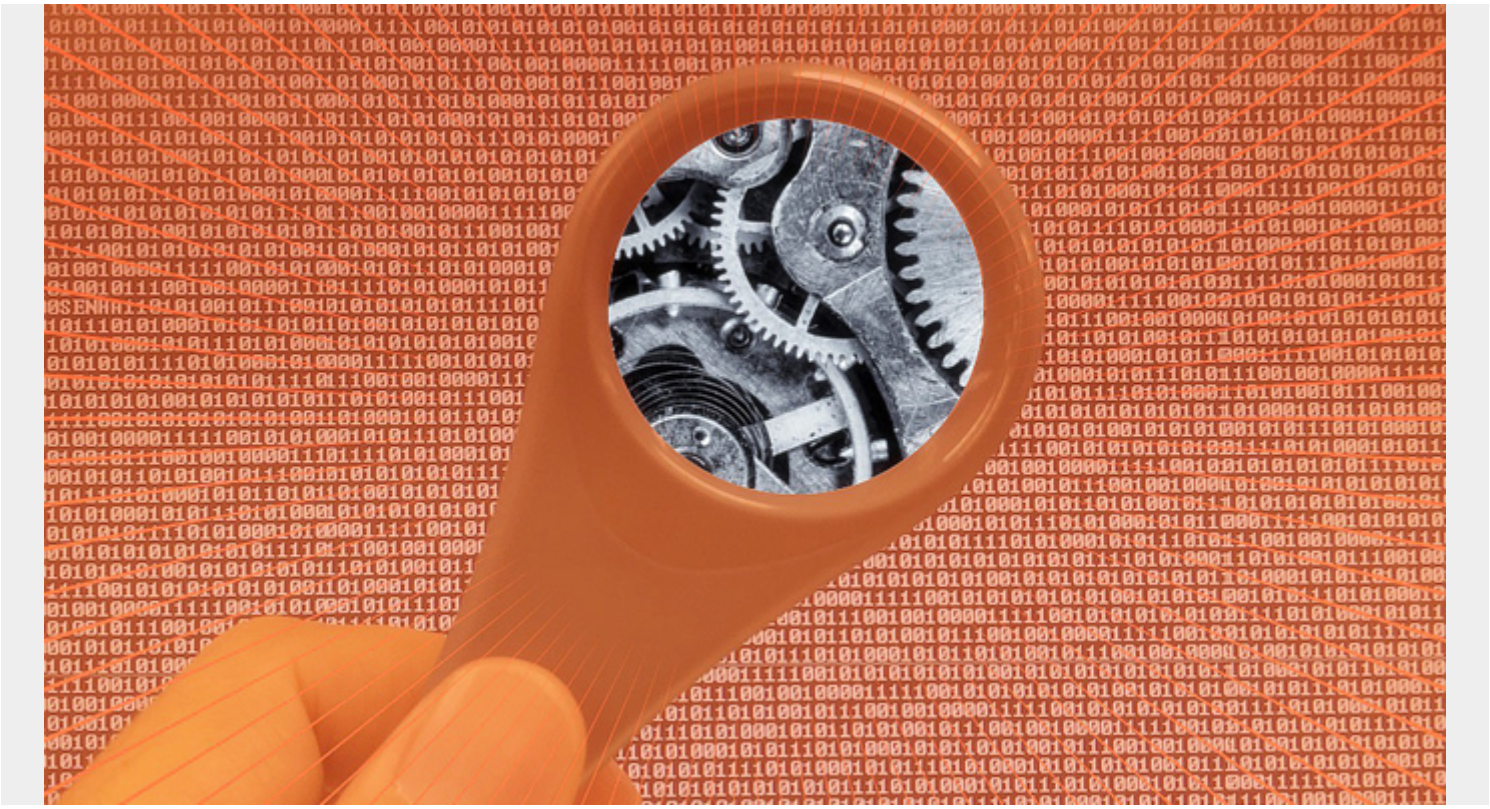
# API/DEVELOPER PORTALS: HOW TO CREATE GREAT API PORTALS



If you're a developer, you've no doubt encountered the terms API portal and developer portal. The goal of these portals is to educate developers so they can use the associated APIs—so what goes into the best API portals? Let's take a look.

## What is an API portal?

The API portal, also called the developer portal, is the intermediary between the API provider and the API consumer. (An application programming interface is how you set up interactions with various software apps.) The API portal will offer documentation about using the API.

The API portal is located on a company's website. It can usually be found under a developer's page, otherwise just search on the website. In the portal, you can get information about:

- The API
- How to use it
- API keys
- When versions are released or being deprecated

The API portal is there to encourage development engagement and innovation with the API. When good communication exists around the product, it is easier for people to adopt and to use. That is what the API Portal is for.

# What makes a good API portal?

The DevPortal awards are a great place to see good, different approaches to the API portal. The success of the API portal depends entirely around communication. When its purpose is to communicate what the API is and how people should and can be using it, the task becomes one of communication.

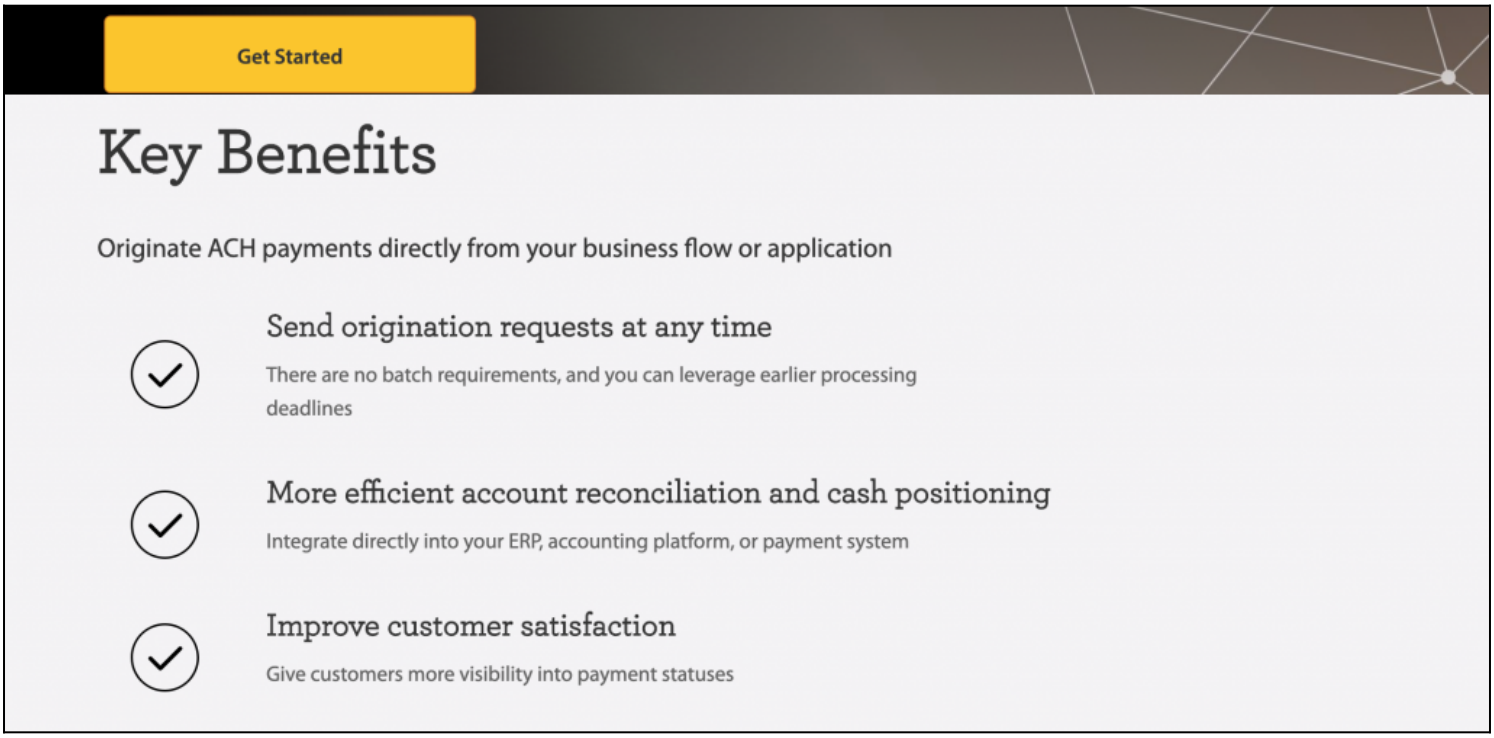
How an API communicates its use is specific to the API. Wells Fargo may have more hurdles to get through than Twitter because of its sensitive account data. But Twitter may have more restrictions about how many calls can be made to its API because of the sheer volume of data that passes through it. Each API is different, and those things that make it different need to be communicated.

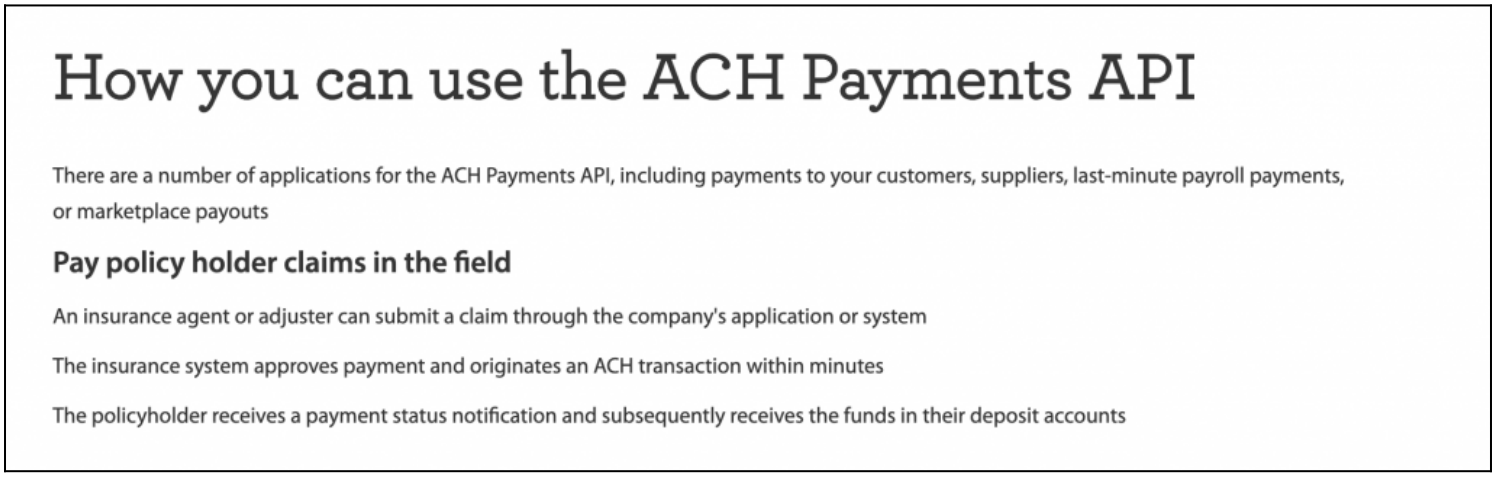
Communication is key to the success of an API Portal. A good API portal has two characteristics:

- A clear statement about the value the API offers the developer
- Steps for getting started

## API value statement

Wells Fargo has multiple APIs on offer. On each offerings, they write a one-liner that states what the API does, and how it helps the developer.






For this ACH payments API, these components form the entire page. Each API gets this one-page treatment.

## Steps for getting started

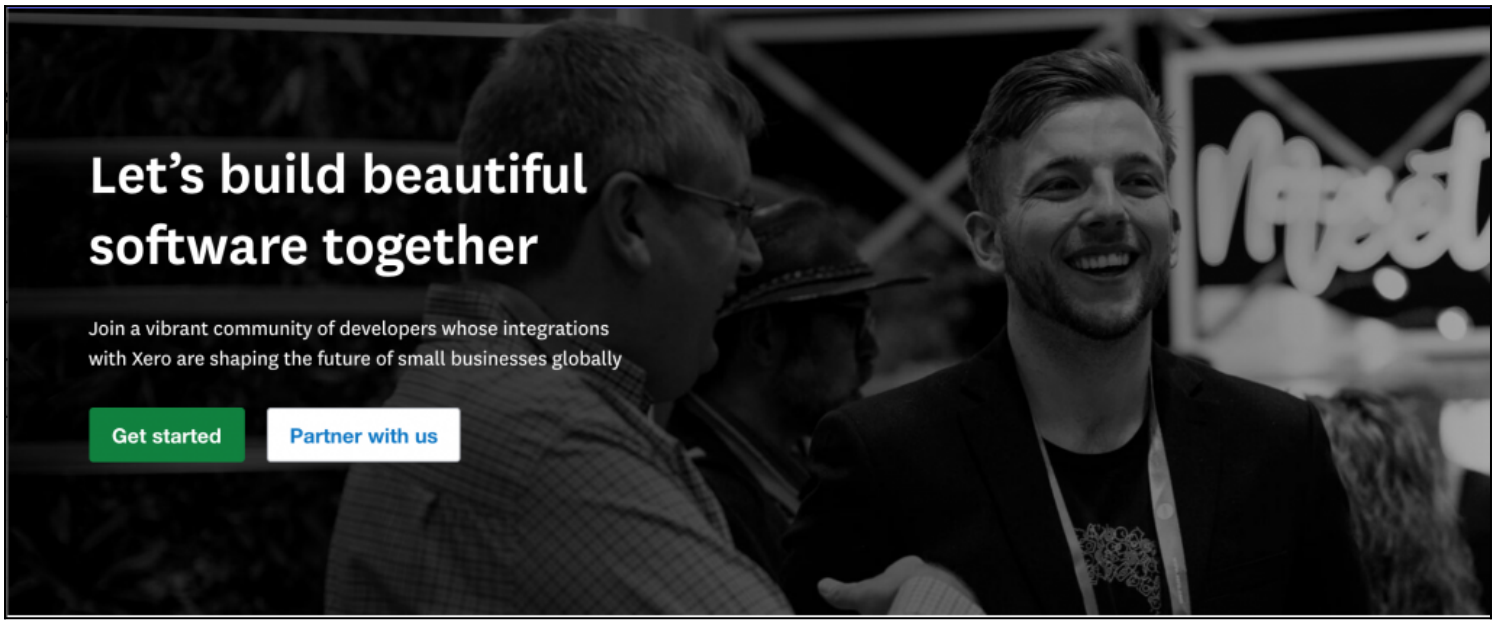
Like good sales, after the API is shown and a developer wishes to try it out, a Getting Started page should be:

- Easy to access
- Easy to follow

The getting started page should be somewhere obvious on the website:

- In a header or footer with other links
- In a popup menu chat agent
- In the body of text with the API service description.

It should be easy for a developer to say, "I want to get started." Then say, "Great, there's the link so I can." Xero's Get started button is built right into their banner, making it take center stage on their developer portal:



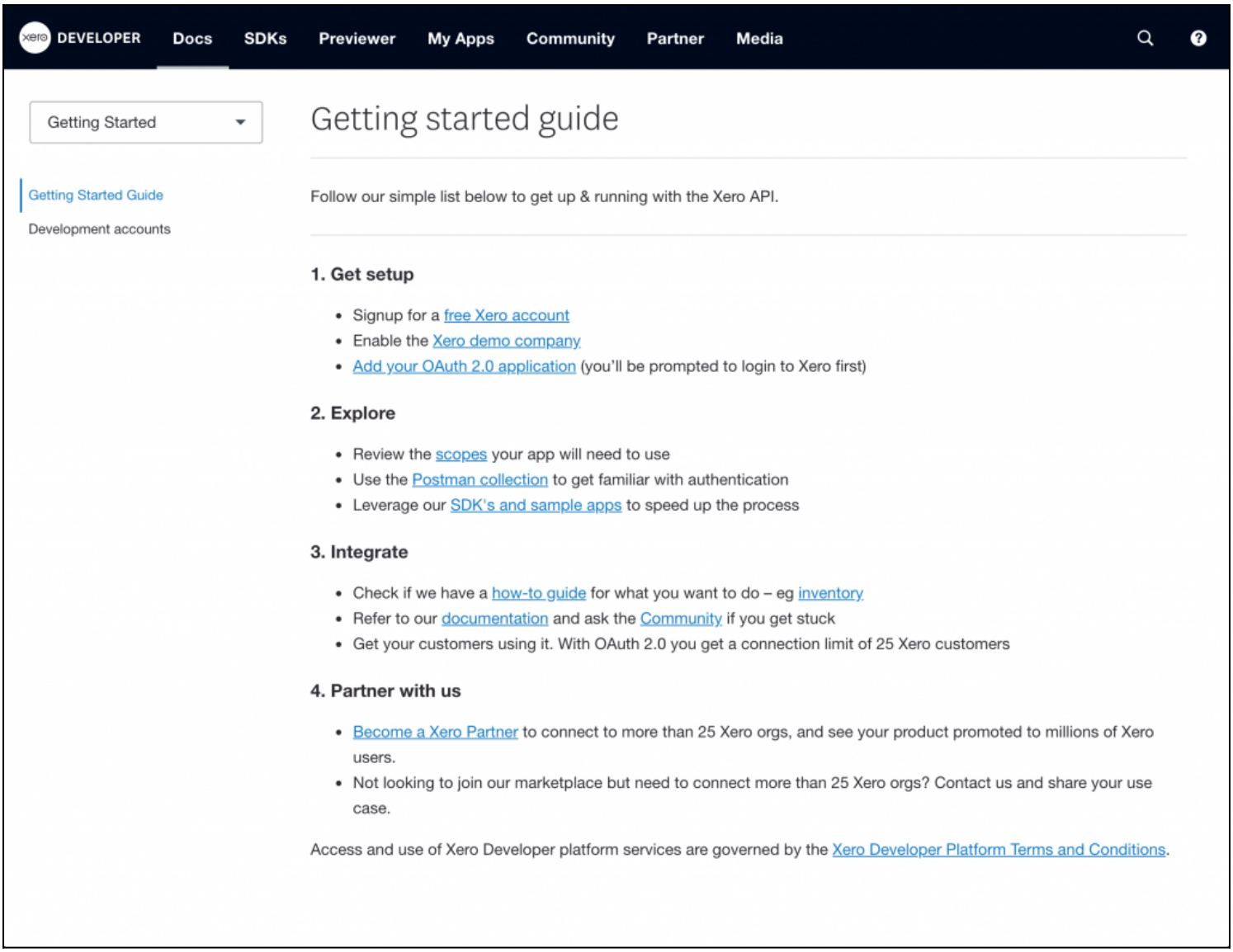

Developers are used to reading instructions. Every software library should have good documentation: it is a principle of a good developer to read, and follow, the directions.

The getting started documentation should be easy to follow:

- Numbered bullet points are a good way to organize information into steps. But caution: you don't want to have too many steps for devs to follow. That gets complex.
- The documentation can be very plain. It should be, right? Because that is the simplest to read.

While it might contain high-tech information, the portal needs to be read and understood by devs, so it is common to see the getting started guide look like a simple Wiki page.

Another way to increase user adoption? Simplicity. These getting started pages can—should—stay short. Xero's entire documentation fits in one screen, contained in 4 bullet points:




# Bonus items: good to great API portals

Most API portals share information about the API: what it does and how to use it. But a great portal includes extra items—here are two:

**Authorization instructions.** Each API usually has an auth provider it works with to see who is using it, and throttle, or charge, for the API use. OAuth2 has become a standard.

**Developer community.** Given the size of the API and the people using it, some companies may put together the resources for where their developers can meet and discuss their use of the API. This

can be done on GitHub, Slack channels, Discord, or self-built forums. In these communities, developers:

- Help each other get started
- Discuss problems they are having using the API
- Request new features

This community is there to help both the developers using the API use the API itself, but also allows the API providers to stay in touch with the people using their service so they know how to tweak their services to make the developers happy.

## Communication for API success

There is no API without an API Portal. And it is essential to have good communication inside your API portal. If you're going to the effort to put an API together to serve a community, then you can go through the effort to explain what it is, what it does, and how to use it.

## Additional resources

For more, check out these resources:

- BMC DevOps Blog
- Java Developer Roles and Responsibilities
- What Is Pair Programming?
- Resilience Engineering: An Introduction