

Node.js Modules Team Meeting 2020-07-15

Links

* **Recording***: http://www.youtube.com/watch?v=LOHDJGs_FNs

* **GitHub Issue***: <https://github.com/nodejs/modules/issues/532>

* **Minutes Google Doc***:

<https://docs.google.com/document/d/1iEMdSLFXlpEwtAzogOOcGQZgS87fZDerdRVnCSldjUM/edit>

Present

* Myles Borins (@MylesBorins)

* Guy Bedford (@guybedford)

* Wesley Wigham (@weswigham)

* Darcy Clarke (@darcyclarke)

* Bruce MacNaughton (@bmacnaughton)

* Julian Londono (@julian-londono)

* Jordan Harband (@ljharb)

Agenda

Announcements

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

nodejs/node

* esm: Modify getFormat and getSource loader hooks

[#34144](<https://github.com/nodejs/node/pull/34144>)

- Crux of the issue, example: an https loader having to make a request in `getFormat` to get the resource format, then again in `getSource` to get the resource
 - Having state throughout the 4 loader hooks may be able to solve this
 - Pushing a tricky GC issue to loader hooks that may not be able to rely on getSource always being called after getFormat (especially once chaining exists).
- PR should delve more into edge cases and document exploration further, covering possible alternatives.
- Make sure PR works with Gus's compositional loader PR
[#33812](<https://github.com/nodejs/node/pull/33812>)

* Special treatment for package.json resolution and exports?

[#33460](<https://github.com/nodejs/node/issues/33460>)

- Refs:

- add API for "package dir" [#516](<https://github.com/nodejs/modules/issues/516>)

- up for debate on the placement (ex. ``module``, ``require``, ``import.meta``)

- proposed interface/method name: ``resolvePackageRoot``

- ex. usage, ``resolvePackageRoot(import.meta.url)`` & ``resolvePackageRoot(process.cwd())``,
``resolvePackageRoot(__dirname)`` (?)

* module: CJS exports detection [#33416](<https://github.com/nodejs/node/pull/33416>)

PR has been updated to use a flag. Should we let the PR work stall at this point? - no active interest in pursuing any escalation for the block.

- Concerns raised based on heuristic not being fully accurate.

- Discussed alternatives briefly.

- Suggestion to use a loader to provide the feature to users for further feedback.

- Snapshotting of CJS modules does not support loader interception - agreement that a PR to disable snapshotting and allow CJS mutation to allow a loader would be preferable to not being able to support CJS named exports.

- Discussion to open meta issue related to group handling of these types of consensus issues.