



Three Principles of API First Design

When developing cloud-native applications, “API First” is a core principle. Putting that principle into practice is harder.



Lars Trieloff · Follow

Published in Adobe Tech Blog · 4 min read · Jun 2, 2017



797



4

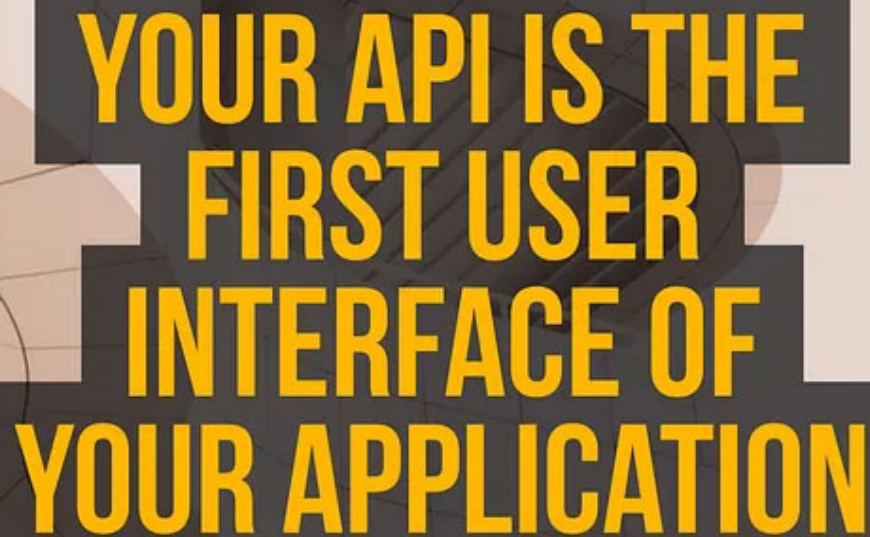


In my conversations with developers who are looking to embrace cloud-native development and openness, one question frequently comes up: what exactly does “API First” mean? All too often, “API First” can come to mean “Yeah, APIs are important, but they’re not essential”.

There are three principles of API First Design:

1. Your API is the first user interface of your application
2. Your API comes first, then the implementation
3. Your API is described (and maybe even self-

Let's take a look what each of these principles mean.




**YOUR API IS THE
FIRST USER
INTERFACE OF
YOUR APPLICATION**

API first thinking means that your API is the first interface of your applications. This means that the people developing against your API are your users, and your API needs to be designed with those users in mind.

Your API is how your product exposes its functionality, so if there is functionality in your product not covered by an API, it can't be covered by a graphical user interface, command line interface, or voice interface either, effectively making the functionality invisible.

Finally, as your API is the first and most important way to access and interact with your product, the API needs to be managed and designed deliberately. **Just as you spend time to design your graphical user interface, invest time to design your API: what it exposes, what it does, and how it grows.**



YOUR API COMES FIRST, THEN THE IMPLEMENTATION

Once you realize that your API is an interface that deserves attention of its own, you begin to realize that the API has – and should have – a life of its own.

Your implementation will change as your application evolves and you optimize, refactor and grow the functionality. Your API, however, should not change frequently, but instead grow slowly and deliberately.

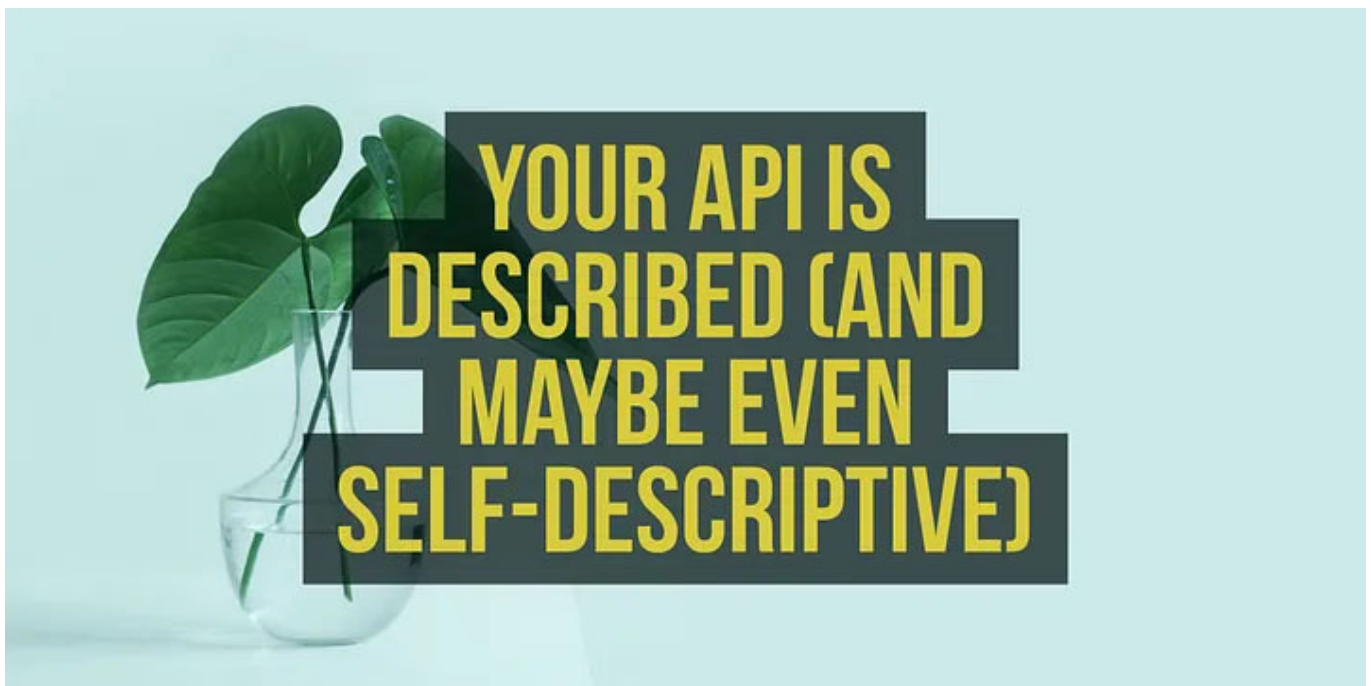
Your implementation will change frequently, your API should not.

Here's another way to think about this approach: If your API is the surface area of your product; the functionality is its volume. Doubling the functionality will only grow your API surface by 25%.

It's important to think of API evolution in terms of growth and increasing flexibility. Graceful API evolution is *additive* in terms of functionality, and

subtractive in terms of requirements. While change is inevitable, planning for a graceful API evolution is a good way to minimize changes that break things. For example: required input may become optional, but not the other way around.

Treating your API as independent from the implementation, even if that's harder, allows you to decouple development of API and implementation. Your API becomes a contract and specification to the implementation, instead of just being a thin veneer on top of your implementation.



The third principle of API First Design is about descriptiveness. In order to be used, your API needs to be easily understood by people that have not been involved in its creation. That means documentation.

Usable API documentation is an essential prerequisite to making it consumable by humans. As robots and AI aren't taking over programming anytime soon, this makes it essential.

When it comes to documentation for APIs, structured documentation beats unstructured documentation. Following a standard pattern for URLs, resource types, request methods, headers, request parameters, and response formats will make it easier to explore and understand functionality, and reduces surprises when your API grows.

Speaking of surprises: in your API design, try to minimize surprises and follow established standards and best practices wherever possible. When it's impossible to do so, document the deviation so that your API consumers won't waste hours chasing a bug introduced by an API that is working just a bit differently than expected.

There is one thing that makes even the best documented APIs stand out: **APIs that are self-descriptive** by using hypermedia constructs like links that allow the discovery of other API resources. This gives you a great way to close the loop from design, over implementation, to documentation.

Follow the [Adobe Tech Blog](#) for more developer stories and resources, and check out [Adobe Developers on Twitter](#) for the latest news and developer products.

API

Api First

Cloud Services

Software Development

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.