Extension API Process

The goal of this process is to be transparent about how APIs are created, what phases a proposal goes through, and to know when proposals will be finalized. While still maintaining flexibility in the API making process.

About Proposed APIs

All proposed APIs are defined in the `vscode.proposed.d.ts` file.

We iterate, both internally and externally, on API proposals before making any APIs stable. With greater transparency and collaboration, we can produce better, flexible, powerful, and more ergonomic APIs. Therefore, ideally, proposals would be reviewed, tried, and tested by extension authors. To try out a proposed API, do the following:

- You must use Insiders because proposed APIs change frequently.
- You must have this line in the `package.json` file of your extension: `"enableProposedApi": true`.
- Copy the latest version of the `vscode.proposed.d.ts`-file into your project.

Note: You cannot publish an extension that uses a proposed API. We may likely make breaking changes in the next release and we never want to break existing extensions.

API Proposal Stages

Here is a rough flow chart of the process.

API Idea   Have an idea for a new API? Great! Ask yourself the following questions:

- How does this fit into the existing API?
- Is this a one-off-API that solves only your one problem, or can it be applied more broadly?
- Does this reuse a concept, like providers or commands, or does this add a new concept?
- Who else could use an API like this? (Multiple use-cases lead to better APIs)

API Proposal Suggested   Before proposing your new API, ask yourself the following:

- Does the proposal alter an existing API? If yes, it **must** be backward compatible with the existing API.
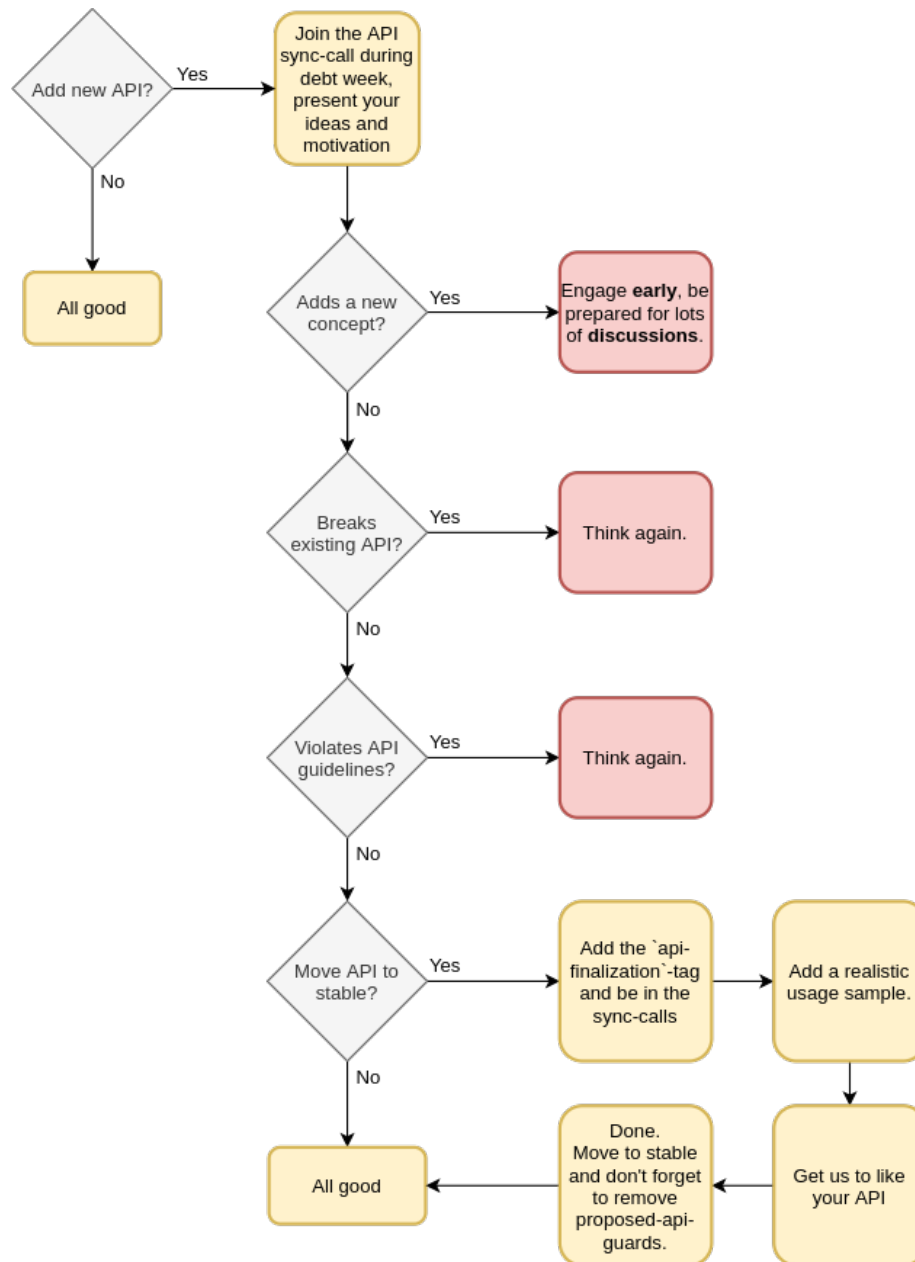- Does it violate our API guidelines?

Figure 1: Flow

An API proposal is suggested, either internally or externally via a GitHub issue.
- The issue will track the life-cycle of the API proposal - The issue will be labeled
with `api` and `api-proposal`

Once you submit your proposal, it will begin gathering interest and
pre-implementation feedback.

API Proposal Implemented / Feedback Requested   Next, if the proposal
garners enough interest, it will likely be implemented. Once implemented, the
proposal is ready for review and feedback (e.g. does it meet requirements, API
guidelines, ergonomics, etc). We will also monitor and gauge continued interest
in the proposal. Ideally, extension authors/contributors will try out the new
API and provide real-world feedback.

- `vscode.proposed.d.ts` file will be updated with the new API
  - Each proposed API is contained within its own `//#region` and contains a short description of the API and a link to the proposal GitHub issue
- The proposal issue will be labeled with `api-feedback`

As an API proposal is reviewed and gathers feedback, it is very likely to evolve
in response. As this continues, look to the `vscode.proposed.d.ts` file for the
latest version.

API Proposal Finalization   Here are some finalization questions that will be
asked before beginning: - Is there a non-trivial usage-sample for this new API? -
Are there multiple (ideally not similar) use-cases that this proposal solves? - Is
the proposal too narrow (not solving enough) or too broad (trying to solve too
much)? - Who would use this new API?

Once a proposal sufficiently answers those questions and is in a good place, it
begins the finalization process for becoming stable. While this signals the intent
to ship the proposed APIs, there isn't any guarantee that there won't be futher
changes or that the proposal will become stable (e.g. in rare circumstances,
we might even deprecate the proposed API). At the same time, ideally, the
finalization of an API will be attached to an upcoming milestone, again as an
intent not guarantee.

And finally, assuming a succesful finalization, the API will move from
`vscode.proposed.d.ts` into `vscode.d.ts` and the API proposal issue will be
closed.

Weekly API call:

- During debt week, *everyone* that plans to work on API joins the call and
  presents their case. This call is about forming a shared understanding
  of the problem and helps us to understand whether the request is one

that can safely be mapped onto prior art or will require custom constructs (which pretty much means more work).

- We introduce a 'api-finalization' tag on GH to clearly identify the items that are about moving API from proposed to stable.
- Everyone who owns an open 'api-finalization' item for the current iteration needs to show up for the API call.
- Everyone who owns an open 'api-finalization' item scheduled for the iteration will present in week 1 of the iteration a concrete proposal and sample code showing how the API is being used. The sample code needs to be realistic. One-liners with artificial parameters etc. usually don't cut it.
- In each API call, we'll work through the remaining open 'api-finalization' items before we look at 'api-proposal' items.
- If an API topic needs more time than what we have in the API call, we'll schedule separate calls for these topics. These calls are open to everyone, just like the API call.