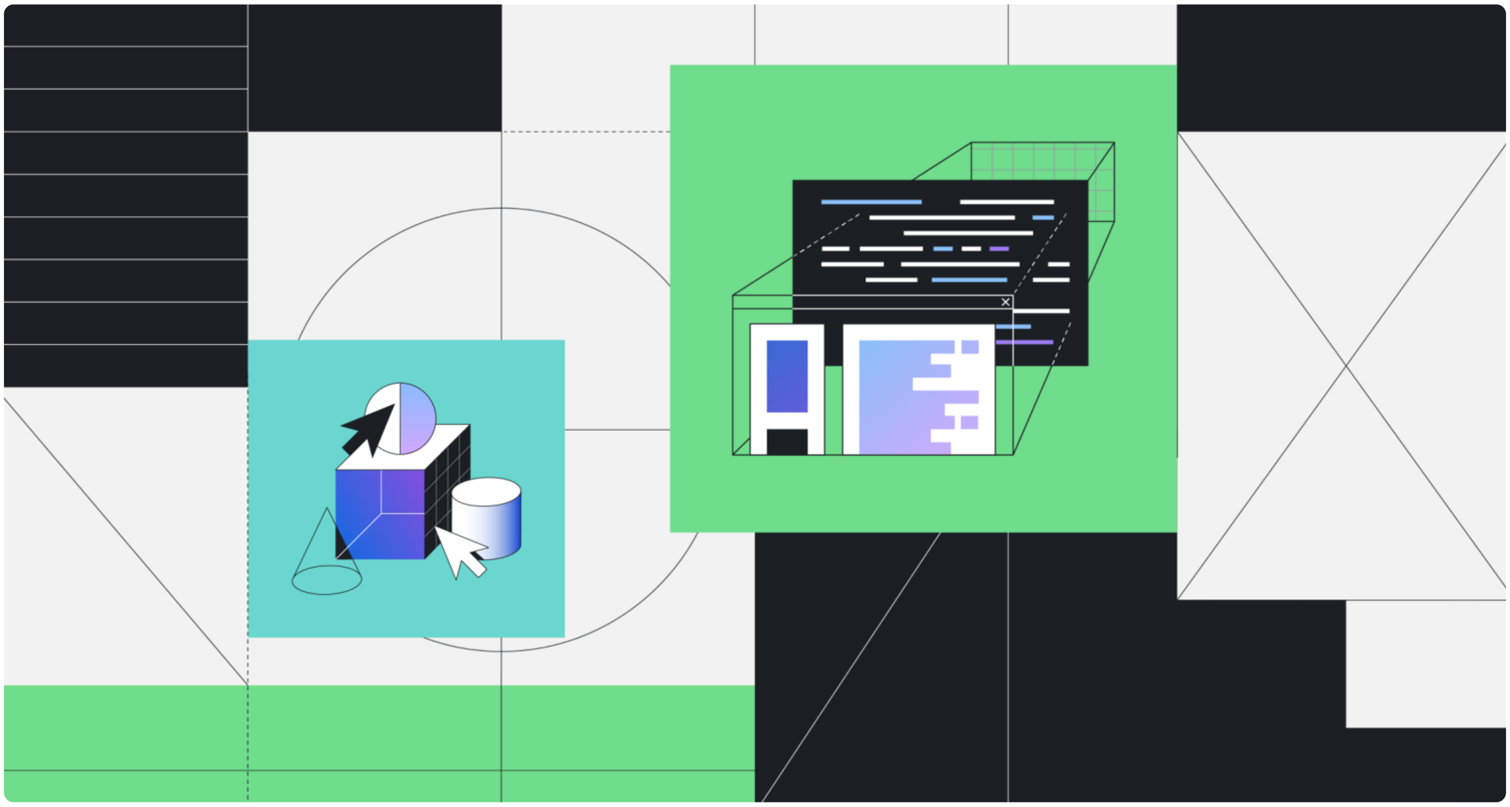
To infinity and beyond: enabling the future of GitHub's REST API with API versioning

We're introducing calendar-based versioning for our REST API, so we can keep evolving our API, whilst still giving integrators a smooth migration path and plenty of time to update their integrations.



Author

Tim Rogers

November 28, 2022

Millions of developers rely on the GitHub API every day—whether they've built their own bespoke integration or are using a third-party app from the GitHub Marketplace.

We know that it's absolutely crucial to provide a stable, consistent API experience. We can't—and don't—expect integrators to constantly update their integrations as we tweak our API.

At the same time, it's crucial that we're able to evolve the API over time. If the API had to stay the same forever, then we couldn't bring the latest and greatest product features to API users, fix bugs, or improve the developer experience.

We can make most changes (for example, introduce a new endpoint) without having a negative impact on existing integrations. We call these **non-breaking changes**, and we make them every single day.

But sometimes, we need to make **breaking changes**, like deleting a response field, making an optional parameter required, or deleting an endpoint entirely.

We launched version 3 ("V3") of our API more than a decade ago. It has served us well, but **we haven't had the right tools and processes in place to make occasional breaking changes** AND give existing users a **smooth migration path and plenty of time to upgrade** their integrations.

To enable us to continue evolving the API for the next decade (and beyond!), we're introducing calendar-based **versioning for the REST API**.

This post will show you how the new API versioning system will work and what will happen next.

How it works—a 60-second summary

Whenever we need to make breaking changes to the REST API, we'll release a new version. Head over to our documentation to learn about the kinds of changes that we consider to be breaking and non-breaking.

Versions will be named based on the date when they were released. For example, if we release a new version on December 25, 2025, we would call that version `2025-12-25`.

When we release a new version, the previous version(s) will still be available, so you won't be forced to upgrade right away.

Picking what version you want to use is easy. You just specify the version you want to use on a request-by-request basis using the `X-GitHub-Api-Version` header.

In our API documentation, you can pick which version of the docs you want to view using the version picker.

We'll only use versioning for breaking changes. Non-breaking changes will continue to be available across all API versions.

Note: calendar-based API versioning only applies to our REST API. It does not apply to our GraphQL API or webhooks.

How often will you release new versions, and how long will they last?

We'll release a new version when we want to make breaking changes to the API.

We recommend that new integrations should use the latest API version and existing integrators should keep their integrations up to date, but we won't frequently retire old versions or force users to upgrade.

When a new REST API version is released, we're committed to supporting the previous version for at least two years (24 months).

After two years, we reserve the right to retire a version, but in practice, we expect to support historic versions for significantly longer than this. When we do decide to end support for an old version, we'll announce that on our blog and via email.

I have an existing integration with the REST API. What does this mean for me?

You don't need to do anything right now.

We've taken the existing state of the GitHub API—what you're already using—and called it version `2022-11-28`.

We encourage integrators to update their integration to send the new `X-GitHub-Api-Version: 2022-11-28` header. Version `2022-11-28` is exactly the same as the API before we launched versioning, so you won't need to make any changes to your integration.

In the next few months, we'll release another version with breaking changes included. To move to that version, you will need to point the `X-GitHub-Api-Version` header to that new version and make sure that your integration works with the changes introduced in that version. Of course, we'll provide a full changelog and instructions on how to upgrade.

Next steps—and what you can do today

If you have an integration with the REST API, you should update it now to start sending the `X-GitHub-Api-Version` header.

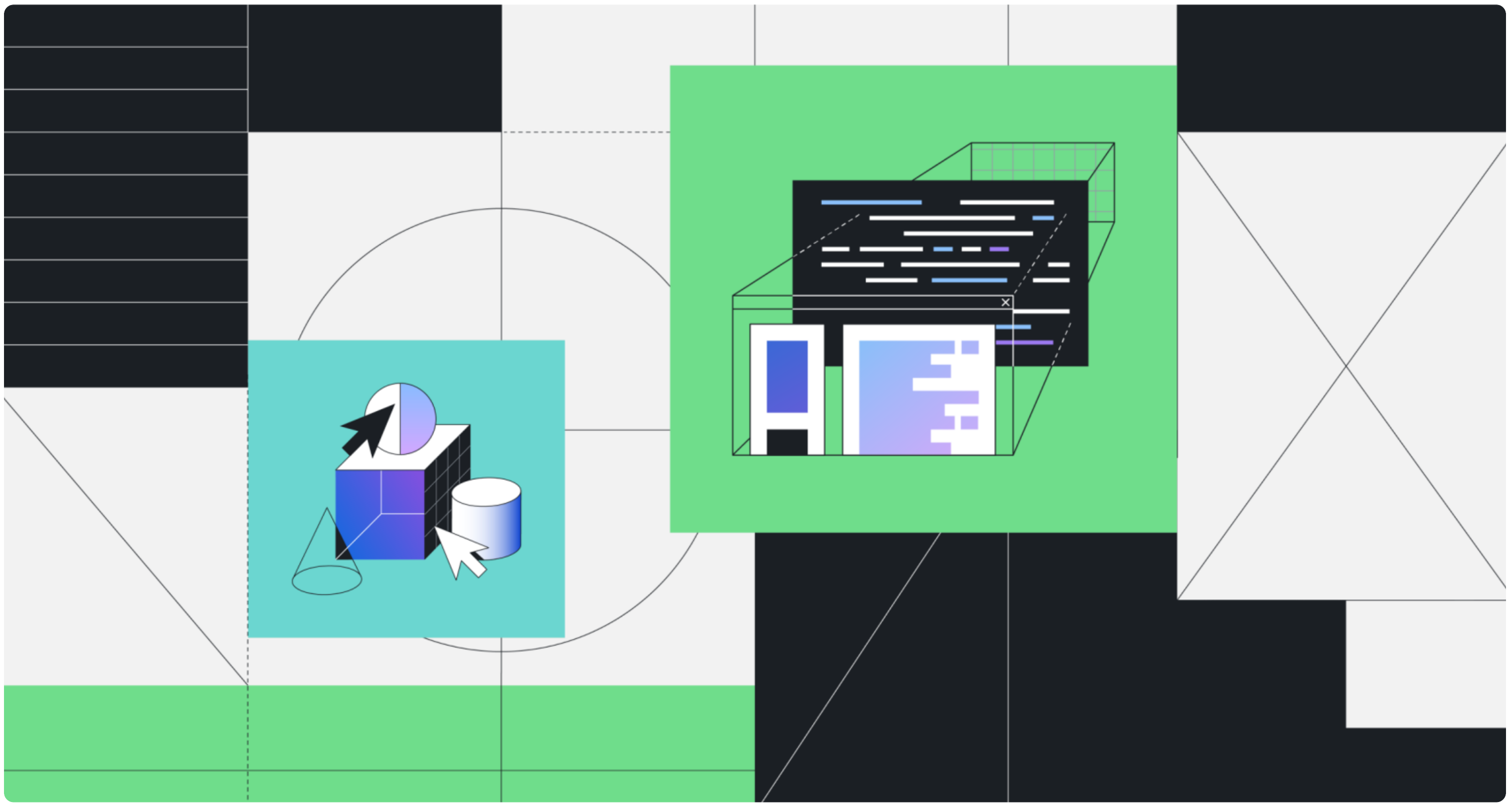
Soon, we'll release a dated version of the API with breaking changes. Then, and whenever we release a new API version, we'll:

- Post an update to the GitHub Changelog.
- Publish the documentation, information about the changes, and an upgrade guide in the GitHub REST API docs.
- Email active GitHub.com developers to let them know about the new release.
- Include a note providing details of the API changes in the release notes for GitHub Enterprise Server and GitHub AE.

We'll also launch tools to help organization and enterprise administrators track their integrations and the API versions in use, making it easy to manage the upgrade process across an organization.

Tags: extensibility, GitHub API, REST

Related posts



Exciting new GitHub features powering machine learning

Discover the exciting enhancements in GitHub that empower Machine Learning practitioners to do more.

Seth Juarez

The power of GitHub in the palm of your hand

GitHub Mobile helps keep work going while you're going. Untether yourself from your office.

Candy Ho

A better way to search, navigate, and understand code on GitHub

Reading code is a hugely important task for developers. That's why we built GitHub's new code search—to help developers search, navigate, and understand code written by them, their team, and the world.

Ryan J. Salva

Explore more from GitHub

Engineering

Posts straight from the GitHub engineering team.

Learn more >

The ReadME Project

Stories and voices from the developer community.

Learn more

GitHub Actions

Native CI/CD alongside code hosted in GitHub.

Learn more

Work at GitHub!

Check out our current job openings.

Learn more

Subscribe to The GitHub Insider

A newsletter for developers covering techniques, technical guides, and the latest product innovations coming from GitHub.

Your email address