Contributing a new API to Node-API

Node-API is the next-generation ABI-stable API for native modules. While improving the API surface is encouraged and welcomed, the following are a set of principles and guidelines to keep in mind while adding a new Node-API.

- A new API **must** adhere to Node-API API shape and spirit.
  - **Must** be a C API.
  - **Must** not throw exceptions.
  - **Must** return `napi_status`.
  - **Should** consume `napi_env`.
  - **Must** operate only on primitive data types, pointers to primitive data types or opaque handles.
  - **Must** be a necessary API and not a nice to have. Convenience APIs belong in node-addon-api.
  - **Must** not change the signature of an existing Node-API API or break ABI compatibility with other versions of Node.js.
- New API **should** be agnostic towards the underlying JavaScript VM.
- New API PRs **must** have a corresponding documentation update.
- New API PRs **must** be tagged as **n-api**.
- There **must** be at least one test case showing how to use the API.
- There **should** be at least one test case per interesting use of the API.
- There **should** be a sample provided that operates in a realistic way (operating how a real addon would be written).
- A new API **should** be discussed at the Node-API team meeting.
- A new API addition **must** be signed off by at least two members of the Node-API team.
- A new API addition **should** be simultaneously implemented in at least one other VM implementation of Node.js.
- A new API **must** be considered experimental for at least one minor version release of Node.js before it can be considered for promotion out of experimental.
  - Experimental APIs **must** be documented as such.
  - Experimental APIs **must** require an explicit compile-time flag (`#define`) to be set to opt-in.
  - Experimental APIs **must** be considered for backport.
  - Experimental status exit criteria **must** involve at least the following:
    * A new PR **must** be opened in `nodejs/node` to remove experimental status. This PR **must** be tagged as **n-api** and **semver-minor**.
    * Exiting an API from experimental **must** be signed off by the team.
    * If a backport is merited, an API **must** have a down-level implementation.
    * The API **should** be used by a published real-world module. Use of the API by a real-world published module will contribute favorably to the decision to take an API out of experimental

status.

* The API **must** be implemented in a Node.js implementation with an alternate VM.