

Node.js Foundation Tooling Group Meeting 2019-03-19

Links

- * **Recording**:
- * **GitHub Issue** : \$GITHUB_ISSUE\$
- * **Minutes Google Doc** : \$MINUTES_DOC\$

Present

- * Tooling team: @nodejs/tooling:
 - * Refael Ackermann (@refack)
 - * Ian Sutherland (@iansu)
 - * Michel Lopez (@MichelLopez)
 - * Ruy Adorno (@ruyadorno)
 - * Christopher Hiller (@boneskull)
 - * Corey Farrell (@?)
 - * Benjamin Coe (@bcoe)

Agenda

Extracted from **tooling-agenda** labelled issues and pull requests from the **nodejs org** prior to the meeting.

nodejs/tooling

- * rimraf progress [#13](https://github.com/nodejs/tooling/issues/13)
- * argument parsing [#19](https://github.com/nodejs/tooling/issues/19)
- * @bcoe has stuff

##z Notes

- Rimraf progress
 - Move some of rimraf into fs library
 - Exclude glob
 - Exclude CLI
 - Proposal to implement Python's shutil module in core (@refack)
 - <https://docs.python.org/2/library/shutil.html>
 - Maybe start with rimraf
 - Incrementally add more functionality?
 - Identify which functionality we want to implement in Node

- There's a userland package shutil on npm
 - <https://www.npmjs.com/package/shutil>
 - What do we do about that?
 - Claim the name to avoid confusion (@boneskull)
 - Create an issue in tooling to track for meetings, etc.
 - Create a PR in node repo
 - Microsoft working on fs improvements on Windows (in libuv) (@MichelLopez)
 - Shouldn't have an impact on JS implementation of fs methods
 - Will be on by default but can be disabled by a flag of some kind
- Argument parsing (@boneskull)
 - Seems to be consensus that having something would be good
 - Keep it simple, anything better than string splitting
 - Trying to identify the minimal subset of argument parsing from other libraries and languages: (@boneskull)
 - <https://github.com/nodejs/tooling/issues/19#issuecomment-467687792>
 - Define what params you accept, parse input into a data structure based on that
 - Even a simple alternative to `argv.slice(2)` would be an improvement
 - Just parsing to an object is complicated and opinionated
 - How do we decide on a good default?
 - Minimalist might be a good example for defaults
 - Consider bringing in an existing library like yargs (@refack)
 - Mixed opinions on willingness to accept such a large change to core
 - Compare yargs parser, commander, boost (C++), argparse (python) and document similarities and differences (@boneskull)
 - Node is more focused on web services (@refack)
 - This group is trying to convince people that CLI is also an important usecase
 - Most other languages provide something for arg parsing
 - Is there a specification for this we can refer to?
 - Breaking changes to argument parsing in the core would mean different versions of Node might not work with a CLI package (@coreyfarrell)
 - Yargs has had a number of breaking changes
 - We would need to define a good subset and keep it as consistent as possible
 - Need to be compatible with Windows
- Sourcemaps (@bcoe)
 - Stack traces are not useful if you're using TypeScript or others
 - Google's SDKs are mostly written in TypeScript (@bcoe)
 - There is some support for sourcemaps in V8 which could be used
 - Stack traces already have line numbers by the time they get to Node (added by V8)
 - TypeScript support is growing
 - npm survey results show over 50% usage but that survey has certain

biases

- Try to get someone from npm on the tooling team
- Maybe a minimal version of npm should be in core (@refack)
 - Ability to link directly to a package URL would be great (see Deno)

Action items

- Take over shutil module name on npm (@boneskull)
- Create an issue in the tooling repo to track shutil work (@iansu)
- Implement rmtree in shutil modules in node lib (@iansu)
- Compare argument parsing libs (@boneskull)
- Kickstart sourcemap discussion (@bcoe)

Q&A, Other

Upcoming Meetings

* **Node.js Foundation Calendar***: <https://nodejs.org/calendar>

Click `+GoogleCalendar` at the bottom right to add to your own Google calendar.