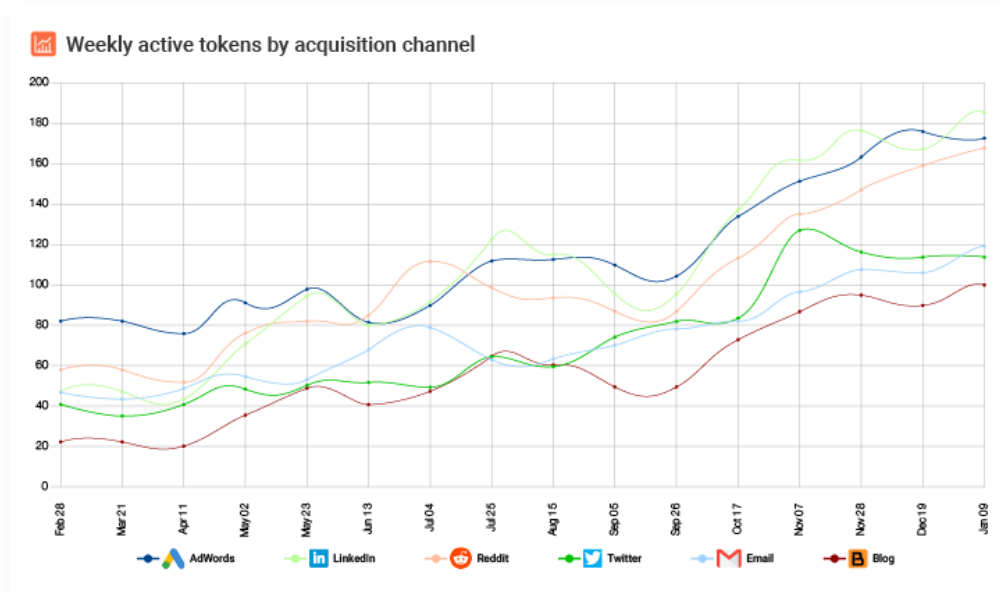
## Recommended Engagement Metrics

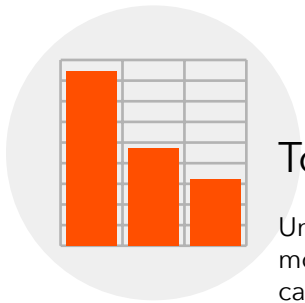
### Weekly active API token

Weekly active API tokens, the number of distinct tokens accessing the API in a given week, is one of the best north star metrics to track for API products. Because a single developer account can create multiple API tokens such as for a sandbox and production environment, a more accurate measurement would be Weekly Active Users or Weekly Active Companies. However, this requires analytics infrastructure that can link API tokens to a respective user or company account.



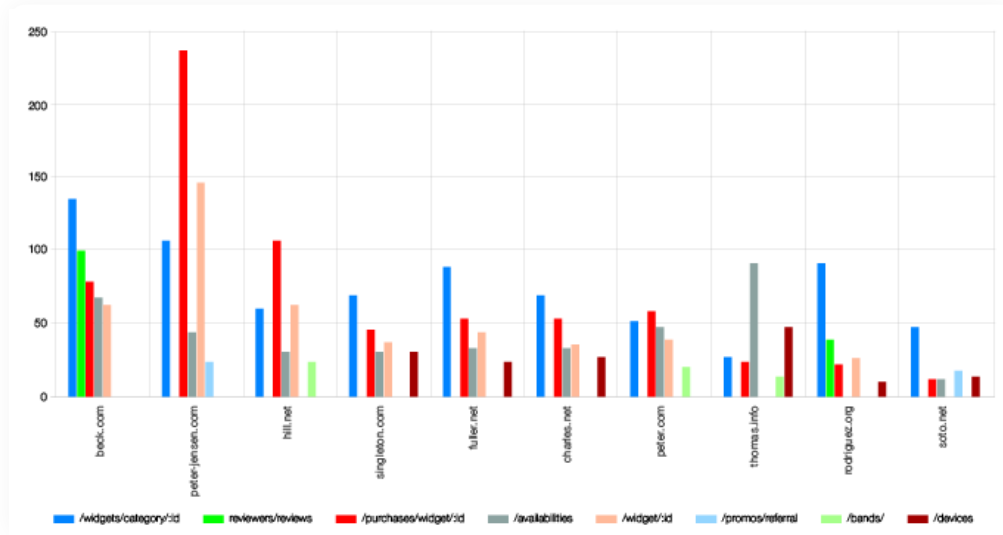
When a developer signs up to use your API, you should also be tracking referring domain and UTM parameters. This enables you to group weekly active tokens by UTM source or UTM campaign to better understand the marketing channels that are contributing to growth in usage and engagement.





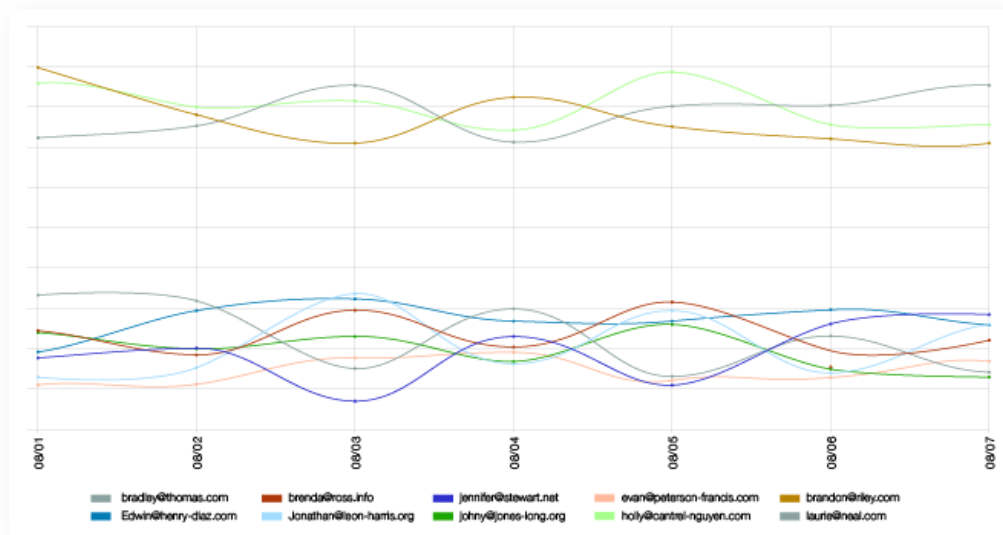
## Top customers by API traffic

Understanding your top consumers of your API helps you understand what your most valuable users want. By breaking it down by endpoints your features, you can look at second-order metrics such as which endpoints are popular with your most engaged users that are not leveraged by users not using your API much. It's possible there is a small group of highly valuable features that need greater exposure.



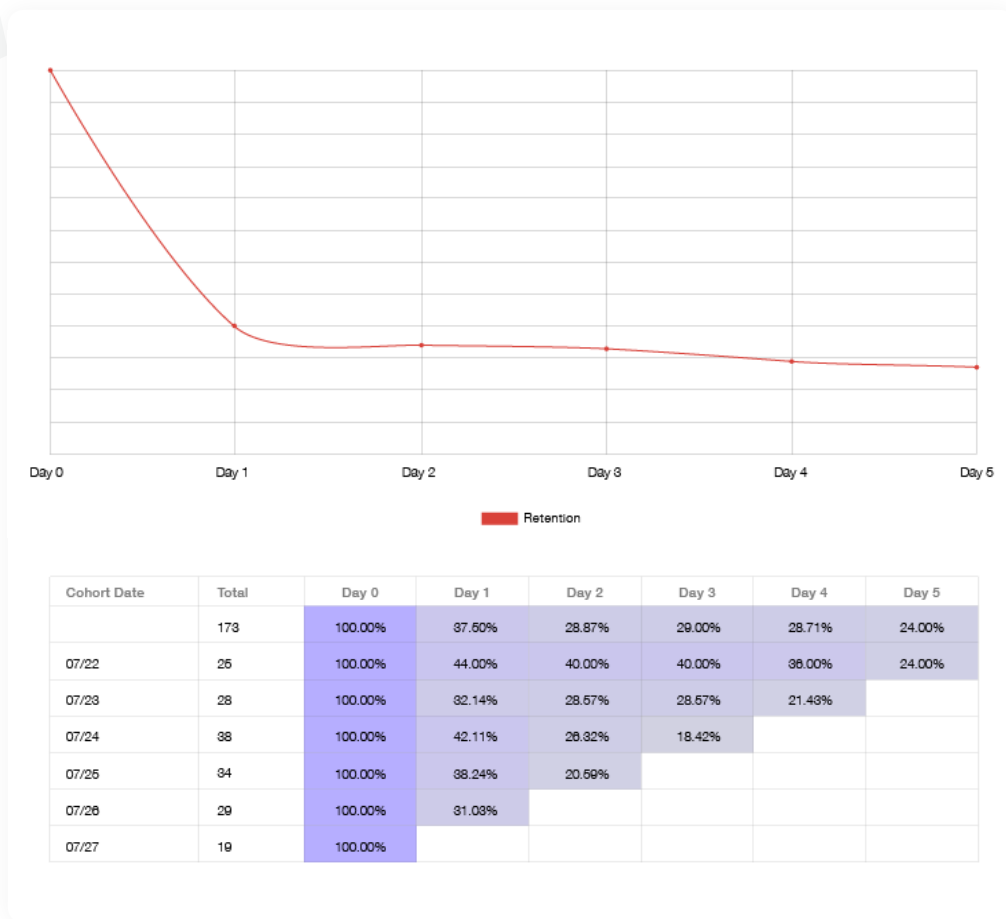
## 90th percentile latency

Average latency can be a false metric since it doesn't show variation in latency. Most users consuming your API can handle an API that's consistently high latency since it's predictable. However, an API with a high variance in latency is harder to work with. Latency variation causes false alarms to be sent, triggers race conditions in code, and makes it harder to capacity plan. One way to measure this is via the 9th percentile latency. You can also group this by the customer, endpoints, or other data to find the hotspots with the largest latency variance.

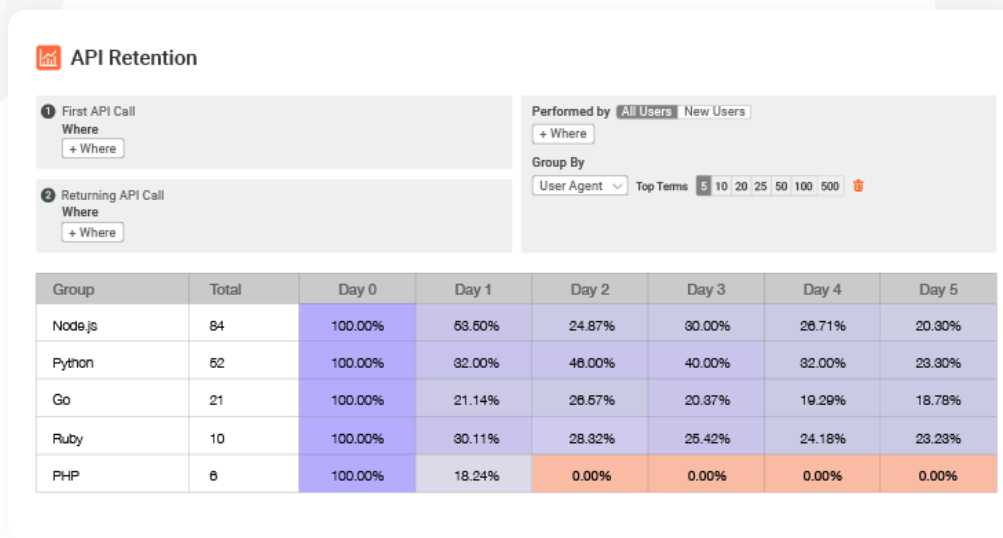


# Monitoring API Retention

Once you have a good understanding of adoption and engagement, it's important to look into API product retention to find areas of the product that require improvement. Product retention is a concept born out of revenue retention and requires segmenting your user base into cohorts such as via sign up date. Then, you track the percentage of each cohort returning to engage with your platform.



While all retention curves slope downward since no product can guarantee 100% retention after they sign up, good products that provide long-term customer value have retention curves that approach a non-zero equilibrium over time like the one above. However, bad products that do not have product/market fit will exhibit retention curves that slope down to zero. This means you have a leaky boat and should prioritize fixing your product retention before further investment in adoption. Otherwise, you are just wasting time and money to acquire users who don't stick around.



In the above metric, we are exploring API retention grouped by the user's SDK. We can see that PHP has a far lower retention percentage than the other SDKs. This could imply that PHP is buggy or has some performance issues that require fixing.

## Concluding Thoughts

Getting developers to adopt your API is hard. It requires having a good intuition of what developers want while not letting personal biases get in the way. The two best ways to build products developers actually love are through 1. looking at real usage data and 2. customer discovery and surveys. Having both qualitative and quantitative data is important to plan and invest in the correct product initiatives.



## Moesif for API Product Management

Build better API products with real-time insights into API usage and customer behavior. Leverage API analytics to track how customers adopt and use your platform from sign up to first API call. Guide customers and keep them informed of issues like subscription quota.