Maintaining types for Node.js

While JavaScript is a weakly-typed language, there are some complementary tools like TypeScript and Flow, which allow developers to annotate the source code of their JavaScript projects. While many people don't annotate their code, or make use of annotations at all, there are enough who do that the project has agreed it's important to work towards having suitable types for end-users.

High level approach

There are a number of ways that types could be maintained for Node.js ranging from shipping them with the Node.js runtime to having them be externally maintained.

The different options were discussed as part of the next-10 effort and it was agreed that maintaining them externally is the best approach. Some of the advantages to this approach include:

- Node.js maintainers do not need to be familiar with any given type system/technology.
- Types can be updated without requiring Node.js releases.

The agreement was that the ideal flow would be as follows:

- APIs are added/documented in the existing Node.js markdown files.
- Automation in the Node.js project creates a machine readable JSON representation of the API from the documentation.
- Automation within external type projects consumes the JSON and automatically generates a PR to add the API.

Generation/Consumption of machine readable JSON files

When you run `make doc` the canonical markdown files used to document the Node.js APIs in the doc/api directory are converted to both an `.html` file and a `.json` file.

As part of the regular build/release process both the `html` and `json` files are published to nodejs.org.

The generator that does the conversion is in the tools/doc directory.

Markdown structure

The constraints required on the markdown files in the doc/api directory in order to be able to generate the JSON files are defined in the documentation-style-guide.

Planned changes (as of Jan 1 2022)

While JSON files are already being generated and published, they are not structured well enough for them to be easily consumed by the type projects. Generally external teams need some custom scripts along with manual fixup afterwards.

There is an ongoing effort to add additional markdown constraints and then update the flow in order to be able to generate a better JSON output.