

Node.js Foundation Modules Team Meeting 2018-07-18

* **Recording**: https://www.youtube.com/watch?v=_k63DkTJpUI

* **GitHub Issue**: <https://github.com/nodejs/modules/issues/154>

<https://docs.google.com/document/d/1zruFPTN9dIpvSEkStxo4fkPenXTdhsLEVkcmoKIVmOA/edit>

Present

- Gus Caplan (@devsnek)
- Jan Krems (@jkrems)
- Ben Newman (@benjamn) (took notes below)
- Michael Dawson (@mhdawson)
- Matt DuLeone (@mduleone)
- Bradley Farias (@bmeck)
- Guy Bedford
- Jordan Harband (@ljharb)
- Geoffrey Booth
- Jeremiah Senkpiel
- Kevin Smith
- Geoffrey Booth (@GeoffreyBooth)
- Saleh Abdel Motaal (@SMotaal)

Agenda

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

Update on Progress (15 minute Timebox)

* Pull request opened for `import.meta.require` on core
[#130](<https://github.com/nodejs/modules/issues/130>)

- 3 minute timebox

- Jordan: philosophical objection to implementing something we haven't come to consensus on yet

* Thinking about deadlines [#123](<https://github.com/nodejs/modules/issues/123>)

- 3 minute timebox

- Jordan: this working group wasn't convened for expediency, so we should make sure we're shipping the right thing above all else
- Bradley: there's been a thought going around about starting a very minimal implementation that we can all agree on, so that we can decide about future features one at a time (more so than now)

* Initiative: Terminology / Historical Decisions documents

[#119](<https://github.com/nodejs/modules/issues/119>)

- 3 minute timebox

- Jan: Myles said probably no progress this week, but next week he may have more time
- Guy: let's get an update on that from Myles next week then

* Developer Survey [#85](<https://github.com/nodejs/modules/issues/85>)

- 3 minute timebox

- Guy: goal should be to create agenda items to discuss in these meetings, and also iterate on between meetings
- Michael Dawson: a GitHub issue would make it easier for folks to add ideas asynchronously
- Bradley: survey shouldn't just be about user expectations, since we can violate those in service of meeting user needs

* Features List Adjustments [#156](#)

- 3 minute timebox

- Guy: removing duplicates and flagging features that weren't on the README page
 - Unfortunately don't have a quorum to merge PRs

Discussion

* Package-Name-Maps a proposal for bare imports in browsers

[#51](<https://github.com/nodejs/modules/issues/51>)

- 5 Minute Timebox

- Jordan: summarizing:
 - Static file that declaratively tells the browser how to resolve bare specifiers to final URLs
 - Has nested structure (scopes) to support deep imports from packages
 - Should be easy to create to adapt Node code to browsers
 - Question: Should Node provide support for optional package name maps, too?
- Bradley:
 - Folks are making assumptions about the limits of package name maps without fully investigating the feature set
 - For example, you *can* resolve directories and missing file extensions
- Michael Dawson:
 - How does this fit into overall priorities? Can we think about this later on?
- Geoffrey Booth:
 - I'm trying to avoid code that behaves differently in Node and browsers, so package name maps are a compatibility concern for me
 - As packages add maps, they could cause breaking changes in resolution, e.g. for code doing deep imports from those packages
 - Lots of value in aligning with the browser on this, though
- Bradley: I don't think this is the same level of compatibility concern

- Always going to be some mismatch between how Node and browsers work, so we should be considering the subset of functionality that works the same in both
 - Bigger compatibility problems with caching, `import.meta.url`, importing JSON, HTML modules
 - Node is not a subset of the web
 - Using tools to generate package name maps solves some of the problems here
 - Guy:
 - Package name maps can only map bare specifiers, so `./` specifiers aren't addressed by the proposal
 - Bradley:
 - Shouldn't make Node dependent on what browsers do here
 - Other approaches to solve the same problems, such as loader hooks
 - Michael Dawson:
 - Node probably would not want to allow mapping bare specifiers to external URLs
 - Undermines Node's security model, though it works in browsers
 - Guy: Relative import specifiers are one of the big unresolved problems
 - Bradley: that's something the designers of package name maps should think about
 - Michael Dawson:
 - Maybe we should be providing input to the design of package name maps rather than discussing whether to implement them as-is
 - Guy
 - If anybody believes there is something we should prioritize on this front please raise, otherwise might be left until we figure out some of the other issues.
- * Have presentation on loaders. [#135](<https://github.com/nodejs/modules/issues/135>)
- 20 Minute Timebox
- Purpose: get folks to stop thinking of loaders as an amorphous API that allows anything
 - Instead, talk about what loaders actually do, and must be able to do
 - Discussion:
 - MD: Are facades not able to change / add things to the module?
 - BF: when we say code is ESM, we don't touch it, we just load it directly
 - MD: Why would we do otherwise?
 - BF: Changing the ordering of imports would be an example of not directly importing an ESM module
 - BF: We're following the spec as long as resolution returns an ESM module record
 - Proposal:
 - Move loaders off the main thread for isolation and performance
 - Command line `--loader` option
 - Environment variables
 - Per-package loader hooks configured in `package.json`? (BF in favor)
 - Works in user-land currently, though difficult to keep rebasing it against Node master
 - *See slides*
 - Differences between this proposal and existing ideas

- *See slides*
- Issues
 - Caching is tricky, since it works so differently in the browser than other tools
 - Would prefer to remove caching from the proposal
 - Raw amount of time spent doing transformations is significant
 - Caching compilation results should therefore be easy/simple
 - Can't solve this problem in this proposal, but should show it can be solved

* transparent-or-not interop [#90](<https://github.com/nodejs/modules/issues/90>)

- 20 Minute Timebox

- *Skipped this due to running out of time*

Saleh Abdel Motaal on terminology document:

- Not much to say about terminology document progress
- Two documents
 - One document went “way overboard” (first draft)
 - Another took a different direction with input from other people
- Guy: can we add one of those documents as a link to the list of resources that we maintain?
- Saleh: yes, will create a markdown document in the repo
- *Sounds good to everyone*