

Notes: OpenTelemetry JavaScript SIG.

See details about this call: <https://github.com/open-telemetry/community#nodejs-sdk>

THESE MEETINGS ARE RECORDED

Notes from the [OpenTelemetry JS](#) SIG meetings. See the [community repo](#) for details on this and other language SIGs, or join us on [slack](#). Related repo: [opentelemetry-js-contrib](#).

Calendar:

https://calendar.google.com/calendar/u/0/embed?src=google.com_b79e3e90j7bbsa2n2p5an5lf60@group.calendar.google.com

Updated zoom link (in calendar):

<https://zoom.us/j/92719952048?pwd=YkJPt3B6eDh1RVloVi8zdjBXamFLdz09>



Cursor Parking Lot:



Please add yourself to the Attendees list and feel free to add any agenda item

2023-06-21

Notes document archived. For today's and all future meetings see [OpenTelemetry JS Meeting Notes](#)

2023-06-14

Attendees

- Nev Wylie (Microsoft)
- Daniel Dyla (Dynatrace)
- Trent Mick (Elastic)
- Purvi Kanal (Honeycomb)
- Martin Kuba (Lightstep)
- Marc Pichler (Dynatrace)

- Jackson Weber (Microsoft)
- Jamie Danielson (Honeycomb)
- Svetlana Brennan (New Relic)
- Karlie L (Microsoft)

Agenda

- **Feel free to add your topics below** 📌 😊
- [Nev] Code minification discussion about Browser / Minification improvement
 - Proposal: Update tsconfig.base.json to “importHelpers” and “noEmitHelpers”, which will require including “tslib” as a dependency
 - By importing the helpers this avoids TypeScript creating unique versions of the __spreadArray() and __read() helpers for every usage of the spread operator for arguments “function xxx(...args)” and also for usages of “theFunc.call(thisArg, ..args)” etc. Thus reducing the overall package size, examples below from the bundles created by the sandbox-web-js build

Package	Current (UMB Module)	With Import helpers (UMD Module)	Saving
API Bundle	Raw: 49,512; Min: 19,706	Raw:43,570; Min: 17,612	~2k
Core Bundle	Raw: 97,026; Min: 36,727	Raw: 85,238; Min: 32,383	~4k
SDK Trace Web	Raw: 175,118; Min: 66,719	Raw: 152,126; Min: 58,094	~8k

- Issue created: [\[Minification\] Add noEmitHelpers, importHelpers and tslib as a dependency · Issue #3913 · open-telemetry/opentelemetry-js \(github.com\)](#)
- [dan] related to minification above: improve import paths using entypoints
 - <https://github.com/open-telemetry/opentelemetry-js/issues/3908>
- [marc] Should we support Node 14 on new packages (like logs)?
 - EOL reached [a little more than a month ago](#)
 - For stable packages we'd most likely drop it in 2.0
 - <https://github.com/open-telemetry/opentelemetry-js/issues/3583>
- [martin] Performance overhead - are there any baseline benchmarking results that have been done in the past? Specifically, CPU instrumentation cost as described [here](#).
- [nev] Question: The “web” examples are importing the node otlp-http exporter (which is not currently imported (as an active building package)into the sandbox) and it doesn't seem to make sense that “web” examples require node??? Is this just for testing?
- PRs needing review:
 - document load instrumentation update:
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1529>
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-06-07

Attendees

- Marc Pichler (Dynatrace)
- Daniel Dyla (Dynatrace)
- Hector Hernandez (Microsoft)
- Purvi Kanal (Honeycomb)
- Jamie Danielson (Honeycomb)
- Trent Mick (Elastic)
- Abinet Debele (Cisco)
- Nev Wylie (Microsoft)
- Karlie L (Microsoft)
- Jackson Weber (Microsoft)

Agenda

- 1.14.0 and 0.40.0 released
 - Adds esm instrumentation
 - add 'grpc' deprecation notice
 - Remaining log exporters merged, logs should be useable now
- OTLP web refactor [\[otlp-exporter\] refactor of sending requests on web · Issue #3845 · open-telemetry/opentelemetry-js · GitHub](#)
 - Action items:
 - Create issue for creating browser support matrix (what should be supported without polyfills)
 - Create issue against spec questioning implementation of the User Agent for browser JavaScript (especially considering browser user agent is really important; if anything we'd probably want to append?)
 - Maybe consider... don't do fetch. Keep sendWithBeacon as default (this is current behavior) (so no user agent)... if headers are already set, then append User Agent with Exporter and use XHR. This will be probably decided after the spec issue/discussion. (browsers are messy)
- **Feel free to add your topics below**  
- [Abinet] asking for ideas on <https://github.com/open-telemetry/opentelemetry-js/issues/3848>
 - Abinet will fix the document load, daniel will create new issues for the other 2
- PRs needing review:
 - gRPC instrumentation bugfix:
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3804>
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-05-31

Attendees

- Daniel Dyla (Dynatrace)
- Marc Pichler (Dynatrace)
- Trent Mick (Elastic)
- Nev Wylie (Microsoft)
- Jamie Danielson (Honeycomb)
- Jackson Weber (Microsoft)
- Martin Kuba (Lightstep)

Agenda

- Semconv breaking changes
 - Our semconv package is 1.0
 - Using entry points we may be able to have separate import paths for experimental and stable
 - Import { ... } from '@opentelemetry/semantic-conventions/experimental';
- Impossible to have spec-compliant beacon sending?
 - https://github.com/open-telemetry/opentelemetry-js/pull/3811#discussion_r1211872594
 - In regards to sendBeacon being default
 - <https://github.com/open-telemetry/opentelemetry-js/issues/3062>
- **Feel free to add your topics below**  
-
- PRs needing review:
 - Logs exporter PRs
 - Protobuf exporter:
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3779>
 - Logattribute map (reviewed just waiting on one change to be resolved)
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3821>
 - gRPC instrumentation bugfix:
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3804>
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-05-24

Attendees

- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Trent Mick (Elastic)
- Jamie Danielson (Honeycomb)

- Martin Kuba (Lightstep)
- Hector Hernandez (Microsoft)
- Abinet Debele(Cisco)
- Svetlana Brennan (New Relic)

Agenda

- Welcome new approvers Jamie, Hector, and Purvi 😊
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3830>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3831>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3832>
- [Unmaintained components](#) deadline 05/24
 - Thanks to Purvi/Jamie for taking express instrumentation
 - All other components on this list are unmaintained and will be marked as such in the readme. No more aggressive action planned yet (deprecations/warnings)
- **Feel free to add your topics below** 📌 😊
- **[Abinet] , Asking for review of** <https://github.com/open-telemetry/opentelemetry-js/pull/3779> , and <https://github.com/open-telemetry/opentelemetry-js/pull/3821>
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-05-17

Attendees

- Nev Wylie (Microsoft)
- Amir Blum
- Marc Pichler (Dynatrace)
- Hector Hernandez (Microsoft)
- Daniel Dyla (Dynatrace)
- Jackson Weber (Microsoft)
- Jamie Danielson (Honeycomb)
- Purvi Kanal (Honeycomb)
- Martin Kuba (Lightstep)
- Trent Mick (Elastic)

Agenda:

- [Unmaintained components](#) deadline 05/24
- OpenCensus shim <https://github.com/open-telemetry/opentelemetry-js/pull/3809>
 - Note for aabmas we updated require-in-the-middle to v7
 - RITM vs Instrumentation
 - RITM slow issue <https://github.com/open-telemetry/opentelemetry-js/issues/3029>

- Check what lambda instrumentation is doing because they previously had a similar issue
 -
- <https://github.com/open-telemetry/opentelemetry-js/pull/3704>
 - Consider making these span events default-off? Breaking telemetry data
- Drop support for `grpc` module?
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3807>
- Call for reviewers/approvers
- **Feel free to add your topics below** 📌 😊
- [martin] question about otlp-transformer dependency on the API
- [martin] Is there any performance/overhead analysis that has been done in the past?
- [JacksonWeber] forceFlush PR ready for final review
 - [feat\(opentelemetry-sdk-trace-base\): Add optional forceFlush property to SpanExporter interface](#)
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-05-10

Attendees

- Marc Pichler (Dynatrace)
- Jamie Danielson (Honeycomb)
- Purvi Kanal (Honeycomb)
- Trent Mick (Elastic)
- Jackson Weber (Microsoft)
- Abinet Debele (Cisco)
- Hector Hernandez (Microsoft)
- Svetlana Brennan (New Relic)

Agenda

- (contrib) Release 1.13 / 0.39.0 (still coming)
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3776>
- (contrib) [Component ownership](#) PR merged
- **Feel free to add your topics below** 📌 😊
- [martin] HTTP Logs exporters PRs
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3764>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3779>
- [purvi] ESM support PR
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3698>
- [JacksonWeber] SpanExporter PR final question
 - [feat\(open-telemetry-sdk-trace-base\): Add mandatory forceFlush property to SpanExporter](#)
-

- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-05-03

Attendees

- Nev Wylie (Microsoft)
- Aaron Abbott (Google)
- Marc Pichler (Dynatrace)
- Martin Kuba (Lightstep)
- Daniel Dyla (Dynatrace)
- Trent Mick (Elastic)
- Jackson Weber (Microsoft)
- Jamie Danielson (Honeycomb)
- Purvi Kanal (Honeycomb)

Agenda

- Release 1.13 / 0.39.0
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3776>
- [Component ownership](#)
 - Next steps: automate issue and PR assignment by tags (see collector contrib for example)
- **Feel free to add your topics below**  
- [aaron] OpenCensus shim <https://github.com/open-telemetry/opentelemetry-js/pull/3751>
- [purvi] ESM PR ready for re-review: <https://github.com/open-telemetry/opentelemetry-js/pull/3698>
- [JacksonWeber] SpanExporter Ready for Review
[feat\(opentelemetry-sdk-trace-base\): Add mandatory forceFlush property to SpanExporter interface by JacksonWeber · Pull Request #3753 · open-telemetry/opentelemetry-js \(github.com\)](#)

2023-04-26

Attendees

- Daniel Dyla (Dynatrace)
- Marc Pichler (Dynatrace)
- Nev Wylie (Microsoft)
- Jamie Danielson (Honeycomb)
- Martin Kuba (Lightstep)
- Trent Mick (Elastic)
- Svetlana Brennan (New Relic)

Agenda

- Announcements:
 - [Logs gRPC exporter PR merged](#)
- **Feel free to add your topics below** 📧 😊
- [marc] Collecting thoughts on removing support for OTEL_TRACES_EXPORTER=jaeger from @opentelemetry/sdk-node (breaking change)
 - @opentelemetry/sdk-node is still experimental
 - OTLP is supported by Jaeger natively now
 - jaeger-client library used by the exporter is unmaintained
 - The spec made support for it optional
- [dan] contrib component maturity
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1474>
- [dan] component ownership audit
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1478>
 - detectors/node/opentelemetry-resource-detector-gcp
 - detectors/node/opentelemetry-resource-detector-github
 - packages/opentelemetry-propagation-utils
 - packages/opentelemetry-test-utils
 - plugins/node/opentelemetry-instrumentation-bunyan
 - plugins/node/opentelemetry-instrumentation-cassandra
 - plugins/node/opentelemetry-instrumentation-connect
 - plugins/node/opentelemetry-instrumentation-dns
 - plugins/node/opentelemetry-instrumentation-express
 - plugins/node/opentelemetry-instrumentation-fastify
 - plugins/node/opentelemetry-instrumentation-hapi
 - plugins/node/opentelemetry-instrumentation-koa
 - plugins/node/opentelemetry-instrumentation-mysql2
 - plugins/node/opentelemetry-instrumentation-net
 - plugins/node/opentelemetry-instrumentation-pino
 - plugins/node/opentelemetry-instrumentation-winston
 - plugins/web/opentelemetry-plugin-react-load
 - propagators/opentelemetry-propagator-grpc-census-binary
 - propagators/opentelemetry-propagator-ot-trace
- [marc] [OTLP Exporter GA Milestone](#)
 - If anyone has some cycles, feel free to pick up one of the issues there. 😊

2023-04-19

Attendees

- Marc Pichler (Dynatrace)
- Trent Mick (Elastic)
- Martin Kuba (Lightstep)
- Nev Wylie (Microsoft)
- Matt Wear (Lightstep)

- Jackson Weber (Microsoft)

Agenda

- Announcements:
 - [1.12.0](#) / [Experimental v0.38.0](#) released 🎉
 - Includes Logs SDK & bug fixes
 - EasyCLA bot currently not accepting commits created via Github UI ❌
 - <https://github.com/communitybridge/easycla/issues/3903>
 - For important PRs please merge main locally until this is resolved
- **Feel free to add your topics below** ✍️ 😊
- [marc] Created a (long overdue) [OTLP Exporter GA Milestone](#)
 - If there are things that you'd like to see before we go GA with the exporters, please create issues and we'll assign it to that milestone. 😊
 - I'll be focusing on moving that along 🎯
 - I'm planning to do this with other components too after the exporters are done
- [your name] your topic
- ESM / import-in-the-middle <https://github.com/open-telemetry/opentelemetry-js/pull/3698>
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-04-12

Attendees

- Daniel Dyla (Dynatrace)
- Marc Pichler (Dynatrace)
- Jackson Weber (Microsoft)
- Svetlana Brennan (New Relic)
- Trent Mick (Elastic)

Agenda

- Zoom link changed
 - <https://github.com/open-telemetry/community/issues/1414> for details
 - <https://zoom.us/j/92719952048?pwd=YkJPt3B6eDh1RVloVi8zdjBXamFLdz09>
- [V1.12.0 proposal](#)
- Logs SDK <https://github.com/open-telemetry/opentelemetry-js/pull/3549>
 - Release with above proposal?
 - <https://github.com/open-telemetry/opentelemetry-specification/pull/3354>
- Component ownership audit not yet complete

- <https://github.com/open-telemetry/assign-reviewers-action> getting updated and will be receiving new features for ownership. Help appreciated
- PR opened and approved for RITM request-promise bug
 - <https://github.com/elastic/require-in-the-middle/pull/63>
- Update on RITM esModuleInterop fix?
 - <https://github.com/open-telemetry/opentelemetry-js/issues/3701>
- Kubecon EU next week
 - Keeping meeting for now :)
- **Feel free to add your topics below**  
- [your name] your topic
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-04-05

Attendees

- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Hector Hernandez (Microsoft)
- Trent Mick (Elastic)
- Svetlana Brennan (New Relic)
- Matt Wear (Lightstep)
- Jamie Danielson (Honeycomb)
- Stephen Belanger (Datadog)
- Jackson Weber (Microsoft)
- Purvi Kanal (Honeycomb)
- Helio Frota (Red Hat)

Agenda

- Note: dan might lose power or internet at any point during the meeting
- [V1.11.0](#) released
- Down to 4 maintainers. Dan doing contrib component ownership audit
- **Feel free to add your topics below**  
- [your name] your topic
- [marc] gRPC Exporter PR that will make the exporter work in bundled Node apps (please review )
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3705>
- [dan] from last week [martin] [Logs SDK PR](#) one more request for review
 - Reviewed and waiting on author changes
- [dan] update from last week on RITM bug?
 - <https://github.com/open-telemetry/opentelemetry-js/issues/3701>
- [stephen] Node.js landed TracingChannel
 - <https://github.com/nodejs/node/pull/44943>

- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-03-29

Attendees

- Daniel Dyla (Dynatrace)
- Marc Pichler (Dynatrace)
- Jackson Weber (Microsoft)
- Nev Wylie (Microsoft)
- Amir Blum
- Hector Hernandez (Microsoft)
- Trent Mick (Elastic)
- Matt Wear (Lightstep)
- Martin Kuba (Lightstep)
- Abinet Debele (Cisco)

Agenda

- [1.11.0/0.37.0](#) release PR is open, release is expected to include Exponential Histograms
- JS Maintainership / Ownership
- Please review:
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1400>
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1444>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3549>
- Kubecon EU
 - [Observability day](#) Tuesday April 18
 - [OpenTelemetry Project Meeting](#) Tuesday April 18 1pm - 3pm (time/date may change)
 - [Maintainer Track Talk](#) Thursday April 20 4:30pm - 5pm
 - Dan will be at the booth most of the time
 - Attendance?:
 - Dan
- **Feel free to add your topics below**  
- [your name] your topic
- [martin] [Logs SDK PR](#) one more request for review
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-03-22

Attendees

- Marc Pichler (Dynatrace)
- Hector Hernandez (Microsoft)
- Matt Wear (Lightstep)
- Jamie Danielson (Honeycomb)
- Nev Wylie (Microsot)
- Svetlana Brennan (New Relic)

Agenda

- Announcements:
 - [1.10.1](#) fix for breaking changes is published, Node 12 (unsupported) should continue to work for now
 - [1.11.0/0.37.0](#) release PR is open, release is expected to include Exponential Histograms
- **Feel free to add your topics below**  
- [your name] your topic
-
-
-
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-03-15

Attendees

- Nev Wylie
- Daniel Dyla (Dynatrace)
- Marc Pichler (Dynatrace)
- Matt Wear (Lightstep)
- Purvi Kanal (Honeycomb)
- Jackson Weber (Microsoft)

Agenda

- [nev] Sandbox update
 - Ready for branching (branch workstreams) - [open-telemetry/opentelemetry-sandbox-web-js: non-production level experimental Web JS packages \(github.com\)](#)
 - Up to date as of last week, has a PR out to update to yesterday (version changes occurred so waiting until this meeting before merging)
 - [open-telemetry/opentelemetry-sandbox-web-js: non-production level experimental Web JS packages \(github.com\)](#)
 - This just merges the code from js/js-contrib to the `auto-merge/repo-staging` and does not perform any validation (it just brings over all of the history and code)

- Asks:
 - What additional (web) packages do you want merged (currently doesn't have any of the examples actively compiled / tested)
 - What branches / workstreams would you like to create
 - Current branch (workstream) plans
 - Minification - Try and minify the existing code to make it "viable" for a web environment. This will also benefit node (if committed back to source repos)
 - Events / logs prototyping / development (once complete this will be merge (via new PR) back to main repo's)
 - Minimal Web JS SDK only (not reusing everything) – eg only using the interfaces. This is a parallel workstream to the minification
 - The `main` branch of the sandbox should NOT have any code changes directly added to it except via the merge scripts
 - This branch represents the "baseline" of the code migrated from the JS and JS-Contrib repos only
 - And the workstream branches will perform `git merge` from the `main` branch to their branch to consume the updated migrated packages
- [dan] API/SDK/experimental released
 - Resource backwards compatibility problem
 - Pull: <https://github.com/open-telemetry/opentelemetry-js/pull/3677>
 - Issue: <https://github.com/open-telemetry/opentelemetry-js/issues/3676>
 - [dan / matt] Exponential Histograms
 - ~~[Part 1 - Mapping functions](#)~~
 - ~~[Part 2 - Accumulation & Aggregator](#)~~ (will be merged after the new release)
 - [Part 3 - Export](#)
 - Draft to show how they all work together - [feat: Add Exponential Histogram](#)
 - Video code walkthrough by matt wear <https://youtu.be/5AkBbc2QXqc?t=630>
 - [martin] Are plugins in contrib experimental? What is the guidance on making breaking changes? See [this PR](#) for AWS Lambda with a breaking change.
 - **Feel free to add your topics below**  
 - [your name] your topic
 - [jamie/purvi] [ESM updates](#)
 - [In Progress PR](#)
 - ...
 - ...
 - ...
 - 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-03-08

Attendees

- Marc Pichler (Dynatrace)
- Hector Hernandez (Microsoft)
- Trent Mick (Elastic)
- Jackson Weber (Microsoft)
- Svetlana Brennan (New Relic)

Agenda

- [marc] New API/SDK/Experimental release in the works
 - [chore: Release API 1.4.1, SDK 1.10.0, Experimental 0.36.0 by dyladan](#)
- [marc] Logs SDK is ready for review
 - [Logs SDK by fuaiyi · Pull Request #3549](#)
- [marc] Please review
 - [fix\(sdk-metrics\): deprecate instrument type in `InstrumentDescriptor` by pichlermarc](#)
- [marc / matt] Exponential Histograms
 - ~~[Part 1 - Mapping functions](#)~~
 - [Part 2 - Accumulation & Aggregator](#) (will be merged after the new release)
 - [Part 3 - Export](#)
 - Draft to show how they all work together - [feat: Add Exponential Histogram](#)
 - Video code walkthrough by matt wear <https://youtu.be/5AkBbc2QXqc?t=630>
- [nev] [fix: Minor typing issues for web sandbox by MSNev · Pull Request #3660 · open-telemetry/opentelemetry-js \(github.com\)](#)
- **Feel free to add your topics below**  
- [your name] your topic
- ...
- ...
- ...
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-03-01

Attendees

- Marc Pichler (Dynatrace)
- Nev Wylie (Microsoft)

- Daniel Dyla (Dynatrace)
- Matt Wear (Lightstep)
- Svetlana Brennan (New Relic)
- Jamie Danielson (Honeycomb)
- Jackson Weber (Microsoft)
- Karlie L (Microsoft)

Agenda

- **Feel free to add your topics below**  
- [your name] your topic
- [marc] Please review:
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3641>
- [marc] Contrib CI is failing
 - Not the same issue as we had in core, looks like adding memory/getting rid of nyc source-map generation does not fix the issue there
 - Amir's PR trying to increase memory: <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1416>
 - Test failure example <https://github.com/open-telemetry/opentelemetry-js-contrib/actions/runs/4294102101/jobs/7483915093>
- [dan / matt] Exponential Histograms
 - ~~[Part 1 - Mapping functions](#)~~
 - [Part 2 - Accumulation & Aggregator](#)
 - [Part 3 - Export](#)
 - Draft to show how they all work together - [feat: Add Exponential Histogram](#)
 - Video code walkthrough by matt wear <https://youtu.be/5AkBbc2QXqc?t=630>
- Logs SDK ready for review <https://github.com/open-telemetry/opentelemetry-js/pull/3549>
- Related to autoinstrumentation PR
 - OTEP for file-based configuration <https://github.com/open-telemetry/oteps/pull/225>
- [mwear] Process Detector improvements:
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3605>
- [nev] Sandbox update
 - Now failing (once sync'd to yesterday) on the resources package due to this
 - "@opentelemetry/resources_1.9.0": "npm:@opentelemetry/resources@1.9.0", specifically on this line
 - `import { Resource as Resource190 } from '@opentelemetry/resources_1.9.0';`
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-02-22

Attendees

- Nev Wylie (Microsoft)
- Daniel Dyla (Dynatrace)
- Marc Pichler (Dynatrace)
- Svetlana Brennan (New Relic)
- Amir Blum (Aspecto)
- Aaron Abbott (Google)
- Karlie L (Microsoft)

Agenda

- **Feel free to add your topics below**  
- [your name] your topic
- [dan / matt] Exponential Histograms
 - ~~Part 1 - Mapping functions~~
 - [Part 2 - Accumulation & Aggregator](#)
 - [Part 3 - Export](#)
 - Draft to show how they all work together - [feat: Add Exponential Histogram](#)
 - Video code walkthrough by matt wear <https://youtu.be/5AkBbc2QXgc?t=630> (timestamp link)
- [marc] windows test failures (out of memory)
- [marc] Question for nev - how hard would it be to apply the rush/rollup sandbox changes to the main repo?
- [nev] Sandbox update, sync'd to Jan 16th
 - Now has Api, Core and semantic conventions merged
 - PR out to add resources, context-zone and context-zone dep (dealing with tsconfig.es5.json rename from core [chore: Update merge main script to handle the renaming of the base tsconfig.x.json files by MSNev · Pull Request #54 · open-telemetry/opentelemetry-sandbox-web-js \(github.com\)](#))
- [sami] [JS Auto Instrumentation](#)
- [martin] [Log SDK](#) - please review high level design
- [svetlana]: [retry pr](#) needs review
- [aaron] fyi planning to work on [OpenCensus bridges](#)
 - Please assign to me <https://github.com/open-telemetry/opentelemetry-js/issues/3461>
 - Will make bug for trace bridge
- [Amir] need webpack help in [this PR](#)
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-02-15

Attendees

- Nev Wylie (Microsoft)
- Marc Pichler (Dynatrace)
- Amir Blum (Aspecto)
- Matt Wear (Lightstep)
- Stephen Belanger (Datadog)
- Karlie L (Microsoft)
- Hector Hernandez (Microsoft)

Agenda

- Exponential Histograms
 - ~~Part 1 - Mapping functions~~
 - [Part 2 - Accumulation & Aggregator](#)
 - [Part 3 - Export](#)
 - Draft to show how they all work together - [feat: Add Exponential Histogram](#)
 - Video code walkthrough by matt wear <https://youtu.be/5AkBbc2QXqc?t=630> (timestamp link)
- **Feel free to add your topics below**  
- [your name] your topic
- [nev] Sandbox update
 - [chore: Add additional required dev dependencies by MSNev · Pull Request #40 · open-telemetry/opentelemetry-sandbox-web-js \(github.com\)](#)
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-02-08

Attendees

- Nev Wylie (Microsoft)
- Marc Pichler (Dynatrace)
- Karlie L (Microsoft)
- Matt Wear (Lightstep)
- Daniel Dyla (Dynatrace)
- Jackson Weber (Microsoft)
- Trent Mick (Elastic)

Agenda

- New maintainer: Marc Pichler
- New Approver: Martin Kuba
<https://github.com/open-telemetry/opentelemetry-js/pull/3594> 🥳
- New Approver: Haddas Bronfman
<https://github.com/open-telemetry/opentelemetry-js/pull/3593>
- Contrib released with new 1.9.1 and 0.35.1 SDK
- Exponential Histograms
 - ~~Part 1 - Mapping functions~~
 - [Part 2 - Accumulation & Aggregator](#)
 - [Part 3 - Export](#)
 - Draft to show how they all work together - [feat: Add Exponential Histogram](#)
 - Video code walkthrough by matt wear <https://youtu.be/5AkBbc2QXqc?t=630> (timestamp link)
- ~~Logs/Events API PR splitting logs and events API merged~~
 - Next step: logs SDK PR
- **Feel free to add your topics below** 📌 😊
- [your name] your topic
- [sami] [sync resource](#)
- [marc] [chore\(jeager-exporter\): deprecate jaeger exporter](#)
- [dan] [GA package deprecation policy](#)
- [dan] [SDK 2.0 Milestone created](#)
- [nev] Sandbox merge to main script was merged yesterday and [produced a merge PR for the api](#)
 - Only minor issue detected, should have CI pipeline activated by next week and
 - new plan is to have this available for branch creation by end of Feb.
 - Config location for the packages to merge and where to get them from and drop them into
<https://github.com/open-telemetry/opentelemetry-sandbox-web-js/blob/faccfa6ddf5c88b9a7ec3725670dd967f7c65633/sandbox-tools/merge-repos/src/config.ts#L109-L125>
 -
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-02-01

Attendees

- Nev Wylie (Microsoft)
- Amir Blum (Aspecto)
- Jackson Weber (Microsoft)
- Karlie L (Microsoft)
- Trent Mick (Elastic)

- Stephen Belanger (Datadog)
- Matt Wear (Lightstep)

Agenda

- 1.9.1 and 0.35.1 bugfix versions released
- Please review 😊
 - Exponential Histograms
 - ~~Part 1 - Mapping functions~~
 - [Part 2 - Accumulation & Aggregator](#)
 - [Part 3 - Export](#)
 - Draft to show how they all work together - [feat: Add Exponential Histogram](#)
 - Video code walkthrough by matt wear
<https://youtu.be/5AkBbc2QXqc?t=630> (timestamp link)
- **Feel free to add your topics below** 📌 😊
- [your name] your topic
- [martin] [Logs/Events API PR](#)
- [sami] [sync resource](#)
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-01-25

Attendees

- Matt Wear (Lightstep)
- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Jackson Weber (Microsoft)
- Marc Pichler (Dynatrace)
- Trent Mick (Elastic)
- Stephen Belanger (Datadog)

Agenda

- Leaked grpc types fix merged. Release this week? Blocking contrib update to latest SDK
- Clock drift fix for metrics caused a flaky test
 - LastValueAggregator > diff > return the newly sampled accumulation
 - Likely also affects others like “merge” section
 - Probably just need to mock Date.now in the tests
 - <https://github.com/open-telemetry/opentelemetry-js/actions/runs/3965220408/jobs/6794748510>
 - <https://github.com/open-telemetry/opentelemetry-js/actions/runs/3999844458/jobs/6864243629>

-
- Please review 😊
 - Exponential Histograms
 - [Part 1 - Mapping functions](#) - Ready to merge?
 - [Part 2 - Accumulation & Aggregator](#)
 - [Part 3 - Export](#)
 - Draft to show how they all work together - [feat: Add Exponential Histogram](#)
 - Video code walkthrough by matt wear
<https://youtu.be/5AkBbc2QXqc?t=630> (timestamp link)
- **Feel free to add your topics below** 🗨️ 😊
- [your name] your topic
- [mwear] - [feat\(lambda\): add OTEL_LAMBDA_DISABLE_AWS_CONTEXT_PROPAGATION environment variable](#)
- [martin] Logs/Events - can this PR be merged this week?
<https://github.com/open-telemetry/opentelemetry-js/pull/3550>
- [stephen] Node.js TracingChannel - <https://github.com/nodejs/node/pull/44943>
- [nev] Sandbox merge to main script is almost ready, gave a talk in the RUM sig yesterday
 - [chore: Add Merge Staging to Main branch script by MSNev · Pull Request #22 · open-telemetry/opentelemetry-sandbox-web-js \(github.com\)](#)
 - Example of generated PR (with more packages enabled) [\[AutoMerge\]\[Main\] Merging staged change\(s\) to main by MSNev · Pull Request #21 · open-telemetry/opentelemetry-sandbox-web-js \(github.com\)](#)
- ...
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-01-18

Attendees

- Marc Pichler (Dynatrace)
- Jackson Weber (Microsoft)
- Trent Mick (Elastic)
- Karlie L (Microsoft)
- Nev Wylie (Microsoft)
- Hector Hernandez (Microsoft)
- Matt Wear (Lightstep)
- Amir Blum (Aspecto)
- Siim Kallas (Splunk)

Agenda

- API 1.4.0 / SDK 1.9.0 / Experimental 0.35.0 is out 🎉

- Performance clock drift fix for both traces & metrics
 - Various fixes and improvements for metrics, instrumentations and more.
- Please review 😊
 - Exponential Histograms
 - [Part 1 - Mapping functions](#)
 - [Part 2 - Accumulation & Aggregator](#)
 - [Part 3 - Export](#)
 - Draft to show how they all work together - [feat: Add Exponential Histogram](#)
 - Video code walkthrough by matt wear
<https://youtu.be/5AkBbc2QXqc?t=630> (timestamp link)
- **Feel free to add your topics below** ✅ 😊
- [marc] Delaying creation of Meter until getter is called first in @opentelemetry/instrumentation
 - Not sure if we talked about this before
 - Realized that we have lots of empty scopes being exported for instrumentation packages that do not create any metrics.
- Needs reviews: [eager exporting for batch span processor](#)
- [martin] Please review - [separate Events API from Logs API](#)
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-01-11

Attendees

- Nev Wylie (Microsoft)
- Daniel Dyla (Dynatrace)
- Amir Blum (Aspecto)
- Trent Mick (Elastic)
- Marc Pichler (Dynatrace)
- Jackson Weber (Microsoft)

Agenda

- Dan out next week
- [API 1.4.0 / SDK 1.9.0 / Experimental 0.35.0 release PR](#)
 - Merging after clock drift (below)
- High Resolution Histogram Support
 - [Part 1 - Mapping functions](#)
 - Approved by 1 maint + 1 approver
 - [Part 2 - Accumulation & Aggregator](#)
 - [Part 3 - Export](#)
 - Draft to show how they all work together - [feat: Add Exponential Histogram](#)

- Video code walkthrough by matt wear <https://youtu.be/5AkBbc2QXqc?t=630> (timestamp link)
- [fix\(tracing\): make spans resilient to performance clock drift](#)
 - Last chance to comment. Merging after meeting
 - Related, but metrics [use Date.now\(\) for instrument recording timestamps](#)
- [fix\(sdk-trace-base\): eager exporting for batch span processor](#)
 - [Spec PR](#) has many approvals, near merge
- [feat\(api-logs\): split out event api](#)
 - Old PR but important, please review
 - Related: The Logs API / Events API issue from Tigran <https://github.com/open-telemetry/opentelemetry-specification/issues/3086>
- [Create sync resource with some attributes that resolve asynchronously](#)
- [marc] browser packaging question related to browser support for otel http proto https://github.com/open-telemetry/opentelemetry-js/pull/3208#discussion_r1060005046
- [nev] Plan to get back to make the sandbox ready for branching and contributions (hoping to get this done by end of next week)
- [Santosh] What is a good place to keep the docs on browser events - spec repo or js repo? See https://github.com/scheler/opentelemetry-specification/blob/rum-events/specification/logs/events/semantic_conventions/browser.md
- [Abinet] Help on PR, <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1238>
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2023-01-04

Attendees

- Daniel Dyla (Dynatrace)
- Marc Pichler (Dynatrace)
- Amir Blum (Aspecto)
- Jackson Weber (Microsoft)
- Trent Mick (Elastic)
- Matt Wear (Lightstep)
- Svetlana Brennan (New Relic)

Agenda

- Contrib released last week <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1301>
- [API 1.4.0 / SDK 1.9.0 / Experimental 0.35.0 release PR](#)
- [Document repo settings](#)
 - Important: 1 reviewer
 - Will be doing same for contrib

- TODO: update contributing.md to reflect 1 reviewer
 - TODO: look into setting that allowed approvers to merge
- High Resolution Histogram Support
 - [Part 1 - Mapping functions](#)
 - [Part 2 - Accumulation & Aggregator](#)
 - [Part 3 - Export](#)
 - Draft to show how they all work together - [feat: Add Exponential Histogram](#)
 - Video code walkthrough by matt wear <https://youtu.be/5AkBbc2QXqc?t=630> (timestamp link)
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-12-21

Attendees

- Amir Blum (Aspecto)
- Daniel Dyla (Dynatrace)
- Matt Wear (Lightstep)
- Abinet Debele(Cisco)
- Marc Pichler (Dynatrace)
-
-

Agenda

- Cancel next week's meeting
- Reduce required approvals to 1
- [Create sync resource with some attributes that resolve asynchronously](#) still needs reviews
- No update on release automation
- Clock drift PR still being worked on. Should finish this week
 - [fix\(tracing\): make spans resilient to performance clock drift](#)
- Logs:
 - Merged [refactor\(api-logs\): refactor Logger api by fuaiyi · Pull Request #3443 · open-telemetry/opentelemetry-js · GitHub](#)
 - Next step: [feat\(api-logs\): split out event api](#)
- [mwear] [Exponential histogram PR](#) walkthrough
- [Abinet] Help on PR, <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1238>
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-12-14

Attendees

- Daniel Dyla (Dynatrace)
- Amir Blum (Aspecto)
- Trent Mick (Elastic)
- Hector Hernandez (Microsoft)
- Marc Pichler (Dynatrace)
- Matt Wear (Lightstep)
- Nev Wylie (Microsoft)
- Martin Kuba (New Relic)
- Jackson Weber (Microsoft)
- Jamie Danielson (Honeycomb)
- Svetlana Brennan (New Relic)

Agenda

- Needs reviews
 - [fix\(tracing\): make spans resilient to performance clock drift](#)
 - [Create sync resource with some attributes that resolve asynchronously](#)
- [mwear] Checking in on the status of: [feat\(lambda\): add OTEL_LAMBDA_DISABLE_AWS_CONTEXT_PROPAGATION environment variable](#)
- [dan] Contrib release automation - no update
- [mwear] Exponential histogram - brief update
 - Matt might walk us through the implementation next week or in January during the SIG Meeting
- [nev] Sandbox is not going to be ready this year 😞
- [martin] Logs SDK - blocked on changes in the API
 - [SDK draft PR](#)
 -
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-12-07

Attendees

- Amir Blum (Aspecto)
- Nev Wylie (Microsoft)
- Daniel Dyla (Dynatrace)
- Trent Mick (Elastic)
- Hector Hernandez (Microsoft)
- Matt Wear (Lightstep)
- Jamie Danielson (Honeycomb)

- Jackson Weber (Microsoft)
- Svetlana Brennan (New Relic)

Agenda

- Needs reviews
 - [fix\(tracing\): make spans resilient to performance clock drift](#)
 - [fix\(api\): use active context as default in NoopTracer](#)
 - [Create sync resource with some attributes that resolve asynchronously](#)
- Contrib release automation - no update
- [Nev] Sandbox update – plan to work on creating the “main” merge script in the next 2 weeks to hopefully get “main” ready for branches
- Logs API
 - thoughts on the proposed change in [this PR](#)
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-11-30

Attendees

- Amir Blum (Aspecto)
- Marc Pichler (Dynatrace)
- Matt Wear (Lightstep)
- Jackson Weber (Microsoft)
- Aaron Abbott (Google)
- Karlie Li (Microsoft)

Agenda

- [dan] New spec PR to clarify what “spec compliant” API/SDK are required and allowed to do
 - <https://github.com/open-telemetry/opentelemetry-specification/pull/2968>
- [marc] Instrument descriptor
 - [Instrument descriptor semantics are unclear when using views](#)
 - dataPoint.descriptor.type is “incorrect”
 - Proposing to deprecate it & document that it refers to the original instrument’s type
- [dan] Still working on contrib release automation
- [dan] New approach to fixing span clock drift (brought over from last week)
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3434>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3384>
- [martin] [Logs SDK PR](#)
 - was out the last two weeks, but planning to make progress on the PR this week
 - Refactor PR opened
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3443/files>

- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-11-23

Attendees

- Daniel Dyla
- Marc Pichler (Dynatrace)
- Abinet Debele
- Aaron Abbott (Google)
- Jackson Weber (Microsoft)
- Amir Blum (Aspecto)
-

Agenda

- Short meeting probably (holiday week)
- New approach to fixing span clock drift
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3434>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3384>
- Log API refactor
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3443>
- Help on merging PR, <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1238>
- [amir] ([sql commenter](#))
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-11-16

Attendees

- Daniel Dyla
- Jackson Weber (Microsoft)
- Amir Blum (Aspecto)
- Matt Wear (Lightstep)
- Marc Pichler (Dynatrace)
- Nev Wylie (Microsoft)
- Svetlana Brennan (New Relic)
- Jamie Danielson (Honeycomb)

Agenda

- Metrics GA finalization

- Contrib release
<https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1269>
- Next priorities
 - Logs
 - API convenience helpers
 - Developer experience
 - Nev out for the next 2 weeks, but working on sandbox when he gets back
- Fix span clock drift
 - Visual diagram of solution
<https://github.com/open-telemetry/opentelemetry-js/issues/3279#issuecomment-1302575571>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3384/files>
 - Now ready for review - covers span events also
 - [t2] No fix yet for if startTime is specified and is a performance timing? (document-load instrumentation)
 - startTime < timeOrigin -> add offset
 - startTime + offset + timeOrigin
(via timeInputToHrTime)
- Logs update?
 - <https://github.com/open-telemetry/opentelemetry-specification/pull/2941>
 - Potential update to events: use cloudevents to define event data
- Sandbox update?
 - Merge script working
 - Next step is to use rush/rollup on the merged repo
 - Current "Merge" PR (from 2 days ago) [\[AutoMerge\] Merging change\(s\) from otel-js; otel-js-contrib; by opentelemetrybot · Pull Request #12 · open-telemetry/opentelemetry-sandbox-web-js \(github.com\)](#)
- All docs move to website including new metrics docs
- [Rename types.ts to config.ts](#)
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-11-09

Attendees

- Daniel Dyla (Dynatrace)
- Marc Pichler (Dynatrace)
- Jackson Weber (Microsoft)
- Matt Wear (Lightstep)
- Svetlana Brennan (New Relic)
- Nev Wylie (Microsoft)
- Hector Hernandez (Microsoft)
- Jamie Danielson (Honeycomb)

Agenda

- Metrics GA release
 - Main release PR: <https://github.com/open-telemetry/opentelemetry-js/pull/3340>
 - API only release PR: <https://github.com/open-telemetry/opentelemetry-js/pull/3399>
 - TC spec review results
 - <https://github.com/open-telemetry/community/issues/1204>
 - 7 Issues opened, 5 already resolved, 2 pushed to after GA
 - After GA milestone <https://github.com/open-telemetry/opentelemetry-js/milestone/16>
- Proposal for long term span timing fix PTAL
 - <https://github.com/open-telemetry/opentelemetry-js/issues/3279#issuecomment-1302575571>
 - Prototype of this diagram: <https://github.com/open-telemetry/opentelemetry-js/pull/3384/files>
- [Svetlana] Needs review - retries pr: <https://github.com/open-telemetry/opentelemetry-js/pull/3207>
- [Martin] Logs SDK update
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3400>
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-11-02

Attendees

- Amir Blum (Aspecto)
- Jackson Weber (Microsoft)
- Nev Wylie (Microsoft)
- Matt Wear (Lightstep)
- Marc Pichler (Dynatrace)
- Purvi Kanal (Honeycomb)
- Martin Kuba (New Relic)
- Hector Hernandez (Microsoft)

Agenda

- Metrics GA
 - [PR opened](#) no longer a draft
 - Blocked on [feat\(api\): merge api-metrics into api #3374](#)
 - [Docs PR](#) (would need to change after 3374 merge)
- ESM PR updated and no longer draft [feat\(instrumentation\): add support for esm module](#)
- Matt working on exponential histogram support
- Anchored clock PR reverted due to negative span duration bug

- <https://github.com/open-telemetry/opentelemetry-js/issues/3279>
- Log SDK update?
- Instrumentations GUIDELINES?
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1261>
 -
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-10-19

Attendees

- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Marc Pichler (Dynatrace)
- Purvi Kanal (Honeycomb)
- Jackson Weber (Microsoft)
- Matt Wear (Lightstep)
- Kesha Mykhailov (Intercom)
- Amir Blum (Aspecto)
- Jamie Danielson (Honeycomb)
- Svetlana Brennan (New Relic)
- Hector Hernandez (Microsoft)

Agenda

- Kubecon next week - meeting canceled
- Governance committee election
 - <https://github.com/open-telemetry/community/blob/main/elections/2022/governance-committee-election.md#voting-process>
- Metrics GA
 - PR (draft) open <https://github.com/open-telemetry/opentelemetry-js/pull/3340>
 - Blocked on:
 - [fix\(resources\): strict OTEL_RESOURCE_ATTRIBUTES decoding #3341](#)
 - Docs
 - Compilation failure - lerna linking does not work for peer deps so metric api 1.x is not found during bootstrap
 - Proposed solution: release only metrics api package, then release all other packages
- After GA
 - In progress
 - ~~Prometheus exporter resource support~~
 - ~~<https://github.com/open-telemetry/opentelemetry-js/pull/3300>~~

- High resolution histogram (exponential) - dan started on a prototype for this
 - No update
 - Exemplars
 - Temporality configuration via environment variable
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3305>
 - 1 more review required
- Waiting on spec
 - Drop unused attributes (fixes potential memory leak concern)
- [dan] Timing issues with anchored clock (+1 from Keshava Mykhalov to chat about this)
 - No objection on the issue so I will change the SDK to use Date
 - <https://github.com/open-telemetry/opentelemetry-js/issues/3279>
 - Merged <https://github.com/open-telemetry/opentelemetry-js/pull/3327> which ensures at least that span durations are positive
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-10-12

Attendees

- Rauno Viskus (Splunk)
- Daniel Dyla (Dynatrace)
- Sebastian Poxhofer (N26)
- Jackson Weber (Microsoft)
- Marc Pichler (Dynatrace)
- Nev Wylie (Microsoft)
- Jamie Danielson (Honeycomb)
- Svetlana Brennan (New Relic)

Agenda

- Metrics GA
 - Still just waiting on TC review
 - <https://github.com/open-telemetry/community/issues/1204>
- After GA
 - In progress
 - Prometheus exporter resource support
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3300>
 - High resolution histogram (exponential) - dan started on a prototype for this
 - No update
 - Exemplars
 - Temporality configuration via environment variable
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3305>

- - Waiting on spec
 - Drop unused attributes (fixes potential memory leak concern)
 - API merge to main repo
 - ~~Clean up repo files~~
 - ~~[Add back to lerna](#)~~
 - ~~Webworker tests currently failing~~
 - ~~Add API to auto published docs~~
 - Automate versioning
 - Experimental, sdk and experimental, all 3
 - ~~Migrate all issues from API repo~~
 - All issues migrated, closed, or reopened with links to old issues
 - All pull requests must be reopened by the original author. Commented on all PRs
 - Archive API repo
 - ~~Test old node versions on API only~~
 - [purvi] Browser proto exporter is ready for review: <https://github.com/open-telemetry/opentelemetry-js/pull/3208>
 - [purvi] is a PR like <https://github.com/open-telemetry/opentelemetry-js/pull/3087> up for grabs? I have cycles to help push this along
 - Assigned to Purvi by Marc
 - [dan] Timing issues with anchored clock
 - No objection on the issue so I will change the SDK to use Date
 - <https://github.com/open-telemetry/opentelemetry-js/issues/3279>
 - [dan] reminder to ask martin if there is logs/events sdk/exporter work ongoing or is it up for grabs
 - issue - <https://github.com/open-telemetry/opentelemetry-js/issues/3061>
 - Requesting feedback on proposal for:
 - SugaredTracer <https://github.com/open-telemetry/opentelemetry-js/pull/3317>
 - Previously <https://github.com/open-telemetry/opentelemetry-js-api/pull/173>
 - API extension (ForcedFlush and Shutdown) <https://github.com/open-telemetry/opentelemetry-js/pull/3323>
 - Previously <https://github.com/open-telemetry/opentelemetry-js-api/pull/172>
 - Contrib PRs looking for reviews
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1224>
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1221>
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1219>
 - [nev] web sandbox
 - Last automerge broke script because of a rename
 - [mwear] Expo histogram progress?
 - 15m triage bugs
 - [Untriaged bugs](#)

- [Untriaged contrib bugs](#)

2022-10-05

Attendees

- Daniel Dyla (Dynatrace)
- Hector Hernandez (Microsoft)
- Jackson Weber (Microsoft)
- Matt Wear (Lightstep)
- Nev Wylie (Microsoft)
- Svetlana Brennan (New Relic)
- Rauno Viskus (Splunk)

Agenda

- Metrics GA
 - Still just waiting on TC review
- After GA
 - In progress
 - Prometheus exporter resource support - in progress (marc)
 - Todo
 - High resolution histogram (exponential) - dan started on a prototype for this
 - Exemplars
 - Waiting on spec
 - Drop unused attributes (fixes potential memory leak concern)
- API merge to main repo
 - ~~○ Clean up repo files~~
 - ~~○ [Add back to lerna](#)~~
 - ~~■ Webworker tests currently failing~~
 - ~~■ Add API to auto-published docs~~
 - Automate versioning
 - Experimental, sdk and experimental, all 3
 - Migrate all issues from API repo
 - Archive API repo
 - Test old node versions on API only
- Timing issues with anchored clock
 - Fixed in express
 - Same problem with XHR
 - <https://github.com/open-telemetry/opentelemetry-js/blob/610808d3b64b9f660f4dd640ad961b8c9f67be66/experimental/packages/opentelemetry-instrumentation-http-http-request/src/xhr.ts#L434-L441>
 - Dan is working on this
 - Created a discussion issue here for the general problem
 - <https://github.com/open-telemetry/opentelemetry-js/issues/3279>

- Proposed solution: Just use Date
- [marc] Demo Application SIG is looking for developers to add metrics to a JS service
 - <https://github.com/open-telemetry/opentelemetry-demo/issues/68>
 - Nev: Web js not on the demo app 1.0 roadmap
- [dan] is anyone working on or planning on working on logs/events SDK/exporter
 - Martin is out this and last week. Not full time on otel. Nev thinks he might be working on something
- [dan] proto update 0.19.0
- [mwear] mongoose and dataloader have deps on the underlying packages for types
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-09-28

Attendees

- Daniel Dyla (Dynatrace)
- Marc Pichler (Dynatrace)
- Nev Wylie (Microsoft)
- Matt Wear (Lightstep)
- Jackson Weber (Microsoft)
- Svetlana Brennan (New Relic)

Agenda

- Metrics GA <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
 - TC review is scheduled
 - <https://github.com/open-telemetry/community/issues/1204>
 - After GA <https://github.com/open-telemetry/opentelemetry-js/milestone/16>
 - Drop unused attributes (fixes potential memory leak concern)
 - High resolution histogram (exponential) - dan started on a prototype for this
 - Exemplars
 - Prometheus exporter - in progress (marc)
- API merge to main repo
 - ~~Clean up repo files~~
 - ~~[Add back to lerna](#)~~
 - ~~Webworker tests currently failing~~
 - ~~Add API to auto published docs~~
 - Automate versioning
 - Experimental, sdk and experimental, all 3
 - Migrate all issues from API repo
 - Archive API repo
 - Test old node versions on API only
- Express timing issues with anchored clock

- <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1210>
- Same problem with XHR
<https://github.com/open-telemetry/opentelemetry-js/blob/610808d3b64b9f660f4dd640ad961b8c9f67be66/experimental/packages/opentelemetry-instrumentation-xml-http-request/src/xhr.ts#L434-L441>
- Created a discussion issue here for the general problem
<https://github.com/open-telemetry/opentelemetry-js/issues/3279>
- RITM singleton in need of more reviews
<https://github.com/open-telemetry/opentelemetry-js/pull/3161>
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-09-21

Attendees

- Daniel Dyla (Dynatrace)
- Jackson Weber (Microsoft)
- Nev Wylie (Microsoft)
- Marc Pichler (Dynatrace)
- Rauno Viskus (Splunk)
- Svetlana Brennan (New Relic)
- Weyert de Boer (Tapico)
- Hector Hernandez (Microsoft)
- Jamie Danielson (Honeycomb)

Agenda

- Sandbox CLA Issue?
- 1.7.0 and 0.33.0 release
 - Likely last release before metrics GA
- Metrics GA <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
 - TC review (issue link here)
 - After GA <https://github.com/open-telemetry/opentelemetry-js/milestone/16>
 - Drop unused attributes (fixes potential memory leak concern)
 - High resolution histogram (exponential)
 - Exemplars
 - Prometheus exporter
- [Proto release 0.19.0](#)
 - Remove InstrumentationLibrary* fields
 - Add scope attributes
 - Add partial success response
- API merge to main repo

- ~~Clean up repo files~~
- [Add back to lerna](#)
 - Webworker tests currently failing
 - Add API to auto published docs
- Automate versioning
 - Experimental, sdk and experimental, all 3
- Migrate all issues from API repo
- Archive API repo
- Test old node versions on API only
- [feat\(instrumentation\): implement require-in-the-middle singleton](#)
 - All contrib tests pass except aws lambda because of the absolute file path intercept
- [svetlana]: <https://github.com/open-telemetry/opentelemetry-js/pull/3143>
 - Question on NoopSpanProcessor
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-09-14

Attendees

- Daniel Dyla (Dynatrace)
- Hector Hernandez (Microsoft)
- Jackson Weber (Microsoft)
- Rauno Viskus (Splunk)
- Nev Wylie (Microsoft)
- Purvi Kanal (Honeycomb)
- Jamie Danielson (Honeycomb)
- Matt Wear (Lightstep)
- Marc Pichler (Dynatrace)
- Svetlana Brennan (New Relic)

Agenda

- Metrics GA <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
 - 2 issues left - documentation and 1 maybe not blocking
- API merged to main repo - next steps - <https://github.com/open-telemetry/opentelemetry-js/tree/main/api>
 - Clean up repo files
 - Add back to lerna
 - Automate versioning
 - Experimental, sdk and experimental, all 3
 - Migrate all issues from API repo
 - Archive API repo
 - Test old node versions on API only

- Add API to auto published docs
- Logs/Events API merged [Logs/events API](#)
 - Should probably release
 - Wait for metrics GA? no
- [Santosh] Removing process.runtime.* attributes from BrowserDetector
 - Ref: https://github.com/open-telemetry/opentelemetry-js/pull/3201#discussion_r969099001
- [feat\(instrumentation\): implement require-in-the-middle singleton](#)
- Sandbox PR needs additional review:
 - [chore: Remove API Repo from auto-merge and add Js Contrib Repo by MSNev · Pull Request #5 · open-telemetry/opentelemetry-sandbox-web-js \(github.com\)](#) – Merged
 - New Auto Merge PR: [\[AutoMerge\] Merging change\(s\) from otel-js: otel-js-contrib: by opentelemetrybot · Pull Request #6 · open-telemetry/opentelemetry-sandbox-web-js \(github.com\)](#)
- 15m triage bugs
 - [Untriaged bugs](#)
 - [Untriaged contrib bugs](#)

2022-09-07

Attendees

- Daniel Dyla (Dynatrace)
- Marc Pichler (Dynatrace)
- Amir Blum (Aspecto)
- Nev Wylie (Microsoft)
- Jackson Weber (Microsoft)
- Matt Wear (Lightstep)
- Hector Hernandez (Microsoft)
- Jamie Danielson (Honeycomb)

Agenda

- Rauno bano back from vacation
- Metrics GA:
 - [docs\(node-sdk\): update outdated metric documentation #3226](#)
 - [feat\(sdk-metrics\): align MetricReader with specification and other language implementations](#)
 - [feat\(opentelemetry-api-metrics\): Adding generics to create{metricType} #3170](#)
- Bring API into main repo
 - <https://github.com/open-telemetry/opentelemetry-js/issues/3221>
 - No objections on the issue. Dan will do this this week
 - Flarna mentioned he thinks the API should not be released with each SDK release. This is achievable fairly easily but we need to make sure

- This part of the sandbox merge does the git merge [opentelemetry-sandbox-web-js/syncReposToStaging.ts at main · open-telemetry/opentelemetry-sandbox-web-js \(github.com\)](#) from the remote branch which is added earlier in the script and the readme and image provides a guide for what it's doing [opentelemetry-sandbox-web-js/sandbox-tools/merge-repos at main · open-telemetry/opentelemetry-sandbox-web-js \(github.com\)](#)
- Logs API test failure
 - Dan and Marc are unable to reproduce locally
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3231> might have fixed this? Just reran the tests to see
- Contrib test failures
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1137> only reviewed by 1 person. Will merge if nobody objects but it touches a lot of files so more eyes would be good
- [Finalize metrics API](#)
 - Metrics API depends on API
 - Context
 - Resource
 - Diag
- [Nev] Web Sandbox PR still blocked:
 - [\[AutoMerge\] Merging change\(s\) from otel-js-api: otel-js: by opentelemetrybot · Pull Request #4 · open-telemetry/opentelemetry-sandbox-web-js \(github.com\)](#)
 - Historical commit with CLA failure
 - Chris said we can disable the check for the initial merge but he doesn't want that to be a long-term solution. CLA check should stay enabled whenever possible.
 - Is it possible to rewrite the history of the sandbox to remove the offending commit?
 - Will need to investigate – But probably not without breaking something :-)
 - Since we are going to merge back the API repo I'm going to pause this as I'll need to rework the merge script anyway.
- 15m triage bugs
 - [Untriaged bugs](#)

2022-08-31

Attendees

- Daniel Dyla (Dynatrace)
- Marc Pichler (Dynatrace)
- Nev Wylie (Microsoft)
- Jackson Weber (Microsoft)
- Svetlana Brennan (New Relic)

- Amir Blum (Aspecto)
- Martin Kuba (New Relic)
- Jamie Danielson (Honeycomb)
- Helio Frota (Red Hat)

Agenda

- Commit in history without CLA authorization
 - CNCF contacted with no response yet
 - Blocking the web sandbox
 - Possible solutions
 - Rewrite history - very impactful
 - Disable CLA check in sandbox repo just for this single merge
 - Dan will reach out to ayme at CNCF directly to ask for guidance
- [marc] We need to finalize the metrics API package.
 - Possibly integrate it into the API package, but we'll need to consider bundle size.
 - Issue: <https://github.com/open-telemetry/opentelemetry-js/issues/3217>
 - Option 1: separate API packages per-signal
 - Option 2: integrate metrics API into the main API package
- [dan] consider moving API back into main repo
 - Dan will create an issue and tag the other maintainers
- PR opened for instrumenting restricted fastly compute@edge environments
 - PR closed in favor of <https://github.com/fastly/compute-js-opentelemetry>
- Metrics GA <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
 - [feat\(sdk-metrics-base\): per metric-reader aggregation](#)
- Contrib test failures
 - [ci: \[Bug\] Web based karma tests are no longer running #1135](#)
 - [ci: \[Bug\] DNS Lookup and Promise Lookup test for error NOT_FOUND failing #1136](#)
- [martin] [Logs API PR](#)
- [amir] upstreamed mongoose instrumentation please review
- [svetlana] triage question -
 - <https://github.com/open-telemetry/opentelemetry-js/issues/568>
 -
- 15m triage bugs
 - [Untriaged bugs](#)

2022-08-24

Attendees

- Daniel Dyla (Dynatrace)
- Jackson Weber (Microsoft)
- Amir Blum (Aspecto)
- Nev Wylie (Microsoft)

- Matt Wear (Lightstep)
- Marc Pichler (Dynatrace)

Agenda

- Welcome Svetlana Brennan to approvers!
- Instrumentation supported node engines
 - https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1122#discussion_r951837079
- PR opened for instrumenting restricted fastly compute@edge environments
 - [feat: add fastly instrumentation lib](#)
 - PA will be joining next week to discuss the work
- SDK release [chore: proposal 1.6.0/experimental 0.32.0](#)
- Metrics GA <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
 - Still reviewing [feat\(sdk-metrics-base\): per metric-reader aggregation](#)
 - Still working on docs
 - New PR to reduce internal exports from metrics SDK
<https://github.com/open-telemetry/opentelemetry-js/pull/3197>
- [Logs/Events prototype](#)
 - Still under review
- Rate limiting still waiting on CNCF to contact github
- Sandbox CLA Issue outcome?
- 15m triage bugs
 - [Untriaged bugs](#)

2022-08-17

Attendees

- Nev wylie (Microsoft)
- Amir Blum (Aspecto)
- Jackson Weber (Microsoft)
- Daniel Dyla (Dynatrace)
- Hector Hernandez (Microsoft)
- Martin Kuba (New Relic)
- Santosh Cheler (Cisco / AppDynamics)
- Marc Pichler (Dynatrace)
- Matt Wear (Lightstep)
- Purvi Kanal (Honeycomb)
- Svetlana Brennan (New Relic)
- Abinet Debele (Cisco / AppDynamics)
- Helio Frota (Red Hat)

Agenda

- Please add any agenda items
- (Santosh) Criteria for a package to move out of experiment stage

- [dan] API 1.2.0 release
- [dan] contrib release
- [dan] stuck on updating web test dependencies
 - [deps: update web dev deps](#)
 - Cannot even run tests locally on macos (some sort of permission issue)
- Metrics GA <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
 - Michael Hausenblas from AWS has volunteered to help and I believe is going to start with docs/examples on the website
 - Last large body of work required for GA [feat\(sdk-metrics-base\): per metric-reader aggregation](#)
- Metrics base renamed to sdk-metrics
- [Logs/Events prototype](#)
 - Still under review
- Build on windows merged and passing
 - https://github.com/open-telemetry/opentelemetry-js/runs/7791131124?check_suite_focus=true
- [amir] Should we come up with standart for type.ts in instrumentations (only public types?)
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1097>
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/blob/main/plugins/node/opentelemetry-instrumentation-graphql/src/types.ts>
- Rate limiting
 - Chris Aniszczyk asked Jeffrey Sica to look into raising limits for us
- [Abinet] Batch Span processor import issue. Only node batch span processor is exported , unable to import the Browser related span processor in browser app
- 15m triage bugs
 - [Untriaged bugs](#)
 - getGlobal Helper [microsoft/ApplicationInsights-JS: Microsoft Application Insights SDK for JavaScript \(github.com\)](#)

2022-08-10

Attendees

- Nev Wylie (Microsoft)
- Daniel Dyla (Dynatrace)
- Amir Blum (Aspecto)
- Jackson Weber (Microsoft)
- Hector Hernandez (Microsoft)
- Matt Wear (Lightstep)
- Jamie Danielson (Honeycomb)
- Helio Frota (Red Hat)
-

Agenda

- Please add any agenda items
- Releases
 - Stable 1.5.0
 - Experimental 0.31.0
 - Contrib (multiple releases)
 - [Contrib release](#) on 1.5.0 and 0.31.0 packages pending
 - [chore: release 1.2.0](#)
- Metrics GA <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
- [Browser support for protobuf exporters](#)
 - There are some other packages that will require browser targets (exporter base packages) but they don't have node specific behavior
- Rename metrics base
 - <https://github.com/open-telemetry/opentelemetry-js/issues/3137>
- [ValueType.INT metric instrument integer overflow](#)
- [Logs/Events prototype](#)
 - Now ready for review
- [Windows build target](#)
 - Test run failing in CI
 - Pass locally
 - https://github.com/open-telemetry/opentelemetry-js/runs/7769857294?check_suite_focus=true
- Contrib rate limiting for releases
 - Created a ticket with CNCF for this CNCFSD-1366
 - No reply yet
- [fix: avoid type imports of the aws-sdk package in the built assets](#)
 - Created a CNCF ticket CNCFSD-1369 to ensure we're handling this properly
- 15m triage bugs
 - [Untriaged bugs](#)

2022-08-03

Attendees

- Daniel Dyla (Dynatrace)
- Jackson Weber (Microsoft)
- Hector Hernandez (Microsoft)
- Martin Kuba (New Relic)
- Matt Wear (Lightstep)
- Nev Wylie (Microsoft)
- Connor Lindsey (Grafana)
- Aaron Abbott (Google)
- Jamie Danielson (Honeycomb)
- Helio Frota (Red Hat)

Agenda

- Please add any agenda items
- Metrics GA <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
 - Some metrics topics (prometheus) may actually not be truly required for GA
- OTLP exporters no longer depend on their JSON counterparts. Trace exporters can be released soon.
- [Browser support for protobuf exporters](#)
 - Pkanal volunteered for this last week
- Call for opinions [Rename @opentelemetry/sdk-metrics-base to @opentelemetry/sdk-metrics?](#)
- [ValueType.INT metric instrument integer overflow](#)
- [Logs/Events prototype](#)
 - request for feedback
 - not sure why tests are timing out
- [Windows build failures](#) merged
- Please review [Prevent double wrapping event emitter listeners](#) p1 bug
- Missing Semantic Conventions for Metrics, missing YAML files in specification repo.
 - [Metrics Semantic Conventions not specified as YAML files · Issue #2666 · open-telemetry/opentelemetry-specification \(github.com\)](#)
- Contrib rate limiting for releases - any update?
- 15m triage bugs
 - [Untriaged bugs](#)

2022-07-27

Attendees

- Ptah Dunbar (New Relic)
- Martin Kuba
- Nev Wylie (Microsoft)
- Svetlana Brennan
- Pruvi Kanal

Agenda

- Maintainer Vacations

2022-07-20

Attendees

- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Marc Pichler (Dynatrace)
- Helio Frota (Red Hat)
- Jamie Danielson (Honeycomb)

- Hector Hernandez (Microsoft)
- Jackson Weber (Microsoft)

Agenda

- Maintainer Vacations
 - Dan July 27
 - Amir July 20 and 27
 - Rauno July 27 and Sept 3
- Meeting next week (July 27) canceled due to lack of maintainer attendance
- Metrics GA <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
- OTLP/JSON short keys closed
 - Combination of protobuf and compression options made this not worth the effort
- proto/http exporter now uses compiled js instead of loading proto files
 - Native webpack support (just js files)
 - Not yet tested in browser but it should work (sendWithHttp supports browser and protobufjs supports browser). Just need to add tests to be sure
- Contrib rate limiting
 - No update
- Add your items here:
- ...
- [Santosh] Review Events and Logs Api spec PR - <https://github.com/open-telemetry/opentelemetry-specification/pull/2676>
- [nev] sandbox
 - Expect to have a sandbox PR to automerge the js/api repos into a staging folder in sandbox to be moved into the final packaged sandbox
 - Example Merge pr [\[AutoMerge\] Merging change\(s\) from otel-js-api: otel-js: by MSNev · Pull Request #40 · MSNev/opentelemetry-sandbox-web-js \(github.com\)](#)
 - Repo-staging branch (created by the merge script and the PR above): [MSNev/opentelemetry-sandbox-web-js at auto-merge/repo-staging \(github.com\)](#)
 - Merge Script; [MSNev/opentelemetry-sandbox-web-js at MSNev/AutoMerge \(github.com\)](#)
 - Dan: related note about possibly moving the api package back into the main repo
- [richard simpson] cloudflare workers <https://github.com/RichiCoder1/opentelemetry-sdk-workers>
 - No global state in cloudflare workers
 - No performance API
 - Implements OTLP/HTTP for json and protobuf
- [nev] additional msft members
 - Hector and jackson
- 15m triage bugs
 - [Untriaged bugs](#)
- Bug assignment

- [High priority bugs](#)

2022-07-13

Attendees

- Daniel Dyla (Dynatrace)
- Rauno Viskus (Splunk)
- Nev Wylie (Microsoft)
- Marc Pichler (Dynatrace)
- Matt Wear (Lightstep)
- Martin Kuba (New Relic)
- Jamie Danielson (Honeycomb)

Agenda

- [dan] out july 27
 - Amir gone next week (july 20) and the week after maybe (july 27)
 - Rauno out 25th of july to 2nd of september
- [Complaint about 1.4.0](#)
- Metrics GA <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
- [dan] API attach/detach methods
 - [Return detach callback proposal](#)
- <https://github.com/open-telemetry/opentelemetry-js-api/pull/163/files>
 - Current will be renamed to active per <https://github.com/open-telemetry/opentelemetry-js-api/issues/162#issuecomment-1182173718>
- Metrics PRs need reviews
 - [feat\(sdk-metrics-base\): distinguish between Sum and Gauge in MetricData](#)
 - [fix\(histogram\): fix maximum when only values < -1 are provided](#)
 - [feature\(views\): move views registration to MeterProvider constructor](#)
- Short keys for OTLP/JSON:
<https://github.com/open-telemetry/opentelemetry-proto/pull/413>
- ...
- Add your items here
- [nev] sandbox
 - Still going
 - Trying to create a github workflow which will automatically merge otel js into the sandbox
 - Using an otel bot that works locally but fails in github
- Contrib rate limiting
 - Dan will look into what tier the CNCF/otel github is on and see if there are options to raise the rate limit
- 15m triage bugs
 - [Untriaged bugs](#)
- Bug assignment

- [High priority bugs](#)

2022-07-06

Attendees

- Rauno Viskus (Splunk)
- Amir Blum (Aspecto)
- Nev Wylie (Microsoft)
- Purvi Kanal (Honeycomb)
- Svetlana Brennan (New Relic)
- Marc Pichler (Dynatrace)
- Jamie Danielson (Honeycomb)

Agenda

- [dan] july 27 dan will not be here
- [Rauno56] I Will be away from 25th of July - 2nd of September
- 1.4.0/0.30.0 released today
- Metrics GA <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
 - [feat: export unit block when unit is set in metric descriptor](#) - waiting on author
 - [feat: add InMemoryMetricExporter](#) - waiting on author
- [dan] New form based bug triage flow feedback?
 - Will use the same process in contrib and link to the bug document from the contrib contributing.md
- [React native support feature request](#)
 - Dan will create a PR for a feature lifecycle doc where we can discuss ideas
 - Cutoff date?
 - Stale flow?
 - Label backlog issues?
 - Pick out specific issues for a release milestone?
 - This is not a good issue to begin closing since it is so popular
 -
- [dan] API attach/detach methods
 - <https://github.com/open-telemetry/opentelemetry-js-api/pull/123>
 - Ruby did something similar for similar reasons - no direct feedback positive or negative
- [dan] <https://github.com/open-telemetry/opentelemetry-js-api/pull/163>
- [marc] [feature\(views\): move views registration to MeterProvider constructor](#)
 - Looking for opinions
 - Decision before Metrics GA.
- 15m triage bugs
 - [Untriaged bugs](#)
- Bug assignment
 - [High priority bugs](#)
 - ~~<https://github.com/open-telemetry/opentelemetry-js/pull/2988>~~ - abandoned
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3076>

- <https://github.com/open-telemetry/opentelemetry-js/pull/3054>
- <https://github.com/open-telemetry/opentelemetry-js/pull/3068>
- Sandbox update
 - Still working on automating the merging (with history) from JS/API -> Sandbox
 - Resolved issues related to multiple renames / deletion of files
 - Hitting issue where during the above git is prompting for “commit” details (i.e. opening editor even with –no-edit :-()
- [mwear] Thoughts on short keys for OTLP/JSON: <https://github.com/open-telemetry/opentelemetry-proto/pull/413>
- [pkanal] <https://github.com/open-telemetry/opentelemetry-js/issues/2786>
- [pkanal] rate limited GitHub API actions, would funding help?

2022-06-29

Attendees

- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Marc Pichler (Dynatrace)
- Amir Blum (Aspecto)
- Aaron Abbott (Google)
- Svetlana Brennan (New Relic)
- Martin Kuba (New Relic)
- Matt Wear (Lightstep)
- Jamie Danielson (Honeycomb)

Agenda:

- [1.3.0 / 0.30.0 release](#)
- Metrics GA <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
 - [feat: export unit block when unit is set in metric descriptor](#)
 - [feat: add InMemoryMetricExporter](#)
 - Forceflush vs reset
 - Force flush was spec'd a while ago, but we're missing it in our span exporter
<https://github.com/open-telemetry/opentelemetry-specification/issues/1451>
 - Related OTel JS issue: [Implement SpanExporter#forceFlush](#)
 - Could be a metric reader to avoid buffering/resetting
<https://github.com/open-telemetry/opentelemetry-specification/issues/2425>
- Request for review of Events/Logs API OTEP
 - <https://github.com/open-telemetry/oteps/pull/202>
- New bug triage workflow
 - [Documentation](#)

- [Template](#)
- 15m triage bugs
 - [Untriaged bugs](#)
- Bug assignment
 - [High priority bugs](#)
 - ~~<https://github.com/open-telemetry/opentelemetry-js/pull/2988>~~-abandoned
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3054>
- PRs awaiting reviews:
 - [fix\(instrumentation-http\): add http.host attribute before sending the request](#)
 - [Fix docs deployment](#)
 - [feat: export unit block when unit is set in metric descriptor](#)
 - [feat: add InMemoryMetricExporter](#)
 - [feat\(metrics-api\): use common attributes definitions](#)
 - [feat: allow extending BasicTracerProvider with custom registered components](#)
 - [fix\(fetch\): pass spanResponse to endSpan](#)
 - [Fixes #2984 instrumentation-grpc: Fix grpc-js success status code](#)
 - Grpc status span attribute was being set to 1 (should be 0 on success)
 - [aaron] [Add or change resource detection to be synchronous · Issue #2912 · open-telemetry/opentelemetry-js · GitHub](#)
 - Alternative draft prototype: <https://github.com/open-telemetry/opentelemetry-js/pull/2933>
 - [dchambers] [feat: support graphql v16 #998](#)
 - [valentin] [feat\(instrumentation\): add support for esm module](#)
- Sandbox update
 - In-progress PR: * [Initial creation by MSNev · Pull Request #1 · MSNev/opentelemetry-sandbox-web-js \(github.com\)](#) (beware that I am using force push for changes as this is WIP)
 - Its having issues running in git vs locally and the linting differences between API and JS repo's
 - Sync'd with Daniel yesterday and changing tack slightly (as this is 637 files) to try and "merge" with history and then move into this structure.

2022-06-22

Attendees

- Daniel Dyla (Dynatrace)
- Keith Miller (Mediavine)
- Aaron Abbott (Google)
- Matt Wear (Lightstep)
- Marc Pichler (Dynatrace)
- Nev Wylie (Microsoft)
- Martin Kuba (New Relic)

Agenda

- Metrics GA
 - <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
 - [feat\(sdk-metrics-base\): detect resets on async metrics](#)
 - [feat\(sdk-metrics-base\): implement min/max recording for Histograms](#)
 - [feat: export unit block when unit is set in metric descriptor](#)
 - [feat: add InMemoryMetricExporter](#)
- Request for review of Events/Logs API OTEP
 - <https://github.com/open-telemetry/oteps/pull/202>
- Dropping old node runtimes
 - <https://github.com/open-telemetry/opentelemetry-js/pull/3048>
- Async resource detection with sync init
 - [feat: allow Resource to be created with some attributes resolving asynchronously](#)
 - This is a breaking change
 - OTEP: Ephemeral Resource Attributes
 - <https://github.com/open-telemetry/oteps/pull/208>
- [marc] fetch instrumentation
 - https://github.com/open-telemetry/opentelemetry-js/pull/3037#discussion_r901742999
- [mwear] type dependencies and the metapackage:
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1017>
- [nev] Sandbox status
 - Still working to get the tests running (attempting to get every package to run both node and browser tests)
 - Not using learna – using rush instead
 - Attempted to use karma-typescript rather than webpack for browser tests – but failed because of the way the platform/[node/browser] works – where karma-typescript doesn't look at the package.json for the package to change the implementation
 - Currently attempting to get rollup to generate “bundles” for size comparisons and potentially size unit tests.
- [martin] plan for logs / question
- [nev] FYI [\[Bug\] The definition of SpanAttributeValue \(and AttributeValue\) does not conform to the proto KeyValue specification · Issue #161 · open-telemetry/opentelemetry-js-api \(github.com\)](#)
 - [\[Common\] Spec inconsistency with proto definition of Attributes #2581 by MSNev · Pull Request #2620 · open-telemetry/opentelemetry-specification \(github.com\)](#)
- New PRs:
 - [fix\(instrumentation-http\): add http.host attribute before sending the request](#)
 - [Add node 18 to tests and remove EoL node versions](#)
 - [Fix gzip compression for OTLP HTTP exporter](#)
 - [Fix docs deployment](#)
 - [feat: export unit block when unit is set in metric descriptor](#)
 - [feat: add InMemoryMetricExporter](#)
 - [feat\(metrics-api\): use common attributes definitions](#)

- Old PRs awaiting reviews:
 - [feat: allow extending BasicTracerProvider with custom registered components](#)
 - [feat\(sdk-metrics-base\): implement min/max recording for Histograms](#)
 - [fix\(fetch\): pass spanResponse to _endSpan](#)
 - [fix\(instrumentation\): use all provided filepatches](#)
 - [Fixes #2984 instrumentation-grpc: Fix grpc-js success status code](#)
 - Grpc status span attribute was being set to 1 (should be 0 on success)
 - [aaron] [Add or change resource detection to be synchronous · Issue #2912 · open-telemetry/opentelemetry-js · GitHub](#)
 - Alternative draft prototype: <https://github.com/open-telemetry/opentelemetry-js/pull/2933>
 - [dchambers] [feat: support graphql v16 #998](#)
 - [valentin] [feat\(instrumentation\): add support for esm module](#)
 - [feat\(sdk-metrics-base\): detect resets on async metrics](#)

2022-06-15

Attendees

- Daniel Dyla (Dynatrace)
- Amir Blum (Aspecto)
- Nev Wylie (Microsoft)
- Svetlana Brennan (New Relic)
- Matt Wear (Lightstep)
- Marc Pichler (Dynatrace)
- Jamie Danielson (Honeycomb)
-

Agenda

- Otel community day on Monday
 - <https://events.linuxfoundation.org/open-telemetry-community-day>
- Current topics of focus
 - Metrics GA - <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
 - Documenting metrics
- Contrib release automation failure
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/actions/runs/2462785069>
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/1043>
 - <https://github.com/google-github-actions/release-please-action/issues/506>
 - <https://github.com/googleapis/release-please/blob/main/docs/manifest-releaser.md>
- Metrics GA update
 - <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
 - [Document latest Metrics SDK](#)

- [feat\(sdk-metrics-base\): detect resets on async metrics](#)
 - [feat\(sdk-metrics-base\): implement min/max recording for Histograms](#)
- User experience improvements
 - SDK setup
 - Global registration implicitness
 - [Make Singleton/Registration explicit](#)
 - Async resources make the setup experience suboptimal
 - Work to improve this is started but not finished ([issue](#), [draft PR](#))
 - There are many packages to install, and it is not always obvious which versions are required
 - Performance enhancements
 - [Auto shutdown in collector-exporter can hurt performance in some browsers](#)
 - Is there some event other than unload we can use to trigger a flush?
 - unload, beforeUnload, visibilityChange (show, hide), pageHide/pageShow (iOS)
 - [\["@opentelemetry/instrumentation"\] `require` performance grows linearly with each instrumentation plugin · Issue #3029](#)
 - Instrumentation authorship
 - The instrumentation module dependencies often cause breaking changes in instrumentations
- Drop old node runtimes
 - Makes sense to do this for metrics before they are released as stable
- [martin] Request for review of Events/Logs API OTEP
 - <https://github.com/open-telemetry/oteps/pull/202>
- New Issues:
 - [Implement `InMemoryMetricsExporter` · Issue #3035 · open-telemetry/opentelemetry-js · GitHub](#)
 - [\["@opentelemetry/instrumentation"\] `require` performance grows linearly with each instrumentation plugin · Issue #3029](#)
 - [Using ZoneContextManager and FetchInstrumentation in browser causes lots of leaked Promises · Issue #3030 · open-telemetry/opentelemetry-js · GitHub](#)
 - [Missing http.host attribute on timed out requests](#)
 - [\[opentelemetry-exporter-prometheus\] Export PrometheusSerializer · Issue #3033](#)
 - [Implement `ConsoleMetricsExporter` · Issue #3036 · open-telemetry/opentelemetry-js · GitHub](#)
 - [Nev] Oddities identified as part of the initial creation of Sandbox structure
 - a ReadableSpan is actually not a Span, and the Sdk requires Sdk "Trace" implementations rather than api.Trace for some arguments.
 - As part of manually copying over the Core API/SDK I've changed this, but should I?
 - [rauno56] Maybe a side-effect of us only focusing on the single SDK and probably something to look into to make it compatible with API - An issue or a PR maybe to host the discussion?

- New PRs:
 - [feat: allow extending BasicTracerProvider with custom registered components](#)
 - [feat\(sdk-metrics-base\): implement min/max recording for Histograms](#)
 - [feat\(opentelemetry-exporter-prometheus\): export PrometheusSerializer](#)
 - [fix\(fetch\): pass spanResponse to _endSpan](#)
- Old PRs awaiting reviews:
 - [fix\(instrumentation\): use all provided filepatches](#)
 - [Fixes #2984 instrumentation-grpc: Fix grpc-js success status code](#)
 - Grpc status span attribute was being set to 1 (should be 0 on success)
 - [aaron] [Add or change resource detection to be synchronous · Issue #2912 · open-telemetry/opentelemetry-js · GitHub](#)
 - Alternative draft prototype: <https://github.com/open-telemetry/opentelemetry-js/pull/2933>
 - [dchambers] [feat: support graphql v16 #998](#)
 - [valentin] [feat\(instrumentation\): add support for esm module](#)
 - [feat\(sdk-metrics-base\): detect resets on async metrics](#)

2022-06-08

Attendees

- Nev Wylie (Microsoft)
- Daniel Dyla (Dynatrace)
- Marc Pichler (Dynatrace)
- Amir Blum (Aspecto)
- Rauno Viskus (Splunk)
- Jamie Danielson (Honeycomb)
- Matt Wear (Lightstep)
- Purvi Kanal (Honeycomb)

Agenda

- [marc]: [Browser tracing without Metrics](#)
 - Possibly related? [Add sideEffects: false to package.json for packages to enable tree shaking](#)
 - Looking for a way forward. Should we split up otlp-transformer, or go with tree-shaking?
- [rauno] (if present): [feat: allow extending BasicTracerProvider with custom registered components](#)
- Bug from last week: AsyncHooksContextManager breaks event emitters
 - Issue: <https://github.com/open-telemetry/opentelemetry-js/issues/2971>
 - Reproduction: <https://github.com/open-telemetry/opentelemetry-js/pull/2970>
 - No update

- Metrics RC cleanup releases (latest is now 1.3.1 and 0.29.2)
- [chore: update core dependencies stable ^1.3.1 experimental ^0.29.2](#) merged/released
- Metrics GA update
 - <https://github.com/open-telemetry/opentelemetry-js/milestone/15>
 - [Document latest Metrics SDK](#)
- [dan] changelog format?
- New PRs:
 - ~~[fix\(grpc-exporter\): use non-normalized URL to determine channel security](#)~~
 - ~~[fix\(otlp-transformer\): remove type dependency on Long](#)~~
 - ~~[fix: remove aws and gcp detector from SDK](#)~~
- Old PRs awaiting reviews:
 - [fix\(instrumentation\): use all provided filepatches](#)
 - [Fixes #2984 instrumentation-grpc: Fix grpc-js success status code](#)
 - Grpc status span attribute was being set to 1 (should be 0 on success)
 - [aaron] [Add or change resource detection to be synchronous · Issue #2912 · open-telemetry/opentelemetry-js · GitHub](#)
 - Alternative draft prototype: <https://github.com/open-telemetry/opentelemetry-js/pull/2933>
 - [dchambers] [feat: support graphql v16 #998](#)
 - [valentin] [feat\(instrumentation\): add support for esm module](#)
 - [fix\(sdk-trace-base\): fix crash on environments without global document](#)
 - [feat: record exceptions in http instrumentation](#)
 - [feat\(sdk-metrics-base\): detect resets on async metrics](#)
- Sandbox Repo: [REQUEST: Create a new repository opentelemetry-sandbox-web-js · Issue #1068 · open-telemetry/community \(github.com\)](#)

2022-06-01

Attendees

- Daniel Dyla (Dynatrace)
- Rauno Viskus (Splunk)
- Amir Blum (Aspecto)
- Marc Pichler (Dynatrace)
- Matt Wear (Lightstep)
- Jamie Danielson (Honeycomb)
- Nev Wylie (Microsoft)

Agenda

- Community Day: [Open Telemetry Community Day | Linux Foundation Events](#)
- Metrics RC released last week
 - [V1.3.0](#) / [experimental/v0.29.0](#)
 - Created [metrics GA milestone](#) to track issues required for official 1.0 release
 - Prometheus exporter still needs some changes for spec compliance

- [Min/Max not implemented for histogram \(required by spec\)](#)
- Release experimental hotfix
 - [fix\(otlp-transformer\): add missing prepublishOnly script #2998](#)
 - [fix\(resources\): fix browser compatibility for host and os detectors #3004](#)
 - [Hotfix release](#) (please review)
- Bug from last week: AsyncHooksContextManager breaks event emitters
 - Issue: <https://github.com/open-telemetry/opentelemetry-js/issues/2971>
 - Reproduction: <https://github.com/open-telemetry/opentelemetry-js/pull/2970>
- Contrib CI is failing
 - Possibly fixed by <https://github.com/open-telemetry/opentelemetry-js/pull/3004> included in hotfix mentioned above
 - Action item: change exports in SDK packages to be interfaces/types instead of class exports. Amir will create an issue
- Flaky tests and codecov +/- 1% issue might be fixed
 - Please bring this up if you notice it still happening
 - https://github.com/open-telemetry/opentelemetry-js/pull/3007/checks?check_run_id=6692659699 :(
- Request: [Add ARM64 Support](#)
- If aaron present: async resources
- [martin] Events API OTEP - call for review
 - [Events API OTEP](#)
- [martin] Experimental Events API
 - [prototype](#)
 - Could we work towards getting this merged as experimental without spec finalized?
- [nev] Sandbox repo for creating experimental Web JS packages
 - This is intended to be for experimental generation (and publishing) of short-lived (will not be long term maintained) packages and will not be supported for production.
 - I would like to have the package names and maybe the repo defined with a name that highlights this experimental nature.
@opentelemetry/opentelemetry-experimental-web-js (or something like that)
 - And this will need a different set of permissions to that of the main repo (different maintainers etc), with the intent of this repo providing a sandbox where any code improvements will then be migrated to the main repos (or we create a new web specific js repo – will depend on how the experiments go)
 - <https://github.com/microsoft/ApplicationInsights-JS/tree/master/AISKU>
 - <https://github.com/microsoft/DynamicProto-JS> - aggressive minification techniques.
- Old PRs awaiting reviews:
 - [fix\(instrumentation\): use all provided filepatches](#)
 - [Fixes #2984 instrumentation-grpc: Fix grpc-js success status code](#)
 - Grpc status span attribute was being set to 1 (should be 0 on success)

- [aaron] [Add or change resource detection to be synchronous · Issue #2912 · open-telemetry/opentelemetry-js · GitHub](#)
 - Alternative draft prototype: <https://github.com/open-telemetry/opentelemetry-js/pull/2933>
- [dchambers] [feat: support graphql v16 #998](#)
- [valentin] [feat\(instrumentation\): add support for esm module](#)
- New PRs:
 - [fix\(sdk-trace-base\): fix crash on environments without global document](#)
 - [feat: record exceptions in http instrumentation](#)
 - [feat\(sdk-metrics-base\): detect resets on async metrics](#)

2022-05-25

Attendees

- Daniel Dyla (Dynatrace)
- Amir Blum (Aspecto)
- Svetlana Brennan (New Relic)
- Nev Wylie (Microsoft)
- Marc Pichler (Dynatrace)
- Matt Wear (Lightstep)
- Trivikram Kamat (Amazon)
- Abhinav Mathur (AppDynamics)
- Andrew Hayworth (Shopify)
- Andy Fleming (Shopify)
- Jamie Danielson (Honeycomb)
- Rauno Viskus (Splunk)

Agenda

-
- Community Day: [Open Telemetry Community Day | Linux Foundation Events](#)
- [andyfleming] Question: What is the state of the browser prototype/client in OTEL-js. How does it differ from the new prototype by the client SIG?
- Contrib release automation failure - release please was being rate limited by the github API for creating too many releases and commenting on the PR too many times
 - Released packages manually
 - Docker resource detector not released because it is marked 1.0 and I think that is incorrect
 - This will happen again in the future any time all packages are updated at once (updating core or api dependencies)
 - [rauno56] Created 2 issues investigating this
 - [Report rate-limiting errors clearly googleapis/release-please#1440](#)
 - [Unhandled promise rejection: HttpError: Server Error google-github-actions/release-please-action#487](#)
- Release core this week

- Dan will create a release PR and please comment on it if there are any issues which block it.
- AsyncHooksContextManager breaks event emitters
 - Issue: <https://github.com/open-telemetry/opentelemetry-js/issues/2971>
 - Reproduction: <https://github.com/open-telemetry/opentelemetry-js/pull/2970>
- Metrics RC
 - Default instrument unit may be a breaking change (/cc andy)?
 - Transformer esm/esnext packaging fix (listed below, marc)
 - [Update Prometheus Exporter with latest Metrics SDK · Issue #2773 · open-telemetry/opentelemetry-js · GitHub](#)
- FYI scope attributes [OTEP](#) merged. JS research issue: [Is it possible to add attributes to "instrumentation scope"?](#)
- New PRs
 - [fix\(instrumentation\): only patch core modules if enabled](#)
 - [fix\(otlp-transformer\): include esm and esnext in package files and update README](#)
 - [fix: remove attributes from OTLPExporterConfigBase](#)
 - [fix\(instrumentation\): use all provided filepatches](#)
 - [test\(fetch\): make prepareData async to enforce fakeTime order](#)
 - Attempt to fix flaky tests
 - [Fixes #2984 instrumentation-grpc: Fix grpc-js success status code](#)
 - Grpc status span attribute was being set to 1 (should be 0 on success)
- Old PRs awaiting reviews:
 - [aaron] [Add or change resource detection to be synchronous · Issue #2912 · open-telemetry/opentelemetry-js · GitHub](#)
 - Alternative draft prototype: <https://github.com/open-telemetry/opentelemetry-js/pull/2933>
 - [dchambers] [feat: support graphql v16 #998](#)
 - [valentin] [feat\(instrumentation\): add support for esm module](#)
- Where should the official instrumentation for Cloud SDKs go to?
 - For AWS, it's in [@opentelemetry/instrumentation-aws-sdk](#)
 - For Azure, it's in [@azure/opentelemetry-instrumentation-azure-sdk](#)
 - For Google Cloud, it's [@google-cloud/opentelemetry-cloud-trace-propagator](#)
 - Relevant question related to AWS SDK on CloudNative Slack #otel-js channel: <https://bit.ly/3PFd5gj>
 - Reach out to @tedsuo on Slack on what should be the official recommendation for library authors from OpenTelemetry.
- Recommendation for handling CJS/ESM code especially if the library being instrumented is dual-published with Isolate state?
 - Thread on CloudNative Slack #otel-js channel: <https://bit.ly/3wLxjxi>
 - The official [Isolate State dual-packaging solution](#) by Node.js suggests using export maps with separate CJS and ESM files.
 - The [InstrumentationNodeModuleFile](#) takes a particular file path. Is there a way for it to select a CJS/ESM file based on the type of the application?

- What should be the next steps? A GitHub issue on opentelemetry-js?
 - ESM loader implementation has been a problem. There was a PR in the past.
 - Probable solution: extend API to accept list of strings. Trivikram to create issue on opentelemetry-js.
- Update bookmarks in #otel-js CloudNative Slack channel, which can help new folks joining the channel?
 - I'd asked a question at <https://bit.ly/3wMhjLD>
 - Add links to good first issues in bookmarks
 - Add link to SIG meeting.

2022-05-18

Cancelled - many of us at the KubeCon!

2022-05-11

Attendees

- ~~New Wylie (Microsoft)~~ - Not able to attend today
- Daniel Dyla (Dynatrace)
- Marc Pichler (Dynatrace)
- Matt Wear (Lightstep)
- Svetlana Brennan (New Relic)
- Amir Blum (Aspecto)
- Jamie Danielson (Honeycomb)

Agenda

- Community:
 - Kubecon
 - Community Day:
 - [Open Telemetry Community Day | Linux Foundation Events](#)
- Open Metrics Issues:
 - [Update Prometheus Exporter with latest Metrics SDK · Issue #2773 · open-telemetry/opentelemetry-js · GitHub](#)
 - [Default Instrument unit does not comply with spec · Issue #2910 · open-telemetry/opentelemetry-js · GitHub](#)
- New issues:
 - [Move Sampler interface to an SDK package?](#)
- [martin] Any plan for implementing Log exporter?

New PRs awaiting reviews:

- [fix\(opentelemetry-instrumentation-http\): handle null ports in options #2948](#)
- [fix\(sdk-metrics-base\): misbehaving aggregation temporality selector tolerance](#)
- [chore\(eslint\): remove airbnb package](#)

Old PRs awaiting reviews:

- [aaron] [Add or change resource detection to be synchronous · Issue #2912 · open-telemetry/opentelemetry-js · GitHub](#)
 - Progress update
- [svetlana] [fix\(otlp-http-exporter\): update endpoint to match spec](#)
- [svetlana] [feat\(trace-otlp-grpc\): configure security with env vars](#)
- [svetlana] [feat\(otlp-exporter\): add timeout env var](#)
- [dchambers] [feat: support graphql v16 #998](#)
- [valentin] [feat\(instrumentation\): add support for esm module](#)
- [amir] [feat\(instrumentation-redis\): add support for redis@^4.0.0](#)

2022-05-04

Attendees

- Daniel Dyla (Dynatrace)
- Aaron Abbott (Google)
- Amir Blum (Aspecto)
- Nev Wylie (Microsoft)
- Rauno Viskus (Splunk)
- Nathan Smith (Elastic)
- Svetlana Brennan (New Relic)
- Andy Fleming (Shopify)
- Marc Pichler (Dynatrace)
- Jamie Danielson (Honeycomb)
- Keith Miller (Mediavine)
- Siim Kallas (Splunk)
- Matt Wear (Lightstep)

Agenda

- [Unable to get current span with trace.getSpan\(\)](#)
- [null port when using ignore matchers in opentelemetry-instrumentation-http](#)
- [Rename InstrumentationLibrary -> InstrumentationScope](#)
 - Required before metrics RC
- [OTLPExporterBrowserBase cannot be invoked without 'new'](#)
- [exporter-trace-otlp-proto cannot communicate with collector version > 0.42](#)
 - <https://github.com/open-telemetry/opentelemetry-collector/releases/tag/v0.43.0>
 - <https://github.com/open-telemetry/opentelemetry-collector-contrib/releases/tag/v0.43.0>
- Drop support for old node versions
<https://github.com/open-telemetry/opentelemetry-js/issues/2721>
 - Dan suggests a policy of EOL + 1 year for the SDK and bumping the minor version when a runtime is dropped.

- API can support much longer as there is less regular development there and less to support.
 - .NET stopped supporting .NET5
<https://github.com/open-telemetry/opentelemetry-dotnet/issues/3147>
 - <https://github.com/open-telemetry/opentelemetry-specification/pull/2510>
- <https://events.linuxfoundation.org/open-telemetry-community-day/>
 - <https://forms.gle/huxhULkrFMN1gzyp7>
- ~~[svetlana] question on [OTEL_TRACES_EXPORTER env var](#)~~
 - Next week
- [aaron] Async resources <https://github.com/open-telemetry/opentelemetry-js/issues/2912>
- ADD AGENDA ITEMS ABOVE THIS LINE
- New PRs
 - [fix\(opentelemetry-instrumentation-http\): handle null ports in options](#)
 - ~~[chore: remove unused InstrumentationConfig#path](#)~~
 - [docs: retrospectively add prometheus counter suffix change](#)
 - [fix\(otlp-http-exporter\): update endpoint to match spec](#)
 - [feat: enable a way to disable amqp tests that require a server running](#)
 - [feat: support graphql v16 #998](#)
 -
- Old PRs waiting for reviews:
 - [siim] [feat: metric aggregation temporality controls](#)
 - Required for Metrics GA
 - [marc] [feat\(exporters\): update proto version and use otlp-transformer](#)
 - Required for Metrics GA
 - [legendecas] [feat\(sdk-metrics-base\): hoist async instrument callback invocations](#)
 - Required for Metrics GA
 - [svetlana] [feat\(trace-otlp-grpc\): configure security with env vars](#)
 - [svetlana] [feat\(otlp-exporter\): add timeout env var](#)
 - [feat\(instrumentation\): add support for esm module](#)
 - [amir] [feat\(instrumentation-redis\): add support for redis@^4.0.0](#)

2022-04-27

Attendees

- Daniel Dyla (Dynatrace)
- Svetlana Brennan (New Relic)
- Marc Pichler (Dynatrace)
- Nathan Smith (Elastic)
- Amir blum (Aspecto)
- Rauno Viskus (Splunk)
- Keith Miller (Mediavine)
- Aaron Abbott (Google)
- Siim Kallas (Splunk)
- Nev Wylie (Microsoft)

Agenda

- End user survey feedback requested:
<https://github.com/open-telemetry/community/issues/1045>
- 1.2.0 / 0.28 release next steps
 - Documentation
 - Examples
 - Website docs (dan)
 - Metrics
 - Required to be useful:
 - Update proto (being done by marc)
 - Required for GA
 - Documentation
 - Examples
 - Spec review (TC)
 - [Define reentrant safety of JavaScript Metrics API callback](#)
 - PR open: [feat\(sdk-metrics-base\): hoist async instrument callback invocations](#)
 - [feat\(sdk-metrics-base\): async instruments callback timeout](#)
 - Blocked on above PR
 - [feat: metric aggregation temporality controls](#)
 - Update prometheus exporter
 - <https://github.com/open-telemetry/opentelemetry-js/issues/2773>
 - After GA
 - Exemplars
- [marc] Anyone else having issues with codecov being inconsistent? (+/-1%)
 - We are currently only testing changed packages in contrib which may confuse codecov
 - May be related to Codecov partial updates
- **Will be dropping support for node 8 and 10**
 - Spec will probably loosen rules around that:
<https://github.com/open-telemetry/opentelemetry-specification/pull/2510>
 - Add testing for node 18 while we are at it
 - Change the support statement to something like “Will support node versions a year past their EOL”
- PRs waiting for reviews:
 - [siim] [feat: metric aggregation temporality controls](#)
 - Please review
 - Required for Metrics GA
 - [marc] [feat\(exporters\): update proto version and use otel-transformer](#)
 - [legendecas] [feat\(sdk-metrics-base\): hoist async instrument callback invocations](#)
 - [svetlana] [feat\(trace-otlp-grpc\): configure security with env vars](#)
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2846>

- [fix\(resources\): extend ResourceAttributes interface to comply with spec](#)
 - New resource detectors [feat: implement OSDetector](#) and [feat: implement HostDetector](#)
 - [amir] [feat\(instrumentation-redis\): add support for redis@^4.0.0](#)
- <https://events.linuxfoundation.org/open-telemetry-community-day/>
 - <https://forms.gle/huxhULkrFMN1gzyp7>
- Issue triage if time

2022-04-20

Attendees

- Nev Wylie (Microsoft)
- Daniel Dyla (Dynatrace)
- Rauno Viskus (Splunk)
- Jamie Danielson (Honeycomb)
- Marc Pichler (Dynatrace)
- Siim Kallas (Splunk)
- Svetlana Brennan (New Relic)
- Matt Wear (Lightstep)
- Amir Blum (Aspecto)
- Andy Fleming (Shopify)

Agenda

- [dan/marc] [OTLP Exporter split/refactor](#)
 - Codecov not running?
 - rauno56: Only for packages that have tests run - lerna tries to be clever about what has changed and what has not. Codecov probably thinks files that don't have their coverage reported are deleted.
- [dan] [Code Coverage paths incorrect](#)
- [svnm] [Agent-like experience with -r](#)
 - Separate -r flags to load instrumentations and the sdk
 - Depend on core set of exporters
 - [Add or change resource detection to be synchronous](#)
- [fix\(propagator-jaeger\):extract 1 digit traceFlag\(0\) return 1](#)
 - Waiting on nit to merge
- [chore: cleanup API dependencies](#)
 - Reviewed by maintainers. Waiting on final comments from author before merge
- [feat: metric aggregation temporality controls](#)
 - Please review
- Input requested from ESM experts [feat\(instrumentation\): add support for esm module](#)
- Needed before release
 - Example of new metrics SDK
 - Prometheus or OTLP
 - Add a view which shows renaming

- One of each metric type
 - This can possibly be reused as an end-to-end or integration test
- Website documentation
- <https://events.linuxfoundation.org/open-telemetry-community-day/>
 - <https://forms.gle/huxhULkrFMN1gzyp7>
- 15m at the end Issue Triage: <https://github.com/open-telemetry/opentelemetry-js/issues?q=is%3Aissue+is%3Aopen+sort%3Aupdated-desc>

2022-04-13

Attendees

- Daniel Dyla (Dynatrace)
- Rauno Viskus (Splunk)
- Amir Blum (Aspecto)
- Nev Wylie (Microsoft)
- Quinn Hanam (AWS)
- Marten Hennoch (Splunk)
- Siim Kallas (Splunk)
- Aaron Abbott (Google)
- Svetlana Brennan (New Relic)
- Jamie Danielson (Honeycomb)
- Andy Fleming (Shopify)
- Matt Wear (Lightstep)

Agenda

- [martinkuba] Browser sessions as resource attributes / question on challenges with updating resources
- Instrumentation stability guidelines merged
<https://github.com/open-telemetry/opentelemetry-specification/pull/2180>
- What is the current status: [feat\(trace-otlp-grpc\): configure security with env vars](#)
- Consider using something like TAV for API in core? Maybe simpler to have a test package for each minor version of the API like we do for the node backcompat tests?
 - API versioning dependency issues - example: sdk-node depends on resource-detector-aws (contrib) which depends on api 1.0.x means hoisted API version is 1.0.x even though all packages reference ^1.0.0 in [this PR](#)
- New PRs:
 - [feat\(sdk-metrics-base\): shutdown and forceflush on MeterProvider](#)
- State of [refactor\(exporters\): introduce packages for shared exporter classes](#)

2022-04-06

Attendees

- Per-Anders “PA” Rangsjö (IKEA)
- Daniel Dyla (Dynatrace)
- Amir Blum (Aspecto)
- Jamie Danielson (Honeycomb)
- Aaron Abbott (Google)
- Svetlana Brennan (New Relic)
- Michal Zajac (Splunk)
- Nev Wylie (Microsoft)
- Chengzhong Wu (Alibaba)
- Marc Pichler (Dynatrace)
- Josephine Liang (Mode)

Agenda

- [Per-Anders] Short presentation of lightweight OTel JS library for Fastly Compute@Edge created by IKEA
 - [aaron] copyright guidelines are here
<https://github.com/open-telemetry/community/blob/main/CONTRIBUTING.md#copyright-notices>
- Instrumentation stability guidelines (affects most contrib packages)
<https://github.com/open-telemetry/opentelemetry-specification/pull/2180>
- [marc] <https://github.com/open-telemetry/opentelemetry-js/pull/2871>
- Feature request PR:
 - [allow span names to be customized with a reserved __spanName__ attribute](#)
- Metrics TODO
 - Update otlp transformer for new metric types - marc is working on this
 - Update OTLP metrics exporters (ongoing pichlermarc)
 - [refactor\(sdk-metrics-base\): add ObservableRegistry and hoist invocations](#) (depends on meter shared states)
 - Meter identity (depends on [refactor\(sdk-metrics-base\): meter shared states](#))
 - [AggregationTemporality Controls](#)
- New PRs
 - [fix: sanitize attributes inputs](#)
 - Bug: [fix\(otlp-http-exporter\): remove content length header](#)
- PRs still open from last week (and the week before)
 - [feat\(trace-otlp-grpc\): configure security with env vars](#)
 - [feat\(otlp-exporter\): add timeout env var](#)
 - [refactor\(sdk-metrics-base\): meter shared states](#)
- [svetlana]:
 - Order of precedence: programmatic vs env var
 - [Endpoint issue](#)

2022-03-30

Attendees

- Rauno Viskus (Splunk)
- Daniel Dyla (Dynatrace)
- Amir Blum (Aspecto)
- Andy Fleming (Shopify)
- Siim Kallas (Splunk)
- Nev wylie (Microsoft)
- Matt Wear (Lightstep)
- Josephine Liang (Mode)
- Marc Pichler (Dynatrace)
- Ken Hamric (Kubeshop)

Agenda

- OTLP metric exporters, OTLP security PR, and moving exporter back to experimental all had dependency issues caused by the split between experimental and stable
 - Update API in all experimental packages for now
 - Consider going back to a single lerna directory with independent versioning
 - More complex versioning for release, but this can be solved with automation.
- Instrumentation stability guidelines (affects most contrib packages)
<https://github.com/open-telemetry/opentelemetry-specification/pull/2180>
- [fix\(sdk-metrics-base\): coerce histogram boundaries to be implicit Infinity](#)
- Metrics TODO
 - Update otlp transformer for new metric types - dan is working on this
 - Update OTLP metrics exporters (ongoing pichlermarc)
 - [refactor\(sdk-metrics-base\): add ObservableRegistry and hoist invocations](#) (depends on meter shared states)
 - Meter identity (depends on meter shared states)
 - [AggregationTemporality Controls](#)
- [chore: moves express examples into its package to establish pattern](#) (tests failing)
 - Failing is related to versioning issue, not the PR changes. PR looks good but is waiting on external factors for merge.
- [Discussion: Drop support for older Node.JS versions](#)
- PRs still open from last week
 - [changelog PR](#)
 - [refactor\(sdk-metrics-base\): meter shared states](#)

2022-03-23

Attendees

- Daniel Dyla (Dynatrace)
- Rauno Viskus (Splunk)
- Marc Pichler (Dynatrace)
- Aaron Abbott (Google)
- Matt Wear (Lightstep)
- Amir Blum (Aspecto)
- Abhinav Mathur (AppDynamics)
- Josephine Liang (Mode Analytics)
- Nev Wylie (Microsoft)
- Andy Fleming (Shopify)
- Svetlana Brennan (New Relic)

Agenda

- 1.1.0 release out
- 1.1.1 release also out
- API version compatibility issues?
https://cloud-native.slack.com/archives/C01NL1GRPQR/p1647951332622529?thread_ts=1647901111.302479&cid=C01NL1GRPQR
- [changelog PR](#)
- [Express example PR](#)
- Metrics update
 - [refactor\(sdk-metrics-base\): meter shared states](#)
 - [feat\(prometheus\): update prometheus exporter with wip metrics sdk](#)
 - [Update OTLP Metrics Exporter with latest Metrics SDK](#)
- [svetlana] [Support for Node versions](#)

2022-03-16

Attendees

- Rauno Viskus (Splunk)
- Daniel Dyla (Dynatrace)
- Andy Fleming (Shopify)
- Nev Wylie (Microsoft)
- Siim Kallas (Splunk)
- Aaron Abbott (Google)
- Svetlana Brennan (New Relic)
- Sidartha Gracias (Cisco)

Agenda

- [dan] OTLP trace exporters

- JSON not stable 🙄
- Splitting transformation functionality
 - Make private
 - Release stable
 - Move to experimental
 - Release experimental
 - Update to use new module
- [dan] [1.1.0 release proposal](#)
- [dan] Consider manual changelogs in core
- [svetlana]: Feedback on [security pr](#)
- [andy]: Discussion on examples PR

2022-03-09

Attendees

- Daniel Dyla (Dynatrace)
- Matt Wear (Lightstep)
- Amir Blum (Aspecto)
- Svetlana Brennan (New Relic)
- Andy Fleming (Shopify)
- Abhinav Mathur(AppDynamics)
- Nev Wylie (Microsoft)
- Sidartha Gracias (Appdynamics)
-

Agenda

- [dan] otlp exporter refactor
 - Create package to do transformations (private)
 - Make exporters experimental (temporarily)
 - Make exporters depend on new package using bundled dependencies
 - Transformation package is not made public, but bundled inside the tar.gz of the exporter
 - Deploy experimental otlp exporters
 - Move trace exporters back to stable
- [legendecas] [fix: enable downleveliteration for es5 targets](#)
 - Es5 does not have Map/Set so es5 bundlers need to require a polyfill
 - Our CI is currently not building the non-default targets
- [dan] fs missing from release in contrib because it is not in the release please config
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/938>
- Metrics
 - Draft: [feat\(prometheus\): update prometheus exporter with wip metrics sdk](#)
 - [refactor\(sdk-metrics-base\): meter shared states](#)

- Replaces [fix\(sdk-metrics-base\): combine concurrent observable callback invocations](#)
 - [refactor\(sdk-metrics-base\): add ObservableRegistry and hoist invocations](#)
 - [feat\(views\): Update addView\(\) to disallow named views that select more than one instrument.](#)
- [dan] http client span names are not descriptive enough - large list of "HTTP GET" is not helpful to users
 - [Spec](#) says to use what we currently have
- [martin kuba] process.runtime.name attribute not being captured, is this an oversight?
- [andy] instrumentation examples reorganization
<https://github.com/open-telemetry/opentelemetry-js-contrib/pull/939>
- [nev] Browser metrics proposal (OTEP draft)
https://docs.google.com/document/d/1bVCOYVgfbfF3Etdj4Y9x3dV2RdA-K_pKRREEKbVyfnY/edit#

2022-03-02

Attendees

- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Amir Blum (Aspecto)
- Martin Kuba (New Relic)
- Svetlana Brennan (New Relic)
- Andy Fleming (Shopify)

Agenda (please add any agenda items you would like to discuss)

- Meeting link appears to be working
- Metrics export data structure refactor
 - [PR open](#) - please review as this blocks exporters being updated for the new SDK
- [dan] [contrib release PR](#) needs review
 - Release automation was updated which broke the release pipeline. The bug in release-please was fixed and released so we should be good to go now
- ~~[dan] aws folks - [feat: support baggage propagation in aws lambda custom context extraction](#) ok to merge?~~
- [dan] [feat: amqplib instrumentation](#)
 - Reviewed, but being left open until next week in case anyone wants to review it
- [dan] [add field name to graphql resolve span name](#)
 - Is this high-cardinality? I am not a graphql expert
- [dan] Please review [combine concurrent callback invocations](#)
- [dan] Please review [timeouts](#)
- [dan] OTLP trace export release blockers
 - [OTLP transformation PR](#)
 - Potential spec breaking issue
 - JSON exporter needs to be moved back to experimental

- Blocking metrics PRs
 - [feat\(views\): handle view conflicts. #2734](#)
 - [refactor: change metrics export data model to match OTLP protos](#)
- [Andy] Instrumentation examples dependency update question

2022-02-23

Attendees

- Nev wylie (Microsoft)
- Amir Blum (Aspecto)
- Aaron Abbott (Google)
- Daniel Dyla (Dynatrace)
- Svetlana Brennan (New Relic)
- Matt Wear (Lightstep)

Agenda (please add any agenda items you would like to discuss)

- Metrics export data structure refactor
 - In progress by Siim
 - PR incoming today or tomorrow hopefully
- [dan] OTLP trace export release blockers
 - Using an old version of the proto
 - Grpc and proto depend on a 0.x version of http
 - Updating grpc/proto dependency on http to 1.0 is broken because of a breaking change in the http exporter
 - Revert the breaking change or update the other exporters?
 - [mwear] recent changes in the proto are backwards breaking in JSON format
 - <https://github.com/open-telemetry/opentelemetry-specification/blob/main/specification/protocol/otlp.md#otlphttp>
 - Should we not release the JSON exporter as 1.0?
 - Proto change has a thread already
<https://github.com/open-telemetry/opentelemetry-proto/pull/362/files#r812109657>
- [dan] please review [multiple async callbacks](#)
- [dan] should we consider [no polyfills targets esnext](#) a breaking change?
- Needs reviews
 - [Resource detector for docker](#)
 - [feat\(trace-otlp-http-exporter\): add compression env vars](#)
 - [test\(sdk-trace-base\): pin core.hrtime dependencies on timeOrigin](#)
 - [HTTP 400 status code should not set span status to error on servers](#)
 - [feat\(otlp-exporter\): add timeout env var](#)
 - [rabbitMQ instrumentation](#)

2022-02-16

Attendees

- Daniel Dyla (Dynatrace)
- Amir Blum (Aspecto)
- Nev Wylie (Microsoft)
- Marc Pichler (Dynatrace)
- Svetlana Brennan (New Relic)
- Siim Kallas (Splunk)
- Rauno Viskus (Splunk)
- Abhinav Mathur(AppDynamics)

Agenda (please add any agenda items you would like to discuss)

- [dan] [refactoring export data structure](#)
 - No responses on the issue. I would like to move forward with this for metrics at least and maybe also traces if we can find a backwards compatible way
- [dan] push/pull exporter interface naming
 - <https://github.com/open-telemetry/opentelemetry-specification/pull/2286>
 - “If the pull exporter is modeled as MetricReader, implementors MAY name the MetricExporter interface as PushMetricExporter to prevent naming confusion.”
 - Decision: use MetricReader and PushMetricExporter interface names
- [dan] prototype of sampling propagation still in progress
- Prototype progress
 - Meter identity - mostly finished locally. Have opened a few clarification comments on the spec issue but nothing appears blocking
 - Multiple callback - <https://github.com/open-telemetry/opentelemetry-js/pull/2785>
- PRs
 - [rauno56] <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/906>
 - Would like to merge asap. Tooling changes.
- [abhinav] working on a resource detector for docker
<https://github.com/open-telemetry/opentelemetry-js-contrib/issues/907>

2022-02-9

Attendees

- Daniel Dyla (Dynatrace)
- Rauno Viskus (Splunk)
- Matt Wear (Lightstep)
- Nev Wylie (Microsoft)
- Vera Reynolds (Honeycomb)
- Marc Pichler (Dynatrace)
- Svetlana Brennan (New Relic)

Agenda

- Metrics:
 - [Project updated](#) (thank you legendecas)
 - [Consider refactoring export data structure](#)
 - Current status:
 - Basic shape of SDK prototype finished
 - Waiting on several spec clarifications
 - [Conflict resolution](#)
 - [Prometheus export](#)
 - [Export interface naming](#)
 - Spec solidifying [multiple callback registration for async instruments](#)
 - Marc Pichler prototyped [view conflict resolution](#), being updated for [latest spec](#)
 - In-progress prototypes
 - Meter identity
 - Dyladan
 - Multiple async observable callback registration
 - legendecas
 - Async observable callback deregistration
 - Legendecas
 - Next steps
 - Maybe refactor export data structure (see above)
 - A simple exporter prototype (console or similar?)
 - Requirements for release
 - Prometheus and OTLP exporters at least prototyped
 - All instruments working
 - Documentation for transition
 - Exemplars and views are optional for alpha
- Tracing:
 - There is renewed interest in attach/detach methods (with requires callbacks)
 - Should we also consider a similar export data structure refactor? May be tricky to do in a backwards compatible way
 - New exporter base class (SpanExporterV2 or similar)
 - Version property in exporter (default to 1 if unspecified)
 - [dan] prototyping [probability sampling in tracestate](#)
 - Amir prototyping rabbitmq instrumentation and migrating it to contrib. Please take a look and review or contact amir if you are interested
- New PRs
 - [feat\(longtasks\): allow callback to add span attributes on collection](#)

2022-01-26

Attendees

- Nev Wylie (Microsoft)

- Daniel Dyla (Dynatrace)
- Vera Reynolds (Honeycomb)
- Matt Wear (Lightstep)
- Svetlana Brennan (New Relic)
- Marc Pichler (Dynatrace)
- Amir Blum (Aspecto)

Agenda

- TODAY'S MEETING IS NOT RECORDED DUE TO AN ISSUE WITH ZOOM
- [amir] Do we have a board with TODO tasks once we drop support for node 8 / 10?
[Example](#)
- [dan] API 1.1 release today? [SDK PR](#) prepared and in draft
- [dan] is SDK ready for release? 1.0 release of OTLP trace exporters included
 - Wait for env var timeout or retry logic?
- New issues
 - [Jaeger exporter does not transform links attributes](#)
 - [Add properties field to the BaggageEntry interface](#)
- New PRs
 - [docs\(otlp-grpc-exporter\): update default url](#)
 - [feat\(opentelemetry-resources\): add runtime version information](#)
 - [docs: specify minimum version of npm to run command in subproject](#)
 - [fix\(exports-jaeger\): transform all links to jaeger reference](#)
 - [feat\(views\): add FilteringAttributesProcessor](#)
 - [feat\(otlp-exporter\): add timeout env var](#)
 - [fix\(resources\): align exported names in different environments](#)

2022-01-19

Attendees

- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Amir Blum (Aspecto)
- Svetlana Brennan (New Relic)
- Matt Wear (Lightstep)
- Rauno Viskus (Splunk)
- Alan West (New Relic)

Agenda

- [dan] calendar event was missing. Morgan recreated it. It is now renamed from Node/Browser Sync to OpenTelemetry JS SIG. Not sure how it got deleted but sorry if it caused confusion.
 - New meeting invite doesn't have a link to this doc.
- [dan] otlp-trace exporter 1.x
 - Waiting on [feat\(otlp-exporter-http\): change otlp-http port to canonical 4318](#)

- Any other PRs/issues blocking?
- [rauno] [Discussion: Drop support for older Node.JS versions](#)
- [dan] API 1.1.0
 - Completed
 - [define common attributes type \(#142\)](#) ([ae9bead](#))
 - [trace: add optional schema url to TracerProvider.getTracer \(#129\)](#) ([aa65fc6](#))
 - [Add TraceState implementation to API](#)
 - Maybe for release
 - [feat: support OTEL_LOG_LEVEL environment variable](#)
 - If this can be ready in time then it can be included
- [svetlana]
 - Waiting on exporter retries ([#1233](#)) - need to work on adding timeout env variable to exporter first ([#2706](#))
 - Issue with OTLP/gRPC endpoint when using http/https scheme - "13 INTERNAL" error - [details in spec](#)
- [alan] retry is not quite specified
 - Added some context [here](#)
 - Ruby impl: <https://github.com/open-telemetry/opentelemetry-ruby/blob/main/exporter/otlp/lib/opentelemetry/exporter/otlp/exporter.rb#L247-L271>
- New PRs
 - [feat\(sdk-trace-web\): web worker support](#)
- New Issues
 - <https://github.com/open-telemetry/opentelemetry-js/issues/2713>
 - bug
 - [@opentelemetry/exporter-trace-otlp-http](#) exporter failing with "RangeError: Maximum call stack size exceeded"
 - <https://github.com/open-telemetry/opentelemetry-js/issues/2718>
 - Bug
 - Jaeger Exporter filters out links having a different link span id than the parent id
- Metrics PRs
 - [feat\(sdk-metrics-base\): implement async instruments support](#)
- Docs PRs
 - [docs\(api-metrics\): add notes on ObservableResult.observe](#)
 - [docs\(context\): Fix links, edit prose](#)
- Other PRs
 - [feat: re-enable TAV](#)
 - [feat\(express\): allow rewriting span names](#)
 - [feat\(otlp-exporter-http\): change otlp-http port to canonical 4318](#)
- Stale PRs and issues
 - Newly Stale
 - [Allow instrumentations to be extendable](#)

- [jaeger-client is now deprecated](#)
- [The batched spans are not sent during unload of browser](#)
- [JaegerExporter caches sender, which locks process service.name for all Spans](#)
- [Can not execute the operation on ended Span opentelemetry-instrumentation-http](#)
- [feat\(@opentelemetry-instrumentation-http\): allow for customization of span name](#)
- [Expose counts for dropped attributes/events/links in ReadableSpan](#)
- [doc: Getting started typescript still uses logLevel](#)
- [grpc collector require protocol or returns empty string](#)
- [Handling potentially missing browser API usage](#)
- [Typescript compatibility](#)
- [Http Spans are not linked / does not set parent span](#)
- [chore: use unknown in catch](#)

2022-01-12

Attendees

- Daniel Dyla (Dynatrace)
- Amir Blum (Aspecto)
- Vera Reynolds (Honeycomb)
- Rauno Viskus (Splunk)
- Matt Wear (Lightstep)
- Nev Wylie (Microsoft)
- Svetlana Brennan (New Relic)
- Hector Hernandez (Microsoft)
-

Agenda:

- [amir] Should we deprecate the non-general http instrumentation config option (ignoreIncomingPaths, ignoreOutgoingUrls)? [link](#)
- [amir] Test multiple versions of instrumentation in PRs CI (mongodb3 / mongodb4) [link](#)
- [dan] will be gone monday/tuesday next week
- [dan] Colorsjs pinned to 1.4.0 for now
 - We depend on colors through karma
 - We do not use the latest version of karma so even if they fix we will not get the fix
 - Updating to latest colors/webpack/etc is blocked by some newer modules and dependencies not supporting node 8
 - Rauno: node 10 may also be an issue
- [mwear] What is the status of CI on contrib?
 - Tag rauno on CI issues in contrib if there are ongoing issues
- Current work (this is just what dan could think of quickly, please update yourself if you want something listed here)
 - Legendecas

- non-node runtimes support [link](#)
 - Async instruments
 - Ignore request hooks for http instr
- Dan
 - proto package
 - API release prep/triage
- Svetlana
 - OTLP exporter retries
- Rauno
 - Contrib CI stability
 - Contrib TAV script
 - [Tedious instrumentation](#) - please review!
- Amir
 - Contrib repo reviews
- [dan] API 1.1.0
 - Completed
 - define common attributes type ([#142](#)) ([ae9bead](#))
 - Waiting for release
 - [Add TraceState implementation to API](#)
 - [feat\(trace\): add optional schema url to TracerProvider.getTracer](#)
 - Candidate PRs
 - [feat\(context\): add attach and detach methods to context API](#)
 - Do not include
 - [feat: add prefix option to DiagConsoleLogger](#)
 - Do not include
 - [feat: support OTEL LOG_LEVEL environment variable](#)
 - If this can be ready in time then it can be included
- New PRs
 - [docs\(api-metrics\): add notes on ObservableResult.observe](#)
 - [feat\(instrumentation-http\): add options to ignore headers](#)
- New Issues
 - [otlp exporters: missing some environment variables from spec in SDK](#)
 - [Allow module to be passed directly into plugins / bypass module hooks](#)
 - [Incorret timestamps of XMLHttpRequestInstrumentation spans after system sleep on browsers.](#)
- Metrics PRs
 - [feat\(sdk-metrics-base\): implement async instruments support](#)
- Other PRs
 - [feat\(otlp-exporter-http\): change otlp-http port to canonical 4318](#)
 - [docs\(context\): Fix links. edit prose](#)
 - [feat: tedious\(SQL Server client\) instrumentation](#)
- Stale PRs and issues
 - [feat\(@opentelemetry-instrumentation-http\): allow for customization of span name](#)
 - [Expose counts for dropped attributes/events/links in ReadableSpan](#)

- [doc: Getting started typescript still uses logLevel](#)
- [grpc collector require protocol or returns empty string](#)
- [Handling potentially missing browser API usage](#)
- [Typescript compatiblity](#)
- [Http Spans are not linked / does not set parent span](#)

2022-01-05

Attendees

- Svetlana Brennan (New Relic)
- Rauno Viskus (Splunk)
- Amir Blum (Aspecto)
- Daniel Dyla (Dynatrace)
- Valentin Marchaud

Agenda:

- Welcome new maintainers:
 - Amir Blum (blumamir), Aspecto
 - Chengzhong Wu (legendecas), Alibaba
 - Rauno Viskus (Rauno56), Splunk
- Is stale bot useful?
 - Not all old issues have even been tagged yet
 - Feature requests generally don't become stale
 - Mostly useful for issues that are opened as "bugs" but are not actually bugs but are misconfigurations and complaints
 - [Logs for a run](#)
- Testing in containers vs setup-node
 - Setup-node runs on the host machine. Potentially less reproducible? Test all versions doesn't run setup-node currently. Services for tests need to listen on localhost
 - Testing in container runs into some permissions issues, but works with existing TAV scripts
 - Rauno56: Should TAV be run scheduled(and release PRs) or every for PR?
 - Argument(s) for scheduled:
 - Allows for open ranges without breaking every unrelated PR from a certain lib release. Only running tests for changed packages helps, but think of an a documentation change to a package that has released an incompatible version
 - Argument(s) for running for every PR:
 - If a PR breaks an old version it's known immediately
 - TAV script needs to be updated to run setup-node
- [WIP @opentelemetry/proto package](#)
 - Depends on protobufjs

- Autogenerated JS with .d.ts declarations
- Static code (no async load of proto definitions)
-
- Metrics PRs
 - [feat\(sdk-metrics-base\): implement async instruments support](#)
 - [fix\(sdk-metrics-base\): remove aggregator.toMetricData dependency on AggregationTemporality](#)
 - [feat\(sdk-metrics-base\): meter identity](#)
 - [feat\(metric-reader\): add metric-reader](#)
- Fix PRs
 - [fix\(propagator-jaeger\): 0-pad span-id to match 16-symbol validation](#)
 - [fix: span attribute count and value limits \(#2671\)](#)
 -
- Other PRs
 - [feat\(otlp-exporter-http\): change otlp-http port to canonical 4318](#)
 - [refactor: unifying shutdown once with BindOnceFuture](#)
 - [docs\(context\): Fix links, edit prose](#)
 - [feat: tedious\(SQL Server client\) instrumentation](#)
- Retry logic in http
 - <https://github.com/open-telemetry/opentelemetry-js/issues/1233>
 -

2021-12-29

Attendees

- Dan
- Amir
- Srikanth

Agenda:

- Holiday week office hours
- Srikanth: tracestate should be in api package, PR on the way

2021-12-22

Attendees

- Nev Wylie (Microsoft) [Will miss the next 2 meetings]
- Amir Blum (Aspecto)
- Daniel Dyla (Dynatrace)
- Vera Reynolds (Honeycomb)
- Rauno Viskus (Splunk)
-

Agenda

- Holiday week
- Released 1.0.4 API
- [Began a style guide](#)
- [Autoinstrumentation span status](#)
 - Dan will reach out to the instrumentation group in slack
 - Amir: it is much easier to leave it unset and only set when there are errors
 -
- [MetricReader interface](#)
- Documentation failing to update automatically
 - Combination of branch protection rules and typedoc updates
 - Dan is looking into it
- Remove bart from maintainers
 - [Core repo](#)
 - [API](#)
 - [Contrib](#)
- Doc JS updates looking for js-approvers (e.g. <https://github.com/open-telemetry/opentelemetry.io/pull/988>)
- [Exemplar Collect] <https://github.com/open-telemetry/opentelemetry-js/pull/2641>

2021-12-15

Attendees

- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Svetlana Brennan (New Relic)
- Aaron Abbott (Google)
- Amir Blum (Aspecto)
- Matt Wear (Lightstep)
- Vera Reynolds (Honeycomb)

Agenda

- Test all versions action PR currently runs on each PR for only changed packages
 - Should we run nightly on all packages?
 - How do we alert on nightly failures? Create an issue? Who gets assigned? How do they get deduplicated (or don't they)?
 - Ext-js repo: aws sdk tends to break instrumentations fairly regularly. Github action runs every day. Failure sends a message to a slack channel
 - Test nightly, create an issue on failures
- Please review metric streams PR <https://github.com/open-telemetry/opentelemetry-js/pull/2636/files>
 - It has had a few reviews and revisions so far and should be ready for final review
- Exemplars PR <https://github.com/open-telemetry/opentelemetry-js/pull/2641>

- <https://github.com/open-telemetry/opentelemetry-js/pull/2658>
 - Examples for WIP packages should reside within the package directory
 - Examples in /examples folder should show released functionality
- Shared attributes definition
 - Definition in spec is shared, but there is no hard requirement to have it be a shared definition in clients
 - Current attribute definition is “SpanAttributes” and has been released as 1.0 so needs to be maintained
 - Metrics attribute definition is currently just “Attributes”
 - Option 1: create new “Attributes” and alias “SpanAttributes” for backwards compatibility, use this everywhere
 - Matt: if we didn’t have the previous name, we would ideally just have a single name
 - Aaron: proto definition is shared
<https://github.com/open-telemetry/opentelemetry-proto/blob/7d4cbec4a940edc92a87b173b8d130b24085607e/opentelemetry/proto/metrics/v1/metrics.proto#L353>
 - Daniel what about logs: looks the same in proto as well
<https://github.com/open-telemetry/opentelemetry-proto/blob/7d4cbec4a940edc92a87b173b8d130b24085607e/opentelemetry/proto/logs/v1/logs.proto#L138>
 - Option 2: alias “SpanAttributes” to “MetricAttributes”. Definition is shared, but use is consistent (use span attrs when doing tracing and metric attrs when doing metrics). Leaves the possibility for diverging implementations in the future without breaking.
 - Option 3: rename metrics attributes to “MetricAttributes” and keep definitions separate. Same advantage as 2
 - Action Item: create issue (never stale) to remove spanattribute name. Create a v2 milestone and put the issue in it.
 - AI: use deprecated js doc tag for span attributes
 - AI: update all call sites to use attributes name

2021-12-08

Attendees

- Valentin Marchaud
- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Amir Blum (Aspecto)
- Svetlana Brennan (New Relic)
- Matt Wear (Lightstep)
- Rauno Viskus (Splunk)
-

Agenda

- Bugfix PR for grpc instr: <https://github.com/open-telemetry/opentelemetry-js/pull/2650>
- Aggregator PR still under review
- Exemplars <https://github.com/open-telemetry/opentelemetry-js/pull/2641>
- Contrib TAV action <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/778>
-
- New feature for API
 - <https://github.com/open-telemetry/opentelemetry-js-api/pull/129>
 - Adds options object to getTracer with schemaUrl
 - We need to determine a branching strategy to release 1.x versions of the API with new features while still backporting fixes to 1.0.x
- Design doc metrics
<https://docs.google.com/document/d/1EluDhj5UanNATS3hR8xx30sv55Se7uYjsGYslvNWHtc/edit#>

2021-12-01

Attendees

- Daniel Dyla (Dynatrace)
- Aaron Abbott (Google)
- Matt Wear (Lightstep)
- Svetlana Brennan (New Relic)
- Amir Blum (Aspecto)

Agenda:

- Slow week due to the US holiday
- Contrib versioning issues
 - Not all instrumentations target the same version of the instrumentation class
 - If the instrumentation class version used by the instrumentation does not match that used by the SDK, there can be a compile time error that the types don't match
 - I updated all instrumentations to the latest instrumentation class. Only one minor code change was required
(<https://github.com/open-telemetry/opentelemetry-js-contrib/pull/769/files#diff-6226210cd15f1d45ccbe5824046775a19fee8b799267b4e737a2e68646220e4cR49>)
 - Release PR is ready for that
<https://github.com/open-telemetry/opentelemetry-js-contrib/pull/771>
- Please review metrics PRs
 - [Aggregation Support](#)
 - Legendecas is creating a design doc for our Metric SDK similar to python's
 - [Exemplars](#)
 - [Metric Streams](#) (draft)

- [Trace OTLP exporters to 1.0](#) still draft?

2021-11-24

Attendees

- Daniel Dyla (Dynatrace)
- Rauno Viskus (Splunk)
-

Agenda:

- No agenda because of US holiday. Dan will hold maintainer office hours if anyone has anything to discuss
- Dan will look into running test all versions daily and on release PRs and have a slack channel notification - [Issue here](#)
 - Rauno reached out to volunteer for this

2021-11-17

Attendees

- Aaron Abbott (Google)
- Amir Blum (Aspecto)
- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Valentin Marchaud
- Svetlana Brennan (New Relic)
- Srikanth
- Matt Wear (Lightstep)
- Vera Reynolds (Honeycomb)
- Rauno Viskus (Splunk)
-

Agenda

- Jaeger client sunset
- Metrics:
 - Old Metrics SDK replaced with WIP metrics SDK
 - Consider removing metrics exporters
 - They depend on the old metrics SDK and API and cause compilation issues when linked with the new API
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2628>
 - [Views bootstrap](#) (legendecas)
 - [Async instruments](#) (legendecas)
 - [Metrics Exemplars](#) assigned to lonewolf3739

- Python Design proposal for metrics SDK (includes overview of Java impl)
<https://docs.google.com/document/d/1FOmpGCiZAWTd6a3vcEHBnm1LKp7HAKuD4BBN3YCkymc/edit#>
- [api] [Add source map files to builds](#)
- Upcoming holiday schedules
 - Should we look at cancelling some meetings preemptively
 - Thinking specifically 11/24; 12/22; 12/29

2021-11-10

Attendees

- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Bartłomiej Obecny (Lightstep)
- Amir Blum (Aspecto)
- Vera Reynolds (Honeycomb)
- Srikanth C
- Aaron Abbott
-

Agenda

- [Release 1.0.1 / 0.27.0](#)
 - PR is failing because the experimental packages point to unreleased stable packages
 - Merge PR with failing build, release stable, re-run tests (they should pass now), release experimental
- [New ground-up metrics SDK](#)
 - Marked `private` in package.json to prevent releasing until it is useable
 - Will not be released until it is in a minimum useable state
 - Can at least collect and export a metric
 - May not have all features for first alpha release
- [Remove old metrics SDK](#)
 - Build currently fails because exporters/etc depend on unreleased sdk changes
 - After release above, this build will pass
- [Close stale issues](#)
- [ESM 2015 release](#)
- Lint warning fixes
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2404>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2451>
- [HTTP: Allow customization of span name](#)
 - Adds a callback to config which is called with each request and returns a name for the span

- Spec on span names
https://github.com/open-telemetry/opentelemetry-specification/blob/main/specification/trace/semantic_conventions/http.md#name
- Decided to wait for now and see if the PR author responds

2021-11-03

Attendees

- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Marten Hennoch (Splunk)
- Vera Reynolds (Honeycomb)
- Bob Evans (New Relic)
- Bartłomiej Obecnny (Lightstep)

Agenda

- [Metrics API](#)
 - 3 todo / 6 in-progress / 1 in-review / 10 done
 - [Update async instruments](#)
 - [Draft: Update sync instruments](#)
 - [Rename labels to attributes](#)
 - Up for grabs: [Updates required for compliant attributes](#)
 - Rename labels to attributes (assigned to pirgeo)
 - Update proto
 - Update API to move attributes to a common folder
 - Use API attribute definition in metrics packages
- [Metrics SDK](#)
 - Project currently has only broad issues. Please help create issues to describe work needed to achieve metrics 1.0 SDK and help break up the large issues into manageable pieces of work
- <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/702>
 - Bart will look into the double-enable
 - 0-length spans are OK for now. Metrics would be better
 - RUM sig might want these to be defined as logs
 - Logging implementation may be a while
 - Marten will take ownership of the package
-

2021-10-27

Attendees

- Daniel Dyla (Dynatrace)

- Nev Wylie (Microsoft)
- Aaron Abbott (Google)
- Amir Blum (Aspecto)
- Bartłomiej Obecny (Lightstep)
- Hector Hernandez (Microsoft)
- Valentin Marchaud
- Rauno Viskus (Splunk)
- Bob Evans (New Relic)
-

Agenda

- Contrib released
 - Sorry this took so long
- [Metrics API](#)
 - 7 todo / 8 in-progress / 2 done
 - Big thanks to legendecas for API work this week
 - [Metrics API issues in need of reviews](#)
- [Question of style](#) which do you prefer?
 - Rejected `getMeter(name: string, options: MeterOptions)`
 - Does not match the trace API style
 - `getMeter(name: string, version?: string, schemaUrl?: string)`
 - Does not require extra types
 - There is only one option right now so an object is weird if none are added in the future
 - No trace APIs have an options object afaik
 - `getMeter(name: string, version?: string, options?: MeterOptions)`
 - Allows for adding more options in the future
- [Linking experimental and non-experimental packages](#)
 - Currently not possible
 - Suggested solution: go back to the way lerna was before with only a single packages directory. Use npm scripts with “stable” and “experimental” in the name which list all packages in the group. For example: “version:stable”: “lerna run version --scope @opentelemetry/core --scope @opentelemetry/sdk-trace-node ...etc”

2021-10-20

Attendees

- Daniel Dyla (Dynatrace)
- Bartłomiej Obecny (Lightstep)
- Yaniv Davidi (Aspecto)
- Amir Blum (Aspecto)
- Rauno Viskus (Splunk)
- Aaron Abbott (Google)

- Severin Neumann (Cisco)
- Nev Wylie (Microsoft)

Agenda

- [dan] meeting link might change
- [dan] Metrics roadmap
 - API compliant with spec around December
 - SDK implementation complete early 2022
- [dan] metrics issue triage and assignment
 - Project <https://github.com/open-telemetry/opentelemetry-js/projects/14>
 - Tracking Issue <https://github.com/open-telemetry/opentelemetry-js/issues/2480>
- [willarmiros] I don't have a strong opinion one way or another (nor am I sure if I'll be able to attend this SIG meeting) but would like to get a conclusion on the ultimate goal of the exporters to unblock the PR
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2485#pullrequestreview-779349890>
 - Rephrase: should we have a single exporter with all stable signals (currently only trace) or should we have one exporter per signal
- Talked about generating a central protobuf handling package
 - <https://github.com/open-telemetry/opentelemetry-js/issues/2554>
- [Nev] Discussion on not using the bleeding edge features (from Maintainers meeting)
 - Specifically TS features that generate different code for older versions etc
 - Question: How do we go about catching/enforcing these types of issues

2021-10-13

Meeting Canceled due to kubecon 2021 conflict

2021-10-06

Meeting Cancelled due to maintainer vacations

2021-09-29

Attendees

- Daniel Dyla (Dynatrace)
- Valentin Marchaud
- Nev Wylie (Microsoft)
- Siim Kallas (Splunk)
- Aaron Abbott (Google)
- Vera Reynolds (Honeycomb)

- Will Armiros (AWS)
- Bob Evans (New Relic)
- Ivan Santos (EverQuote)
- Amir Blum (Aspecto)

Agenda

- [dan] API to attach/detach current context
 - Currently we only allow with() to change context which does not allow for control flows which are not inherently nested
 - Draft <https://github.com/open-telemetry/opentelemetry-js-api/pull/123>
 - Biggest question: how do we develop/test new API features without releasing them as part of the 1.x series?
 - 1.x.y-alpha?
 - On a branch
 - Experimental namespace within API which may be unstable?
 - Simply document unstable methods as unstable?
 - This is what node itself does
 - New api-experimental package which wraps and extends the API?
 - Something else?
 - Steps
 - Reach out to TC to validate the plan outlined below
 - Early development of the api method on 1.1.0-alpha
 - Implement SDK methods
 - Release 1.1.0 with methods clearly documented unstable
 - Unstable methods may change with minor version changes
 - What is the behavior if a method cannot be implemented by SDK
 - Dan believes it should be up to that SDK to implement some fallback
- [blumamir/jonbcampos-alto] How should we approach otel instrumentations which collide with other non-otel instrumentations?
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/666>
- [willarmiros] <https://github.com/open-telemetry/opentelemetry-js/pull/2485>
 - Split trace & metric exporters rebased & ready for review/merge **after** 1.0 & experimental release
- [willarmiros] <https://github.com/open-telemetry/opentelemetry-js/issues/2413> - how often are the docs at <https://open-telemetry.github.io/opentelemetry-js/> updated?
- 1.0 release
 - Any open PR/issue which should block?
 - Packages which will be released as 1.0
 - @opentelemetry/context-async-hooks
 - @opentelemetry/context-zone-peer-dep
 - @opentelemetry/context-zone
 - @opentelemetry/core
 - @opentelemetry/exporter-jaeger
 - @opentelemetry/exporter-zipkin

- @opentelemetry/propagator-b3
- @opentelemetry/propagator-jaeger
- @opentelemetry/resources
- @opentelemetry/sdk-trace-base
- @opentelemetry/sdk-trace-node
- @opentelemetry/sdk-trace-web
- @opentelemetry/semantic-conventions
- @opentelemetry/shim-opentracing
- Steps
 - [Release 0.26](#) core
 - Release 0.26 experimental
 - Release 1.0 core

2021-09-21

Attendees

- Daniel Dyla (Dynatrace)
- Marten Hennoch (Splunk)
- Amir Blum (Aspecto)
- Aaron Abbott (Google)
- Vera Reynolds (Honeycomb)
- Rauno Viskus (Splunk)
- Ivan Santos (EverQuote)
- Bartłomiej Obecny (Lightstep)
- Will Armiros (AWS)
- Hector Hernandez (Microsoft)
- Severin Neumann (Cisco/AppDynamics)
-
-

Agenda

- 1.0 prep
 - Prometheus moved to experimental
 - Will working on splitting OTLP exporters into separate packages
 - Replacing lodash.merge with vendored version
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2484>
- [dan] examples/docs often reflect unreleased state. Have had multiple users approach me recently who have been confused when they are trying to get started with otel and using the examples. When the examples show unreleased functionality they get stuck.
 - Suggestion: root examples folder should show only released functionality. With 1.0, they should continue to work for a long time without changes. These examples should be treated as documentation. Currently they are used as development tools.

- Rename examples to test-apps
- Link to website for getting started
- Remove getting-started from repository
- Version headers for website docs
- Review current website docs to make sure links back to the repository are properly contextualized
- “Full” example application which shows a fully working instrumented application in the contrib repo. Possibly this is a polyglot example in a dedicated repository?
- [dan] release-please automation working in contrib
 - Changes to package A which depend on changes in package B must be done in separate PRs. Referencing unreleased changes in package B will cause the tests to fail in the release-please PR.
 - Specifically relates to test utils usually
 - The best way to ensure you are using published versions is to run npm install in the directory of the package you are developing instead of bootstrapping/linking
 - First automated release in contrib successful
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/releases/tag/instrumentation-aws-lambda-v0.26.0>
 -
- [will] working on splitting exporters, just one last bug I can't figure out
 - PR: <https://github.com/open-telemetry/opentelemetry-js/pull/2485>
 - Will make a separate PR which *only* removes metrics from the exporters without adding anything new. Then stable can be released (as 1.0), rebased on existing PR, and then new metric exporters can be merged
- [marten] Has anyone done anything with Browserstack already?
 - Bart is working on this. Can open a draft PR so others can see progress
-

2021-09-15

Attendees

- Rauno Viskus (Splunk)
- Marten Hennoch (Splunk)
- Aaron Abbott (Google)
- Bartłomiej Obecny (Lightstep)
- Valentin Marchaud
- Daniel Dyla (Dynatrace)
- Vera Reynolds (Honeycomb)
- Severin Neumann (Cisco/AppDynamics)
- Weyert de Boer
- Amir Blum (Aspecto)
- Will Armiros (AWS)
- Hector Hernandez (Microsoft)

- Ram Thiru (Microsoft)
- Diana Thayer (New Relic)

Agenda

- 1.0.0 release prep
 - [Renaming collector exporter packages](#)
 - Rauno: instrumentation enabling with uninitialized fields
 - This package is not a 1.0 package
 -
- Contrib release-please automation
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/659>
 - Will to add NPM and GH publishing steps, then should be reviewed
- [dan] looking into contrib 0.25.0 release failure
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/662>
 - Manually publishing
- ESM targets es2017 vs es5
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2472>
- [Require usages in ESM builds](#)
- Upcoming otel GC elections
- [Carlos] Please review gzip compression PR (we want to release Spec 1.7 today/tomorrow):
<https://github.com/open-telemetry/opentelemetry-specification/pull/1923>
- [Will] can we expect 1.0.0 release this week (after collector renaming is done)?

2021-09-08

Attendees

- Bartłomiej Obecny (Lightstep)
- Daniel Dyla (dynatrace)
- Rauno Viskus (Splunk)
- Aaron Abbott (Google)
- Nev Wylie (Microsoft)
- Will Armiros (AWS)

Agenda

- Slow week due to Labor Day holiday vacations. Apologies for slow reviews and merges
- [dan] follow up on grouping packages with multiple lerna.json files
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2448>
 - Current problems
 - There is no method of selecting a lerna.json file when running lerna commands.
 - Current solution: add lerna.json to gitignore and copy lerna-stable.json or lerna-experimental.json to lerna.json locally. Add npm scripts use:stable and use:experimental for this reason.

- Possible solution: rush.js if it supports package groups
 - Node 8 fails on the update-ts-references step
 - Stop testing on node 8?
 - Pre-generate tsconfig.json files with references array prepopulated?
 - Update the ts references project. Copy into our directory if needed to unblock for now and create an upstream PR
 - Integration test not ported
 - Will move this into the trace-base package where it should have been from the beginning
 - Bootstrapping with one group hoists dependencies into node_modules folder. You must delete the node_modules folder and bootstrap again each time you switch from stable to experimental
 -
- PRs
 - [Implemented general limits of attributes #2430](#)

2021-09-01

Attendees

- Amir Blum (Aspecto)
- Daniel Dyla (Dynatrace)
- Valentin Marchaud
- Rauno Viskus (Splunk)
- Aaron Abbott (Google)
- Nev Wylie (Microsoft)
- Will Armiros (AWS)
- Bartlomiej Obecny (Lightstep)

Agenda

- [dan] release-please may not turn out to be the correct solution for the core repo
 - When it updates a module version, it does not update the dependencies of its dependent modules
 - All SDK modules should share a version to reduce confusion. Release please wants to only bump the versions of those packages which have changed
 - Lerna has no good way to group packages and filter by a group.
 - Suggested solution: nested lerna monorepo
 - Remove release-please
 - Packages directory keeps all packages which are to be released as 1.0
 - New experimental directory is a nested lerna monorepo which keeps version 0.x packages
 - experimental/lerna.json
 - experimental/CHANGELOG.md

- experimental/packages/metrics-api
 - experimental/packages/instrumentation
 - experimental/packages/instrumentation-http
 - ... and so on
- Suggested solution: multiple lerna.json files
 - Lerna-experimental.json
 - Lerna-stable.json
 - lerna.json
- Suggested solution: maintain a list of stable and experimental packages and use --scope
 -
- Suggested solution: script which updates versions in package.json files
 - <https://github.com/microsoft/ApplicationInsights-JS/tree/master/tools/release-tools>
- [will] contrib: Is the CLA signing issue for GH Actions bot resolved?
 - As a workaround, we are using Dan's GH token which is CLA signed
 - In future, CNCF will either allow-list GH Actions bot or create a new bot whose token we can use

2021-08-25

Attendees

- Nev Wylie (Microsoft)
- Bartłomiej Obecny (Lightstep)
- Amir Blum (Aspecto)
- Yaniv Davidi (Aspecto)
- Aaron Abbott (Google)
- Daniel Dyla (Dynatrace)
- Will Armiros (AWS)
- Vera Reynolds (Honeycomb)

Agenda

- [dan] contrib update to core 0.25 stalled due to failing tests in one component. How do we want to handle this in general so that these types of issues don't stall all components and prevent timely releases? Should component owners be responsible for updating their own components? Should the person doing the upgrade skip components which cause a problem?
 - Try to update all packages
 - Skip packages which fail for non-obvious reasons
 - Create an issue and assign to component owner for package which failed to update
- [dan] still waiting on github actions exemption for release-please. Not sure what is taking so long, but I hope they are not thinking of denying the request

- [dan] soliciting feedback from component owners on the first 1-2 weeks. Do you feel your PRs are being merged quickly enough? Are there pain points we have not yet addressed?
 - Amir: what is the policy if i create a component which I am an owner of
 - Should get 1 more reviewer just to make sure there is an additional set of eyes. Github action will be extended to add labels to PRs
- [dan] [fix event target undefined](#) LGTM but I would like a browser expert to review. There is no component owner for this component currently.
- [aaron] HTTPTraceContextPropagator should not have "HTTP" in the name <https://github.com/open-telemetry/opentelemetry-js/issues/2428>
 - Do we want to make this late rename and cause churn?
 - `TraceContextTextMapPropagator`
- [amir] <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/639>

2021-08-18

Attendees

- ~~Nev Wylie (Microsoft)~~ - Out of Office
- Daniel Dyla (Dynatrace)
- Rauno Viskus (Splunk)
- Amir Blum (Aspecto)
- Yaniv Davidi (Aspecto)
- Will Armiros (AWS)
- Ivan Santos (EverQuote)
- Vera Reynolds (Honeycomb)
- Bartlomiej Obecny (Lightstep)

Agenda

- [dan] automated release
 - [Example release PR](#) (waiting on [easycla exemption](#))
- [dan] [Manual release PR for now](#)
- [dan] Congrats to component owners
 - NathanielRN and willarmiros
 - id-generator-aws-xray
 - instrumentation-aws-lambda
 - propagator-aws-xray
 - resource-detector-aws
 - blumamir and naseemkullah
 - instrumentation-ioredis
 - Blumamir
 - instrumentation-redis
 - Legendecas

- resource-detector-alibaba-cloud
- rauno56
 - instrumentation-restify
 - instrumentation-router
 - instrumentation-memcached
 - instrumentation-knex
 - Instrumentation-generic-pool
- [dan] Looking for someone to help co-maintain the github action
- [dan] Rules for owners
 - 1 owner from each changed package must approve
 - 1 additional reviewer or no?
 - Bart: should we hold changes for 24 hours for substantial changes
 - Rauno: I would not support this. The component owners are already the trusted maintainer for the package. Suggested alternative: 24 hours minimum only if an owner has not reviewed yet
 - Amir: bugfixes need to be merged quickly
 - Urgent Bugs: merge after owner approves
 - Minor Bugs: owner + 24 hour wait
 - Features: owner approval + 24 hour wait
 - Breaking changes: owner approval + 24 hour wait + 1 more approver
 - What things can block a PR even if owner approves?
 - Explicit spec violation
 - Known security issue
 - Potential PII violation
 - 80% test minimum should be fairly easy to meet
 - Exceptions to these must be communicated and documented in the README
 - Should new packages be required to have an owner?
 - DD: votes for yes
 - Rauno: this may prevent some people from contributing when they dont want to make long term commitments
 - No package to 1.0 without an owner
 - If component owner does not review the PR
 - 2 official reviewers, at least 1 must be a maintainer
 - No hard rule, but generally after a few days we will ping an owner if they are unresponsive. After a week or more if they have still not responded, the PR may be merged.
 - Pattern of non-responsiveness may be grounds for removal of ownership at maintainer discretion
 - Only maintainers merge or approvers too
 - Anyone with write access can merge a PR that has met the ownership approval requirements above
 - The approver should not be the one who merges in order to get a second eye on the PR

-
- New instrumentations
 - Fastify
 - Obecny
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/611>
 - NestJS
 - Rauno56
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/553>
- Bugfixes
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/627>
- PRs
 - [Spec: implement attribute value length limit option](#)

2021-08-11

Attendees

- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Amir Blum (Aspecto)
- Yaniv Davidi (Aspecto)
- Rauno Viskus (Splunk)
- Bartłomiej Obecny (Lightstep)
- Severin Neumann (Cisco)
- Will Armiros (AWS)
- Vera Reynolds (Honeycomb)
- Ivan Santos (Everquote)

Agenda

- [dan] merged large package renaming PR in core. Release PR today (or we can merge the release-please PR and let it make the PR)
 - Tried to test it on a branch but the queue is taking a long time
<https://github.com/open-telemetry/opentelemetry-js/actions/runs/1120709659>
- [dan] contrib packages can now be “owned” by adding an entry to [component_owners.yml](#)
 - Component owners must be members of the github org (non-members cannot be assigned or receive review requests)
 - If an owner is non-approver, their check will still be grey but they are still counted by us as “owner”
 - Still defining what exactly ownership means. It should be clearly documented.
 - Dependabot PRs will be ignored after [this](#)
- [blumamir/willarmiros] <https://github.com/aspecto-io/opentelemetry-ext-js/issues/169> - move aws-sdk instrumentation to contrib, along with some new workflows
 - Requirement of 1 non-author, component-owning approver before merge?
 - Sounds good

- Does a component owning must be an 'approver'?
 - No - just a member of otel org
- What could be a realistic expectation for timeline between approval and merge by maintainer, especially if "approved" label was added to PRs by workflow?
 - approved-by-owner and authored-by-owner labels will likely be added
 - Potentially will allow approvers to merge approved PRs - dyladan to check if this violates any OTEL policy
- What would be the cadence of releases with release-please on core repo and contrib?
 - Will start with daily releases manually, potential to move to automatic releases
- Ok with daily build workflows in contrib repo? How do we alert on their failure - slack message to #otel-js, open an issue, something else?
 - Daily builds should be ok, can potentially have longer runs
- Test on multiple versions of an instrumented library as part of CI, even if CI time goes up to a few minutes?
 - Should look into lerna --changed or similar for only running contrib tests on changed packages
 - Can run tests on only a small sample (3-5) of versions on PR builds - need to make sure CI use isn't too heavy due to limited resources
 - Could possibly allow Aspecto-hosted CircleCI account or similar - Dan to ask TC if that's allowed & Amir to ask if they could host
- [dan] release-please merged in API
- [dan] [release-please for core](#)
 - Each time a PR is merged, the release PR is automatically updated
 - When the release PR is merged, there is a release
 - Exact cadence to be determined, but it should follow a regular and predictable pattern (daily, weekly, monthly, every merge, etc)
 - Contrib is next, will follow the same pattern as core with possibly a different release cadence

2021-08-04

Attendees

- Rauno Viskus (Splunk)
- Daniel Dyla (Dynatrace)
- Valentin Marchaud
- Bartłomiej Obecny (Lightstep)
- Vera Reynolds (Honeycomb)
- Will Armiros (AWS)
- Nev Wylie (Microsoft)
- Ivan Santos (Everquote)
- Severin Neumann (Cisco)
- Marten Hennoch (Splunk)

Agenda

- Welcome Amir Blum to approvers
- [dan/vmarchaud] [SDK renames](#)
 - PR is large and touches many many files. Will require almost every other PR to rebase when it is done. We should try to approve and merge quickly so it doesn't get stale as valentin has already put a lot of work into it
 - @opentelemetry/tracing -> @opentelemetry/sdk-trace-base
 - @opentelemetry/node -> @opentelemetry/sdk-trace-node
 - @opentelemetry/web -> @opentelemetry/sdk-trace-web
 - @opentelemetry/metrics -> @opentelemetry/sdk-metrics-base
 - @opentelemetry/node-sdk -> @opentelemetry/sdk-node
- [rauno56] Consider setting up something like [Close Stale Issues GHA](#) to keep better hygiene among Issues and PRs.
 - K8s uses something called “fejta-bot”, which is custom to their use case, I think.
 - Among other things def setup [exempt-{issue.pr}-labels](#),
 - I'd add [delete-branch](#),
 - and initially set the stale period long(or use [start-date](#)), not to get overwhelmed with everything reported stale at once(unless someone commits to attend those reports).
 - Nev mentioned [Close Stale Issues · Actions · GitHub Marketplace](#) (looks like the same one) ?
 - Rauno56 will look into both and open an issue with a recommendation
- [willarmiros] Blockers for contrib trace components 1.0
 - <https://github.com/open-telemetry/opentelemetry-js/issues/2348> - lerna independent versioning (in contrib as well)
 - See release automation below
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/blob/main/CONTRIBUTING.md#contributing-vendor-components> - Probably want to start using CODEOWNERS in contrib for independent ownership
 - Cannot use codeowners without granting write access
 - Maybe we can set up a GHA to auto-assign contrib component “owners”
 - Instrumentation API 1.0? (sorry I forgot if this was being included in trace GA)
 - no
 - Anything else?
 - <https://github.com/open-telemetry/oteps/pull/165>
 -
- [dan] release automation
 - [API release-please](#)
 - [SDK release-please](#)
 - [docs](#)
 - Changes to independent versioning. Each package
 - Remove version.ts from version control and generate them at precompile
 - Strict requirement for [conventional commits](#)

- Fix: bumps patch version
 - Feat: Bumps minor version
 - Fix! And feat! Both bump major version (breaking change)
- PRs in need of review
 - [Follow spec defined port 4318 for HTTP exporter](#)
 - [fix: respect sampled flag in Span Processors, fix associated tests](#)
 - [refactor: move detectors to opentelemetry-js-contrib repo](#)
 - [feat: move aws/gcp detectors from opentelemetry-js repo](#)
 - [Fix Lint warnings in api-metrics and context-zone-peer-dep](#)
 - [feat: more specific types for getConfig and setConfig](#)
 - [fix\(@opentelemetry/exporter-collector\): remove fulfilled promises cor...](#)
 - Has breaking test and conflicts
 - [chore: fix lint errors on opentelemetry-context-async-hooks](#)

2021-07-28

Attendees

- Jon Church @jonchurch (otel@jonchurch.com)
- Daniel Dyla (Dynatrace)
- Rauno Viskus (Splunk)
- Nathaniel Ruiz Nowell (AWS)
- Bartłomiej Obecny (Lightstep)
- Aaron Abbott (Google)
-

Agenda

- [dyladan] API bugfix merged. [1.0.2 release PR](#)
 - Patch release, please review
- [dyladan] 0.24 released
- [dyladan] [PR to update core deps to 0.24 in contrib](#)
- [dyladan] SDK package renaming [#2340](#)
- [dyladan] browserstack is taking longer than i'd hoped
- [dyladan] documenting dependency strategy recommendations
 - Which deps to pin, which to use ^
 - Which deps should be peer vs regular
 - Etc
 - [Renovate is now my sworn enemy](#)
- [rauno56] publishing [test-utils](#) - For? Against?
 - [Discussion](#)
 - Action Item: Rauno will create a readme.md for it
- PRs to review
 - [opentelemetry-js-contrib#553](#)

2021-07-21

Attendees

- Dan Dyla (Dynatrace)
- Aaron Abbott (Google)
- Nev Wylie (Microsoft)
- Vera Reynolds (Honeycomb)
- Bartłomiej Obecny (Lightstep)
- Valentin Marchaud
- Marten Hennoch (Splunk)
- Rauno Viskus (Splunk)
- Will Armiros (AWS)
-

Agenda

- [dyladan] release 0.24 today?
 - Will make the release PR today
- [dyladan] welcome rauno to approvers
- [dyladan] [SDK packages renaming](#)
 - Made a decision, commented on issue
- [dyladan] Docs and examples update?
- [dyladan] still going through the process to get browserstack. Shouldn't be much longer I hope
- [dyladan] Linking to non-open-source community resources in our docs
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2344>
- ... please add agenda items if you have things to discuss
- Please review:
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2345>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2365>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2358>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2332>
-

2021-07-14

Attendees

- Rauno Viskus (Splunk)
- Daniel Dyla (Dynatrace)
- Vera Reynolds (Honeycomb)
- Nev Wylie (Microsoft)
- Aaron Abbott (Google)
- Jon Church (Open Source Contributor)
- Amir Blum (Aspecto)

- Severin Neumann (Cisco)
- Will Armiros (AWS)
- Weyert (Tapico)
- Bartłomiej Obecny (Lightstep)
- Nathaniel Ruiz (AWS)

Agenda

- [dyladan] Browserstack testing
 - Unlimited manual and automated testing
 - 5 user licenses and parallels
 - Full platform access to Live, Automate, App Live and App Automate
- [dyladan] [SDK GA project](#)
 - Please add any issues/PRs to this project if they should block GA
 - Bugs
 - Required tooling
 - Changes required by spec 1.0
 - Anything that changes the instrumentation API
 - Anything that changes the setup/configuration
- [dyladan] Docs consolidation update from svnm?
 - Working with weyert to make docs and examples consistent
 - Website docs updated to be working
 - PR open with some minor improvements
- [dyladan] Examples update from weyert?
 - Still looking into this
- [dyladan] Would we prefer to release all tracing packages as 1.0 together or release one at a time in a rolling release fashion as they are ready (start with easy packages like semconv)?
 - [aaron] semconv is probably not a good candidate
https://github.com/open-telemetry/opentelemetry-specification/tree/main/specification/trace/semantic_conventions
 - [NathanielRN] What does "Experimental" mean if Specifications [are already 1.0](#)?
 - It means that Trace/Context/Baggage APIs (think methods and such) are stable, but Attributes (like above) *are not*
- [dyladan] Should the instrumentation API be part of the initial stable release?
- [svnm] Worth to continue the discussion on
<https://github.com/open-telemetry/opentelemetry-js-contrib/issues/552?>
- PRs
 - [NathanielRN] [Change default HTTP exporter port to 55681 #2331](#) - Ready to merge?
 - [dyladan] [ignore backwards compatibility in renovate](#)
 - This one comes from a suggestion from the renovate folks
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2343>

- Required by spec
- [feat: updated spec to v1.5.0 and renamed resource class](#) (no need to discuss just as reference)

2021-07-07

Attendees

- Aaron Abbott (Google)
- Bartłomiej Obecny (Lightstep)
- Nev Wylie (Microsoft)
- Will Armiros (AWS)
- Vera Reynolds (Honeycomb)
- Severin Neumann (Cisco)
- Rauno Viskus (Splunk)
- Amir Blum (Aspecto)
- Yaniv Davidi (Aspecto)

Agenda

- Dyladan out today for wisdom tooth extraction
- Examples still to be decided how we want to structure them
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/issues/552>
- Api & core Docs update - Dyladan is working on that
- Website docs -> <https://github.com/open-telemetry/opentelemetry-js/pull/2324>
- [improve naming sdks](#)
- core and contrib 0.23 released
-
-

2021-06-30

Attendees

- Aaron Abbott (Google)
- Daniel Dyla (Dynatrace)
- Nev Wylie (Microsoft)
- Bartłomiej Obecny (Lightstep)
- Alan Storm (Elastic)
- Severin Neumann (Cisco)
- Weyert de Boer (Tapico)
- Rauno Viskus (Splunk)

Agenda

- [Core 0.23.0 proposal](#)

- [obecny] Please create an issue when you want to start working on a new instrumentation - this will allow to avoid duplications
- [dyladan] Is anything blocking
<https://github.com/open-telemetry/opentelemetry-js/pull/2266> ?
- [svnm] there are a few open tickets around documentation and also <https://github.com/open-telemetry/opentelemetry-js/issues/1683>, can we chat on what to do with the docs (have them in the repo, on the website?)
- [dyladan] what is the current state of our metrics effort? Is anyone working on figuring out what needs to be done to bring us up to date?
- [dyladan] I am working on documentation for the API repository and core repository (including website docs). What can we do to make sure these docs are kept more up to date?
- [dyladan] I need a volunteer to update the examples in the core and contrib repositories, and ideally come up with a process to ensure they don't become out of date as quickly as they seem to do.
 - Weyert is volunteering to try to come up with a process to keep examples up to date
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/issues/552>
 -
- [dyladan] Blockers for SDK RC/GA
 - Tooling does not allow us to release experimental modules separately right now
 - Microsoft beachball
(<https://microsoft.github.io/beachball/overview/configuration.html#options>)
 - Documentation is extremely out of date. Looking for a volunteer to help us in this area. Anyone willing to help with this please reach out to me since I am actively working on this at the moment.
 - Others?
 - [vmarchaud] [improv: naming sdks](#) ?
- [dyladan] Contrib repository often gets left behind. Since work on API/SDK is slowing down, it is time to address this. Ideas?
 - Assign an approver/maintainer to "own" contrib?
 - Auto-assign each PR to one of a list of approvers? Assigned approver would be responsible to shepherd the PR through the process.
 - Other ideas?
- [weyert] semantic conventions: [ResourceAttributes](#)
- If you have PRs that need review, put them here:
 - [svnm] <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/498/>

2021-06-23

Attendees

- Nev Wylie (Microsoft)
- Aaron Abbott (Google)
- Bartłomiej Obecny (Lightstep)

- Vamsee Lakamsani (ServiceNow)
- Daniel Dyla (Dynatrace)
- Vera Reynolds (Honeycomb)
- Alan Storm (Elastic)
- Severin Neumann (Cisco)

Agenda

- [Contrib 0.22.0 release PR](#)
- New instrumentation PRs
 - [Memcached](#)
 - [Cassandra-driver](#)
- Response Hook support
 - [Postgres](#)
 - [Mongo](#)
- [OT-shim logging update \(required by spec\)](#)
- [Grpc default URL](#)
- [Cannot instrument -beta versions](#)
- [b3 extract check for array](#)
- Metrics: a Dynatrace engineer will be assigned to bring metrics up to date
- [aaron] are you looking for more approvers?
 - Requirements
<https://github.com/open-telemetry/community/blob/main/community-membership.md#approver>
 - Pick reviewers automatically from the list
<https://docs.github.com/en/organizations/organizing-members-into-teams/managing-code-review-assignment-for-your-team>
 -

2021-06-16

Attendees

- Rauno Viskus(Splunk)
- Aaron Abbott (Google)
- Severin Neumann (Cisco)
- Daniel Dyla (Dynatrace)
- Weyert de Boer (Tapico)
- Valentin Marchaud
- Vera Reynolds (Honeycomb)
- Will Armiros (AWS)
- Marten Hennoch (Splunk)
- Nev Wylie (Microsoft)
- Alan Storm (Elastic)

Agenda

- Bart on vacation
- Dan vacation mon-tues next week
- <https://github.com/open-telemetry/opentelemetry-js-api/releases/tag/v1.0.0>
 - <https://github.com/open-telemetry/opentelemetry-js-api/pull/96>
- [0.22.0 release](#) (uses API 1.0)
- [Include source files in package](#) ?
 - -- inline source maps
- [Enable esModuleInterop](#) ?
 - Nev has a flag that will help this user
- [Collector auto-shutdown performance concerns](#)
- [ReadableSpan should export dropped counts](#)
- [rauno56] Steps to publishing 1.0?
- [rauno56] Metrics signal - plans, roadmap, focused effort, etc?
- [vreynolds] <https://github.com/open-telemetry/opentelemetry-js-api/issues/95>

2021-06-09

Attendees

- Nev Wylie (Microsoft)
- Daniel Dyla (Dynatrace)
- Rauno Viskus (Splunk)
- Bartłomiej Obecny (Lightstep)
- Aaron Abbott (Google)
- Will Armiros (AWS)
-
-

Agenda

- Maintainer vacations
 - Dan this friday the 11th
 - Bart 10th to 20thg
- [API 1.0.0 release proposal](#)
 - <https://github.com/open-telemetry/opentelemetry-js-api/issues/91>
- [Core 0.21 release proposal](#)
- [Contrib update core/api to 0.21.0](#)
 - Depends on core 21 release obviously
 - Automated release following merge
- [Node 16 tests broken in API](#)
 - Fix: <https://github.com/open-telemetry/opentelemetry-js-api/pull/93>
- Pull Requests
 - [Auto-flush batch span processor on visibility change](#)
- API track breaking changes with tooling
 - Any update from msnev?
- API changes can break SDK compatibility

- No update from Rauno: haven't had a moment to put much thought into it.
- No update WRT independent versioning in core/contrib
- Why is this failing for node 16 -> <https://github.com/open-telemetry/opentelemetry-js/pull/2261> valid failure or ?

2021-06-02

Attendees

- Nev Wylie (Microsoft)
- Daniel Dyla (Dynatrace)
- Vera Reynolds (Honeycomb)
- Severin Neumann (Cisco)
- Bartłomiej Obecny (Lightstep)
- Rauno Viskus (Splunk)
- Alan Storm (Elastic)

Agenda

- All spec review issues closed
- After [API#78](#) there will be no breaking changes to API
 - Does not affect SDK core/contrib
 - Additive changes and bug fixes still allowed
 - 0.21 will be released
 - Expect 1.0 to be a re-release of the same code this time next week
- [Core 0.20.0 Release Proposal PR](#)
- Browser tests fixed by rolling back karma-webpack to 4.x
- Before 1.0 release should we rename metrics and metrics api to be `metrics-api-experimental` and `metrics-experimental`?
 - All of core currently versions together. If we don't update names, the metrics API and SDK will also get version 1.0 which will be confusing since they are not stable. If we do update names, we will have metrics-api-experimental@1.0 which may break even on minor version bumps.
 - Alternatively, we could version core packages separately? The [versioning and stability doc](#) in the spec discourages this
- Versioning doc is no longer really required since there is a doc in the spec which already says everything we would say in more detail. Updated to link to that doc <https://github.com/open-telemetry/opentelemetry-js/pull/1737>
 - API and SDK may have different major versions
 - Within one major version of the SDK, a single API major version should be targeted
 - Within one major version of the API, a single spec version should be targeted
- [Tests fail on node 16](#)
 - Certs need to be regenerated to not use md5 signature
- [Auto-flush batch span processor on visibility change](#)
- Experimental packages to keep 0.x version

- Metrics
- Metrics API
- Opentracing shim
- API track breaking changes with tooling
 - Msnev will create an issue
- API changes can break SDK compatibility
 - Rauno will create an issue to track

2021-05-26

Attendees

- Daniel Dyla
- Bartłomiej Obecny (Lightstep)
- Valentin Marchaud
- Vera Reynolds (Honeycomb)
- Rauno Viskus (Splunk)
- Will Armiros (AWS)
- Nev Wylie (Microsoft)
- Severin Neumann (Cisco/AppDynamics)
- Marten Hennoch (Splunk)

Agenda

- API 0.20.0 released
- Browser tests fail locally in core. Are they working for anyone else? They work in CI
 - <https://github.com/open-telemetry/opentelemetry-js/issues/2231>
 - **npm run test:browser**
- Core updated to use API 0.20.0
- Spec review
 - Only API issue remaining from review is to remove Noop classes and singletons
 - <https://github.com/open-telemetry/opentelemetry-js-api/issues/50>
 - Core PRs to remove references:
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2229>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2230>
 - SDK
 - Default service.name in resource
 - Use string "<unknown OpenTelemetry JS service>"
 - Implement [startActiveSpan](#)
- Prettier is removed. Linting might not be as "smart" for auto fixes but we now have more control over the rules
- Last API breaks
 - [API#78](#) is waiting for an opinion from the TC.
 - [API#46](#) removing singletons waiting on core PRs above
 - Once they are resolved there will be no more breaking changes made to API unless required by spec.

- Need reviews (please add your PRs here if you feel they are ready for review and have not received one)
 - [feat: add browser extension](#)

2021-05-19

Attendees

- Nev Wylie (Microsoft)
- Amir Blum (Aspecto)
- Marten Hennoch (Splunk)
- Will Armiros (AWS)
- Aaron Abbott (Google)
- Rauno Viskus (Splunk)
- Vera Reynolds (Honeycomb)
- Bartłomiej Obecnny (Lightstep)
- Martin Kuba (New Relic)

Agenda

- API 1.0.0-rc.1 release issue
 - <https://github.com/open-telemetry/opentelemetry-js-api/issues/74>
- API 0.19.0 released
 - Core update in progress. Blocked by broken test and lint
 - Core update depends on <https://github.com/open-telemetry/opentelemetry-js/pull/2202> which has failing tests because of grpc exporter
- GRPC exporter tests currently broken
 - Appears to be related to TLS because the tests only fail if TLS is enabled
 - Exporter itself seems to work. only the tests are failing
- Linting failure
 - Prettier updated their brace style. This is not configurable
 - Remove prettier - <https://github.com/open-telemetry/opentelemetry-js/pull/2204>
- Metrics protocol and data model declared stable. API/SDK spec progressing and experimental implementations are starting. We should make an effort to bring our metrics up to date

2021-05-12

Attendees:

- Daniel Dyla
- Nev Wylie (Microsoft)
- Bartłomiej Obecnny (Lightstep)
- Vera Reynolds (Honeycomb)
- Valentin Marchaud

- Will Armiros (AWS)
- Rauno Viskus (Splunk)
- Matt Wear (Lightstep)
- Amir Blum (Aspecto)

Agenda:

- Release API rc.1?
 - Merge suppress instrumentation changes first
- startActiveSpan() helper: decide whether defined in interface or purely in api, ref.: https://github.com/open-telemetry/opentelemetry-js-api/pull/54#discussion_r627867589
 - Add to interface
 - No need for a base class right now
- [dan] I believe api-extension package not needed at this time
 - Suppress tracing in SDK/core
 - startActiveSpan in API (allowed explicitly by spec)
 - Instrumentations can use suppress tracing by either depending on core or using symbol.for
 - This makes it clear that suppressTracing only works with our SDK, and may have no effect if used with third party SDKs
 - Are there other “helper APIs” we would want to add to an API extension package? If so, we can still make one
- [dan] default service.name causes a lot of problems
 - All exporters have their own mechanism to set service names including some default logic. This logic is not consistent between exporters.
 - Enforcing default service.name in resource constructor (current PR) causes issues with merging resources together. Name is frequently overwritten.
 - Proposal: put default service.name logic in the TracerProvider. When the resource is passed to the TracerProvider, check to see if a service.name is set. If not, set one. Remove all service name logic from exporters.
- [vmarchaud]: [proposition about baggage utils](#)
- Bugs
 - [Sampler doesn't honor SpanOptions#root](#)
 - [PR](#)
- Spec review issues
 - [TracerProvider should expose forceFlush\(\)](#)
 - [PR](#)
 - [Rename TracerParams to SpanLimits](#)
 - [PR](#)
 - [TracerProvider's newly set Resource should have service name set](#)
 - See above discussion
 - [Consider not exposing type + singleton](#)
 - DD working on this. Requires changes in at least 2 repos. API should merge last
 - API PR done, core in progress

- In order to remove these singletons and noop implementations we need wrapSpanContext. PR is blocked by API release
- [\[spec\] Do not export NOOP implementations](#)
 - Assigned to DD. May also require core changes
- [Will] Timeline for SDK RC?

2021-05-05

Attendees:

- Marten Hennoch (Splunk)
- Rauno Viskus (Splunk)
- Vera Reynolds (Honeycomb)
- Daniel Dyla (Dynatrace)
- Mitar Milanovic (Dynatrace)
- Bartłomiej Obecny (Lightstep)
- Valentin Marchaud
- Matt Wear (Lightstep)
- Aaron Abbott (Google)
- Nev Wylie (Microsoft)
- Severin Neumann (Cisco)

Agenda:

- [Remove suppress instrumentation](#)
- Bugs
 - [PostgreSQL instrumentation uses wrong attributes](#)
 - [bug: tracing sdk depend on src dir from core](#)
 - [PR](#)
 - [Sampler doesn't honor SpanOptions#root](#)
 - [PR](#)
 - <https://github.com/open-telemetry/opentelemetry-js/issues/2162>
 - Last week assigned to obecny. Any chance to look at it?
 -
- Env config PRs need review [Issue](#)
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2100>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2117>
- Spec review issues
 - [TracerProvider should expose forceFlush\(\)](#)
 - Assigned to weyert
 - [Rename TracerParams to SpanLimits](#)
 - Assigned to weyert
 - [Implement Baggage support for Jaeger propagation](#)
 - Done pending conflict resolution
 - [TracerProvider's newly set Resource should have service name set](#)
 - Done pending tests and reviews

- [Consider hiding/moving Event/TimedEvent](#)
 - Obecny working on this. Requires changes in 3 repos. API should merge last
 - PRs all open. Please review
- [Consider not exposing type + singleton](#)
 - DD working on this. Requires changes in at least 2 repos. API should merge last
 - API PR done, core in progress
- [Use whole spancontext for link](#)
 - Want maintainer review if he has time
- [\[spec\] Do not export NOOP implementations](#)
 - Assigned to DD. May also require core changes
- [Rauno] [Span hierarchy if HTTP Instrumentation is not installed](#)
 - Severin will make an issue/pr to document the http dependency
- [vmarchaud]: [Express instrumentation depends on our SDK](#)
 - Create a new context key for HTTP span inside instrumentation pkg
 - Open spec issue to discuss if this needed across SIGs

2021-04-28

Attendees:

- Daniel Dyla
- Vera Reynolds (Honeycomb)
- Nev Wylie (Microsoft)
- Valentin Marchaud
- Will Armiros (AWS)
- Bartlomiej Obecny (Lightstep)

Agenda:

- Previous document somehow became private and we're not sure who owns it. Our current best guess is a previous maintainer who is on vacation right now. Sorry for the inconvenience.
- From spec meeting/review
 - Suppress instrumentation context is not an approved API
 - We should be exporting ONLY spec'd APIs if possible
 - This means removing noop exports we currently have if possible
 - [Spec PR](#) to add it meeting some resistance
- Bugs
 - [Sampler doesn't honor SpanOptions#root](#)
 - [ParentBasedSampler should not drop invalid SpanContext](#)
 - [Spans from Node.js code NOT registering with the collector.](#)
 - [instrumentation-grpc: Undefined path attribute in clientMethodTrace](#)
- [improv: naming sdks](#)

- [SDK spec review issues](#)
- [API spec review issue](#)

2021-04-21

Attendees

- Daniel Dyla
- Valentin Marchaud
- Nev Wylie (Microsoft)
- Rauno Viskus (Splunk)
- Bartlomiej Obecny (Lightstep)
- Weyert de Boer
- Amir Blum (Aspecto)
- Will Armiros (AWS)
- Marten Hennoch (Splunk)

Agenda

- Release
 - Core 0.19 PR approved ~~but waiting on~~ <https://github.com/open-telemetry/opentelemetry-js/pull/2135>
 - Tests passed. Merge/release after meeting
 - Contrib update to 0.19 done locally, waiting for release to make PR
 - Contrib release automation should make this feasible to do today
- Spec ToC review - Thanks to Carlos
 - API - Tasks to consider/do/close before 1.0.0
 - [Consider renaming Span.context\(\) to spanContext\(\)](#)
 - [Rename NoopSpan to NonRecordingSpan](#)
 - [Consider moving setSpan\(\)/getSpan\(\) to its related subsection](#)
 - [Consider moving the diag/ section to core](#)
 - [createContextKey\(\) should use the string as debug information only](#)
 - [Consider not exposing type + singleton](#)
 - [Consider using SpanContext for Link](#)
 - [Consider hiding/moving Event/TimedEvent](#)
 - SDK - tasks to consider/do/close before 1.0.0-rc.0
 - Required by spec 1.0
 - [Implement Baggage support for Jaeger propagation](#)
 - [Rename TracerParams to SpanLimits](#)
 - [Consider not exposing getActiveSpanProcessor at Tracer level](#)
 - [TracerProvider's newly set Resource should have service name set](#)

- Required by spec 1.1
 - [TracerProvider should expose forceFlush\(\)](#)
 - Recommended
 - [Consider adding the Propagator suffix to tracecontext/baggage propagators](#)
 - [TraceContext should exist under trace/](#)
 - [Consider renaming JaegerHttpTracePropagator to JaegerPropagator](#)
- Env config for exporters - required for 1.0-rc
 - SDK ([issue](#)): [PR opened](#),
 - @vmarchaud: [Clarification from the spec](#), need to update PR
 - Collector ([issue](#)): [PR opened](#) [new PR](#)
 - pls review
- Continued work on Lambda
 - PR: <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/431>
 - Pls review

2021-04-14

Attendees

- Bartłomiej Obecny (Lightstep)
- Valentin Marchaud
- Vera Reynolds (Honeycomb)
- Will Armiros (AWS)
- Nev Wylie (Microsoft)
- Matt Wear (Lightstep)
- Weyert de Boer
- Rauno Viskus (Splunk)
- Vinod Vydier (Newrelic)
- Jakub Malinowski (Splunk)

Agenda

- <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/417>
 - Concerned about NPM PUT 404 issue: https://github.com/open-telemetry/opentelemetry-js-contrib/runs/2272045649?check_suite_focus=true#step:6:543
 - npm login helps, but not best practice on CI: <https://github.com/npm/cli/issues/1637#issuecomment-678030871>
- <https://github.com/open-telemetry/opentelemetry-js/pull/2102>
 - Lambda Resource detector - small PR looks like review is in progress
 - pls review
- <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/421>
 - X-Ray propagator being upstreamed
 - pls review

- towards SDK RC
 - ◉ ~~All plugins converted to instrumentation~~
 - ◉ ~~Release core - DD~~
 - ◉ ~~Release contrib - DD~~
 - ◉ ~~NPM deprecate - DD~~
 - ◉ ~~Remove plugins (Core)~~
 - ◉ ~~Remove base plugin and loader - bart~~
 - ◉ ~~Env config for propagators: [PR merged](#)~~
 - ◉ Remove plugins (Contrib)
 - Assigned to DD , Bart will continue updating examples and readme
 - ◉ ~~Remove old metapackages~~
 - ◉ Env config for exporters
 - SDK ([issue](#)): [PR opened](#), vmarchaud will raise an issue with spec.
 - Basically we can either drop the support or try to come out with some registering mechanism to achieve that, but for web it would not make sense to bundle everything, with node it might vary but still
 - Collector ([issue](#)): [PR opened](#)
 - pls review
 - ◉ Env config for sampling:
 - <https://github.com/open-telemetry/opentelemetry-js/issues/1675#issuecomment-817188179>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/2111>
 - OTEL_TRACES_SAMPLER and OTEL_TRACES_SAMPLER_ARG

2021-04-07

Attendees

- Nev Wylie (Microsoft)
- Daniel Dyla (Dynatrace)
- Vera Reynolds (Honeycomb)
- Will Armiros (AWS)
- Michael Haberman (Aspecto)
- Bartlomiej Obecny (Lightstep)
- Valentin Marchaud
- Marten Hennoch (Splunk)
- Weyert de Boer (Tapico)

Agenda

- [dyladan] has will had time to look into the publishing step failure?
 - ◉ Lerna-changelog does not support package directories other than 'packages'
 - ◉ Changelog in second workflow is regenerated even if the changelog.md file was manually fixed

- Install and bootstrap step was never run for the publishing step
- towards SDK RC
 - All plugins converted to instrumentation
 - ~~○ Release core - DD~~
 - ~~○ Release contrib - DD~~
 - ~~○ NPM deprecate - DD~~
 - Remove plugins - Bart core DD contrib
 - Working on this today
 - Remove base plugin and loader - bart
 - [Remove Base Plugin](#)
 - [Remove plugin loader](#)
 - Work is partly done, PR waiting on release
 - Remove old metapackages
 - Env config for propagators
- Presenting ext-js repo (<https://github.com/aspecto-io/opentelemetry-ext-js>)
- [weyert] working on autgeneration of the semantic conventions
 - <https://github.com/open-telemetry/build-tools>

2021-03-31

Attendees

- Daniel Dyla
- John Bley (Splunk)
- Vera Reynolds (Honeycomb)
- Aaron Abbott (Google)
- Bartłomiej Obecny (Lightstep)
- Weyert de Boer (Tapico)
- Nev Wylie (Microsoft)
- Will Armiros (AWS)
- Vinod Vydier (Newrelic)
- Matt Wear (Lightstep)

Agenda

- [Contrib create release PR action failing](#)
 - Contrib release required in order to finish plugin deprecation
 - [Manual PR](#) for the time being
- towards SDK RC
 - All plugins converted to instrumentation
 - ~~○ Release core - DD~~
 - Release contrib - DD
 - Blocked by release action fail
 - NPM deprecate - DD
 - Blocked by release
 - Remove plugins - DD

- Blocked by release
- Remove base plugin and loader - bart
 - [Remove Base Plugin](#)
 - [Remove plugin loader](#)
 - Work is partly done, PR waiting on release
- Remove old metapackages
 - Blocked by release
- [NoopContextManager infinite loop](#)
 - Diag log on incompatible API versions easily missed
 - Span exports should not be traced in manual tracing situations (no context manager)
 - <https://github.com/open-telemetry/opentelemetry-js-api/issues/27>
- [0.18.1 was actually breaking](#)
 - Fixed with 0.18.2
 - 0.19.0 PR after [1.0.0-rc.0 upgrade](#)
- <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/399>
 - Lambda instrumentation PR

2021-03-24

Attendees

- Nev Wylie (Microsoft)
- Daniel Dyla (Dynatrace)
- Valentin Marchaud
- Bartłomiej Obecný (Lightstep)
- Matt Wear (Lightstep)

Agenda

- [Add ESM build](#)
 - Increase size of npm module tarball
 - Better for bundlers/minifiers
- [Context activation for outgoing requests](#)
 - User wants the “data” event callback to use the outgoing request as a parent
- SDK using RC version of API release this week
 - Release PR today. If there are blocking issues comment there please
 - Contrib to follow immediately
- towards SDK RC
 - All plugins converted to instrumentation
 - Release core - DD
 - Release contrib - DD
 - NPM deprecate - DD
 - Remove plugins - DD
 - Remove base plugin and loader - bart
 - [Remove plugins - deprecate them in npm](#)

- DD will npm deprecate today
 - Plugins
 - Plugin metapackages
 - DD will create a PR to remove plugins
 - Cannot be merged until contrib release with metapackages
- [Remove Base Plugin](#)
- [Remove plugin loader](#)
- Remove old metapackages
- Metapackages for web and node done
- Update compatibility matrix - done
- <https://github.com/open-telemetry/opentelemetry-js/pull/2031>
 - Simple documentation pr needs review
- <https://github.com/open-telemetry/opentelemetry-js/pull/1931> becoming stale?
 - Move jaeger propagation to core
- [DD will try to get a TC member to review the RC](#)

2021-03-17

Attendees

- Bartłomiej Obecny (Lightstep)
- Rauno Viskus (Splunk)
- Will Armiros (AWS)
- Nev Wylie (Microsoft)
- Weyert de Boer (Tapico)
- Vinod Vydier (Newrelic)

Agenda

- API RC released last week
- towards SDK RC
 - Mysql Plugin - being worked on by DD
 - this is the last plugin that left before we can start removing old plugins and all other pieces of code that were related to old plugin system
 - [Remove plugins - deprecate them in npm](#)
 - [Remove Base Plugin](#)
 - [Remove plugin loader](#)
 - Remove old metapackages
 - Meta package for node done
 - [Meta package for web](#) ready for reviews
- Update compatibility matrix

2021-03-10

Attendees

- Daniel Dyla
- Nev Wylie (Microsoft)
- Valentin Marchaud
- Rauno Viskus (Splunk)
- Will Armiros (AWS)
- Bartłomiej Obecny (Lightstep)
- Matt Wear (Lightstep)

Agenda

- Contrib Timelines
- API RC
- RC timeline
 - API RC - pending all maintainer approval
 - [First documentation PR](#)
 - More incoming
 - API RC PR today?
 - Maintainers consent unanimously
 - Deprecate plugins in favor of instrumentations
 - Depends on instrumentation conversions
 - ~~[refactor: migrate express to instrumentation #252](#)~~
 - ~~[refactor: migrate mongodb to instrumentation #250](#)~~
 - ~~[chore: converting pg plugin and pg-pool to instrumentation](#)~~
 - Koa: [Moving to Koa Instrumentation](#)
 - Mysql being worked on by DD
 - New version to support mysql and mysql2 if possible
 - Hapi waiting for review: [Moving Hapi Plugin to Instrumentation](#)
 - Redis waiting for review: [feat: migrate redis to instrumentation #246](#)
 - Instrumentation auto-installation and/or auto-enable
 - [Meta Package PR](#) please review
 - SDK RC
 - [Remove plugins - deprecate them in npm](#)
 - [Remove Base Plugin](#)
 - [Remove plugin loader](#)
 - Remove old metapackages
 - Depends on auto-installation packages
 - Possible TC review
 - API/SDK release

- Contrib repo release automation:
<https://github.com/open-telemetry/opentelemetry-js-contrib/pull/366>
- Additional contributing guidelines for vendor-specific components:
<https://github.com/open-telemetry/opentelemetry-js-contrib/pull/338>

2021-03-03

Attendees

- Nev Wylie (Microsoft)
- Aaron Abbott (Google)
- Daniel Dyla (Dynatrace)
- John Bley (Splunk)
- Bartłomiej Obecny (Lightstep)
- Weyert de Boer (Tapico)

Agenda

- API RC
- RC timeline
 - ~~Diag logger merge - after reviews~~
 - ~~Release 0.17~~
 - ~~Move API to own repo~~
 - <https://github.com/open-telemetry/opentelemetry-js-api>
 - Dyladan is refreshing API documentation/readme for the new repo
 - ~~Remove API and Context base packages from core~~
 - [#1942](#)
 - ~~Use semver for API compatibility checking~~
 - [1772](#) will be moved to the new repository
 - [#6 Opened](#)
 - ~~Closed because it contained diag logger refactor as well as semver global changes~~
 - [#9 Diag Refactor](#)
 - [#10 Use semver to determine global compatibility](#) depends on #9
 - API RC - pending all maintainer approval
 - Deprecate plugins in favor of instrumentations
 - Depends on instrumentation conversions
 - ~~[refactor: migrate express to instrumentation #252](#)~~
 - ~~[refactor: migrate mongodb to instrumentation #250](#)~~
 - ~~[chore: converting pg plugin and pg pool to instrumentation](#)~~
 - Up for grabs
 - ~~Hapi~~
 - Mysql
 - DD is planning on taking this after API changes are made

- New version to support mysql and mysql2 if possible
- Redis
 - Koa being worked on by obecny
- Instrumentation auto-installation and/or auto-enable needs to be designed and implemented for the new instrumentations
- SDK RC
- API/SDK release
- [aaron abbott] peerDependencies?
 - <https://github.com/open-telemetry/opentelemetry-js/issues/723>
 - GCP exporter is using peerdependencies
 - <https://github.com/GoogleCloudPlatform/opentelemetry-operations-js/blob/ba78ad813e000708fda9d8954fd2af0880f0d7a2/packages/opentelemetry-cloud-trace-exporter/package.json#L64-L68>

2021-02-24

Attendees

- Daniel Dyla
- Bartlomiej Obecny (Lightstep)
- Nev Wylie (Microsoft)
- John Bley (Splunk)
- Will Armiros (AWS)

Agenda

- [will] contrib update
 - Workflow PR for contrib should be today or tomorrow
 - Xray id gen and propagator coming soon as well
 - <https://github.com/willarmiros/test-package-lock-repo/pull/18>
- RC timeline
 - ~~Diag logger merge after reviews~~
 - ~~Release 0.17~~
 - ~~Move API to own repo~~
 - <https://github.com/open-telemetry/opentelemetry-js-api>
 - Dyladan is refreshing API documentation/readme for the new repo
 - Remove API and Context base packages from core
 - [#1942](#)
 - Use semver for API compatibility checking
 - ~~[1772](#) will be moved to the new repository~~
 - ~~[#6 Opened](#)~~
 - Closed because it contained diag logger refactor as well as semver global changes
 - [#9 Diag Refactor](#)
 - [#10 Use semver to determine global compatibility](#) depends on #9

- API RC - pending all maintainer approval
- Deprecate plugins in favor of instrumentations
 - Depends on instrumentation conversions
 - [refactor: migrate express to instrumentation #252](#)
 - [refactor: migrate mongodb to instrumentation #250](#)
 - [chore: converting pg plugin and pg-pool to instrumentation](#)
 - Up for grabs
 - Hapi
 - Mysql
 - DD is planning on taking this after API changes are made
 - New version to support mysql and mysql2 if possible
 - Redis
 - Koa being worked on by obecny
- Instrumentation auto-installation and/or auto-enable needs to be designed and implemented for the new instrumentations
- SDK RC
- API/SDK release
- <https://github.com/open-telemetry/opentelemetry-js/pull/1931> - import history or no?

2021-02-17

Attendees

- Daniel Dyla
- Nev Wylie (Microsoft)
- Valentin Marchaud
- Bartlomiej Obecny (Lightstep)
- John Bley (Splunk)
- Will Armiros (AWS)

Agenda

- Enabling propagators with ENV variables (moving contrib propagators to core)
 - 3rd party propagators
 - <https://github.com/open-telemetry/opentelemetry-specification/blob/main/specification/context/api-propagators.md#propagators-distribution>
 - Jaeger propagator should move to core because it is required by spec
 - Ot trace and xray will continue to live in contrib as they do now
 - Propagators will be optional peer dependencies of the node sdk package
 - Propagators will require manual installation in the web
- RC timeline
 - Diag logger merge - after reviews
 - Release 0.17
 - Move API to own repo
 - Dyladan is refreshing API documentation/readme for the new repo
 - Use semver for API compatibility checking

- [1772](#) will be moved to the new repository
- API RC - pending all maintainer approval
- Deprecate plugins in favor of instrumentations
 - Depends on instrumentation conversions
- SDK RC
- API/SDK release

2021-02-10

Attendees

- Daniel Dyla
- Nev Wylie (Microsoft)
- Bartłomiej Obecny (Lightstep)
- John Bley (Splunk)
- Will Armiros (AWS)
- Rebecca Rodriguez (New Relic)
- Valentin Marchaud
- Alan Storm (Elastic)
- Matt Wear (Lightstep)

Agenda

- API move to dedicated repository
 - Waiting on diag logger
 - Repository layout
 - Context base package?
- GA Schedule
 - API RC - this week i hope
 - SDK RC - next week i hope
 - Should this include instrumentation modules?
 - @vmarchaud must have i think, only need [grpc-js](#) for core
 - Spec 1.0 announcement 2/16
 - Contrib
- Contrib tooling behind core - PR incoming to make them match
 - 1 tsc process
 - Bring dev dependencies to root
 - Run version script on version change not on build
 - Cache lockfiles
- Discuss goals for automated CD workflow
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/issues/345>
 - 2 automated processes that “sandwich” a manual PR
- [Nev] Random question: Do we have the support browser/ environment matrix documented somewhere?

- And what packages are supported on each. If not it would be worth having this stated in the readme for each package to avoid random issues for packages we are explicitly not supporting in those environments

2021-02-03

Attendees

- Daniel Dyla
- Nev Wylie (Microsoft)
- Bartłomiej Obecny (Lightstep)
- Matt Wear (Lightstep)
- John Bley (Splunk)
- Will Armiros (AWS)

Agenda

- Currently when Span ends it sends the whole [self](#) to the span processor. Span Processor expects the span to be ReadableSpan but in fact the span that is being sent is a whole object with methods etc. I think we should send only an object in correct format. We could have for example a private function "_serialize" or "_toReadableSpan" which will send only required data and nothing more. We should also think about sealing the data so it cannot be modified.
 - [Spec](#)
 - Will create an issue to copy the object before onEnd
- Proposal for shared maintenance
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/issues/333>
- Status of OTel Core GA and Contrib GA?
- [Baggage](#)
- <https://github.com/open-telemetry/opentelemetry-js/pull/1855>
- Diag-logger "quick" discussion - timeboxed don't want to go berserk
 - forcedInfo
 - VERBOSE / TRACE log level
 - For loop creation of logger (is for both code reuse and minification)
 - Separate filter logger so that the "filter" logic is not duplicated

2021-01-27

Attendees

- Bartłomiej Obecny (Lightstep)
- Daniel Dyla
- John Bley (Splunk)
- Nev Wylie (Microsoft)
- Jakub Malinowski (Splunk)
- Will Armiros (AWS)

Agenda

- Meeting passcode is 77777
- New release - add very specific note about upgrading and connection between core repo and contrib repo to avoid things like
 - [issue 1](#)
 - [issue 2](#)
 - [issue 3](#)
 - Add info to readme
 - Add API as a peerDependency to each instrumentation with pinned version
- Remind of merging guidelines, please don't rush with merging allow other people to see changes - no merging on weekends?, wait this 24 hours at least.
 - Update merge guidelines - DD
 - No merge on weekends
- Documentation is broken [issue](#)
 - Create release PR this evening - Obecny
- Openness to including vendor-specific packages in contrib repo if their build/release systems are made separate from instrumentation libs. Precise design would come next week, but this would likely involve migration to GitHub Actions for workflows, decoupling the publishing responsibilities, and using CODEOWNERS like in collector-contrib to not increase maintainer burden
 - [Issue in JS repo](#)
 - [Proposal in .NET](#)
 - AWS is prepared to do the required work to support this
- [More specific api type names](#)
- Our baggage implementation is woefully far from spec
 - DD is creating a baggage PR
- [MSNev] Quick update on the api.diag progress
 - Assumption: main is now master?
 - Do we want to close this Issue an"e release.
 - Known possible breaking issues
 - functions that pass Logger as a parameter (and not last parameter) Plugin enable() with new OptionalDiagLogger type (was previously just Logger)

```
export type OptionalDiagLogger = Logger | DiagLogger |
null | undefined;

enable(
  moduleExports: T,
  TracerProvider: TracerProvider,
  logger?: OptionalDiagLogger,
  config?: PluginConfig
): T;
```


- Deprecation and removal of ResourceDetectionConfigWithLogger interface from constructor passing the base interface instead, which defines an optional logger rather than a mandatory one. (This affects the AWS implementations)
- General approach to deprecation

```

    /** Optional Logger.
     * @deprecated This property will be removed prior to
     * v1.0 and instead api.diag will be used
     * for logging internal diagnostic details.
     * @see {@link DiagAPI} and {@link DiagLogger}
     */
    logger?: Logger;
    /** Optional Logger. */
    diagLogger?: DiagLogger;

```

- Environment type change for OTEL_LOG_LEVEL: api.DiagLogLevel.INFO, Was previously core LogLevel.INFO

2021-01-20

Attendees

- Daniel Dyla
- Nev Wylie (Microsoft)
- Bartłomiej Obecny (Lightstep)
- Valentin Marchaud
- Matt Wear (Lightstep)
- Marten Hennoch (Splunk)
- John Bley (Splunk)
- Will Armiros (AWS)

Agenda

- Welcome to approvers John Bley and Gerhard Stoebich
- [chore: 0.15.0 release proposal](#)
- [setAttribute value should be required #1848](#)
- Should tracestate implementation be moved to API?
 - Need clarification about TraceState with the W3C spec being the only implementation or it need to be a interface with the W3C implementation available
 - @dyladan will ask for clarification
- New API repo is created and empty

- Metrics API is now a separate package
- Behavior of getLogger
 - Current PR is quite a large change including removing NoopLogger from API
 - PR removes NoopLogger, but leaves LoggingFunction type. Seems inconsistent
 - Should logging be an API concern at all or should it be entirely moved to SDK/instrumentation?
 - Packages with built-in otel - Use API only, likely have their own logging
 - Instrumentations - Use API only, would need logging helpers
 - Applications manually tracing - Use API and SDK, likely have their own logging
- Status of contrib repo accepting new packages
 - New vendor-specific packages likely won't be considered til after tracing API stabilization
- [Status Span is wrong and needs to be converted before exporting. We should think of aligning this](#) Unset - 0, OK - 1, Error - 2, But it was changed to Unset - 1, OK - 0, Error - 2

<https://github.com/open-telemetry/opentelemetry-proto/blob/master/opentelemetry/proto/trace/v1/trace.proto#L304>

- PRs to look at (if you have some time):
 - [feat: merge grpc-js into grpc instrumentation](#)
 -

2021-01-13

Attendees

- Daniel Dyla
- Nev Wylie (Microsoft)
- Matt Wear (Lightstep)
- John Bley (Splunk)
- Jakub Malinowski (Splunk)
- Bartłomiej Obecny (Lightstep)

Agenda

- Combating test flakiness
 - Http instr tests use localhost for integrations instead of public internet
 - [chore: remove race from tests](#) (needs approval)
 - Bad canary deploy script is next
- API 1.0 prep
 - [chore: split metrics into its own api package](#)
 - [chore: change SpanOptions startTime to TimeInput](#)
 - [fix\(api\): add public 'fields' function to api.propagator](#)
 - Next step is to move API out of lerna

- New repo or just a folder? [Discussion](#)
- Once semver change is merged, each release will break the API compatibility until we move it out of lerna. Recommend holding this change until API is separate.
- Automatically merge PRs [Discussion](#)
- Replace gitter with discussions [Discussion](#)
- Release PR today
 - Contrib patch
 - Core minor
 - Contrib minor
- PRs in need of review
 - [feat: merge user supplied into default tracer config](#)
 - [feat: user config + default config merge for MeterProvider, Meter](#)
 - [fix\(plugin-fetch\): check if PerformanceObserver exists](#)
 - [Allow zero/negative performance timings](#)
 - [feat\(exporter-zipkin\): per-span service name](#)

2021-01-06

Attendees

- Bartłomiej Obecny (Lightstep)
- Nev Wylie (Microsoft)
- Daniel Dyla (Dynatrace)
- Matt Wear (Lightstep)

Agenda

- Most of this agenda is from the 12/23 meeting which had no quorum
- API 1.0 Prep
 - [feat!: use semver to determine API compatibility](#)
 - [chore: clean up API for release](#) (draft)
 - [Pass through context in ProxyTracer.startSpan\(\)](#)
 - Long review process discouraged paul
 - [Add some namespacing in API](#)
 - Other issues? <https://github.com/open-telemetry/opentelemetry-js/issues/1765>
 - [\[API\] `SpanOptions #startTime` is of type `TimeInput` #1793](#)
 - [Span api feels unintuitive and underdocumented](#)
 - [\[API\] Change enums to be const enum -- unless necessary to have the full enum name at runtime · Issue #1753 · open-telemetry/opentelemetry-js \(github.com\)](#)
 - [\[API Suggestion\] Provide getLogger helper and remove NoopLogger class implementation · Issue #1754 · open-telemetry/opentelemetry-js \(github.com\)](#)

- [\[Discussion\] Hide all private properties and functions in constructor closure for classes / implementations · Issue #1756 · open-telemetry/opentelemetry-js \(github.com\)](#)
- API Release Steps
 - Split out metrics API package
 - Move trace/baggage/context API out of lerna
 - Modify build pipeline to build/release these packages separately
- Small tooling fixes
 - ~~[chore: fix docs workflow](#)~~ merged
 - ~~[chore: remove fix option from husky lint](#)~~ merged
 - [chore: improve dev setup for windows](#)
- Larger tooling change
 - [chore: manually hoist dev dependencies](#)
- [feat: creating one auto loader for instrumentation and old plugins](#)
 - Has enough reviews, but it's important so 1-2 more would be nice
 - 29 days old at this point
- [Browser time skew](#)
 - Curious what the thoughts are here

2020-12-30

Attendees

- Daniel Dyla

Agenda

- No agenda. Maintainer will be here to answer questions
- Call closed after 10m with no attendees

2020-12-23

Attendees

- Daniel Dyla
- Jakob Malinowski

Agenda

- Maybe nothing if we don't have a quorum of maintainers/approvers :)
- No quorum. Held office hours
- ~~API 1.0 Prep~~
 - ~~[feat!: use semver to determine API compatibility](#)~~
 - ~~[chore: clean up API for release](#)~~
 - ~~[Pass through context in ProxyTracer.startSpan\(\)](#)~~
 - ~~[Add some namespacing in API](#)~~
 - ~~Other issues?~~ <https://github.com/open-telemetry/opentelemetry-js/issues/1765>

- ~~API Release Steps~~
 - ~~Split out metrics API package~~
 - ~~Move trace/baggage/context API out of lerna~~
 - ~~Modify build pipeline to build/release these packages separately~~
- ~~Small tooling fixes~~
 - ~~[chore: fix does workflow](#)~~
 - ~~[chore: remove fix option from husky lint](#)~~
- ~~Larger tooling change~~
 - ~~[chore: manually hoist dev dependencies](#)~~
- ~~[feat: creating one auto loader for instrumentation and old plugins](#)~~
 - ~~Has enough reviews, but it's important so 1-2 more would be nice~~

2020-12-16

Attendees

- Daniel Dyla
- Nev Wylie (Microsoft)
- Bartłomiej Obecny (Lightstep)
- Valentin Marchaud
- Matt Wear (Lightstep)
- Nick Tzaperas (New Relic)

Agenda

- [feat: creating one auto loader for instrumentation and old plugins](#)
 - Do we still need [Feat/plugin injection](#)
- [chore: speed up builds](#)
- [Important Fix](#) (Fixing Span status when exporting span) after this release fix for core, and then update contrib (again) and release contrib)
- [chore: replace direct process.env usage with getEnv\(\)](#)
 - Scoped down to only the change in the title
- Discussions enabled
- GA <https://github.com/open-telemetry/opentelemetry-js/projects/12>
 - Split metrics API into its own package
 - Move API package out of lerna
 - Still open issues for env var changes
- Consider moving all top-level helpers into an api namespace
- [feat\(grpc-instrumentation\): migrate grpc to instrumentation](#)
 - Need review
 - Will merge with grpc-js inside it
-

2020-12-09

Attendees

- Daniel Dyla
- Bartłomiej Obecny (Lightstep)
- Nick Tzaperas (New Relic)
- Carlo Pearson (New Relic)
- Rebecca Rodriguez (New Relic)
- Hongbo Miao (Tesla)
- Nev Wylie (Microsoft)
- Marten Hennoch (Splunk)
- Valentin Marchaud
- Matt Wear (Lightstep)
- Hector Hernandez (Microsoft)

Agenda

- [Instrumentation autoloader](#)
- [feat: migrate http and https plugin to instrumentation #1658](#)
- [Migrate CircleCI unit test workflow to Github Actions](#)
- Contrib release with core v.013
 - want to include host metrics
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/issues/275>
- [v1.0.0 Versioning Proposal](#)
 - [OTEP](#)
- Open question: is anyone working on a browser distro? (jtmalinowski)

2020-12-02

Attendees

- Daniel Dyla
- Kelvin Lo (AWS)
- Valentin Marchaud
- Matt Wear
- Eunice Kim (AWS)
- Bartłomiej Obecny (Lightstep)

Agenda

- ~~Env vars PR for GA is ready~~
<https://github.com/open-telemetry/opentelemetry-js/pull/1680>
- ~~Before it's merged we need~~
<https://github.com/open-telemetry/opentelemetry-js-contrib/pull/268> merged and released

- ~~— Or: postpone out of the box support for Jaeger~~
- [dyladan] how is it possible that the core PR depends on a contrib PR?
 - [jtmalinowski] partially answered in the PR, probably best if discussion continues in the PR?
- [Prometheus Exporter](#) - discuss best way of handling this case
- [PluginConfig and plugin param](#) - is this something we really want to have and support for longer time ?
- startTime + duration vs startTime + endTime
 - [Spec](#)
 - [Implementation](#)
- Wanted to discuss when a release for issue attached is possible [Kelvin]
 - <https://github.com/open-telemetry/opentelemetry-js/issues/1623>
 - Blocking testing for AWS
 - Today
- RC
 - Timeline
 - Waiting to move plugin config out of API
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1625>
 - Waiting on 1 more review
 - [chore: rename CorrelationContext to Baggage](#)
 -
- Holiday schedule
 - Hold or cancel meeting 12/23?
 - DD will be out 12/30
 - 12/23 and 12/30 meetings will have short/no agenda, but if contributors have something to bring up there will be a maintainer at least for the start of the meeting
- 2 more reviews wanted [Kelvin]
 - Should vendor-specific changes be fast-tracked when contributed by the vendor itself?
 - Do we need to file another issue for migrating the detectors to contrib repo after this is merged in?
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1669>
- [Instrumentation Loader](#)
-

2020-11-25

Attendees

- Daniel Dyla
- Jakub Malinowski
- Bartłomiej Obecny (Lightstep)

- Hongbo Miao (Tesla)

Agenda

- [RC Progress](#)
- [Fix broken lint build step](#)
- Metrics planning
 - Will prepare PR soon with
 - updated [aggregator](#)
 - updated exporter collector to change the temporality
 - rename batcher to processor
 - added missing definition to api (sumobserver, updownsumobserver)
 - Update host metrics in contrib
 - Update runtime metrics
 - Next step after this we should create/rename/refactor to have accumulator [Accumulator](#)

2020-11-18

Attendees

- Daniel Dyla
- Kelvin Lo (AWS Intern)
- Bartłomiej Obecny (Lightstep)
- Nick Tzaperas (New Relic)
- Matt Wear (Lightstep)
- Jakub Malinowski (Splunk)

Agenda

- Baggage context and trace feature freeze
- Where should resource detectors live (core vs contrib)
 - [feat: Add GitHub Actions Resource Detector](#)
 - [Feat: Added Amazon EKS Resource Detector](#)
 - [Opentelemetry-resource-detector-aws](#)
 - [opentelemetry-resource-detector-gcp](#)
- Trace RC todo list
 - Finish environment vars
 - [Correlation context => baggage rename](#)
 - Automated build/release pipeline
 - Integration testing?
 - How to get paid browserstack account
 - Free for OSS
- Reviews wanted:
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1669> (Kelvin)
- Question about Content-Encoding on PR

- <https://github.com/open-telemetry/opentelemetry-js/pull/1625>
- After trace RC
 - Metrics
 - Next week planning?
- Spec requires jaeger propagator as one of available options - does it mean we should move it into the core repo from contrib?
 - May need to use https://nodejs.org/api/module.html#module_module_createRequire_filename
 -

2020-11-11

Attendees

- Daniel Dyla
- Bartłomiej Obecny (Lightstep)
- Carlo Pearson (New Relic)
- Hongbo Miao (Tesla)
- Jakub Malinowski (Splunk)
- Matt Wear (Lightstep)
- Valentin Marchaud
- Srikanth Chekuri

Agenda

- [bart] Let's discuss if we want to create some instrumentation auto loader and how we could achieve that. Do you think we could reuse the plugin loader and try to accept the plugins and instrumentation together (similar to web) or we should create some metapackage etc. or ?
 - Benefits: when users will be upgrading it will be a matter of renaming plugin name.
 - Autoloaders are difficult to make work with yarn pnp. We will have to make our own require using [module.createRequire](#) or similar
- [dyladan] [trace spec freeze](#) imminent. Top priority is ensuring we are compliant with trace 1.0 spec
 - [update js spec compliance matrix](#)
 - "Allow samplers to modify tracestate"
 - Environment variables
- [Plugin deprecation after release - be aware of this ticket](#)
- [vmarchaud] [feat: migrate http plugin to instrumentation #1658](#)
- [jmalinowski] Environment Variable Configuration Spec #1392 (GA Burndown)
 - Span limits done
 - <https://github.com/open-telemetry/opentelemetry-js/issues/1267>
 - Next week coming: propagators, sampler, exporters
 - Week after: jaeger, BSP, prometheus, zipkin

2020-11-04

Attendees

- Daniel Dyla
- Valentin Marchaud
- Matt Wear
- Jakub Malinowski (Splunk)

Agenda

- [tedsuo] User Research / Dogfooding Guide
 - https://docs.google.com/document/d/1rXT2eEAniVM7r8KiikzgJJ-r36shOg_twZP6GPzNvao/edit
 - Now that the tracing spec is stable, we want to ensure the getting started experience for new users is clear and understandable. I've created a user research exercise that walks through the basic operations an end user must understand in order to use OpenTelemetry tracing.
 - Asks:
 - List the version to be tested in the list of languages in the user research guide.
 - Please run through the user research exercise, in order to identify where we need to improve the experience. 🙏🙏🙏
 - Please recruit 1 - 2 people each (perhaps from where you work) to also try the exercise. 🙏🙏🙏
 - Completed studies can be submitted here:
<https://forms.gle/VrrPnCKpMcL2mBQ6A>
- [tedsuo] configuration hella confusing without documentation
 - Options themselves are poorly documented
 - Config option example with every option listed
- Load order
 - Gcp-metadata uses node-fetch. GCP resources load node-fetch before instrumentation, and instrumentation fails
 - Gcp-metadata will have minimum node version 10
- GA Readiness - [Update JS spec compliance matrix](#)
 - [Environment Variable Configuration Spec](#)
 - Jtmalinowski PR maybe tomorrow
 - [fix: change status codes from grpc status codes](#)
 - Needs review
 - Breaking (status codes change meaning)
 - [feat\(opentelemetry-js\): add content size attributes to HTTP spans](#)
 - Waiting on minor refactor from nijotz
 - [docs: document null and undefined attribute values undefined behavior](#)
 - [Rename plugin to instrumentation](#)
 - Dyla will create individual issues for each plugin

- Unassigned
- [Context leaks](#)
 - Some modules are known to breaking context, we should document them

2020-10-28

Attendees:

Please add your name to the attendees list

- Kelvin Lo (AWS Intern)
- Bartlomiej Obecny (Lightstep)
- Daniel Dyla (Dynatrace)
- Matt Wear (Lightstep)
- Valentin Marchaud
- Jakub Malinowski (Splunk)
- Carlo Pearson (New Relic)
- Nick Tzaperas (New Relic)

Agenda

- get a ticket for OTel Community Day:
 - <https://events.linuxfoundation.org/open-telemetry-community-day/>
 - If you would like to get involved, contact austin parker
- [JS Documentation for Website](#)
 - Volunteer wanted
- [mwear] contrib repo
 - [pg plugin bug fix](#)
 - [Timeline for next contrib release](#)
 - Idea: independently version contrib packages, release automatically every time a change is made
 - DD will try to build a GHA for this
 - Also update the canary GHA
- CNCF would like to discontinue supporting (paying for) CircleCI
 - Move all tests to GHA
 - Volunteer wanted
- Other SIGs looking for test drivers if you have experience with other languages and time
- Would like to release contrib
 - [Dependency update PR first](#)
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/221>
- Bugs
 - [exporter-collector-grpc traceId and spanId values in protobuf are incorrect](#)
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1627>
 - [@opentelemetry/metrics: The option "boundaries" for metrics is not propagated to the HistogramAggregator](#)
 - <https://github.com/open-telemetry/opentelemetry-js/issues/1626>

- [BatchSpanProcessor ExportResult other than SUCCESS causes UnhandledPromiseRejectionWarning: 1](#)
 - <https://github.com/open-telemetry/opentelemetry-js/pull/162>
- [GA Burndown](#)
 - Reviews wanted
 - [chore: remove explicit parent option](#)
 - [chore: sampler gets a full context](#)
 - [feat\(opentelemetry-js\): add content size attributes to HTTP spans](#)
 - [feat: span processor onstart receives context](#)
 - Unassigned
 - [Rename plugin to instrumentation](#)
 - [Spec: Document null values as invalid for Attributes](#)
 - [Environment Variable Configuration Spec](#)
 - [spec: isRecording becomes false after span.end\(\)](#)
 - [Set status with StatusCode \(Unset, Ok, Error\)](#)
- Context - after latest changes please explain the "correct" approach to use certain context. Differences between full context and "context" etc.. Why do we have to use now setActiveSpan (which doesn't set active span) - more confusion.
<https://github.com/open-telemetry/opentelemetry-js/pull/1612>

If this function needs to be always used now, why our sdk cannot do it "inhouse" do avoid code "copy paste"

- [mwear] RE: json vs grpc exporter perf (grpc 7-8x faster)
 - Should we consider
 - Gzipping spans
 - Using http keep-alive
 - Re: protobuf in web:
<https://github.com/open-telemetry/opentelemetry-js/issues/515>

2020-10-21

Attendees:

Please add your name to the attendees list

- Daniel Dyla
- Carlo Pearson (New Relic)
- Martin Kuba (New Relic)
- Nick Tzaperas (New Relic)
- Eunice Kim (AWS Intern)
- Jakub Malinowski (Splunk)

Agenda

- Released contrib repo does not work with current core
 - <https://github.com/open-telemetry/opentelemetry-js/issues/1606>

- Release updated contrib
- [Compliance matrix \(Spec PR\)](#)
 - [chore: remove explicit parent option](#)
- [GA Burndown](#)
 - Please review [GA Burndown PRs](#)
- Switching from JSON to protobuf gives a 7-8x speedup. Not available in web

2020-10-14

Attendees:

Please add your name to the attendees list

- Daniel Dyla
- Carlo Pearson (New Relic)
- Nick Tzaperas (New Relic)
- Bartłomiej Obecny (Lightstep)
- Matt Wear (Lightstep)
- Kelvin Lo (AWS Intern)
- Eunice Kim (AWS Intern)
- Hongbo Miao (Tesla)
-

Agenda:

- [TraceState specification non-compliant](#)
- Quick question about [issue 1249](#).
- [chore: 0.12.0 proposal](#)
- [feat: simplify active span logic](#) required for ga
- [feat: add keys operation to getter](#) required for ga

2020-10-07

Attendees:

Please add your name to the attendees list

- Daniel Dyla
- Bartłomiej Obecny (Lightstep)
- Hongbo Miao (Tesla)
- Valentin Marchaud
- Kelvin Lo (AWS Intern)
- Eunice Kim (AWS Intern)
- Matt Wear (Lightstep)

Agenda:

- [mwear] Any thoughts or feedback on: <https://github.com/nodejs/node/pull/34895>
 - We can't really support this well right now
 - Depends on support from the community as well
- [User has requested we publish a new version](#)

- Waiting on context propagation bug
- Async resource detection (discussion from maintainers mtg)
- [bobecny] [Refactoring and updating metrics with collector due to upgrade to proto ver. 0.5.0](#)
- [Proto over JSON should represent ids with hex](#)
- [GA Progress](#)

2020-09-30

Attendees:

Please add your name to the attendees list

- Bartlomiej Obecny (Lightstep)
- Matt Wear (Lightstep)
- Carlo Pearson (New Relic)
- Eunice Kim (AWS Intern)
- Kelvin Lo (AWS Intern)
- Martin Kuba (New Relic)
- Nick Tzaperas (New Relic)

Agenda:

Tentative goals for the next meeting, feel free to add your own!:

- Planning to release in the end of the week, so please help in reviewing some remaining stuff so it can go into next version, the same will go then with contrib repo
- New Plugin system merged - now only package @opentelemetry/instrumentation will be needed
 - once released we can try to start upgrading plugins
- [GA Tracker Update](#)

2020-09-23

Attendees:

Please add your name to the attendees list

- Bartlomiej Obecny (Lightstep)
- Kelvin Lo (AWS Intern)
- Daniel Dyla (Dynatrace)
- Eunice Kim (AWS Intern)
- Mark Wolff
- Matt Wear (Lightstep)

Agenda:

Tentative goals for the next meeting, feel free to add your own!:

- 2 new AWS interns

- Kelvin taking over Cong's x-ray extensions
 - Eunice not working directly on Otel, but on grafana so has some overlap
- Dan Dyla out next week again
- Many PRs, pls review
- GA progress
 - [global error handler](#)
 - 2 reviews from maintainers. Would be nice to get 1 or 2 more from approvers
- Many PRs merged this week. Release?
 - DD will work on release PR this afternoon
 - Try to merge <https://github.com/open-telemetry/opentelemetry-js/pull/1541> before release
- [feat\(node-tracer\): use AsyncLocalStorageContextManager by default starting Node 14.8 #1511](#)
 - Question of which versions should be enabled by default, currently just 14.8+. V14.8 fixed a bug in async local storage. The change introducing the bug was not introduced to v12 yet, but a fix is requested to be backported anyway in case.
 - Leave as 14.8+
- [Plugins](#) v2
 - Does not affect any current functionality. Think we should merge and iterate it as we start actually using it to write plugins. Pls review
- PRs ready for merge soon
 - [feat: add deep-merge util](#)
 - Merge after meeting
 - [feat: make prometheus config startServer mandatory](#)
 - Becoming quite old now
 - 3 approvals incl 2 maintainers
 - If conflicts are fixed merge?
- Duplicate PRs?
 - [fix\(http\): do not set outgoing http span as active in the context #1479](#)
 - [chore: fixing http context propagation for created span on outgoing request](#)
- Please review <https://github.com/open-telemetry/opentelemetry-js/pull/1541> for inclusion in release
- [Feat: Added AWS ECS Plugins Resource Detector](#)
 - Very old PR from an intern. 2 approvals
 - Looks like good work. Merge? No conflicts
- [GA Tracker Update](#)

2020-09-17

Attendees:

Please add your name to the attendees list

- Bartłomiej Obecny (Lightstep)
- Matt Wear (Lightstep)

- Martin Kuba (New Relic)
- Ziv Salzman (AppDynamics)

Agenda:

Tentative goals for the next meeting, feel free to add your own!:

- Many PRS, pls review
- GA progress
 - [global error handler](#)
 - ready for reviews
 - span attributes [PR open](#) - 3 reviews will merge it soon
 - rest are not started yet -> [GA](#)

2020-09-09

Attendees:

Please add your name to the attendees list

- Daniel Dyla
- Bartlomiej Obecny (Lightstep)
- Valentin Marchaud
- Matt Wear (Lightstep)

Agenda:

Tentative goals for the next meeting, feel free to add your own!:

- [CircleCI failing because of downgraded plan](#)
 - <https://github.com/open-telemetry/opentelemetry-js/issues/1259>
 - Try to downgrade to medium first - consider move to actions
 - Create an issue and tag Brandon Kimberly
 - Consider flags to limit resource use
 - NODE_OPTIONS: --max_old_space_size=2000
 - Document strategies for using lerna on slower machines
- V0.10 contrib release
- Contrib now testing node 14
- [dyladan] [Request setup code in bug template](#)
- [dyladan Discuss] [ZoneContextManager doesn't work with async/await code that targets ES2017+](#)
 - <https://github.com/angular/angular/issues/31730>
 - <https://github.com/nodejs/diagnostics/issues/375>
 - async/await works in modern browsers, but there is no API to hook the async context lifecycle
- [dyladan Discuss] [Wreck library content mixup](#)
- [Node 10 context incorrect with setTimeout](#)
- [Enable AsyncLocalStorageContextManager by default?](#)
- [dyladan] [Sync SDK - Async resource detection](#)

- Implementing TestMemoryExporter which yields to event loop before getting finished spans
 - [Broken test](#) was broken but asymptomatic before this PR
- GA progress
 - Dan D
 - attribute spec compliance
 - span attributes [PR open](#)
 - resource attributes PR today
 - Propagation getter.keys next [1486](#)
 - Propagator fields after that [1485](#)
 - Matt Wear
 - Global error handler in progress

2020-09-02

Attendees:

Please add your name to the attendees list

- Daniel Dyla
- Cong Zou (AWS Intern)
- Mark Wolff
- Bartłomiej Obecny (Lightstep)
- Michael Goin (New Relic)
- Matt Wear (Lightstep)
- Hongbo Miao (Tesla)
- Valentin Marchaud
- Martin Kuba (New Relic)

Agenda:

Tentative goals for the next meeting, feel free to add your own!:

- [v0.11.0 Release](#)
- [JS Spec Compliance Matrix update](#)
- [Sync SDK - Async resource detection](#)
- [Unassigned required GA issues](#) triage
- [Bug](#) triage
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/issues/193>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1474>
- Add to issue template “how are you initializing the SDK” so we can ensure it is used correctly when triaging bugs
- Reviews wanted
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1404>(Cong)
 - This PR was on last week’s list - just needs 1 more approval
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1439> 16 days
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1399> 27 days

2020-08-19

Attendees:

Please add your name to the attendees list

- Daniel Dyla
- Bartłomiej Obecny (Lightstep)
- Michael Goin (New Relic)
- Matt Wear (Lightstep)
- Mayur Kale (Google)
- Cong Zou (AWS Intern)

Agenda:

Tentative goals for the next meeting, feel free to add your own!:

- [Error loading plugin when it is already being patched by require-in-the-middle](#)
 - Caused by importing the module within the plugin
 - [Easy to do](#) (sorry to pick on naseem here)
 - Dyladan will update the plugin guide
 - Add a link to the guide from the PR template core and contrib
 - Will also review existing plugins to ensure this bug is not in many places (add import type where appropriate)
- Release new collector exporter modules today
- Sync SDK update
- Plugins 2 update
- Proxy tracing
- Reviews wanted
 - [fix: updates ValueRecorder to allow negative values](#)
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1389>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1404>(Cong)
 -

2020-08-19

Attendees:

Please add your name to the attendees list

- Daniel Dyla
- Bartłomiej Obecny (Lightstep)
- Cong Zou (AWS Intern)
- Mark Wolff
- Hongbo Miao (Tesla)
- Jonah Rosenblum (Google Intern)
- Michael Goin (New Relic)
- Matt Wear (Lightstep)

Agenda:

Tentative goals for the next meeting, feel free to add your own!:

- missing "update button" for contrib repo - fixed
- Canary builds publishing [correctly](#)
 - Except when they [don't](#)
 - Npm install @opentelemetry/api@alpha
- Collector split PR [1446](#)
- Plugins v2 [1449](#)
- Proxy Tracers [1448](#)
 - Allows plugins to instrument modules before tracer provider is constructed/registered
- Shutdown unification [1439](#)
 - All shutdown methods now return promises
- Split resource detectors merged

2020-08-12

Attendees:

Please add your name to the attendees list

- Daniel Dyla
- Cong Zou (AWS Intern)
- Mark Wolff
- Caroline Evans (Google)
- Jonah Rosenblum (Google)
- Matt Wear (Lightstep)
- Bartłomiej Obecny (Lightstep)
- Nina Stawski (Splunk)

Agenda:

Tentative goals for the next meeting, feel free to add your own!:

- [mwear] Discuss approaches to move Resource auto detection out of the initialization code path: <https://github.com/open-telemetry/opentelemetry-js/issues/1410>
 - PR [1400](#) resolves this
- [dyladan] reduce required reviewers by 1?
 - Currently 4, including "both maintainers"
 - Suggest 3, including 2 maintainers
 - 24 hours waiting period so everyone can see the PR - after 3rd review ??
 - Many PRs have 3 approvals right now, but it seems difficult to get the 4th
 - <https://github.com/open-telemetry/opentelemetry-js/pulls?q=is%3Apr+is%3Aopen+sort%3Aupdated-desc+review%3Aapproved>
- GA update
- HAPI and React plugins approved, release this afternoon
- Koa - changes requested unblocked by another PR
 - Ready for review and maybe merge
- Canary publish failing - DD is looking into this
- Michael goin still moving metapackages?
- Mark Wolff still moving resource detectors?

- Simply cut/paste introduces a circular dep
- [jonahrosenblum] Meter/Trace Provider configs
 - Default values do not propagate
 - Defined in spec?
- DD is working on a backwards incompatible-ish Plugins v2
 - should solve the load order issue
 - Should solve yarn pnp issue
 - Should allow a single tracer provider to be shared between web and node
 - Should allow a single plugin to patch multiple modules
 - Configuration will be more straightforward
 - Should support metrics and traces
 - Default to global providers, option to pass a specific provider
 - Request - Support the idea of a different plugin per-version of an instrumented module

2020-08-05

Attendees:

Please add your name to the attendees list

- Bartłomiej Obecnny (Lightstep)
- Daniel Dyla (Dynatrace)
- Michael Goin (New Relic)
- Tina Gao (Google)
- Reza Karimi (Google)
- Terrence Turner (Google)
- David Witten (Google)
- Caroline Evans (Google)
- Cong Zou(AWS Intern)
- Jonah Rosenblum (Google)
- Reg McDonald (Google)
- Mark Wolff

Agenda:

Tentative goals for the next meeting, feel free to add your own!:

- [dyladan]: [Contrib repo - plugins-node-all - broken](#)
 - For now, we need to release contrib
 - We need a long-term solution to fix this
 - Move plugins-all metapackage to contrib
 - michael goin
 - If plugins depended ONLY on the API, the strong version coupling would be avoided
- [dyladan]: W3C Trace Context test suite
<https://github.com/open-telemetry/opentelemetry-js/pull/1380>
 - Currently we fail many of the tests around parsing invalid headers

- How strict should we be?
 - Michael Goin will look into the test failures
- [dyladan]: Many plugins issues to resolve before GA
 - Metrics support [PR](#)
 - Load order issue
 - Currently node-fetch CANNOT be instrumented as the gcp-metadata resource detector loads it too early in the process
 - Mark Wolff will move resource detectors into their own packages
 - Plugins depend on core
 - Plugin loading slows down startup significantly
 - This is the cause of a flaky test
 - Dyladan will create PR to increase the timeout
 - Plugin loading and default context manager the only difference between web and node tracer providers
 - [carolinee21]: Issue regarding BasePlugin support for multiple module names
 - Plugins can only instrument a single module. Many cases like http/https, grpc/grpc-js, and mysql/mysql2 it would be desirable to have a single plugin for both modules
 - Hapi renamed to @hapi/hapi. The plugin should instrument both
- [carolinee21]: Request to merge PR [#144](#): Adding Koa instrumentation
- [dyladan]: Vendor Specific Propagators
 - <https://github.com/open-telemetry/opentelemetry-specification/blob/master/specification/context/api-propagators.md#propagators-distribution>
 - Additional Propagators implementing vendor-specific protocols such as AWS X-Ray trace header protocol are encouraged to be maintained and distributed by their respective vendors.
- Review requests
 - [thgao]: [React Plugin](#)
 - [carolinee21] PR for Hapi instrumentation
 - [dyladan] [chore: deploy canary releases](#)
 - Metrics support [PR](#)

2020-07-29

Attendees:

Please add your name to the attendees list

- Daniel Dyla
- Bartłomiej Obecny (Lighstep)
- Mark Wolff
- Tina Gao (Google)
- Michael Goin (New Relic)
- Bryan Clement (New Relic)
- Rebecca Rodriguez (New Relic)

- Terrence Turner (Google)
- Matt Wear (Lightstep)
- Reza Karimi (Google)
- Mayur Kale
- David Witten (Google)
- Reg McDonald (Google Intern)
- Cong Zou (AWS Intern)
- Eric Mustin (Datadog)
- Caroline Evans (Google intern)
- Nina Stawski (Splunk)

Agenda:

Tentative goals for the next meeting, feel free to add your own!:

- [dyladan] 0.10.0 and 0.10.1 released
 - 0.10.0 had a bug which required an immediate release of 0.10.1
 - Would a canary release process be interesting to anyone? (@0.x.x-`{git-hash}` or @next/@nightly)
 - End to end testing has been open for a long time, will be required for GA
 - We could use the existing examples to test
 - Reliability env test for canary releases before prod release?
- [dyladan] contrib release PR today
 - Currently at 8.x, skip to 10 to sync with main or no?
 - 0.9.0
 - Dyladan will make an issue to add version compat to contrib README
 - This will depend on either GA or on e2e tests
- [dyladan] rename plugin-instrumentation and add contrib
 - Dyladan will handle this this week
- [dyladan] aws xray propagator package
 - <https://github.com/open-telemetry/opentelemetry-js/issues/1335>
 - What name should it have?
 - @opentelemetry/contrib-aws-xray-propagator?
 - Should we have a separate npm namespace for contrib/vendor code?
 - [Cong] approver for <https://github.com/open-telemetry/opentelemetry-js/pull/1331>
- [dyladan] Resource detection interferes with auto instrumentation of node-fetch
 - <https://github.com/open-telemetry/opentelemetry-js/issues/1315>
 - @otel/tracing -> @otel/resources -> gcp-metadata -> node-fetch
 - Dan will create an issue about load order
 - Recent OTEP splits auto detectors into their own modules
 - <https://github.com/open-telemetry/oteps/pull/111>
 - Load order will be very important for async resource detectors that depend on third party modules
- [dyladan] Unassigned issues and PRs waiting review required for GA in the [GA Burndown Project](#)
 - These are high priority

- Please review
 - [feat: Collector Metric Exporter for the Web](#)
 - [feat: Add React Plugin](#)
 - [Adding Koa Instrumentation](#)
 - [feat: Added Summary Data Points to the Collector](#)
 - [fix: OTel breaks apps using @types/node v8](#)
- [rezakrimi] Metric support for plugins:
<https://github.com/open-telemetry/opentelemetry-js/pull/1305>
- [eric mustin] trace id injection in logs
 -

2020-07-22

Attendees:

- Daniel Dyla (Dynatrace)
- Bartłomiej Obecny (Lightstep)
- Eric Mustin (Datadog)
- Tina Gao (Google)
- Reza Karimi (Google)
- Jonah Rosenblum (Google)
- Michael Goin (New Relic)
- Nina Stawski (Splunk)
- Bryan Clement (New Relic)
- Terrence Turner (Google)
- Caroline Evans (Google)
- Cong Zou (AWS Intern)
- Mark Wolff
- Matt Wear (Lightstep)
- David Witten (Google)
- Rebecca Rodriguez (New Relic)
- Valentin Marchaud

Agenda:

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- [dyladan] 0.10.0 release PR this afternoon
- [dyladan] Will merge PR [contrib/110](#) eslint
 - Very large 148 files
 - 3 approvals (2 maintainers)
 - Speak now or hold your peace :)
- [dyladan] would also like to merge [contrib/157](#)
- [dyladan] [discuss] Should plugins use `_only_` the global tracer/meter providers?
- [dyladan] contrib merge guideline doc
- [Cong] [question] Hosting vendor code in otel repos [#1335](#)

- [davidwitten][request for review] [Collector Metric Exporter for Web](#)
- Please review
 - More advanced typing for toPoint avoids casts
<https://github.com/open-telemetry/opentelemetry-js/pull/1325>
 - OpenTracing Baggage shim
<https://github.com/open-telemetry/opentelemetry-js/pull/918>
 - SDK package <https://github.com/open-telemetry/opentelemetry-js/pull/1187>

2020-07-15

Attendees:

- Daniel Dyla
- Bartłomiej Obecny (Lightstep)
- Jonah Rosenblum (Google)
- Tina Gao (Google)
- Mark Wolff
- Reza Karimi (Google)
- David Witten (Google)
- Matt Wear (Lightstep)
- Caroline Evans (Google)
- Eric Mustin (Datadog)
- Connor Lindsey (AWS)

Agenda:

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- [dyladan] [GA Burndown Project](#)
- [dyladan] [discuss] Change minimum required reviews to match [contribution guidelines](#) (4)?
- [dyladan] Lambda instrumentation PR incoming today (contrib)
- [dyladan] Code style guide PR incoming this week (tomorrow?)
- [dyladan] Open PRs in need of just one more review
 - [refactor\(metrics\): metrics provider implements controller #1017](#)
 - [feat: add OTEL_LOG_LEVEL env var #974](#)
- [dyladan] Very old PR (baggage OT shim)
<https://github.com/open-telemetry/opentelemetry-js/pull/918>
 - Ruben commented he will fix the build/conflicts
- [rezakrimi] New plugin design review:
<https://github.com/open-telemetry/opentelemetry-js/pull/1305>
 - Use global tracer/meter provider or injected?
 - No need for modification of current plugins
- [rezakrimi] Linter for contrib repo:
<https://github.com/open-telemetry/opentelemetry-js-contrib/pull/110>
- [carolinee21] Rename of plugin → instrumentation (issue [here](#))

- + Hapi plugin - [issue](#)
- [mwear] SDK Package: <https://github.com/open-telemetry/opentelemetry-js/pull/1187>

2020-07-08

Attendees:

- Bartłomiej Obecny (Lightstep)
- Morgan McLean (Google)
- Connor Lindsey (AWS)
- Tina Gao (Google Intern)
- Steven Jin (Google)
- Jonah Rosenblum (Google)
- Mark Wolff
- Caroline Evans (Google)
- Eric Mustin (Datadog)
- David Witten (Google)
- Bryan Clement (New Relic)
- Reza Karimi (Google)
- Daniel Dyla (Dynatrace)
- Matt Wear (Lightstep)
- Terrence Turner (Google)
- Reg McDonald (Google Intern)

Agenda:

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- [jonahrosenblum] Support for graceful shutdown for Cloud Run.
 - Any additional functionality requested?
- [obecnyn] default aggregator for CollectorExporter for ValueObserver and ValueRecorder is now MinMaxLastSumCount
- [obecnyn] release - waiting for last metrics updates
- [Morgan] GA planning
 - GA later this year, preceded by one or many release candidates
 - [Project planning doc](#) - please review
 - [Spec Kanban board](#)
 - Basic process:
 - Assign *release:required-for-ga* label to outstanding issues that are required for GA
 - File new issues for work that is required for release candidates / GA and tag them with *release:required-for-ga*
 - Look for newly filed spec issues tagged with *release:required-for-ga* and file JS issues to track the implementation of these

- Create a GitHub project board under the JS repository, matching the org-level specs project board, and pull in any issues and PRs tagged with *release:required-for-ga*
- PR:
 - [UpDownSumObserver](#)
 - [@grpc/grpc-js](#)
 - [OTEL_SAMPLING_PROBABILITY](#)
 - [OTEL_LOG_LEVEL](#)
- <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/110> Daniel will help

2020-07-01

Attendees:

- Caroline Evans (Google)
- Daniel Dyla
- Michael Goin
- Bryan Clement
- Mayur Kale
- Terrence Turner (Google)
- Mark Wolff
- Matt Wear
- Connor Lindsey (AWS)
- Reza Karimi (Google)
- Eric Lee (AWS)
- David Witten (Google)
- Bartłomiej Obecný
- Rebecca Rodriguez

Agenda:

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- [maintainers] Obecný promoted to maintainer status 🎉
- [ted] GA requirements
 - Request: please use “release:required-for-ga” to help track GA issues and PRs
 - GA is getting put together in maintainers meeting
 - <https://docs.google.com/document/d/1-tGqaT9fn9Bpuy4roLS1YDfLzIWHodUFfyrBlyJVz9g/edit#>
 - GA Spec Project Board (spec/oteps):
 - <https://github.com/orgs/open-telemetry/projects/4>
 - GA Planning Sheet:
 - <https://docs.google.com/spreadsheets/d/1Hhw26JzUldj9OhsAaEWm50q-3L8qIBdo6M6R3W4nNeM/>
 - Request: please devise a system for tracking GA progress within the SIG
 - Projecto board, milestone, tracking issue, etc

- [dyladan] Will be on vacation next week. Available if needed. Someone else will need to run this meeting, but I will try to attend
 - Mayur available
 - Obecny available
- [dyladan] ValueObserver and BatchObserver merged. Release?
 - Contrib needs to be released - PR today
- [dyladan] Resource detection logging will be merged
- Metrics
 - UpDownSumObserver needs to be implemented (on obecny's todo list)
 - Prometheus exporter needs to be able to export all metric types
 - SumObserver PR incoming (bryan clement)
 - Resources still not piped through metric export
 - @mattwear thinks this is done and on metric records. Available for export, just not actually exported by prometheus. Collector?
 - Other SIGs not exporting with prometheus (maybe?). Go might be a good starting point for research
 - <https://github.com/open-telemetry/opentelemetry-js/issues/867>
 - Metrics code cleanup session
 - Still ongoing spec discussion as to which default aggregator should be used
 - MinMaxAvg changed to LastValue, but may change again
 - Prometheus changes depend on this
- Resources
 - <https://github.com/open-telemetry/opentelemetry-js/blob/master/packages/opentelemetry-resources/README.md>
 - There is still work to be done to ensure that resources are exported in a consistent way between exporters and between SIGs. Possibly some spec is required for this.
- [dyladan] (rezakrimi) Metrics plugin support update [Design doc #1207](#)
 - Need to decide how node plugins are being loaded
 - Proposal: web plugins loaded by passing an instance (as is done now). For node plugins, new module NodePluginEnabler. NPE loads default plugins as is done now.
 - Trying to keep changes minimal. Plugin constructor needs to be changed to accept Providers
- [dyladan] Bug: [gRPC TypeError](#) (mark wolff?)
- [dyladan] [NodeJS SDK Package](#)
- [mayurkale22] [Fix Renovate Configuration](#)
- [mayur] Semantic Convention package not used in contrib. Contrib generally needs more attention
- [mayur] How are interns finding the project, is there anything we can do to help them succeed?
 - Caroline is looking for guidance on the PR process, DD is going to make sure contributing.md is up to date

2020-06-24

Attendees:

- Adam Egyed (Google Intern)
- Caroline Evans (Google intern)
- Eryn Schots (Google)
- Bartłomiej Obecny
- Mayur Kale
- Bryan Clement
- Tina Gao (Google Intern)
- Reza Karimi (Google intern)
- David Witten (Google intern)
- Daniel Dyla
- Jorrie Brettin (Google)
- Nina Stawski
- Terrence Turner (Google)
- Eric Mustin (Datadog)
- Matt Wear (Lightstep)
- Valentin Marchaud
- Connor Lindsey (AWS)

Agenda:

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- [dyladan] please put your name/github handle before items you add to the doc :)
- [obecny] How can we speed up review process?
 - Possibly third maintainer
 - Dep bot will be merged with 1-2 approvers if build passes to reduce spam
 - Reduce days dep bot runs
- [obecny] With regards to this question [React Native](#) who has knowledge or power to decide about it ? Is there already a plan / strategy to what extent otel wants to support new things, plugins etc. ?
 - (@vmarchaud) Same request for [Cloudflare Workers](#) has been made
 - DD will bring this up at the maintainers meeting to get guidance and see what other SIGs are doing from this perspective
 - When a plugin is added, when does it get added to the list of default plugins
- [@carolinee21] Plugin proposals for new database and web frameworks instrumentation (SQL Server/Elasticsearch vs. Koa/Hapi)
 - Elasticsearch uses http transport, so using the http/https plugins should give you some telemetry already
 - Express plugin mainly adds the path of the request, and Koa/Hapi would likely be similar
- [@rezakrimi] Adding metric support to plugins: [Design doc](#)
 - <https://github.com/open-telemetry/opentelemetry-js/issues/1207>

- Plugins currently do not support metrics/meters
- [Opencensus implementation](#) maybe be helpful to check out
- Will we go back and add metric support to previously created plugins?
 - If so, who will do this?
- DD will check that this does not conflict with spec
- Metrics sdk still not ready
- (@adamegyed) Resource auto-detection logging for individual detectors.
 - Current state after detection it logs what each detector found
 - Should each detector log in a context-specific manner?
 - Mwear - purpose of logging is not typically to show what was found, but to discover why things were not found.
 - Env not found, env not parseable
 - aws/gcp detectors make async calls that could potentially fail
- (@connorlindsey) [Pull Controller](#) - Current status?
 - Related PR <https://github.com/open-telemetry/opentelemetry-js/pull/1017>
 - DD will reach out to the PR author after this meeting
- (@connorlindsey) Prometheus Exporter
 - Support for Summary / Histogram data types in Prometheus exporter?
 - TODO in [this file](#)
 - @vmarchaud:
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1038>
 - PR needs to be reworked for new metrics SDK. waiting for SDK to be more complete
- [mwear] Add additional transport options for the collector exporter for Node (OTLP/HTTP).
 - [OTLP/HTTP: HTTP Transport Extension for OTLP · Issue #99 · open-telemetry/oteps](#)
 - <https://www.npmjs.com/package/protobufjs#performance>
- [emustin] Would a vendor specific exporter PR be welcome/accepted in the [contrib repo](#)?
 - Python creating PR in the core repo that never gets merged just to get reviews, accidentally got merged :/
- Mayur reached out to brandon and rochdev and they have not responded
 - If no response soon, remove from approvers

2020-06-17

Attendees: Daniel Dyla (Dynatrace, Maintainer), Caroline Evans (Google intern), Connor Lindsey (AWS Intern), Tina Gao (Google Intern), Bryan Clement, Michael Goin, Eryn Schots (Google), Tiger He (Google), Bartlomiej Obecny, Steven Jin (Google), Reza Karimi (Google intern), David Witten (Google intern), Nathan Lehrer (Google), Reg McDonald (Google Intern), Nina Stawski (Splunk), Matt Wear

Agenda:

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- 0.9.0 release PR today
 - Includes plugins-all-node
- Node SDK package <https://github.com/open-telemetry/opentelemetry-js/pull/1187>
- [Batch observer](#) needs reviews
- [feat: add instrumentation library and update collector exporter](#) needs reviews
- Contrib repo merge requirements
 - <https://github.com/open-telemetry/opentelemetry-js-contrib/pull/43> open for 3 weeks
 - DD will open a contrib merge guideline PR
- Adding local linting to opentelemetry-operations-js repo: config file to match CircleCI?
- Google intern project questions: proposing our own projects
 - Adding new plugins
 - Implementing zPages like OpenCensus [OTEP](#)
 - Potential intern projects <https://github.com/open-telemetry/oteps/pull/113>
- TODO: remove `component` attribute from all semantic conventions
- Nested maps as attribute values (OTel specs):
<https://github.com/open-telemetry/opentelemetry-specification/issues/376#issuecomment-643488960>

2020-06-10

Attendees: Daniel Dyla, Bartlomiej Obecny, Mayur Kale, Reza Karimi (Google Intern), Nina Stawski, Bryan Clement, Connor Lindsey (AWS Intern), Eryn Schots (Google), Caroline Evans (Google Intern), Tiger He (Google intern), Tina Gao (Google Intern), Adam Egyed (Google Intern), Jonah Rosenblum (Google Intern), David Witten (Google intern), Jorrie Brettin (Google), Reg McDonald (Google Intern), Michael Goin, Nathan Lehrer (Google Intern Host), Matt Wear, Terrence Turner (Google Intern Host), Cunjun Wang (AWS Intern), Mark Wolff

Agenda:

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- New CLA tool. If you have trouble getting your PRs to pass the new check, please let Daniel or Mayur know
 - Old CLA check still fails and makes all PR builds look like they're failing. This is fine for now
- Dependabot enabled for both repositories
 - [discuss] dependabot configurations
 - Combined dependency PR to reduce spam
<https://github.com/open-telemetry/opentelemetry-js/pull/1180>
- [feat: add node-plugins-all package](#)
- SDK package PR incoming today

- PRs
 - [Batch observer](#)
 - [feat: add OPENTELEMETRY_NO_PATCH_MODULES](#)
 - [refactor\(attributes\): move enums to core](#)
 - [feat: add instrumentation library and update collector exporter](#)
 - feat: add the UpDownCounter instrument
- [Metric instrument names](#)
 - Obecny opened a spec issue to discuss MinMaxSumAvg vs LastValue
<https://github.com/open-telemetry/opentelemetry-specification/issues/636>
- Resource API
 - <https://github.com/open-telemetry/oteps/pull/111>
- [ted] Discuss using github projects for managing GA requirements
 - Sprints, even?
 - GA requirements tracking:
<https://docs.google.com/spreadsheets/d/1Hhw26JzUldj9OhsAaEWm50q-3L8qIBdo6M6R3W4nNeM/>

2020-06-03

Attendees: Daniel Dyla, Steven Jin (Google), Marius Kintel (Google), Jonah Rosenblum (Google intern), Adam Egyed (Google intern), Connor Lindsey (AWS Intern), Bartlomiej Obecny, Amir Blum (Aspecto), Matt Wear, Mark Wolff

Agenda:

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- Google Interns would like to contribute to Open Telemetry JS
- v0.8.3 Release 🎉
- Welcome naseem ullah to approvers
- [discuss] Should plugins-node-all be in core or contrib
 - Decided to keep in core
- [discuss] [Create an SDK package which exports all SDK components](#) (except plugins)
 - MW thinks this sounds helpful, but has questions about configurability
 - BO thinks we should start with just a node package because web packaging confuses the situation
 - DD will open a PR
 - @opentelemetry/sdk-node
- [discuss] [Rename createMetric to newMetric](#)
 - BO “new” sounds weird without the new keyword
 - Matt thinks this is a common go paradigm
 - DD would like to keep create name
 - Mark has concerns that we are named different than spec
 - DD will get mayur’s opinion offline
- Technical committee has gotten approval to use a new CLA tool which should allow us to use renovate

- <https://github.com/open-telemetry/community/issues/306#issuecomment-638289838>
- Pending PRs
 - <https://github.com/open-telemetry/opentelemetry-js/pulls?q=is%3Apr+is%3Aopen+sort%3Aupdated-desc>
- Unassigned metrics Issues
 - [Rename Observer to ValueObserver](#) (obecny)
 - [Implement UpDownSumObserver](#)
 - [OTLP Metrics Exporter](#)

2020-05-27

Attendees: Daniel Dyla, Valentin Marchaud, Bartlomiej Obecny, Amir Blum, Tiago Martins Napoli, Bryan Clement, Matt Wear

Agenda:

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- [Metrics - contrib repo](#)
 - Python and go have similar functionality
 - Python: <https://github.com/open-telemetry/opentelemetry-python/pull/652>
 - Go: <https://github.com/open-telemetry/opentelemetry-go-contrib/pull/9>
- [feat: spec compliant sampling result support](#)
- [feat\(prom\): expose measure as summary #982](#)
- [Asynchooks refactor](#)
 - `with` currently broken in async hook context manager, but works in 7.0.0
- Latest @types/node
 - incompatible with:
 - webpack-env
 - Zone.js
 - DD will open an issue/pr on zone.js if it needs to be
 - Installed by a dependency of lerna, which means it is in the root node_modules and the global node_modules
 - Workaround
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1110/files#diff-b6529bd137afebc7916756da2a214017R6>
- Bring metrics up to date
 - [Implement UpDownSumObserver](#)
 - [Implement SubObserver](#)
 - [Rename Observer to ValueObserver](#)
 - [Rename Gauge to ValueRecorder](#)

- [Implement UpDownCounter](#)
- [OTLP Metrics Exporter](#)
- Currently only one metric can be updated at a time (except observer)

2020-05-20

Attendees: Daniel Dyla, Tiger He, Bartlomiej Obecny, Kanika Shah, Mayur Kale, Matt Wear, Bryan Clement, Nina Stawski

Agenda:

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- Google internship program - 3 to 4 interns for js
 - Nathan and Tiger
 - SDK and exporters
 - Looking for large projects that could span possibly the whole summer
- Discussion - no more force push for PR's, hard to see changes, links from emails doesn't work anymore etc. etc. WDYT?
 - This also breaks the history in the web ui making it difficult to see if things are resolved or not
 - DD will create an issue/PR to update contributing.md
 - Mayur will open merge/PR guideline PR from the doc from last week
- 0.8.1 released 🎉
 - API docs deployed automatically (finally) - [docs](#) - [automation](#)
- Additional release automation
 - Create releases? What should trigger this
 - Move tests from circle to actions
 - Probably not at this stage
 - Deploy to NPM on release publish
 - Mayur saw some google projects with a release bot that creates a release PR
 - Works similar to renovate where it keeps the PR up to date
- Async Hooks update?
- Metrics
 - Label keys removed from API
 - Instruments are changed drastically
 - Counter - We have this now
 - UpDownCounter
 - ValueRecorder - I think this is our gauge
 - SumObserver
 - UpDownSumObserver
 - ValueObserver - I think this is our observer behavior now
- End to end testing
 - Good task for interns?
- PRs

- bugs
 - [fix: missing global in browser environments](#)
 - DD will create a PR to change linting rules to disable global
 - End to end testing will hopefully capture things like this
- [feat: merge user supplied and default plugin configs](#)
- [feat: spec compliant sampling result support](#)
- [feat\(prom\): expose measure as summary #982](#)
- [feat\(opentelemetry-exporter-jaeger\): http sender](#)
- [OpenTelemetry was the second most active CNCF project over the past quarter](#), with even more contributions than gRPC, Envoy, or Prometheus! Only Kubernetes was ahead.
- [Gartner's quarterly APM magic quadrant report](#), published in late April, called out OpenTelemetry multiple times.
 - Each vendor's section noted whether they were contributing to OpenTelemetry; the section for AppDynamics listed their lack of participation in OpenTelemetry as a key risk to their business
 - Something that stood out and is great for Opentelemetry - "By 2025, 50% of new cloud-native application monitoring will use open-source instrumentation instead of vendor-specific agents"
- Some contributors/approvers have been inactive
 - Remove rochdev and brandon?
 - Add matt as an approver
 - DD will reach out to naseem
 - Possibly also legendecas, DD will reach out to determine his anticipated level of contribution moving forward

2020-05-13

Attendees: Daniel Dyla, Mayur Kale, Bartlomiej Obecny, Nina Stawski

Agenda:

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- 0.8.0 release 🎉
- Bugs 🐛 - cut 0.8.1 after fixes?
 - issue/[batch span processor/ioredis crasher](#)
 - pr/[fix: include proto files in deployment package](#)
- Merging rules - with regards to [this comment](#) is there a definition when the merge can be done as majority of merges are done with 3 approves even in [latest draft for release](#)
 - Question (@vmarchaud): After the 3 approves, can any contributor merge ? currently only maintainers do the merge
 - Initial Draft:

<https://docs.google.com/document/d/1Q84zozyBOPz11o0ghdzPQUNmmvyYgHuaBtLTE2sW3TI/edit>
- Github Actions to deploy API docs [chore: deploy docs using github action](#)

- Plugin metapackages for web and node core
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1050>
 - Release with 0.8.1
- Cls-hooked update
 - @vmarchaud Started working on it [there](#), currently stuck with some bugs but will work more this weekend and next week (got time dedicated for this at work)
- Make Labels Optional for CounterMetric::add
 - <https://github.com/open-telemetry/opentelemetry-js/pull/1032>
- Sketch aggregator update
 - @vmarchaud didnt had the time yet, but same as cls-hooked i will have time allocated for this at work

2020-05-06

Attendees: Daniel Dyla, Valentin Marchaud, Bartlomiej Obecny, Mayur Kale, Andrew Hsu (Lightstep),
Matt Wear

Agenda:

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- Plugins-all package
 - Creating 3 packages (plugins-all-node-contrib, plugins-all-node-core, and plugins-all-node)
 - PR open for first package so we can agree on structure
<https://github.com/open-telemetry/opentelemetry-js-contrib/pull/25>
 - Test
 - Not for now, maybe integration later
 - Add a README
 - Mayur: maybe not contrib, just core and all. For now we can start in core
 - If we need we can add a separate contrib one later
 - Plugins-all-web should be done the same way, but they need to be exported or have some init-all method
- Test on node 13 and 14
 - [Reintroduce node 13 tests](#)
 - [chore: test on node 14](#)
 - Tests broken on 14 in contrib
 - pg needs to be updated 8.x.x to support node 14
 - For now update pg module to 8
 - Create an issue to discuss more advanced version matrix testing later
- Administration
 - CODEOWNERS cannot be used for contributors without write access
 - We can use CODEOWNERS for now because it is mostly just us contributing

- Other projects have this issue and we can see how they handle it
 - Not an issue yet so we can probably leave this as it stands
- Discussion issues
 - [MetricExporter.export return promise instead of callbacks](#)
 - Does spec require this to be a callback?
 - Possible performance impact?
 - [bug: asynchookscopemanager's `withAsync` leak context](#)
 - @vmarchaud is looking into cls-hooked or possibly just lifting their code
 - [bug: collector-exported v0.7.0 doesnt contains proto](#)
 - DD is looking into this
 - [metrics: implement Sketch aggregator](#)
 - Not enough info to make a decision right now, but vmarchaud is still doing research
 - [SimpleSpanProcessor depends on implementation specific method](#)
 - [Resource is missing in MetricRecord for MetricExporter](#)
 - <https://github.com/open-telemetry/opentelemetry-specification/issues/508>
 - Java and Go pass resources as part of the spandata
 - Resources are on instruments, but may not make it all the way to the exporter at the moment. Matt will look into adding resources to the metricrecord
- PRs
 - [fix: use system clock for timestamps and perf clock for durations #1019](#)
 - [refactor\(metrics\): metrics provider implements controller](#)
 - [fix\(lint\): move tslint to eslint](#)
 - [feat: implement child span count](#) (@mayur very old, needs rebase)
 - Is this needed any more?
 - See: <https://github.com/open-telemetry/opentelemetry-proto/issues/105> and <https://github.com/open-telemetry/opentelemetry-proto/pull/107>
 - stalled [Base Plugin](#)
- Release some time next week
 - 0.8.0 because I think there are breaking changes

2020-04-29

Attendees: Daniel Dyla, Mayur Kale, Nina Stawski, Bartlomiej Obecny, Valentin Marchaud, Matt Wear

Agenda:

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- [andrewhsu] <https://github.com/open-telemetry/opentelemetry-js-contrib/issues/20> (plugins-all)
 - @opentelemetry/plugins-all-node
- [refactor: merge opentelemetry-base to opentelemetry-core](#)

- Please review
- Metrics [Metrics API and SDK github project](#)
 - <https://github.com/open-telemetry/opentelemetry-js/issues/796> (pull controller)
 - [Feature Request: Metrics with non-default aggregators](#)
 - Prometheus exporter incomplete
 - [prometheus-exporter: export measure · Issue #982 · open-telemetry/opentelemetry-js](#)
 - [Support Resource in the Prometheus Exporter · Issue #867 · open-telemetry/opentelemetry-js](#)
 - View API
 - Not sure about spec status
 - Python prototype for spec exists
 - <https://github.com/open-telemetry/opentelemetry-python/pull/596>
- Administration
 - Create a process for addition of new modules to contrib
 - Minimum number of weekly downloads on npm?
 - We want to limit CI resource consumption and plugins-all package deployment size
 - Define contrib module ownership
 - CODEOWNERS file on github may be used
 - Define a release cycle and process for contrib
 - Automated releases
 - Weekly?
 - Every friday morning
 - PR merge to master?
 - <https://github.com/open-telemetry/opentelemetry-js/issues/419>
 - Ruby has this already
 - <https://github.com/open-telemetry/opentelemetry-ruby/blob/master/.circleci/config.yml#L201>
 - <https://github.com/open-telemetry/opentelemetry-ruby#release-process>
 - The Rakefile generally has related code: <https://github.com/open-telemetry/opentelemetry-ruby/blob/master/Rakefile#L144-L159>
 - Release automation
 - [fix\(lint\): move tslint to eslint](#)
 - [Source file naming convention](#)
 - <https://github.com/open-telemetry/opentelemetry-js/issues/38>
 - <https://www.npmjs.com/package/eslint-plugin-filenames>
 - Document package naming convention
 - Renovate (?)
 - <https://github.com/apps/forking-renovate>
 - Daniel will reach out to bogdan

- [Proposal: Separate release pipeline for core and contrib repositories. · Issue #7 · open-telemetry/opentelemetry-js-contrib](#)
- Configuration
 - Naseemkullah has requested OTEL_LOG_LEVEL and OTEL_SAMPLING_PROBABILITY configuration env vars
 - Looks like other sigs have implemented ENV configurations
 - Let's wait on spec for this
 - [feat: merge user supplied and default plugin configs](#)
- Bug Triage <https://github.com/open-telemetry/opentelemetry-js/projects/10>
- PRs
 - ["Bug" with observer](#) but spec says there should not be bound instrument [see here](#)
 - Daniel will read spec after meeting same for mayur
 - [chore: splitting BasePlugin into browser and node](#)
 - [feat\(http-plugin\): add options to disable new spans if no parent #931](#)

2020-04-22

Attendees: Daniel Dyla, Olivier Albertini, Andrew Hsu (Lightstep), Bartlomiej Obecny, Nina Stawski

Agenda

Please add your name to the attendees list

- Contrib repo
 - Should the versions follow the core repo?
- Cross-language concerns
 - Integration testing
- Instrumentation
 - Span names for DB - Currently using sql command (insert, update, etc)
 - Plugins-all package? Do we, or should we have one?
 - No, but we will soon
 - Need to check semconv compliance for db plugins
- PRs which need reviews
 - <https://github.com/open-telemetry/opentelemetry-js/pull/892>
 - [andrewhsu] <https://github.com/open-telemetry/opentelemetry-js/pull/964> ← will be helpful for other work building upon observable metric
 - <https://github.com/open-telemetry/opentelemetry-js/pull/943>

2020-04-15

Attendees: Ted Young, Daniel Dyla, Bartlomiej Obecny, Matt Wear

Agenda

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- Usability testing feedback [ted]
 - Positives
 - Seems to work
 - Experienced programmers get working pretty quickly
 - Some struggle trying to figure out how to configure different propagators
 - Negatives
 - Required to blow away node modules sometimes
 - Difficult to get started for people with little experience
 - Installing the plugins themselves is difficult for newbies
 - Named tracers are confusing
 - Possible action items
 - Improve docs around propagators
 - Create an all-in-one package which includes sdk and plugins
 - Use a post-install script to suggest users install plugins
- Plugins repo update
 - <https://github.com/open-telemetry/community/issues/331>
- Create web/node specific base plugins
 - DD will create an issue - assign to obecny
- PRs which need reviews
 - [fix\(mongodb\): avoid double patching when enable is called twice #956](#)
 - [Use global API instances](#)
 - [feat\(context\): implement withAsync #752](#)
- PRs which will merge after call today
 - [fix\(plugin-express\): fix double span end #908](#)
 - [feat\(aggregators\): implement histogram aggregator #927](#)
 - [chore: adding metric observable to be able to support async update](#)
 - <https://github.com/open-telemetry/opentelemetry-js/pull/899>

2020-04-08

Attendees: Daniel Dyla, Bartlomiej Obecny, Nina Stawski, Mark Wolff, Valentin Marchaud

Agenda

Please add your name to the attendees list

Tentative goals for the next meeting, feel free to add your own!:

- Schedule time to review global API PR
 - Will stick around on this call after meeting ends
 - DD will add high-level explainer to README
- Core plugins vs contrib plugins
 - Currently have ~15 plugins in the repo
 - All plugins loaded by default in node
 - MK proposal: split to 2 packages
 - Plugins-core and plugins-contrib (naming could change)
 - Core: installed and enabled by default
 - Http/https

- contrib/community: enabled by default if installed by user
 - Possibly have some process to graduate from contrib to core
 - Possible tool to autoinstall plugins
 - Postinstall that lists available not installed plugins
 - VM: privacy issue?
 - CLI tool that lists or installs available not installed plugins
 - Actionable now:
 - Create a plugins repo - opentelemetry-js-contrib
 - Keep git history
 - /plugins/node/*
 - /plugins/web/*
 - Lerna managed?
 - Keep it for now, revisit if it is painful
 - Move all plugins except http/https and xhr to instr repo
 - Why?
 - Maintainability
 - Install core plugins by default (?Maybe, next step) for
 - auto-instr.
- Express PR
 - https://github.com/open-telemetry/opentelemetry-js/pull/910#discussion_r401083456
 - Example with the patch: <https://imgur.com/myZdJG9>
- withAsync PR
 - VM will work on getting zone manager working
- MeasureMetric state
 - Not supported by prometheus - pr coming
 - Naming mismatches spec in some places - VM will look into this
 - Custom batcher PR
 - Histogram aggregator PR
 - Only adds to sdk, still will need to be exported
- New Issues / PR rundown
 - Resource Collection
 - Collector Exporter

2020-04-01

Attendees: Daniel Dyla, Andrew Hsu (Lightstep), Mayur Kale, Matt Wear, Nina Stawski, Bartlomiej Obecnny

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Upcoming v0.6.0 release: after [Remove label set from metrics API #915](#)
 - Release soon
- [Add @opentelemetry/api as peerDependencies on @opentelemetry/sdk](#)
 - Multiple versions of API may exist in SDK

- Use some combination of peer deps
 - Uses existing module structure
 - Use global API instance
 - Ensures true global
- [time / performance / hibernate / problem ?](#)
 - Calculate time offset when new time is generated
 - Makes span timings dependent on system clock
 - Record startTime with system clock, duration/end with perf
 - Could report incorrect end time if sleep during span
 - Dave Raff suggestion - [detect only for now](#)
 - Determine scope
- Core plugins vs contrib plugins
- PR rundown
 - [fix\(plugin-express\): fix double span end #908](#)
 - [grpc for node and support for new proto format for node and browser](#)
 - [feat: resource auto-detection](#)

2020-03-25

Attendees: Daniel Dyla, Bartlomiej Obecny, Matt Wear, Nina Stawski

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Pull requests
 - [feat: resource auto-detection](#)
 - [feat: start a root span with spanOptions.parent = null](#)
- OTLP/gRPC export
 - Finishing up unit tests
 - No image available for end to end testing
- time / performance / hibernate / problem ?
 - Calculate time offset when new time is generated
- Pushed to next week
 - <https://github.com/open-telemetry/opentelemetry-js/issues/885>
 - Core plugins vs contrib plugins

2020-03-18

Attendees: Daniel Dyla, Mayur Kale, Bartlomiej Obecny, Andrew Hsu (Lightstep)

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Beta launch (0.5.0) status update
- Take away from first beta planning meeting
 - Remove labelset from metrics API
 - Mayur taking this

- Consolidate metrics examples
 - Daniel taking this
- Simplify examples
 - Mayur will open an issue
- Who is using OTel? In production? How many deployments/users?
 - Azure
 - helper APIs have been instrumented - Azure SDK - API only
 - Planned exporter
 - GCP
 - Exporter implemented
 - Lightstep
 - Putting docs together to use otel with lightstep
 - Lightstep will be dogfooding go and python
 - Will have a path for users but may not recommend production use yet
 - <https://opentelemetry.io/registry/?s=lightstep>
 - Lightstep javascript exporter: <https://github.com/lightstep/opentelemetry-exporter-js>
 - City of Montreal ([VilledeMontreal](#))
 - Switched from OpenCensus
 - <https://github.com/VilledeMontreal/workit/pull/110>
 - Cazoo Telemetry
 - <https://www.npmjs.com/package/@cazoo/telemetry>
 - ~53k weekly downloads for API
 - ~2-3k weekly downloads for SDK
- Performance and Integration testing would be a huge win
- 2020-03-18 (Wed) OpenTelemetry Beta planning mtg 2:00pm-3:00pm PDT (public otel calendar)
 - <https://docs.google.com/document/d/1zPm4PgGtXNkOBJodHSNUCXIYxT53nOTysfiRI-R7bJQ/edit#heading=h.cfj8n8ydrq6m>
- Context propagation
- Github beta [milestone](#) 100% complete.
 - Q: Do we need another milestone before 2020-03-30?
 - A: another milestone is not necessary to satisfy beta launch plan
- Next steps
 - Auto-instrumentation story
 - Discovering plugins
 - Configuration APIs and default configs
 - Metrics View API
 - Performance and integration tests

2020-03-11 - LAST MEETING BEFORE BETA

Attendees: Daniel Dyla, Mayur kale, Bartlomiej Obecny

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Review [Beta Readiness Tracker](#)
 - Unknown: release version?
 - `0.5.0`
 - Beta Milestone : [72% \(24/33\) complete](#)
- Resources implementation
 - [SDK Resource](#)
 - [export resource to exporters](#)
 - Spec Issue: [Change Processor and Exporter interface to pass resource](#)
- Review Metrics View API
 - <https://docs.google.com/document/d/1ZyPNOqByj4GFxLhXeB8bENR4uWmR3NPYiaS46yrlWTQ/edit>
 - OTEP: <https://github.com/open-telemetry/oteps/pull/89>
- Discussion issues
 - [Which plugins are loaded by default](#)
 - [Rename Scope to Context](#)
 - [Rename formatter to propagator](#)
- Issue / PR walkthrough

2020-03-04

Attendees: , Bartlomiej Obecnny, Matt Wear, Mayur Kale, Andrew Hsu (Lightstep)

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Review [Beta Readiness Tracker](#)
- Composite Propagator [pull/821](#)
 - Runs all propagators in the order they are configured no matter what
 - Logs thrown errors and continues running
- Getter/Setter in Propagator API [pull/827](#)
 - Fixes a spec nonconformance
 - Default getter/setter is just a simple property access
- Metric Observer [pull/828](#)
- Simplify Initialization [#823](#) [pull/837](#)
- Correlation Context Propagator [#834](#) [#838](#)
 - [andrewhsu] suggestion: off by default, document user to use config option to turn on
- Async Hooks scope manager fails with mixed-async use case
 - Possible fix - add withAsync / bindAsync

2020-02-26

Attendees: Daniel Dyla, Matt Wear, Bart Obecnny, Nicholas Moorcroft, Nina Stawski

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- New person nick moorcroft (zappi.io)
- Critical dependency warning in webpack [#814](#)
 - Dan D will try to reproduce this today
- Propagation API merged
 - context and propagation APIs are still completely unused in instrumentation libraries. Fixed in Context based tracing PR
- Remaining beta work [tracking issue](#)
 - Context based tracing PR opened [pull/816](#)
 - Resources Draft PR [pull/815](#) (@mwear)
 - Please review [pull/802](#) add force flush (spec nonconformance fix)
 - Metrics
 - Add observer instrument (@obecny working on this)
 - Docs
 - Examples
 - Prometheus Exporter (@dyladan working on this)
 - Pull controller [#796](#) (looking for a volunteer)
 - gRPC support for collector exporter [#567](#) (@Ema93sh working on this)
- Lightstep plugin hosting?
 - We will host our community plugins here:
<https://github.com/lightstep/opentelemetry-js-plugins>
- Ruby Setup / Plugin Installation:
 - <https://github.com/open-telemetry/opentelemetry-ruby/pull/171>
 - <https://github.com/open-telemetry/opentelemetry-ruby/pull/164>

2020-02-18

Attendees: Daniel Dyla, Andrew Hsu (Lightstep, mtg rm Bear Claw), Mayur Kale, Matt Wear (Lightstep), Bartlomiej Obecny, Ted Young, Nina Stawski

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Context and Metrics merged
- Remaining beta work
 - Context API [pull/792](#)
 - Propagation API (depends on context API)
 - Use Context/Propagation API for tracing
 - Remove Gauge ([PR Merged](#))
 - metrics
 - Add Observer instrument @obecny
 - Documentation
 - Examples
 - Prometheus exporter @dyladan
 - Update for observer

- Measure
 - Histogram
 - Have push controller, would like pull controller
 - Daniel will create an issue for this
- Resources ([spec](#))
 - Assign to @mwear
- Express [pull/685](#)
- Find all @todos and create issues/fix
 - Mayur and Daniel
- Remove spec nonconformities
 - Force flush assigned to @obecny
- gRPC support for collector exporter
- Plugins:
 - Core - stored in the repo with the sdk code
 - Contrib - tbd, clean up before beta
 - Community - repo in the contributor's github org, e.g. lightstep, can ping daniel for code reviews; also add plugin to the otel [registry](#)
 - TODO: need an “upgrade” path (aka promotion path) to bring a plugin from community to contrib or core
 - Let's tackle this after beta

2020-02-12

Attendees: Daniel Dyla, Mayur Kale, Morgan McLean, Matt Wear, Andrew Hsu (LightStep), Nina Stawski, Mark Wolff

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Stackdriver exporter likely to move under <https://github.com/GoogleCloudPlatform>
 - This will be done for all languages
 - Probably next week
- Auto instrumentation: Two repos, what approach do we want to take?
 - Maybe do this before we get to beta
 - 3 levels of plugin
 - Core - http, grpc, sql
 - Enabled by default
 - Non-core
 - Community - registry only
 - Morgan will bring up with TC
- <https://github.com/open-telemetry?utf8=%E2%9C%93&q=auto&type=&language=>
- OpenTelemetry beta release plan:
 - <https://docs.google.com/document/d/1eviggolguOS89dgKL-8ntUOCfPP6FLWAoQkGoBDsld0/edit#>
 - [Beta Readiness Tracking Issue](#)
 - Mayur will clean up milestones for beta

- Metrics SDK: [pull/738](#)
 - Please review
 - This is a starting point to iterate on
- [Separate context propagation \(OTEP 66\)](#)
 - API now depends on Scope Base
 - Would we like Context to be its own entity (separate from sdk or api but consumed by both)
 - Right now this is sort of the case if you take the scope manager to be the Context package - Scope Base is a “leaf” package already.
 - TODO (in future PRs):
 - Add API methods to get and set currently active Context
 - Add global propagator API
 - startSpan should automatically determine parent from currently active Context unless forced to be a root span
 - Correlations API
 - CorrelationContext (baggage) API
- BG: last week SIG was basically only lightstep people

2020-02-05

Attendees: Bartlomiej Obecny, Brandon Gonzalez, Nina Stawski, Andrew Hsu (LightStep), Matt Wear, Olivier Albertini, Mayur Kale

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Rollover from last meeting
 - User Interaction Plugin - stucked from 2 weeks
 - Please review this plugin
 - Note: new plugins, maybe have a contrib repo for these?
 - Find out how other SIGs do this.
 - Ruby has a separate repo for plugins, see tracking [issue](#)
 - Question: what to consider a core plugin?
 - Olivier : plugin-http should be included as a core plugin
 - Create a triage process for plugins that should be in monorepo or in a contrib repo (could be in CONTRIBUTING.md)
 - [New Shimmer \(#618\)](#)
 - Current shimmer prevents us from using shimmer by us, won't be able to unpatch
 - if there will be 2 plugins from opentelemetry and both of them will be patching the same code for example mongo and mongoose it will face the same problem
 - Issue triage if time
 - [Need more reviewers](#)
- [0.4.0 release coming.](#)

2020-01-29

Attendees: Daniel Dyla, Mayur Kale, Bartlomiej Obecny, Morgan McLean, Carlos Cortez, Nina Stawski, Olivier Albertini

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Morgan: beta progress, release plan
 - We want Java, JS, Python, Go, .Net, and the Collector to go beta at Kubecon
 - We're working on a [list of requirements](#) for each SIG's beta release
- New API package - 0.4
- Switch from yarn to npm
- Provider rename spec merged
- Metrics SDK: [pull/738](#)
 - Please review [563](#)
- Winston plugin
- [OTEP 66 - Context Propagation](#)
 - Move to next week
- User Interaction Plugin - stucked from 2 weeks
 - Please review this plugin
- New Shimmer
 - Move to next week
- Issue triage if time

2020-01-22

Attendees: Daniel Dyla, Bartlomiej Obecny, Nina Stawski, Mayur Kale, Olivier Albertini

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- PRs
 - [PRs with no reviews](#)
 - [ioredis supported version](#) (bug, XS change)
 - [unref jaeger socket](#) (bug [#712](#), XS change)
 - [lint examples](#)
- New plugin PRs
 - [web user interaction](#)
 - [express](#)
- Minimum supported version ([issue/711](#))
 - Could potentially use stack scope manager in older versions of node 8?
 - <https://github.com/open-telemetry/opentelemetry-js/blob/b21ed9706a3fdf1d6c6da812bb9797c8253fef40/packages/opentelemetry-web/src/StackScopeManager.ts>
- Separate observability from context propagation
 - Merged otep - spec coming soon
- Spec 0.3 by the end of the month - this will be the beta featureset afaiK

- In my opinion (daniel dyla) we should change focus from adding plugins to completing the SDK
 - Will include context prop spec
- Issue triage if time
- New Shimmer
 - Bart will publish this to his own github

2020-01-15

Attendees: Daniel Dyla, Bartlomiej Obecny, Mayur Kale, Nina Stawski, Mark Wolff

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Updates from Spec
 - Named meters/tracers in flux
 - “Registry” likely renamed “Provider”
 - Provider does not need to keep a reference to tracers/meters but may construct a new one on each getMeter/getTracer call
 - Configs should be stored in the Provider, not on individual tracers
 - Metric namespace may or may not be the same as the meter name [spec/#410](#)
 - getTracer/getMeter name parameter is *required* but if it is called incorrectly we can use some default value as the name and return a working tracer/meter
- New Issues
 - Add support for jaeger propagation formats
 - Use buffer instead string for traceId and spanId
 - support in web -> [uint ArrayBuffer](#)
 - dev branch?
- Stuck PRs
 - feat: lint examples
 - feat: warn user when an instrumented package was already required #636

2020-01-08

Attendees: Daniel Dyla, Nina Stawski, Olivier Albertini, Bartlomiej Obecny, Mayur Kale, Vinod

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- New people
 - Vinod - new relic
- Possibly a spec issue - decoupling sdk components - moved from previous meeting since we ran out of time
 - Current export pipeline is tightly coupled to SDK making it difficult for a third party to replace individual components of the SDK without also building their own full export pipeline

- We already had this discussion about plugins and it was concluded that plugins should only use public api methods and properties. For very similar reasons, I believe the same should be true of the export pipeline.
- Needs one (or more) approver / reviewer
 - Named Tracers / Tracer Registry [pull/582](#)
 - stackdriver trace exporter [pull/648](#)
 - plugin user interaction for web [pull/658](#)
 - Advanced plugin configuration [pull/628](#)
 - TypeScript version for Getting Started Guide [pull/673](#)
- New issues
 - [#672](#) support for common promise libraries - q, promise, bluebird
 - [otep/72](#) merged - Observer metric type - spec will start on this soon and merge quickly so we will need to implement
- Alpha v0.4 release date?
 - Current date : December 31 2019 (already passed).
 - Going to target end of january. Need to update readmes to reflect
- X-Lang question: package/repo layout
 - Does one repo feel good?
 - Automation/Installers
 - Instrumentation
- FFI implementation using C++
 - Any interest?
 - Dan was interested in this, but unsure of its actual practical value
- Performance testing
 - Would like to develop some standards for OTel Beta

2019-12-18

Attendees: Daniel Dyla, Mayur Kale, Bartlomiej Obecny, Nina Stawski, Oliver Albertini

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Community feedback
 - We'd like to improve the project
 - Please provide feedback!
 - (anonymous submissions are ok)
 - (one separate submission per problem/suggestion)
 - <https://forms.gle/m2iY9rYm1Xvc6fUs9>
- New bugs
 - [HTTP attributes are noncomformant](#) (original [#373](#))
 - Plugin author oliver
 - Oliver is handling this
 - Mayur is looking at gRPC to see if it matches spec
 - [gRPC client methods are not patched correctly](#) (PR)
 - Approved by plugin author

- [Web request spans not in correct order](#)
 - Stale
- More advanced examples
 - Mayur will make an issue
 - Possibly use docker to run multiple services
 - Multi-language deployments
- Spec nonconformance
 - Adding a nonconforming label
 - SIG is supposed to generally track the spec but is not required to match version by version until beta at least
- New spec in-process
 - [Rename metric handle to bound instrument](#)
 - DD will take this if noone does
- Project hygiene
 - Enforce lint on examples folder
 - Enforce code coverage threshold
 - Moving towards beta, want to make sure coverage is good for that launch
 - Auto upgrade dependencies (renovate / greenkeeper)
 - Configure benchmark suite
 - Not run by the ci
 - DD and mayur will collab on this
 - Release every week until Beta
 - Bump patch version number
 - This should be automated when we get to beta
 - Use bot to assign labels automatically?
 - [Pro bot - by github](#)
- advanced plugin configuration
 - <https://github.com/open-telemetry/opentelemetry-js/pull/628>
- Dns plugin can get into a loop with the exporter
 - Dns plugin cannot be set up automatically. Needs to be manually configured
- Replacing shimmer
 - [remove shimmer, implement new one](#)
 - Shimmer doesn't support multiple patching. Especially an issue with unpatching
 - Doesn't copy some properties like length which may affect application behavior
- Missing for beta
 - Metrics sdk is not complete
 - Context propagation
 - Baggage missing
- Possibly a spec issue - decoupling sdk components
 - Current export pipeline is tightly coupled to SDK making it difficult for a third party to replace individual components of the SDK without also building their own full export pipeline

- We already had this discussion about plugins and it was concluded that plugins should only use public api methods and properties. For very similar reasons, I believe the same should be true of the export pipeline.
- Next meeting
 - Next week Wednesday is a holiday
 - Week after is new year's day in the US
 - Next meeting 2020-1-8

2019-12-11

Attendees: Daniel Khan, Daniel Dyla, Olivier Albertini, Mayur Kale, Nina Stawski, Bartłomiej Obecny

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Daniel Dyla taking over maintainership from Daniel Khan
<https://github.com/open-telemetry/opentelemetry-js/pull/607>
- 0.3.0 RC <https://github.com/open-telemetry/opentelemetry-js/pull/599>
 - Jaeger crash [PR](#)
- More common release schedule after 3.0
 - weekly
- <https://github.com/open-telemetry/opentelemetry-js/issues/612>
 - Remove dns from default list for 0.3
 - Release dns as default when dns plugin is fixed
 - Short term fix would be to have users configure ignore urls
 - Long term fix is to share ignore urls with the exporter some way
 - Only an issue with exporters that use http
 - Http plugin fixes this with a special header to ignore calls, no such mechanism with dns
- Configuring plugins for tracer replaces config instead of merging with the default config
 - Merge config seems less surprising to users
 - Mayur will dig up the old config issue and reopen or create a new one
- Obfuscate/mask confidential trace info
 - Url = https://myurl.com/api/v1/user?user=****&password=****
 - Dynatrace will make a recommendation on this next week
 - Opencensus provides an interface that lets users define how they want to extract data from urls
 - Mysql should respect enhanced db reporting.
 - @daniel dyla will open this issue
- Supporting older TS versions [#588](#)
 - We can add an “old ts versions” recommendation to the readme explaining how to ignore the generated types
- Plugin guidelines
 - Call vs apply
 - Using arguments keyword

- Use public api only or ok to assume our sdk?

2019-12-04

Attendees: Daniel Dyla, Daniel Khan, Olivier Albertini, Mayur Kale, Nina Stawski, Mark, Bartłomiej Obecny, Brandon

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Follow-up from last meeting
 - PR merge guidance:
 - New Labels: <https://github.com/open-telemetry/opentelemetry-js/labels>
 - Please add any missing or useful labels.
 - update contributing.md : Still pending
 - CHANGELOG and Prometheus Export : merged
 - Named Tracers : new PR is opened (draft)
- Examples
 - Examples are complex with package.json files and full working examples
 - Some single-file examples that show simple concepts that all run out of a single package.json might be easier to maintain long term
 - Bart will make an issue
- Oliver integrating ot with a project
 - Difficult because they are using an old ts version. There is a compilation error because OT is using too new a ts version
 - How do other libraries handle this?
 - Oliver will open an issue with a minimal reproduction
- PR rundown, if any
 - <https://github.com/open-telemetry/opentelemetry-js/pull/501> - stuck for a long time (~28 days).
- New issues
 - Default log level [issues/578](#)
 - Mayur: Maybe we can use `LogLevel.Error` by default, makes sense for the production use-case.
 - We are going to make the change when we go from alpha to beta
 - [http-plugin] Missing traceparent header if WHATWG URL is used [issues/583](#)
 -
 - ignoreMethods vs ignoreIncomingPaths and ignoreUrls vs ignoreOutgoingUrls [issues/585](#)
 - Do not want node package to take dependencies on plugins, maybe config can just be a hash
 - TS Getting started guide [issues/586](#)
 - Nina will take this

2019-11-27

Attendees: Daniel Dyla, Daniel Khan, Mayur Kale, Mark Wolff, Nina Stawski, Brandon Gonzalez, Bartłomiej Obecny,

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Prometheus Export
<https://docs.google.com/document/d/1kiNscleTbW-CTiepbY3fy2NXjkuy4FXqzJfxcihuoY/edit?usp=sharing>
 - Decided to add monotonic flag to interface, not proto
 - Only change non-monotonic counters
- CHANGELOG
https://github.com/open-telemetry/opentelemetry-js/pull/574#discussion_r350478918
 - Automated
 - Pls add labels to PRs
- New contributor intro
 - Nina's first day
- Named Tracers [#420](#)
 - No activity 2+ weeks
 - Spec [#354](#)
 - Will modify brandon's pr to fit the new naming
 - Consolidate into a single tracer and move platform specific logic into the registries
- New Issues
 - Alias static `Meter.labels` to `meter.labels` [issues/572](#)
 - Allow parent span to be null [issues/565](#)
 - Change enums to `const` to make types package compile away [issues/538](#)
- PR merge guidance
 - Mayur will add a few labels and update contributing.md
- Metric Workflow:
https://docs.google.com/document/d/1-bWIOCmyoce11icRBj_VO_XqachbLuyid2qYZW027A/edit
 - Rahul from go repo made this workflow

2019-11-20

Attendees: Daniel Khan, Daniel Dyla, Mayur Kale, Bartłomiej Obecny, Mark Wolff, Brandon

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Stuck PRs
 - [feature\(trace\): adds named tracer factory](#)
 - Please review. After review we can merge but it might change
 - Build broken. Master needs to be merged into the PR

- There will be changes due to confusion between different SIGs about Named Tracers/Meters. Possible changes include Factory => Registry name change.
- The question is how does a library get a correctly configured tracer. There needs to be some sort of global factory or registry to get tracers.
- Use cases
 - Disable instrumentation for a particular version of a library
 - Automatically adding component name to span in the exporter
- Milestone review
 - <https://github.com/open-telemetry/opentelemetry-js/milestone/7>
 - Deadline was 11/15. Kubecon delayed a lot of SIG work. Deadline was originally for Kubecon, but it is passed now.
 - Mayur or Daniel will talk to Ted about deadline
 - [#499 Collector Exporter for Web](#) will be for not only web but also node. Bart pushing PR today. Also updating examples
 - [#329 Baggage for OT Shim](#) will not be ready for v3. Moved to v4
 - [#58 Allow span type everywhere](#) mayur is going to take this and open a PR if required
 - [#452 createMeasure](#) Mark Wolff is working on this. Split into multiple PR: initial skeleton, and aggregation later.
 - [#410 Add README for tracer-web example](#) Good intro task. Daniel may have a person to take this
 - [#458 Node 13 CI](#) For now remove from v3
 - [#402 Metric SDK](#) Just waiting on createMeasure
 - [#416 Getting Started](#) Daniel D will get this finished
 - [#488 record start timestamp for counter](#) mayur will take this. PR this week
 - [#489 mysql](#) needs reviewers
 - [#440 API Docs](#) Daniel D will reach out to bogdan or sergey
 - [#540 Prometheus example](#) Daniel K will take this
- What is next on web?
 - XHR still needs to be done
 - If SDK still needs implementation it is high priority
 - Test coverage
- How do we know when it is ready for prod use
 - As larger users start to use otel, it will become more obvious
 - Multi-language tests and examples
 - Benchmarking
 - Feature coverage/examples/documentation/testing
- Build is much faster now, kudos to Daniel Dyla.
- Is it a good time to start CHANGELOG.md?
 - Vote yes: Dan D
- Daniel Dyla needs a deploy key for the gh-pages documentation
 - Will work with Sergey or Bogdan to get a key
- New Issues
 - Change enums to `const` to make types package compile away [issues/538](#)

- Bart taking this
 - Possibility of adding a linting rule
- CircleCI fails on Node 13 [issues/458](#)
- Codecov only shows browser report
 - Speculating that this is caused by a race
 - Mayur can look at this
- Triage issues: <https://github.com/open-telemetry/opentelemetry-js/issues>

2019-11-13

Please join on new Conference link: <https://zoom.us/j/8287234601>

Attendees: Daniel Khan, Mayur, Mark, Bartlomiej Obecny, Daniel Dyla

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Stuck PRs
 - [feature\(trace\): adds named tracer factory](#)
 - Mayur: reviews needed
 - [feat: add mongodb plugin](#)
 - Mayur: reviews needed
- OpenTelemetry KubeCon demo : <https://kccncna19.sched.com/event/6b33fe008d49d19f2a8469548bef4697>
- PR rundown
- New Issues
- Bartlomiej: Trying to set up collector, missing something there. Problem setting endpoint.
 - <https://github.com/open-telemetry/opentelemetry-collector/blob/aea4392e4aa5293afde9f25cdaae35896cb97a1e/examples/demo/otel-agent-config.yaml#L2>
 - <http://localhost:55678/>

2019-11-06

Please join on new Conference link: <https://zoom.us/j/8287234601>

Attendees: Daniel Dyla, Mayur Kale, Mark, Brandon, Bartlomiej Obecny, Dave

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Released 0.2.0 version
 - Next Milestone - [Alpha v0.3](#)
 - Unofficial: v0.3 release by kubecon. Want to demo during kubecon (or qcon ?) (python/node?). Velocity con in berlin using go for workshop
 - Split named tracer/meter into 2 issues for v0.3
 - Tag v3 issues if you want them in the milestone

- Prometheus Exporter
 - Issue with Counter recording:
 - <https://docs.google.com/document/d/1eEL1x1Rjsbm7Cok-NjpLKcrW6zdC8mGxYq0rygcboO4/edit>
 - Decided to reregister for now, in the future the exporter should pull the metrics on demand from the sdk
 - How to handle null label values
 - Decided to skip for now (opencensus approach)
- Consider how to deal with already registered metric
 - <https://github.com/open-telemetry/opentelemetry-js/issues/474>
 - For now we will skip and log. This will be documented
- Enhance CircleCI build
 - <https://github.com/open-telemetry/opentelemetry-js/issues/481>
 - Action items: merge lint and doc tasks, only run coverage on latest node and browser build, only install doc and lint dependencies (not a full install)
 - Daniel Dyla's metric and prometheus builds are failing during compile step
 - Node 8, 10, 11, 12 all fail with "unreachable code" in DNS plugin
 - Browser fails with "condition will always return true" in gRPC plugin
 - Works locally (node v10.15.0)
 - Other people's PRs seem to be working? Mayur is failing locally. He will update his tsc version and see if it resolves the issue, if so he will make a PR to bump tsc on all packages
- Browser explorer to opentelemetry-collector
 - <https://github.com/nstawski/opentelemetry-browser-test>
 - First step is to make an exporter that writes to the collector, in the future we can explore ideas where the browser would export to the already running server
 - Zipkin exporter actually works in the web directly as long as you set CORS headers properly. Makes sense to keep this compatibility possible.
 - Mayur will open issue to create web/node agent collectors
 - Web-collector, web-exporter, or web-exporter-collector?
 - Suggest package names
 - Ot-exporter-web-collector and Ot-exporter-node-collector
- PR rundown
- New Issues

2019-10-30

Please join on new Conference link: <https://zoom.us/j/8287234601>

Attendees: Olivier Albertini, Mayur Kale, Daniel Dyla

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Language agnostic wg: <https://gitter.im/open-telemetry/language-agnostic-wg>
- OT Community: Decentralize membership lists [issues/233](https://github.com/open-telemetry/opentelemetry-js/issues/233)

- Final decision to move membership from community to language repos
- Release automation [issues/440](#)
 - Current process does not provide a guarantee of tag to trigger doc builds
 - Daniel and Daniel will come up with a release automation strategy
- Metrics (Gauge and Counter); support for long and double.
 - <https://docs.google.com/document/d/1g0O7TZ4KH82iwbJmEdBh3qWONnHbap3ajjD2HWA0mY/edit>
 - Solution 3 ftw
 - Should we use bigint to store internal value for integer metrics?
- Heads up: Protobuf uses unsafe evals [opentelemetry-proto/32](#)
 - Possible fix to hand roll types. This is what metrics did.
- New Issues
 - Validate Metrics Name [issues/465](#)
 - Throw Error or do nothing or fallback to default when invalid name.
 - Log and return noop implementation? log and continue?
 - Log and continue could cause errors later
 - Log and provide noop could cause migration issues from systems that allow “invalid” names
 - Log and make name valid? Could be confusing to users if the name changes.
 - For now we can log and noop, possible to revisit later
 - Web browser span exporters should support navigator.sendBeacon [issues/464](#)
 - So far there is no exporter for web
 - CircleCI fails on Node 13 [issues/458](#)
- PR rundown
 - feature(trace): adds named tracer factory
 - Brandon: second thoughts about using singleton for TracerFactory. Runtime may have multiple applications. Instead we can init a global tracer but allow users to new TracerFactory if they need more than 1.
 - DK: default tracer factory should return a BasicTracer not a NodeTracer.
 - Singleton makes a single place to track all created tracers. Multiple tracer factories may require creation of a tracer registry which would then be likely a singleton
 - feat: implement labelset
 - feat(scope-zone): new scope manager to support async operations in web
 - feat(metrics): add registerMetric and getMetrics
 - feat: add mongodb plugin
 - fix: karma coverage
- Missed v0.2 release deadline
 - Other SIGs in the same boat.
 - Python and Go release partial v0.2 without prometheus exporter
- Prometheus exporter
 - Go is doing aggregation in the exporter
 - Other SIGs do aggregation in sdk and the exporter just responsible for sending

- Review open PRs!

2019-10-23

Please join on new Conference link: <https://zoom.us/j/8287234601>

Attendees: Daniel Dyla, Daniel Khan, Olivier Albertini, Mayur Kale

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Review milestones: <https://github.com/open-telemetry/opentelemetry-js/milestones>
 - DK: Prometheus metrics exporter will be done next week. How close is metrics spec to being done? We will probably need interfaces for pull and push metrics backends
 - Mayur: prefer pull mechanism for prometheus. Prometheus wants you to send raw measurements or deltas for counters, rather than real values. This is only an issue for the JS client.
 - Mayur: Waiting for reviews on metrics sdk. For v2 we are targeting counter and gauges
 - DK: (named tracers) Factory pattern makes sense because not all use cases can be covered with a singleton. If you have a singleton the factory will always use the same config. We have a global tracer factory that can be called with an argument to override its configs. Waiting for brandon to come back from vacation
 - Pg plugin
 - Mark: pretty much done and working. Ready for merge
 - Getting started plugin:
 - DD: working in the next day or 2
 - Web tracer doc load plugin
 - This is mostly done. Examples added. PR failing CI on node v13
 - Issue template and README
 - Waiting on review
 - Release targeting october 28, if not ready we may push by a week
- PR Merge Criteria?
 - Do small PRs have to go through the same process as larger PRs? Maintainer should have some discretion to decide.
 - Possibly label PRs that are smaller and require less review
 - Maintainers can comment on PRs to tell PR author why their change may not have been merged yet even if it has received some positive reviews. It is the responsibility of the PR author to make sure all ongoing discussion has been resolved before a PR is merged.
 - Approval from different companies. 3 is a good number, but we also need to make sure not to merge when there may still be dissent.

- Add labels for small/big change, minimum 3 approvals (different companies). There is still some level of discretion to maintainers and code owners. Can we add a line to the PR template prompting author to add appropriate label.
- Only approvers and maintainers can change labels. Mayur is going to see if this can be changed
- PR rundown
 - Basic tracer karma tests - close
 - Merge document load and trivial prs. Possibly also web doc load
- New Issues
 - #456 Resource timing spans
 - Span for each resource the page loads including network timing
 - This should be relatively straightforward
 - #457 When the user loads html, browser makes http but there is no trace header sent. Server generates a trace id and makes a sampling decision, and sends it back to client. Even if the server span is created first, the client should be the parent.
 - In opencensus, when the server renders the template, it injects a global variable into the window. Was called traceparent in opencensus. Possibly we should prefix with ot_? If we decide on a prefix now, it gives us the option to add more variables later.
 - Server could write a server timing header
 - #442 is already done (daniel khan)

2019-10-16

Please join on new Conference link: <https://zoom.us/j/8287234601>

Attendees: Daniel Khan, Mayur Kale, Brandon, Bill Revak (Intuit), Ilan Rabinovitch (Datadog), Bartlomiej Obecny, Midori Kobayashi (LightStep), Mark Wolff, Xiao Li

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Released 0.1.1 version
 - Mayur: Releases yesterday because of some issue. Good thing - someone from Microsoft is building their own SDK and they raised the issue. Now everything looks stable in terms of Alpha release.
- Host API docs somewhere?
 - Bart: GitHub pages might help
 - Mayur: OpenTracing API hosting might provide a blueprint
 - Daniel: I'll try it with GitHub pages
 - Mayur: Will they be regenerated on every pull request
 - Brandon: Should we only do for every release?
 - Daniel: I think it's a good idea to do as Brandon says - create them with every release through a yarn script
- Benchmarking automation

- Mayur: Should we run it with every release?
- Daniel: Let's pragmatically do them during release process and compare with the latest
- Bart / Mayur: With every merge would indicate problems
- Daniel: Might slow down build process
- Mayur: Can we publish the numbers on GitHub pages somewhere
- Daniel: Taking that with me
- Mayur: right now benchmarking < 1 min or 2. 10 samples every pull request should do. Eventually after v3 or v4 alpha we can move around
- PR rundown
 - Initial Metrics SDK
 - postgres plugin
 - add lerna-changelog
 - feat(plugin-dns): add lookup patches
 - Named Tracers (this should go last)
 - Mayur: I believe tracerfactory should be singleton. That was not the case in Brandons PR. If I have to stop all my tracers reusing the factory gives me that.
 - Brandon: At lightstep we have the options to create tracers with their own options.
 - Mayur: Two http instances. One B3 one TracContext. Right now not available with tracer factory.
 - Brandon: Don't understand why this needs to be a factory. Why don't we just do tracer.withName.
 - Mayur: We are targeting named tracer for v3 so we have time
 - Daniel: What will we do with those 2 PRs
 - Brandon: Will merge Mayur's into mine
 - Ted: It should be 0.2 but if this holds back it can be 0.3. Could be experimental.
 - Context Propagation (Ted)
 - <https://github.com/open-telemetry/oteps/pull/42>
 - Mayur: We will go through this and comment
 - Named tracers are a sticking point
 - Context switching is also an issue
- New Issues:
 - Create the transport layer and refactor exporters so that they can be used both in node and web [#422](#)
 - Daniel: Don't we need the collector here
 - Mayur: There is a reference implementation for Zipkin
 - Mayur: is it easy to add the transport layer for zipkin and node
 - Bart: I think it would be easy
 - <https://github.com/census-instrumentation/opencensus-web/tree/master/packages/opencensus-web-exporter-ocagent>
 - <https://github.com/open-telemetry/opentelemetry-collector>

- Add README.md for tracer-web
ex<https://github.com/census-instrumentation/opencensus-web/tree/master/packages/opencensus-web-exporter-ocagent>ample
 - Add new ZoneScopeManager for web
 - Request(plugin-postgres): pg-pool ([#418](#) how folder organize/test all postgres plugins)
 - gRPC plugin does not work with lerna linked examples ([#384 comment](#))
- Alpha 0.2
 - Prometheus exporter will be done this week
 - Metrics SDK will be merged first shot
 - Postgres plugin: Is not mandatory, but always good to have.
 - Getting started guide. Daniel Dyla will do this
 - Mayur will triage other issues

2019-10-09

Please join on new Conference link: <https://zoom.us/j/8287234601>

Attendees: Daniel Khan, Daniel Dyla, Mayur Kale, Ted Young, Mark Wolff, Bartlomiej Obecny, Valentin Marchaud, Olivier Albertini, Brandon, Dave, Xiao

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Release update and next milestones
 - <https://github.com/open-telemetry/opentelemetry-specification/blob/master/milestones.md#alpha-v02-release-validation>
- Browser merged [pull/334](#)
 - Next steps zonejs scope manager, xhr, more plugins
 - AI: ensure issues are created for these items
 - React next? Browser metrics?
 - Recording page loads
 - Recording native XHR requests
 - React timing info
- Metrics? What's the status?
 - Spec is almost there: <https://github.com/open-telemetry/opentelemetry-specification/pull/250>
- Benchmark suite [pull/390](#)
 - Run on every pr? Nightly? Depends on CI resources
 - No state. Possibly compare to current master build
- Changelog management [issues/372](#)
 - Olivier: wanted to know if we go full automatic (via circle-ci) or semi manual (updating manually the changelog)
 - Valentin: I believe Dave was talking about having implemented the fully automatic way with opencensus-web that we could reuse
 - Dave: I will open an issue about that so we can discuss about it

- Example/docs
 - Write your own exporter
 - Write your own SDK
 - Write your own propagator etc.
 - Getting started guide (possibly a sample from the ground up)
 - Daniel Dyla will make an issue and do this
- New issues
 - Implement Named Tracers and Meters
 - <https://github.com/bg451/opentelemetry-node/pull/14> chipping away at it, should have a PR open eod
 - Combine https/http plugins
 - [plugin-http] Create a span builder
 - [Plugin]: Add postgres plugin
 - WebTracer - auto instrumentation - plugins
- Going to kubecon? Come to Observability Summit!
 - <https://observabilitysummit.com>
 - Add to your kubecon registration
- Default behavior:
 - <https://github.com/open-telemetry/opentelemetry-specification/issues/208>

2019-10-02

Attendees: Daniel Khan, Daniel Dyla, Mayur Kale, Olivier Albertini, Peter Marton

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Alpha release check in
 - Cut release as 0.1.0
 - Python: 0.1a.0
 - Notes:
 - Mayur: I heard it should match the spec version, which is currently 0.2.0. Should we release it as 0.2.0?
 - Daniel: until 1.0 the numbers don't matter as much
 - Npm publish after pr is merged. Use "alpha" in github release name

Next Steps for Alpha v0.2

In scope of SDK Alpha v0.2 release are:

- * Basic telemetry pipeline for traces complete
 - * Built-in samplers (percentage sampler).
 - * SpanProcessor interface and implementations:
 - * Default and built-in processor
 - * Batching processor
 - * Block when the queue is full processor
- * Exporter interface
 - * Jaeger and/or Zipkin exporter

- * Distributed context
 - * Basic implementation
- * Metrics
 - * Default and built-in processor
 - * aggregation implementation
 - * Metrics exporting interface
 - * Prometheus exporter
- Required Spec RFCs for Alpha v0.2:
 - * Metrics
 - * Named Tracers
- Next Steps for Alpha v0.3
- In scope of SDK Alpha v0.3 release are:
 - * Collector support for Otel protocol
 - * Otel exporters
- Required Spec RFCs for Alpha v0.3:
 - * Global Init
 - * Context (separate baggage, renaming, etc)
 - * Protocol
 - * Semantic Conventions

-
- Shared SDK types since some are moving to the SDK level per [issues/376](#)
- Metrics SDK update (if any)
 - Daniel: too many open questions
 - Prometheus Exporter update (if any)
- WebJS
 - Request for reviews as most comments have been responded to
 - Examples in the README
 - Add comment that says it's a wip?
 - ScopeManager
 - add autoinstrumentation for scope manager
 - <https://github.com/obecny/opentelemetry-js/blob/tracer-web/packages/opentelemetry-tracer-web/src/StackScopeManager.ts>
 -

2019-09-25

Attendees:

Olivier, Mayur, Daniel, Valentin, Daniel Dyla, Daniel Khan, Jake Riesterer, Mark Wolff, Bartlomiej Obecny

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Alpha release check in

- Metrics SDK
 - Tag as private
 - Untag as alpha
- Prometheus exporter
 - Daniel: some additional work needed for this. Added base package with types. Merged today. Unsure about process. Exporter is relatively easy, e2e demo may be more difficult. Metrics sdk needed.
 - Brandon: other projects don't have metrics sdk yet either
 - Daniel: will take care of prometheus exporter and can work on metrics but metrics will not be ready for alpha. Will sync with Brandon to get sdk out as we go.
 - Daniel untagging metric as alpha
 - Mayur: should we add private tag to the package (prometheus and metrics)
 - Daniel: yes
- Browser
 - Bart: Not sure browser will be part of alpha. Reluctant to push by the end of the month.
 - Dave: not part of alpha
- PRs
 - Does anyone have PRs that they feel are blocked/should get extra attention?
 - Daniel: Add milestone if you think it should be in alpha
 - <https://github.com/open-telemetry/opentelemetry-js/pull/335>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/333>
 - <https://github.com/open-telemetry/opentelemetry-js/pull/325>
 - Mayur: READMEs still need work esp node and web tracers. Will create issue after meeting
 - Mayur: test api and sdk for last minute issues? Found some issues with http.
 - feat: add BatchSpanProcessor [pull/238](#) has been open for a while
 - Approved just needs merge
- Jaeger Exporter issue
 - Mayur: http example issue. We use jaeger client which has batching logic. Data is not sent immediately. If you run the http example you may not see the span in the jaeger exporter because the jaeger export buffer limit is not reached.
 - Oliver: tested opencensus example and is able to see the span in jaeger
 - Mayur: when we call the exporter from the span exporter we are only sending to the client. Opencensus flushes immediately. Opentelemetry does not flush. One possible solution is to flush every time we export to jaeger. When the exporter is flushed the connection is closed so this may be a problem.
 - Dave: may not be a problem to flush every time because we are doing our own batching
 - Mayur: maybe we should allow jaeger to batch

- Dave: may be confusing. Afraid someone may forget to put the batch exporter wrapped around jaeger or zipkin etc. If the default exporter batches then you can't forget by composing batch processor with a lower level not batching exporter. User can dive deeper into implementation if they need a more advanced use case.
 - Mayur: maybe not possible before alpha
 - Dave/Brandon: agree should not block alpha
 - Brandon: if this is specific to the example, should we just add a flush to the example. Production systems may not want this behavior.
 - Mayur: could be issue in prod. 65k default limit for span batch. Until you hit the limit, the batch won't be sent.
 - Brandon: we have a tick on our batch
 - Mayur: jaeger does not. Not a blocker for alpha. Will open pr with recommendation.
- If people are feeling like CircleCI builds are queueing up, we can start spending money on it to increase capacity (we meaning @brandon/lightstep)
 - General agreement
- Exercise the API and SDKs?
- New issues
 - ExportResult should not be exclusive to tracing
 - Add Usage instructions for OpenTracing shim package
 - scope-manager: describe how they works and their limitations
 - Plugin/gRPC: patching does not work when used with makeGenericClientConstructor
 - Plugin/gRPC: Metadata usage prevents span/client propagation
 - Add baggage support to the opentracing shim
 - Avoid infinity loop when exporter use http to send spans
- Release Process
 - Mayur: should we use opencensus release process? Npm org exists.
 - <https://github.com/census-instrumentation/opencensus-node/blob/master/RELEASE.md>
 - Dave: opencensus web is entirely automatic when you create a tagged release. github/circle trigger a release build automatically. Less risk of human error.
 - Uses an automatic publish key.
 - Dave/Mayur will discuss
 - See the OpenCensus Web script that does this:
 - CircleCI publish task:
 - <https://github.com/census-instrumentation/opencensus-web/blob/68aa988732e03f2de49ebc0213c3b2525aa5b68b/.circleci/config.yml#L46>
 - <https://github.com/census-instrumentation/opencensus-web/blob/master/.circleci/publish.sh>
 - Mayur: github apps may be something worth looking at
 - Oliver: how does versioning and changelog work in opencensus

- Mayur: it is a manual process using npm scripts
- Bart: could be automated using circleci

2019-09-18

Attendees:

Peter, Mayur, Olivier, Dave, Brandon, Dan Dyla, Daniel Khan, Mark Wolff

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Alpha release
 - Goal is to stabilize codebase and get these PRs listed to master:
 - Also need:
 - Prometheus, exporter, code samples, readme, documentations.
 - Integration tests?
 - Daniel to make a Milestone for Alpha launch
- Reviews of the PRs
 - OpenTracing Shim
 - feat(metrics): adds NoopMeter implementation [pull/278](#)
 - rename node-tracer to node-sdk [pull/266](#)
 - feat(baseplugin): add internalfilesloader [pull/240](#)
 - feat(Node-Tracer): initialize plugin loader [pull/252](#)
 - fix(http-plugin): ensure exceptions are handled [pull/273](#)
- Chat about next steps for browser tracer [issues/276](#)
 - Core and basic-tracer work in both Node and browser
 - See draft pull request for Karma tests in basic-tracer: [pull/280](#)
 - Where should internalfilesloader live? Currently BasePlugin, but n/a for browser
 - Dave and Bart to meet to chat about browser tracer
- I'm getting errors when I run "yarn test" or "yarn compile" on master ... can we set up branch protection to make sure that doesn't happen?
 - Let's wait until this happens a second time to find a best practice for this
 - Branch protection option "Require branches to be up to date before merging"
 - Action item on Mayur.
- Http request 'error' and 'aborted' handling [comment/273](#)

2019-09-11

Attendees:

Peter Marton, Mayur, Olivier, Brandon, Daniel Khan

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Can we make scopeManager as an optional param?
(<https://github.com/open-telemetry/opentelemetry-js/issues/194>)

- Daniel: generally prefer it to explicit but not a strong opinion on this
 - B: optional with noop default is fine (but also no strong opinion)
 - Open issue for discussion
- Need more reviews on OpenTracing Shim and gRPC plugin.
 - <https://github.com/open-telemetry/opentelemetry-js/pull/212>
- Plan for alpha release (~20 days remaining)
 - Integration tests
 - Readme
 - Daniel can take this
 - Examples
 - Daniel can take this
 - Testing with HTTP, gRPC (mongodb?) plugins
- New Issues:
 - node-tracer: review if config can be optional when creating it
 - All fields in the config are currently optional, can we just make the config optional as well?
 - Should make it optional
 - Unable to create multiple child spans with the same traceid for requests
 - SDK: Add Prometheus metrics exporter
 - Daniel would like to take care of this.
 - Exporter/Jaeger: use correct startTime and duration from ReadableSpan
 - Will be fixed by peter's PR.
 - Avoid side effects by installing event listeners
 - Can we create a span without adding a listener?
 - Avoid changing polymorphic structure during patching
 - Ensure exceptions are properly handled and propagated

2019-09-04

Attendees:

- Brandon, Mayur, Dave, Mark

Agenda

Tentative goals for the next meeting, feel free to add your own!:

Announcement: We decided to shoot for an alpha at the end of this month, not a beta! Whew! :)

The goal of an alpha is to start writing instrumentation and create a feedback loop for the project.

Alpha means an initial working version of the following:

- Tracing, Metrics, and Context propagation APIs
- Jaeger and Prometheus exporters
- OT/OC Bridges

Alpha does not mean:

- conforming to a particular version of the spec
- promising that we won't break things

Tedsuo's request: We'd like to start communicating more clearly about the state of each implementation. Please update the README to have section at the top with the following:

- Current state of the project, with a focus on the items listed in the alpha description
- Installation and a super basic Quick Start guide

I'm sure there will be questions about deliverables, and about how to communicate about the current state of the repo. Please send questions my way! Since we're all working in parallel, we'd like to try things in different SIGs, and then share best practices.

- Big Board:
 - We'd like to add something to the website which shows the state of the SIGs at a glance.
 - Please leave comments and give feedback on this prototype <https://docs.google.com/spreadsheets/d/1FHt3uJgQWN-UWbgV4uMIPUr36JiJCfZ7ryQbHfurjKA/edit#gid=0>
 - "I don't understand what X means" and "Y appears to be missing" are great pieces of feedback.
 - Mutations is contracting for lightstep, hoping to get them started on web JS work.
 - Basic tracer
 - Code examples and documentation
 - Initial page load instrumentation
 - Automatic XHR tracing
 - Execution context, e.g. tracing the initial interaction that kicks off multiple xhr requests.
 - Zones.js was used for census
 - Rest of various plugins
- Open PRs
 - Review and merge zipkin exporter [pull/192](#)
 - OpenTracing shim [pull/212](#)
 - Tslint rules for license headers [pull/106](#)
 - Review gRPC instrumentation [pull/215](#)
 - Jaeger Exporter [pull/231](#)
 - feat: add InMemorySpanExporter, MultiSpanProcessor [pull/234](#)
- Open Issues
 - <https://github.com/open-telemetry/opentelemetry-js/issues/219>
 - Will close <https://github.com/open-telemetry/opentelemetry-js/pull/221>

2019-08-28

Attendees:

Peter Marton, Mayur, Brandon, Olivier (Montreal city), Daniel, Dave, Mark

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- fix(basic-tracer): performance.now() is relative from start time
 - <https://github.com/open-telemetry/opentelemetry-js/pull/206>
 - Peter: could we convert relative to absolute before exporters? How about ReadableSpan?
 - Dave: I like the idea, could be a method
 - Dave: with performance.now we get nanosecond accuracy
 - Dave: can't just use timeOrigin to add with performance.now, you would lose accuracy
 - Peter: accuracy can be exporter specific
 - Dave: Stackdriver accepts nanosecond
 - Mayur: when you pass startTime, how would we handle relative times?
 - Dave: OT could subtract the Date.now()
 - Peter: how about storing absolute in tuple
 - Dave: would be the API a tuple?
 - Peter: API could be just nanosecond
 - Dave: it could accept multiple things and do the smart thing
 - Dave:
 - API accepts: number|Date|[number,number], can be performance.now, epoch millis, Date or process.hrtime result. Smaller timestamps are performance.now, larger are epoch (b/c no Node or browser processes were running in 1970)
 - Internally, it stores a tuple like process.hrtime
 - ReadableSpan has accessors for the tuple
 - Agreement on this
 - Peter: I'll change my PR
- Proposal: Plugin configuration options
 - <https://github.com/open-telemetry/opentelemetry-js/issues/214>
 - Mayur: I'm looking for feedback on these APIs
 - Daniel: I prefer the first one because it allows individual config
 - Peter: any reason why it's an object not an array?
 - Mayur: Object key is the name of the plugin
 - Daniel: we could have a global config with approach #1
 - Peter: there could be a pluginLoader.config({ ... }) API for global plugin config
 - Mayur use-cases for enabled: in OC we load all the plugins by default, maybe you want to disable some if you have an issue with some of the plugins
 - Mayur: Option #1 looks good to me use-case wise
 - Mayur: I'll open a PR with option #1
 - OPTION 1:

```

pluginLoader.load({
  Http: {
    enabled: true,
    ignoreMethods: ['POST'],
    ignoreUrls: ['/ignored/string'],
    package: "@opentelemetry/plugin-http"
  },
  mongodb: {
    enabled: true,
    enhancedDatabaseReporting: true
  },
  grpc: { enabled: true },
})

```

- OPTION 2:

```

PluginConfig {
  enabled?: boolean,
  plugins: {
    http: 'package name/path',
    mongodb: '@opentelemetry/plugin-mongodb',
    grpc: '@opentelemetry/plugin-grpc',
  },
  ignoreMethods?: string[],
  ignoreUrls?: Array<string | RegExp>,
  enhancedDatabaseReporting?: boolean,
}

```

- [BasePlugin]: implement internalFilesExports
 - <https://github.com/open-telemetry/opentelemetry-js/issues/216>
 - Mark: In OC you can patch on internal file exports
 - Some implementation not ported over, do we want to include it in the baseplugin?
 - Mayur: when I created it it was an initial wireframe, we can port over
 - Mayur: Would it be configurable?
 - Mark: configuration would support semver ranges and list of file paths
 - Mayur: does it make sense to port over the plugin configs?
 - Mark: yes
 - Mark: would be plugin config chosen by the user and us?
 - Mayur: yes, we would have one package that comes with defaults by usn and another where they need to specify
 - Mark: maybe user wouldn't gain any value from changing these configs
- Continue discussion on intiGlobberTracer.
 - <https://github.com/open-telemetry/opentelemetry-js/issues/194>
 - Can we make scopeManager as optional param?
 - Mayur:

- Personally, I don't like passing compulsory param (NoopScopeManager) See line # 7.
 - If I use NodeTracer, I should not have to manually pass the parent span it should fetch it internally from getCurrentSpan() - from the current context. See line # 16-18.
- Mayur: Should we allow users to pass TracerConfig while opentelemetry.getTracer().start()? See line # 32.
 - Peter: how about adding restart() which reuses initial start's config and you can only call start once with configs
 - Daniel: would it be pause() than?
 - Peter: maybe, stop could be kind of destroy
 - Mayur: how does it look like in OpenTracing?
 - Brandon: I think() start could create a completely new tracer
 - Brandon: it's a rare case
- Brandon: about static tracer method...
- Zipkin PR Merge?
 - Peter: let's fix relative/absolute times first
 - Mayur: is spec a blocker?
 - Peter: I don't think so
- Oliver HTTP PR update:
 - Looking for final consensus
- gRPC Plugin PR:
 - Mark: I'll open up for review soon
- B3
 - Mayur: I implemented Daniel's recommendation

2019-08-14

Attendees:

Daniel Khan (Dynatrace), Roch Devost (Datadog), Brandon Gonzales (Lightstep), Mayur Kale (Google), Dave (Google), Mark (Microsoft), Valentin, Peter Marton (Netflix)

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Exporter HTTP Client
 - Peter: Question: Do we want to include a HTTP library for the HTTP calls or should we roll our own one. Right now I'm using raw.
 - Conclusion Mayur, Roch, Daniel, Peter: Let's make it the responsibility of the package.
 - Roch: We should recommend one.
 - Daniel: Should we recommend one now because we need it already for zipkin?
 - Peter: I use raw right now and it's sufficient
 - Mayur: We should some kind of retry logic in place though.

- Peter: Could it be that this logic may be on a higher level of the tracer?
 - Mayur: I have to check with the spec
 - Peter: the exporter has different error returns like retryable error. May this be the retry we are talking about?
 - Roch: The logic itself should be up to the exporter
 - Peter: I kind of agree bc the exporter may know better. Let's clarify this.
- Supported Capabilities
 - User defined headers? (authentication etc.)
 - Follow redirect?
 - Configurable timeout?
 - Retry logic?
 - TLS?

<https://github.com/open-telemetry/opentelemetry-specification/issues/142>
- Built-in vs. library
- Zipkin Exporter Questions
 - Exporter Status: Based on exporter status code: action item to clarify
 - Peter: Zipkin does not have status code. How will we handle this.
 - Peter: As we are not clear on all of that, let's raise this at the spec level.
 - Span.status mapping
 - Span.shared, how to determinate?
 - *"True if we are contributing to a span started by another tracer (ex on a different host)" - Zipkin*
 - How to determinate it?
 - Shutdown behaviour
 - No op?
 - Opportunistic? (issue a request)
 - Graceful shutdown? (we don't have a callback, why?)
 - Brandon: we should definitely have something like a graceful shutdown
 - Peter: Same opinion
 - Mayur: Spec says content should be flushed but non blocking
 - Daniel: Dynatrace also wants flushing
 - Roch: Should be exporter specific. We would go for a shutdown.
 - Peter: the conclusion is that we still need a callback.
 - Daniel: It would be fire and forget if we don't flush
 - Peter: In some cases it is a debugging tool and it may not be so important so both makes sense. So making it configurable would make sense.
 - Conclusion: Callback
 - Testing
 - Integration? (Zipkin server accepts span)
 - Peter: Do we want to make an integration test. Might be easy with circle CI.

- Dave: I also thought about integration tests for web and actually building this all, loading it into the browser etc but I guess this would be something for hardening and can be deferred for a while
 - Peter: So it's desired and not required
 - Mayur: When we go GA we should have it in place. We didn't have it for OC but we should have it for plugins and exporters.
- Exporter Batching
 - Will we provide a reusable batching logic for exporters?
 - Peter: As Jaeger and Zipkin both use batching it may still make sense to have a generic exporter.
 - Mayur: We should raise that on the spec.
 - Peter: ok let's do it on the spec.
 - Mayur: How can we make this decisions all compatible to OC and OT
 - Dave: I would think that we should have a detailed documentation for each of those libraries on how to make the migration. This would free us from being 100% backwards compatibility. That should build enough trust
 - Peter: I agree with Dave that it's better to inform instead of aiming for 100%.
 - <https://github.com/open-telemetry/opentelemetry-specification/pull/205>
- New package names (proposal)
 - **opentelemetry-api** = opentelemetry-types + opentelemetry-core
 - **opentelemetry-basic-sdk** = Implements opentelemetry-api i.e. trace, meter, logs, etc.
 - **opentelemetry-node-sdk** = opentelemetry-basic-sdk + scope_manager_impl
 - **opentelemetry-node-agent** = opentelemetry-node-sdk + plugins (PluginLoader)
 - Peter I don't know if the distinction between node-agent and node-sdk makes sense.
 - Mayur: If we have more than 10 plugins and maybe a user is only interested in MongoDB and express this would bloat the package.
 - Roch: not a fan of this structure because the distinction because types and core is important. For the node agent - let's not use the term agent because it's confusing depending on vendor.
 - I am not sure if we want to separate out metrics, traces and logs
 - Dave: What if basic sdk exists?
 - Roch: That's what I had in mind.
 - Roch: Is there anything on top of the node agent or would it be the top package?
 - Mayur: Would think to have it in separate package.
 - Roch: As a user I go to the Otel documentation, would this be the module to install
 - Mayur: Then just opentelemetry-node
 - Peter: I like that
 - Roch: Advanced users can use the SDK but basic just go for *-node
 - Mayur: Will there be something like OpenTelemetry web?
 - Dave: Yes I'd imagine that there'll be something like that

- Daniel: SDK is a bit confusing but everything else is fine
- Dave: Another option would something like *-all
- Roch: If we just call it 'opentelemetry' and npm install opentelemetry would install it and it would just work independently of browser or Node
- Mayur: Summarize: We will keep the types and core like it's. I will create issues for the rest. Going for node-all that will load all the plugins by default.
- Dave: would it be ok to make basic-sdk just a shell package
- Peter: -all really?
- Roch: maybe confusing
- Dave: let's make *-all the default and not name the package like that
- Peter: Agree with Roch
- Mayur: I will create issues after the meeting
- <https://github.com/open-telemetry/opentelemetry-js/pull/201>
 - Do we want to return a NoopSpan when no span is found in the context ?
 - Valentin: It's possible that you don't find the current spsn because of context loss in third party modules, in this case do we want to return a NoopSpan or directly null ?
 - Roch: I'd just go with returning null to avoid confusion
 - Dave: How does it work in OC?
 - Valentin: it's not handled, can even crash like with redis plugin
 - V: that's this is the use-case I want to solve
 - Dave: I'm okay with null, until it's documented and we change the TS types
 - Dave: we need to create awareness that it can happen
 - Agreement on returning null.
- OpenCensus / OpenTracing bridge
 - Is this still a plan?
 - Do we have anything that's not compatible?
 - Brandon: I think span logs are the only thing that are not compatible today
 - Brandon: we need to check OC events, maybe they are compatible with logs
 - Brandon will create an issue to discuss span logs compatibility.
 - Brandon start to work on shim on Monday.
 - Lightstep will staff on this.
 - Mayur: isn't it too early to work on this? API is not stable yet.
 - Brandon: I think there are components that makes sense to start work on
 - Mayur: my idea for OC was not a shim, it's more like calling OT APIs from OC APIs
 - Brandon: sweet
 - Mayur: any concern
 - Peter: just keep in mind that OT is less mature so we can jeopardize OC stability if we do too early
- Make Span implement ReadableSpan via getters instead of regular properties
 - <https://github.com/open-telemetry/opentelemetry-js/issues/196>

- Dave: I'm not suggesting sd I'd recommend it, it's more an idea to prevent API abusing
- Roch: don't have a strong opinion
- Dave: I like the idea of a wrapper that exposes certain things via getters
- Peter/Dave: some part of confusion is that these are non underscored properties made hidden only by TS types
- Conclusion: leave it as it is for now
- API/SDK usage based on current implementation
 - <https://github.com/open-telemetry/opentelemetry-js/issues/194>
 - Mayur: please take a look and express your feedback
 - Dave/Mayur: Global Tracer
 - Roch: what happened to start
 - Mayur: currently start doesn't accept any parameters
 - Running out of time...

2019-08-07

Attendees

- Mayur Kale (Google)
- Roch Devost (Datadog)
- Peter Marton (Netflix)
- Oliver Albertini (Montreal city)
- Daniel Khan (Dynatrace)
- Brandon Gonzalez (Lightstep)
- Thomas Chan (Lightstep)

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- New member
 - Oliver is coming from Jaeger background
 - Thomas Chan: work for Lightstep with Brandon, primary focus is the spec (metrics, context propagation)
- SpanProcessor (BatchSampledSpanProcessor and SimpleSampledSpanProcessor)
 - Mayur: some conclusion, worth to discuss
 - Span processor is required to attach to async callback when span started and ended
 - BatchProcessor: number of spans or timeout => pass it to exporters
 - Idea is that exporter is not called with every single span
 - SimpleProcessor: no queue, as soon as span ended, pass it to exporter
 - Peter: what about multiple exporters with different needs (one batch one simple)
 - Mayur: currently one processor but multiple exporters
 - Khan: should it be exporter - processor pairs
 - Peter: that would support mixed scenario
 - Mayur: is this a use-case?
 - Peter: what about z-pages or logging with zipkin

- Brandon: yes, there are use-cases
- Roch: maybe always call exporter with every span and leave it to vendor
- Peter: agree, but still could provide reusable batching logic so it's easy for exporter to use it
- Daniel: is processor part of the spec?
- Daniel: exporter could do throttling and other control, it's very backend/vendor specific
- Roch: yes, it's very vendor specific
- Mayur: Would it be a memory overhead having multiple batchings?
- Peter: can we implement in a way that it's not duplication, like no cloning
- Roch: yes, I think it's possible, it would be reference and array has low overhead
- Daniel: how is it handled today?
- Mayur/Daniel: so references are how by exporters
- Agreement on sending all the spans when they ended to all the exporters
- Peter: if we see value in providing batching logic, it could be an exporter base class like ExporterBatch and used by Zipkin, Jaeger etc OSS exporters
- Jaeger will be the first one implemented
- SpanExporter interface
 - Implement Jaeger/Zipkin exporter for end-to-end testing and to see how all the pieces fit together?
 - Zipkin, Jaeger and ? are the default exporters
 - Brandon: we should ask Jaeger folks if they want specific things or we can go ahead
 - Peter: do we have Zipkin representative? Probably not
 - Action item: reach out to Zipkin/Jaeger folks
 - Peter: I can work on Zipkin exporter based on OpenCensus exporter
- HTTP plugin [pull/161](#)
 - Mayur: what about shimmer?
 - Roch: I think it's fine in this use-case. It can be expensive when you call it frequently, because it checks monkey patching.
 - Roch: ScopeManager can be called thousands of times a second
 - Roch: TLDR; only use shimmer when it's called infrequent
 - Oliver: I'm planning to increase the coverage by Friday
- Release OC Web 0.0.4 for future merge on OT JS
 - Representative is not present, skipped.
 - Blog post
- Code coverage
 - Mayur: code coverage regression, we should put a threshold
 - Roch: coverage is not a goal, the goal should be feature coverage
 - Mayur: current threshold is 80%, do you want to go up to 90%
 - Roch: I think 80% is low
 - Roch: 90 or 95% would make sense
 - Mayur: now it's 92% I'll change it to 90 or 92 and enforce in PRs
 - Agreement on increasing from 80%

- SDK Beta timeline
 - Brandon: end of September is a big milestone
 - Should we do alpha, beta etc? How about involving customers?
 - In 15 days we should have one exporter in place
 - Brandon: I agree I feel we are pretty close to have a working instrumentation (least on the trace side)
 - Mayur: should we do a beta and testing on that?
 - Brandon: We could give it to customers to collect feedback
 - As vendors using internally is probably doesn't give the same value
 - Mayur: is someone on the Metrics meeting?
 - Thomas: will schedule the meeting and put on the calendar today
- Readable Span
 - Mayur: I've a PR open with some open comments, but quite lost, let's chat
 - Mayur: name is confusing as it meant for exporters not from API point of view. It's a convenient feature.
 - If we implement Readable interface it means the user will also able to use these APIs
 - Roch: user should be able to write but not read, exporter should not be able to write but should be able to read
 - Two interfaces, one class but you pass interface as a type. So you control visibility through type
 - Daniel: what is the idea behind preventing reading?
 - Peter: if we do it through interfaces and you don't use TS is it useful?
 - Roch: it's JS, you could do anything anyway
 - <https://github.com/open-telemetry/opentelemetry-js/blob/master/packages/opentelemetry-types/src/trace/span.ts#L28>
 - Brandon: we should make it clear in the docs
 - Mayur: should we still implement Readable interface?
 - Roch: I'll take a another look
- Spec vs. JS
 - Roch: there were some cases when spec was merged even when it was not ideal for JS/Node
 - Roch: what would be the way to hear our voice more? Not necessary blocking the spec.
 - How do we make sure the API make sense before it's locked?
 - Peter: could you please call our things when happen in Gitter, not everyone follows spec. It would give an opportunity for more people to show up on the PR.
 - Roch: What's the timeline after initial release? How easy would be to change API after September? What if it's breaking?
 - Thomas: original goal was 1.0, so breaking should happen before it
 - Roch: solution could be to make 2.0 possible in couple of months if we want to move fast with 1.0

2019-07-31

Attendees

Daniel Khan (Dynatrace)

Mayur Kale

Dave

Brandon (LightStep)

Roch

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Move BasePlugin out of core package?
 - Dave: how does the concept of Plugin apply if there is a build on the web like webpack such that code is flattened?
 - Daniel: there are some web use cases for this, e.g. React plugin
 - Dave: leaving it in core is fine for now and we can refactor it
 - Mayur: Open an issue
- Who should generate traceId (and spanId)? Is it Tracer or Span?
 - Mayur: currently we generate trace/span ID in the Span, but I think it should be in the Tracer. E.g. if the trace is coming from the wire.
 - Daniel: we just talked about this in the trace-context SIG yesterday. Are we sure that OpenTelemetry will be in charge of sampling decisions. Should it be done by the user in some way?
 - Brandon: I strongly disagree. I think as a user I shouldn't need to worry about sampling, unless I really want something to be sampled
 - Daniel: if we as trace-context working group is not sure about whether to do this, it's a conversation that we need to have
 - Brandon: I'm unsure about how sampling fits in
 - Daniel: there will be more follow-ups on this
 - Brandon: I think trace ID and span ID should be handled by the tracer.
 - Dave: can we allow a way for the user to specify a span ID
 - Brandon: yes, there should be a way to do that as part of the conversation about SpanData
 - Roch: there could be different span implementations with different logic for IDs. E.g. the NoRecording span always creates a trace ID but only one span ID. My opinion is that it should be in the Span or the SpanContext, wherever the trace ID lives. But it should also be overridable.
 - Brandon: historically with LightStep tracers, the tracer is responsible for creating the trace ID, but if the received trace context lacks an ID, a span is responsible for creating its ID
 - Mayur: in OpenCensus, the tracer is responsible for creating a trace ID for the sampling decision, whereas span ID is the responsibility of the Span.
 - Roch: how does that work with the no-op span?
 - Brandon: if we have no-op singleton, but if we have no-recording span, that's different

- Mayur: we are going with no-recording span
- Dave: does the OC approach where tracer generates trace ID by default and span ID by span by default fit your use cases?
- Roch: the no-op just ignores everything anyway. For no recording, generally you would have a single span ID for everything, because you don't care about instrumenting. I would say that works. I think we should just go with an approach and if it doesn't work well we'll find out.
- Mayur: Daniel, do you agree on this?
- Daniel: this is a larger topic anyways
- Support for sending spanID in the Span constructor?
 - Dave: any updates on the SpanData replacement?
 - Brandon: RC out for removing SpanData. Felt it added a lot of extra complexity. You can create options for start and end options, e.g. start timestamps and explicit end timestamp. If we want to fully meet the requirements.
 - Dave: the use case is for browser initial page loads. Is there any way I can support the RC?
 - Brandon: I imagine a thumbs up would look good to me.
 - Dave: I'll comment to say I want this
- Related question about Span class
 - Mayur: there are currently no Span "get" methods, only "set" methods. We will need that to use in the exporters
 - Brandon: I think I see your point. The Span object then becomes exported
 - Mayur: if I see Java and the other implementations, they convert the Span object to the Span proto, and then send to the exporters
 - Dave: could we use a similar approach as toString, and have something like toJSON?
 - Roch: but what's the argument for not just passing the Span. There is conversation around making a lot of things immutable. But we want vendors to be able to do things to the span after its done. I think it would be a lot easier for everyone if you don't make it immutable. I would just pass around either Span or SpanContext
 - Mayur: that's what we do in OpenCensus. But currently attributes are private.
 - Roch: I would go with exposing things on the span and maybe even have a SpanData interface.
 - Mayur: I have one PR in the pipeline that adds a SpanData
 - Dave: I really like Roch's idea is to just allow mutation of direct variables
 - Daniel: there is a spec around SpanData being immutable. Is there progress?
 - Roch: I feel like immutability is language-specific to an extent. It should be language compliant though. In JS it's very common to not make anything immutable. Monkey-patching is all over the place. That's the norm due to the nature of the language.
 - Daniel: for that we need to understand the intent. I have a use case in mind of SpanData whether or not it's immutable. Immutability could be made clear in comments. If you store a SpanData somewhere and the underlying data

changes, then you might export something that's different than you thought you would export. I'm trying to understand the reasoning on the spec why they point this out. We need to clarify this at some point.

- Roch: we also have TypeScript in there, we could always make the types `readonly` but so that if you respect the TS contracts you won't mutate anything, but it's still possible JS-wise to do it. We need to be able to add things to a common object that is received everywhere. We need that at Datadog due to some features we have that are unrelated to the core tracing. Could be span or span context that is passed everywhere.
- Daniel: yes, of course, every vendor has some similar use case
- Brandon: to tease this apart:
 - Span interface, add attributes/events/etc
 - SDK interface implements
 - Have SpanData interface that SDK implements
- Roch: I think that would make sense. If there is a strong need from users. Other than to implement hacks around functionality that is not release
- Brandon: it feels funky to have SpanData interface on the API span
- Dave: how does that work with the exporter
- Brandon: you have the tracer as a middle layer, similar to Java with proto
- Dave: so you have something like this:
 - Span interface is write-only
 - SpanData is read-only
 - e.g. BasicTracer sends SpanData instances to exporters, but BasicSpan implements both Span and SpanData
- Roch: we would need to document this so that it's very clear, so people don't get confused
- Dave: could this be the way to have the best of both worlds, both spec-compliant and very Java-Scripty?
- Mayur: should it still be called SpanData?
- Brandon: yes, it's triggering
- Dave: what about ReadonlySpan?
- Mayur: in Java it's called ReadableSpan. I like both names
- Roch: but is it confusing that Span you can only write to it? Do we need WritableSpan, REadableSpan and Span?
- Dave: is there a case where a user wants to read data from a span?
- Brandon: not besides a customer exporter. Then again this is coming from Joshua saying "user should never need to read instrumentation data"
- Roch: I try to remove as many limits as possible. Even if I can't think of a use case that the user needs. My preference is to not have the distinction between readable and writable. But the user should be able to do what they need. My general feeling is that there should not be a distinction. Span should be readable and writable. Not sure if that deviates too much from the spec b/c of the immutability discussion

- Brandon: coming from the OpenTracing world, if the user ever wanted to do something weird, they could always cast away from the OT span to the vendor specific span.
- Daniel: I prefer to make APIs to be rather strict, you can always work around it. I like to think about the intent of this thing, then I consider “maybe I shouldn’t use this for any type of baggage”, but then I know I’m moving away from the API intent. In Java everything would be different. You are more constrained.
- Roch: I totally agree with that, but as long as we don’t limit it. I’m fine with the types being very strict, but don’t use `Object.freeze`
- Daniel: if it’s even in the comments, then an implementer knows what we would expect.
- Dave:
 - `Span` - write/only via TS types
 - `ReadOnlySpan` (or `ReadableSpan`) is readonly by TS types
 - Both are implemented by tracers with the same object
- Brandon: I think so, would love to get this in an issue and sleep on it
- Mayur: I would like to use `Object.freeze` for `ReadableSpan` when the span is ended. The user might mutate it.
- Dave: I’d be concerned about the performance impact of a deep freeze and the limit of it
- Mayur: but do we allow mutation of the span after it’s ended?
- Roch: I think it’s still about the types, but it can’t be altered from that point through types, but not really freeze it, but at least not freeze it deeply. Then if vendors could mutate the child properties of the span, then the vendor has a vendor specific property even if the span itself is frozen
- Dave: guard against a user who calls `span.end()` and then tries to add an attribute or a bad exporter
- Mayur: more concerned about the user
- Dave: could we handle this via a logging message or a warning
- Roch: something I’m wondering `Object.freeze` behavior if you try to alter it.
- Dave: just has no effect (in non-strict mode)
- Mayur: it throws an error in strict mode
- Daniel: the use case you’re also talking about is propagating along the way. Then shouldn’t this be in the span, e.g. some metadata property on spans where you can put in your stuff?
- Roch: you mean a way to have vendors or users to have their own space in the span.
- Daniel: this sounds like a legit use case to me. Then we could formalize that
- Roch: I think there is value in that and it would enable users and not just vendors. But how do we make that spec complaint? Same as I think that `TraceState` should allow users to modify it. The user is just another vendor in the end.
- Daniel: there are other plans for another header that will propagate the user baggage data, so I don’t see a user as a vendor. I will discuss this internally. I remember our architects wanted something similar.

- Roch: to get back to the user is not a vendor, the vendors are also users of the library
 - Dave: could we make a distinction that at a minimum a vendor runs their own exporter?
 - Daniel: I would see an OpenTelemetry system is itself a vendor
- Add CircleCI workflows for the different Node versions as well as browsers.
 - Brandon: probably
 - Dave: for browser testing a lot of people use SauceLabs
 - Roch: just to prove it can work in any browser
 - Dave: we do already have Chrome for Karma tests
 - Mayur: someone want to take ownership of this?
 - I can open this issue on GitHub and see if someone can help us
- Event emitter to stream all methods of a Span? Should this be a new class?
 - Dave: I'm nervous about the extra JS bytes being added or a Node.js specific EventEmitter
 - Daniel: if we are talking bytes, it's really minor. We are just adding one more property that is a callback. But it would really provide a lot of benefit.
 - Dave: are people comfortable with a simple callback and not a Node event emitter?
 - Roch: anything that allows people to hook into the browser. There are event emitters that hook into the browser. Maybe it could be done externally?
 - Dave: sounds like the value outweighs the cost, but I would prefer a simple callback
 - Roch: the callback is also arguably faster
 - Dave: so would the callback take a string and then the arguments to that method?
 - Roch: if you hook into startSpan, of course you want the span at some point. Maybe it's something that could come in, hook that runs before gets the arguments and the return value. The vendor should just call one function like "initializeVendor" and then the vendor works fully with the hooks and exporters, etc. to the tracer
 - Dave: maybe once we get a PR out there we can have more specific discussion about the hooks
- Looking for new Maintainer:
 - <https://github.com/open-telemetry/community/blob/master/community-members.md#java-script>
 - Responsibilities and privileges:
 - <https://github.com/open-telemetry/community/blob/master/community-membership.md>

2019-07-24

Attendees

- Roch Devost (Datadog)
- Mayur Kale
- Valentin
- Mark
- Dave
- Hamid (Elastic)

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- AsyncHooksScopeManager: [pull/103](#)
- GlobalTracer API [pull/118](#)
- Which tracers should be browser compatible?
 - Basic Tracer vs Node Tracer vs Web Tracer.
- Basic-Tracer initial implementation, CLA issue
 - Blocker for Real Span & Auto-Tracer
- NoopSpan vs NoRecordingSpan vs RecordingSpan
 - <https://github.com/open-telemetry/opentelemetry-js/pull/99#issuecomment-513325536>
- Metrics Chat: https://docs.google.com/document/d/1zOk27wOMTAm4iv-2MEoTtq3eJUy5KehtXi_UToRqgVk/edit

Notes

- AsyncHooksScopeManager: [pull/103](#)
 - Valentin: discuss the AsyncHookScopeManager. Should we move `bind`, etc. methods to be base?
 - Roch: this isn't a blocker, but was a general recommendation
 - Valentin: we can refactor in the future if needed
 - Valentin: when we bind a specific function, we need to change its length. We should use a switch statement to have more of a fast path if possible. Do we have a benchmark to show the impact of this?
 - Roch: this code is repeated in many libraries, this is not pretty, but it's much faster. I can provide a benchmark. We should eventually have actual benchmarks in the library.
 - Valentin: if it's < 3 arguments, it's not too complicated, but if we have to put in 10 or more arguments it starts to get messy.
 - Roch: eval is also an option, but would we be triggering lint issues. I've seen most people go to 10 arguments.
 - Valentin: can we come back to this?
 - Roch: yes, that would be OK. We have performance issues in other places that we can come back to.
 - Valentin: let's open an issue with the performance problems we have so we can address later
 - Mayur: is the PR still in draft mode?
 - Valentin: no it's a regular PR now

- Valentin: scope for event emitter?
- Roch: the scope for the event emitter should be in the context of the originator rather than the `.emit` method. What if we just don't support event emitters initially? The function is the easiest one by far.
- Valentin: I've added support for promises, but it could be easier to come back to event emitter in another PR. I will edit the PR to remove all promise and event emitter stuff and will open a different PR for that.
- GlobalTracer API [pull/118](#)
 - Mark: OpenTracing initializes the global tracer in the object rather than the `getTracer`. I'm leaning toward not keeping the delegate, but if anyone knows of an issue.
 - Mayur: in OpenCensus we created the singleton instance of the tracer and we expose that as part of the package, so there is no method to initialize the tracer.
 - Roch: we don't return a delegate, it's an internal thing. Init delegates to something else. If you don't init, you get a no-op tracer. It allows the user to disable the tracer and then they get the no-op instead.
 - Mark: then that allows the user to disable the tracer later on?
 - Roch: yes, it allows you to disable it later on. The advantage of the proxy is that you can disable it while the process is running. That can be used for things like the tracer detects that it's causing issues, e.g. a memory leak or something. Then the user can disable the tracer at run time. If you have to return the tracer, then you may be holding the tracer at run time.
 - Dave: do we have a stop method or disable method on the tracer. Would it work for them to call `.stop()` on the tracer at runtime?
 - Dave: I'm fine with the global tracer and some TypeScript type funkiness
 - Roch: should it be hidden from the user?
 - Dave: I like that, they just import one tracer and they don't have to think too hard about it. So maybe the `GlobalTracerDelegate` could just be the tracer.
 - Mark: yeah, that's true.
 - Dave: if we just add a comment then I'm fine with it.
 - Mayur: the default tracer is the no-op tracer. Do we want to consider making it move to the basic tracer?
 - Dave: I like it this way, but in the docs you could explain it. What if each package exports a tracer that is wrapped in a global tracer delegate?
 - Mayur: I like that approach
 - Dave: can you pass in any tracer in the constructor?
 - Mark: yes, you can pass in any tracer
 - Mayur: my other question is, in OpenCensus we had a singleton instance.
 - Dave: you would export a singleton instance of the tracer
 - Roch: where do you pass the options for the tracer?
 - Mayur: in OpenCensus you pass the options to `.start()`
 - Dave: one slight difference is that something like `.start()` is on the instance of the tracer not a global function

- Roch: I think it would be best if you just grab the global tracer and you call .start(and it starts tracing. I've seen a lot that when there are multiple things in the tracer that you have to call, people will get it wrong. The simpler the better. On the global, no need to worry about what's a delegate or not.
- Mark: how it works now this boils down to the default fallback is the no-op tracer, so that's why you need to initialize the tracer. There was some discussion to default it to the basic tracer.
- Roch: I don't think it's necessarily different. It's a possible race condition if you initialize it by default. Say there is some behavior in the tracer that does something that requires config. I think having an explicit method for any configurable initialization code.
- Mark: if the proxy/delegate method, do you see any issues where someone swaps out the tracer and that causes race conditions or known effects?
- Roch: in JS shouldn't be an issue
- Dave: what do people think about the idea of making this PR become more of a utility wrapper?
- Roch: I'm not sure I fully understand, but makes sense
- Dave: we use this wrapper class
- Mark: makes sense and sounds good
- Mayur: can you update your PR with this kind of implementation?
- Mark: I'll follow up with Dave for minor details
- Which tracers should be browser compatible? Basic Tracer vs Node Tracer vs Web Tracer.
 - Mayur: we have Basic tracer everyone is agreed on the naming convention. But for the automatic tracer, what should be the name?
 - Roch: I suggest that the tracers should just compose things rather than doing things on their own. For example, the Node tracer would just compose anything that is Node-specific and anything that is common. It would be limited to composition.
 - Dave: part of my feeling with it is to avoid suffixes
 - Roch: I would expect the Node tracer would import the scope manager and all the different plugins, the rest should be common functionality. There should be something like auto-tracer that would be imported by the Node tracer. Three things that should be platform specific:
 - Utilities
 - Scope
 - Plugins
 - Dave: part of me wants the plugin architecture to be Node-specific
 - Roch: right, any plugin loader should be platform specific
 - Mayur: I agree with Roch. But in terms of package name and all should we go with -node at the end?
 - Roch: if all the actual functionality can be in the basic tracer.
 - Dave: I like -node on the package, which will encourage making
 - Roch: types core and basic tracer is shared between Node and the browser

- Dave: I will add Karma tests to the basic tracer
- CLA discussion
 - Roch: what are we gaining with the CLA?
 - Dave: I think it's for legal coverage
 - Roch: small PR's would not happen with such a difficult CLA process
 - Mayur: I'll talk to Peter offline and see if I can get help on this
- NoopSpan vs NoRecordingSpan vs RecordingSpan:
 - Mayur: there are different types of spans. Suggested approach:
 - NoopSpan: singleton, does nothing
 - NoRecordingSpan: just does context propagation
 - RecordingSpan: does context propagation and recording
 - Roch: you may want to do some kind of deferred sampling, e.g. you are storing some kind of object hash. The implementation might be completely different in a RecordingSpan.
 - Dave: could you have all three?
 - Roch: it's not a super valid use case, if you remove the context propagation
 - Dave: could you have a situation where you just use a singleton for performance
 - Roch: I would go with NoRecording span, and have the three different classes
 - Mayur: option would be NoopSpan or Span (can record or not), but then you need a bunch of if statements in the methods
 - Roch: personally I prefer to avoid conditions wherever possible, and also since the implementations could be completely different. Then the entire implementation would be different, and at that point it should be two different classes.

2019-07-17

Attendees

- Mayur
- Dave
- Roch
- Brandon
- Peter
- Hamidreza
- Mark Wolff

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- ScopeManager related function's behaviour in NoopTracer and BasicTracer
 - Tracer behaviour without ScopeManager: throw, log, swallow
 - Roch: one option could be to move out these APIs from the Tracer
 - Roch: in DataDog / OpenTracing these API's are accessible through scope manager
 - Peter: I like the idea of not having the API on tracer, but specs requires it

- Dave: the idea is that package like instrumentation using it don't have to worry about it, my preference is to reduce the usages of null
- Roch: my worry it's the same confusion, you call withSpan but you don't get the span what you would expect
- Brandon: I don't think we should return null as it can break process
- Do we know how other dynamic languages handle this?
- Roch: what would be the option to make it required so you have to pass it and when you pass Noop it's clear that you picked it
- Dave: maybe noop could just use a simple variable (global)
- Roch: that would be a SyncScopeManager as Noop shouldn't allocate anything
- Dave: so the noop wouldn't call the scope manager with the span
- Brandon: looks like python returns null when the scope is not available
- Dave: our noop implementation still calls the fn
- Roch: I'd expect it calls but doesn't pass span
- Dave: I'd be open to both null and not null
- Peter: withSpan would still be a blackbox and hide that it doesn't do anything, should we warn log?
- Roch: I think making it required pass a scope manager would eliminate the require for logging
- Dave: one of our example could be passing a WarningScopeManager
- Peter: I opened an issue about strict mode, could it be a layer top of NoopScope manager?
- Dave: I'm thinking about the same for browser
- Roch: strict should be the default, but for browser it can be challenging to access those logs
- Dave: in production people could tune the settings
- Peter: consensus on making scope manager required, let's decide logging and strict mode behaviour later
- Roch/Peter: for NoopTracer default can be NoopScopeManager
- Default ScopeManager in NoopTracer: no scope manager? -> NoopScopeManager
- ~~Default ScopeManager in BasicTracer (no scope manager, noop scope manager etc-)~~
- https://github.com/open-telemetry/opentelemetry-js/pull/99#discussion_r303208542 -> required to pass ScopeManager
- Metrics: Support for longs (bigint)
 - <https://github.com/open-telemetry/opentelemetry-js/issues/104>
 - Relevant PR: <https://github.com/open-telemetry/opentelemetry-js/pull/105>
 - Dave: I'm a little bit concerned about using bigint as it would may require polyfill for the browser
 - R: I think it boils down that do we need support for more than 52 bits
 - Brandon: I'm leaning towards not supporting bigint, API would be simpler and we don't need to polyfill

- Mayur: in OC we went with Number and there were no complaints
- Mayur: we need to specify behaviour around decimals
- Roch: I'd just use double always (that's me), but if spec defines long, I'd just strip it and drop the decimals
- Mayur: that could create a different type of histogram
- Mayur: need to define type
- Roch: for long it's fine as there wouldn't be decimal anyway
- Mayur: can we show a warning or something? I'm worried about data loss.
- Mayur: so you say logging is not necessary as user picked long anyway
- Roch: yes
- Brandon: default is double
- Peter: I think logging not logging is the same topic as for tracer scope manager and any other tradeoffs we make here
- Mayur: RFCs around metrics API, will keep you updated about changes
- Brandon: I hope there will be some changes they are fairly complicated
- Roch: I agree, reviewing PR was also time consuming
- Brandon: one concern is that some API looks more compatible with Prometheus like API than statsd
- Roch: my feeling is we would use "measure" for everything
- Conclusion: not supporting bigint and use Number, discuss logging later for wrong type scenario
- Where should Real Span rest?
- Sampled flag in case of Real Span and NoopSpan?
 - Mayur: what should be the value?
 - Mayur: For real span it's "always sampled", right
 - Mayur: for Noop should it be "not sampled"?
 - Roch: I think it's reasonable
 - Mayur: maybe other process decided not to sample a sampled spans
 - Roch: I think the sampling decision should be taken once, except some edge cases
 - Mayur: there is an isRecordingEvents option, in my understanding if it's false we need to create NoopSpan
 - Brandon: used case is when trace is sampled but specific span is not, it's a problem with singleton NoopSpan as propagation should be always correct
 - Roch: but why would you drop one span in a recorded trace?
 - Brandon: a use case could be a "verbose" spans, when you record additional more verbose spans when you enable some debugging, but normally you wouldn't record those spans
 - Roch: but in that case you don't need to drop as you never created those spans
 - Roch: it's a requirement for us to reduce allocation in some way. Doesn't have to be singleton but it has to be possible
 - Mayur: in OpenCensus we had a concept of z-pages, which shows spans even when they are not reported
 - Roch: we could have a third type of span for that use-case

- Conclusion: it's not safe to say that real span is always sampled as we could have this third type of span
 - Mayur: I'll open an issue about fulfilling the use-case of no reported sampled spans
- What's in core? Where should real span go?
 - Core is basic concepts with Noop implementations
 - Core is empty shelf
 - Other languages doing the same
 - RealSpan: Probably it should live in basic-tracer
- Future topic: hooks
 - I think we should have hooks everywhere so vendors can hook to specific places and run their own code

2019-07-10

Attendees

- Mayur Kale
- Peter Marton
- Roch Devost (Datadog)
- Mark
- Brandon Gonzalez (lightstep)
- Hamidreza Jahtalabziabari (Elastic)

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Review file and folder structure casing and relationship
 - <https://github.com/open-telemetry/opentelemetry-js/issues/38>
 - Mayur: looks like we are not following it completely, we should agree on this
 - Dave: I started to work on a rule to enforce it, I think it would be easier to follow if we have rules. Will work on it in the next few days
 - M: we need a PR to fix existing files
 - D: maybe it's need to be in the same to keep CI green
 - M: will it have the header fix as well?
 - D: yes it will use gts and check if header is there and add if not
 - R: they provided a format we can use
 - D: maybe ask CNCF
 - M: I'll do that
- Separate package for **basic-tracer** and **automatic-tracer**?
 - Mayur: I think we should have a separate library as these are flavors / SDKs
 - Roch: I'd agree on separation
 - R: we have metrics as well, not sure how it would work together
 - D: we could have MetricsTracer and combine them
 - P: in OC BasicTracer was born to separate out CLS, it was baked in Tracer
 - R: Maybe automatic tracer is what need to be renamed

- D: it could be a constructor option to pass NoopScopeManager / ScopeManager / ZoneScopeManager
- P: getCurrentSpan and withSpan interfaces are not applicable in node
- D: it could do nothing like store it in a global variable, wouldn't be really useful in node
- R: Node / JS automatic and basic looks different enough that they could be extensions
- P: do you have inheritance in your mind?
- R: no specific implementation in my mind
- D: it doesn't have to be inheritance specifically but it could be just multiple TS interfaces
- BasicTracer and ScopeTracer, interfaces NodeTracer implements ScopeTracer, BasicTracer implements BasicTracer
- R: I think it's okay if AutoTracer is spec compliant but BasicTracer is not necessary have to, these can be more lightweight
- D: what if you call it SpanCreator so it's clear what is it for
- R: maybe not export BasicTracer for direct consumption, or combine it from multiple small pieces
- P: basically three solutions:
 - 1. Make a pluggable scope manager that the constructor takes, e.g. NopScope that can return NopSpans
 - 2. Have a non-spec compliant BasicTracer (or SpanCreator)
 - 3. Make the BasicTracer interface be hidden
- R and M: like option 1
- P: you can make the NopScopeManager log warnings to let the user know this is not expected usage
- Tracer: SpanOptions.parent?: Span | SpanContext, why a union type?
 - P: other languages don't seem to allow this
 - R: this was nice in OpenTracing, you don't need to worry about extracting SpanContext. It makes it easier for the user because they don't need to extract the span context, it saves the user from making those calls everywhere. The public method just ensures the private method gets the right format.
 - M: does that mean everywhere when we pass those types that we need to do a condition check?
 - R: sometimes you do need to do more validation, but in this case you will need to check. We need to do it in multiple places, e.g. library does it 5-10 times vs. users have to do it thousands of times
- Review: [HttpTraceContext](#) Format vs [Binary](#) Format
 - inject/extract vs toByte/fromBytes?
 - M: we have different interfaces which are not compatible, this is how it is in the spec
 - M: there will be a spec meeting this week, where we will discuss it
 - D: Roch, you talked about array buffer, is it native to JS Type or is a polyfill? In Node Buffer, in JS you have Typed arrays. Using an

ArrayBuffer can be resolved by the environment and it would be compatible between Browser and Node.

- D: we had a conversation in OC with Mayur earlier and I remember that the buffer type was expected with gRPC which added an extra friction
- D: in Browser even if it's binary, you need to encode it in the header
- R: for me a generic binary format shouldn't be biased by specific protocol like gRPC as binaries are very generic, although I'd be okay to add something gRPC specific
- D: even if we wouldn't use ArrayBuffer in the Browser as it's compatible I'd be okay with it
- M: if you are not using it, isn't it an overhead for bundling
- D: it's a separate project, right?
- M: no, currently it's not
- D: tree shaking would remove unused code, most people do it. So it doesn't matter if it's part of the same npm package.
- M: it's kind of provided by the spec as default
- D: make the default a function in the platform, so in Browser it can return binary and in Browser it can return text. And then Node can be spec compliant and web a special case. A function: give me the formatter
- R: I'm not sure, Node would need both per use-cases (HTTP, gRPC etc.)
- R: maybe browser also would need binary ~gRPC
- D: are you saying we shouldn't have defaults per environments?
- R: I don't understand the goal of the default
- D: I think it's coming from the spec
- D: in my understanding default pushes people towards best practices
- R: I'm not sure why binary would be preferred for Node, 99% will be HTTP headers
- R: I'd just use text as a default for both Browser and Node
- D: what was the notion about default?
- M: not sure, need to check with the team
- P: don't we already have two methods on Tracer getBinaryFormat and getHttpTextFormat, why do we need default?
- R: the idea was just to call inject and vendor can create specific things. Changing vendor would have no impact on API.
- D: would it be possible to have both inject method and the getBinaryFormat, getHttpTextFormat
- R: I don't mind exposing them, the issue is there is no common interface between them. You have to explicitly use these two.
- M: in Java it's really hard to combine both, so spec is maybe driven by that there
- R: it's more about safety, but this concern is more language specific
- R: incorrect format wouldn't break just would return null
-

- HttpTraceContext: How was it decided to make carrier an object vs. the previous HeaderSetter/HeaderGetter approach?
 - getBinaryFormat, getTextFormat
 - getHeaderFormat(format: string): unknown
 - inject(format: string, target: {[key: string]: unknown}) {
 - getHeaderFormat(format).inject(target);
 - }
 - R: in OpenTracing, it finds the format, and then calls inject on the format
 - R: convenience and extensible to vendors

Running out of time...

- [BaseScopeManager](#) interface
 - What's the motivation for a separate package for the base scope?
 - Review and merge the interface.
- Specs improvement:
 - data format breakout session (PDT+CEST)
 - <https://doodle.com/poll/6pxifbcc72t8hdu> Please vote by EOD Thursday.
 - Metrics meeting [notes](#)
- Up-for-grabs:
 - <https://github.com/open-telemetry/opentelemetry-js/issues?q=is%3Aopen+is%3Aissue+label%3Aup-for-grabs>

2019-07-03

Attendees

- Daniel Khan (Dynatrace)
- Roch Devost (Datadog)
- Mayur Kale (Google)
- Peter Marton (Netflix)
- Justin Smith (Omnition)
- Mark Wolff (Microsoft)
- Dave Raffensperger
- Cristian Gonzalez

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Insight into TraceState - @danielkhan
 - Daniel: Trace context contains two headers (trace parent and trace state)
 - traceparent: version number and trace id
 - tracestate: list of vendor keys (needed as traceparent cannot have this data)
 - Tracestate's encoding and value is up to the vendor
 - Use-case: mixed vendor Dynatrace -> Azure -> Dynatrace

- Tracestate shouldn't be discarded so the data will traverse through vendors/services
- Tracestate example: `az=azureOpaquePayload, dt=dynatraceOpaquePayload`
- At the beginning of trace state you will always see last vendor
- Sampling is informal as you don't trust upstream system outside of your boundaries
- We need to figure out how to propagate sampling in OpenTelemetry
- Zipkin has some debugging flags to force record traces
- Tracestate-limit: how long it will be? It can be a concern for small messages. There is a limit now to reduce the size. Vendor is MAY truncate to meet limit.
- We need to decide if in OT we passthrough, truncate or how to handle
- Mayur: I checked other languages and they throw exception when limit reached
- Daniel: I don't think tracer should ever throw an exception, especially limit is MAY not MUST
- Roch: for us it's a requirement that a tracer should never throw, but it can log
- Daniel: same for Dynatrace
- Daniel: we need to clarify for the spec
- Dave: what about correlation context, user baggage
- Daniel: it should become a baggage container
- R: does it mean trace state should be only touched by vendors
- Daniel: yes
- M: last time we were speaking about a data structure to store arbitrary data, can this be used?
- R: I expect vendor instrumentation will have many hooks and metadata
- R: we would need something where vendor can attach a metadata and it would be available everywhere from the tracer
- R: the problem with current span context is that it's not guaranteed today as it can be converted and immutable
- R: immutable is not enforced so we can workaround today
- Daniel: we have similar requirements, Isabel from Lightstep talked about it
- Daniel: we need to clarify from the spec side, like a dedicated property
- M: do you think it's an SDK level thing?
- M: vendors can implement their own SDK, so they can extend it
- R: I think the problem is that you need some kind of API to expose this data
- M: I heard from Bogdan that we want to keep it simple (only these four properties) but we can add something specific
- Dave: Java has thread local storage but JS doesn't so maybe we need different API
- M: do we have a decision?
- R: we could open an issue just mention what we need as Datadog and how it is not met by current API. To open the discussion.
- M: I think specification repo is a good place to open the issue

- Dave: is basically LocalContext is this arbitrary context what we are talking about?
- SpanContext vs LocalContext vs SpanData
 - Daniel: one discussion: from spec requirement the SpanData has to be immutable
 - Daniel: other discussion ...
 - Daniel: should we discuss it here or should we pop up conversation to spec?
 - Daniel and Mayur try to join spec meetings to carry these topics over
 - R: we need to show up for spec conversations as Java/.net is very different than Node
 - Daniel: PR workflow should help with this
 - Dave: one question I had about span end time
 - Preference would be to use plain objects and be consistent
 - Immutable vs readonly
 - Dave: I think readonly could be used to show it
 - Passing around data in JS is easy but different than in other languages
- Review and discuss: ContextManager interface [pull/66](#)
 - Dave: the challenge I had with this was that there was a non-specific wrap function for the NodeJS event emitter
 - Roch: simpler than what it is currently proposed.
 - What is proposed, it's similar to CLS. What I'm thinking is something in-between. CLS works by prototypical inheritance from all contexts, and you can override things on the sub-contexts.
 - Idea is to do one layer under. We should simply store a single object and that's it. When you activate a context, then it will be stored as-is. You don't need to get/set for different properties.
 - Just store an object and retrieve an object and you're done
 - Then we will need some type of concept of namespacing so that it could be compatible with CLS.
 - Daniel: everyone including NewRelic got rid of CLS. There are ways to just wrap well-known libraries. Then you wrap setTimeout, setInterval, and anything async/await you use async hooks. CLS has a bad reputation.
 - Roch: I didn't mean CLS itself, but rather an API that has a similar surface to CLS. Even if the user doesn't use OpenTelemetry for tracing, they could still use the context. We could later add a CLS style API on it.
 - Dave: you talked about having just one "wrap" function. Could you talk about that?
 - Roch: we would have "wrapFunction", "wrapEmitter" and "wrapPromise", maybe we add a few other things later on. Say as a vendor, I implemented some kind of object that works similar to an event emitter that has the same API. Now I can't add an additional method on this. If you add a method, then you are no longer OpenTelemetry compliant.
 - A value that takes a function, or event emitter, or function or unknown
 - Performance of this: it's low impact to just check for simple things

- Ability to differentiate different things: what if it's a function and a promise and an event emitter. It would just wrap all of them.
 - In general this approach makes for a simpler API to use. As a user, you just bind it and it works, and more extensible.
- Dave: I like this single function approach. We could use `unknown` for the generic context propagation, and then the more specific implementations could have more specific types
- Review and discuss: <https://github.com/open-telemetry/opentelemetry-js/milestone/1>
 - Daniel: I'm working on the Tracer API. What is "WithSpan"? Is that the same as "wrap"? When I look at the Go implementation, it looks like a wrapping function.
 - Mayur: it's like a scope span. You set that span to the current context, and then you restore back to the original state.
 - Daniel: using SpanBuilder, yes or no
 - Mayur: in OpenCensus we had SpanOptions that was a POJO

2019-06-26

Attendees

- Hamidreza Jahtalabziabari (Elastic)
- Mayur Kale (Google)
- Roch Devost (Datadog)
- Brandon Gonzalez (LightStep)
- Peter Marton
- Dave Raffensperger
- Cristian Gonzalez

Agenda

Tentative goals for the next meeting, feel free to add your own!:

- Join mailing lists: <https://github.com/open-telemetry/community#mailing-lists>
- Is there a "lock down date" for specs? Is it possible to deviate from spec like extending it?
 - no changes after mid-August to 1.0 of spec.
 - try prototyping in different languages to see if spec will work for, e.g., Node
- Pick up for grabs issue: <https://github.com/open-telemetry/opentelemetry-js/issues?q=is%3Aopen+is%3Aissue+label%3Aup-for-grabs>
 - Metrics API
 - Context-* package and implementation
 - M: Feel free to open PRs
- What are the priorities for this Week?
 - Add Tracer API [issues/28](https://github.com/open-telemetry/opentelemetry-js/issues/28)
 - M: Daniel is working on Tracer API, no PR yet
 - M: blocked on the first draft
 - Review/Discuss DistributedContext PR : [pull/36](https://github.com/open-telemetry/opentelemetry-js/pull/36)

- M: can I merge it and go ahead, what about comments?
- R: I don't see the point of that TTL
- R: I don't see the use-case to control how many hops for the propagation
- R: passing the propagation is the other service's decision
- M: I see you opened an issue in spec, but no comment yet
- M: TTL is copied from OC
- M: I think we should move forward with what we have and modify later when your proposal is accepted
- R: it would be beneficial if the APM provides could access this API and use it as the general propagation API through OpenTelemetry
- R: would need an additional abstraction, this is my personal goal
- In the browser bundle size matters, simple object would be preferred
- Review/Discuss Propagator interface : [pull/55](#)
 - M: this is currently copied from OpenTracing
 - R: there is a binary format and an HTTP text format
 - R: I think there should be a distinction
 - R: adding some gRPC / amqp headers it could be applied, even if they are not http headers but same format
 - M: I can make the changes just please remind me (to Roch)
 - B: is binary and HTTP the ones are called out in the spec?
 - M: yes but B3 and other formats are coming
 - Current API looks like chaining, could we do something more functional
 - R: it would mean you move the logic to the caller
 - R: would be more challenging with manual instrumentation, because you don't need to know what happens during injection
 - It makes sense
 - I'm wondering if it could be done as a utility function, one for HTTP headers for example and then you could use the utilities to compose these functions
 - It would make sense to make these changes later and not block on this now
 - M: I was thinking about individual packages for this formats, or should they live in the core package?
 - R: no strong opinion, separation makes sense
 - M: in OpenTelemetry we want to give an option to the user choose between formats, lest for HTTP, gRPC only works with binary formats
 - M: I'm not sure about the web
 - D: the web should support the same format but the injection will be different Node HTTP Request -> XHR etc.
- Review/Discuss Resource API: [pull/59](#)
 - M: I updated the PR based on latest commits
 - M: Roch please take a look again
 - M: will be compatible with AWS, K8S...
 - R: I'll take another look, fine in general. Only initial concern was labels...

- Review/Discuss BaseSpan PR : [pull/45](#)
 - M: Main question is to store the whole SpanContext
 - R: span context here is not the same than in OpenTracing, there is no vendor specific things here.
 - R: The W3C span context is meant for propagation outside of context, so maybe there should be a separation here and exposed as immutable
 - R: The next point is sampler, it accepts many values that should be in the span context. We have many layers of sampling today. Avoid duplication like you don't need to pass spanId and traceId as they are already in the span context. Span context becomes this thing you need many places to make sampling decisions.
 - H: user input was discussed in W3C yesterday, Trace State and can be already used for vendor data
 - H: we are trying to make definition between Trace State and Correlation Context. Correlation Context can have user defined data.
 - H: Correlation Context is a header, early stage, meant to pass user provided data to the next Tracer
 - H: it is an HTTP header, designed to propagate over the wire
 - R: some challenges around reserved characters, it's also very restrictive, but I guess we can't change it as it's W3C not OpenTelemetry
 - D: what's the diff between arbitrary context and attributes
 - B: ??attribute is span only, context is over the wire??
 - D: would it make sense to have a new "in-process-metadata" like bucket
 - B: people will be very opinionated about it, let's discuss async
 - R: how about name "local context"?
 - R: span context should contain local context, propagation context etc. in this way we can always reference to it
 - D: wouldn't it be easier literally pass the whole span to the sample?
 - M: it wouldn't be compatible with lightweight NoOpSpan
 - R: it's a lightweight span but it's still allocated. For some of our users it's not acceptable so we are reusing the same NoOpSpan (~zero allocation, singleton)
 - R: any consideration? we would need that at DataDog?
 - D: great idea
 - M: what would be the trace id?
 - R: we use zero but it shouldn't really matter because there is no relation with other spans as they are no ops
 - M: basically we provide some invalid ids, right?
 - R: something like that
 - M: it sounds like it makes sense to provide two context for sample: local and over the wire
 - R: in general yes, but if we can combine them as a user you don't need to know the difference
 - M: concern is for that we need to create span first and pass values

- M: for now I will change sample to use span context now to move forward
 - M: I'll make a change to make span context optional
 - H: do you consider sampling flags coming over the wire?
 - M: Yes, because it will be part of the span context
 - D: is this each sampler's responsibility to respect those fields? Is this defined in the spec?
 - B: it can be challenging to make a decision with conflicting sampling decision in process vs. over the wire
 - M: in OpenCensus we take over the wire first, than we fallback to user configured... (3-4 layers)
 - H: [spec](#) says these flags should be considered as recommendations not strict rules
 - D: OpenCensus Go has an API to say this is a public endpoint and uses linking
 - R: cross-vendor sampling decisions is an interesting question
 - M: plan is to get API ready this month, than we can start to work on SDKs
- Review/Discuss Samplers PR : [pull/48](#)
 - See comments in previous section.
- CUT:
 - Configure CircleCI and typedoc (docs script)
 - Add no-op or default or base tracer API.

2019-06-19

Attendees

- Mayur Kale
- Roch Devost
- Peter Marton
- Mike Marcacci

Agenda

Follow up on action items from last time?

- Renamed to `opentelemetry-js`
 - M: It's done
- Community Meeting [Recording](#)
 - M: Slides on OpenCensus performance improvement
- Added CircleCI configuration, Thanks @bg451.
 - M: PR to add CircleCI
 - M: There is one issue on npm, still working on it
 - R: I thought we decided on Yarn
 - M: yes, should we close it?
 - R: let's continue with Yarn
- MIT vs Apache 2.0
 - R: I'm looking for how we could make it easier to use, maybe linter

- R: or just reduce the size of the header to like one line, it's noise
- P: who decided Apache?
- R: CNCF projects ?must? be Apache
- R: Node projects are usually MIT
- R: I'll reach out to CNCF
- M: do you want to follow up with CNCF and carry this over?
- R: I'll open one-two PRs (header size and TSLint)
- Please file an issue [here](#) so that I can add you as a reviewer.
 - M: It is need to being able to assign issue, please if if you haven't done yet
- Timestamp format
 - OTel spces:
 - <https://github.com/open-telemetry/opentelemetry-specification/pull/111/files>
 - M: we need to finalize it, it's going for weeks now
 - R: discuss was about format
 - R: I think it should be milliseconds because that's what JavaScript supports
 - R: we could use duration instead of end time, that could increase precision
 - R: there are scenarios where you want to use both start and end time, for example OpenTracing support
 - R: Apollo tracing has a proxy which does tracing. Your timing is not the correct one it's the one reported by the service, you need to pass an end time
 - M: I opened an issue to pass timestamp at the end but it was rejected
 - MI: I was working recently lot with GraphQL, these used through layers like Apollo. It's hard to support relemetry natively.
 - MI: I believe if projects like OT could provide best practices, it could improve library ecosystem
 - R: In my experience libraries doesn't really follow these concepts and you end up with monkey patching anyway
 - R: very few libraries emit telemetry events
 - R: challenges around using existing telemetry events in Apollo for example, as APM provider we need to monkey patching
 - P: Node.js Diagnostics WG would like to make apis required for distributed tracing first class citizen, but not much progress yet
 - P: Diagnostics WG also discussed possibility to have an instrumentation API and work with library contributors together, no progress either yet
 - MI: In terms of start/end/duration it's important that event can come in different order, we should be sure OT can handle these situations and doesn't depend on correct order.
 - R: I think it's important that events are pretty much immediately, which also means these are in ~order, naturally in order
 - P: it sounds like duration can be more precise but end time can work with external systems, any concern around clock sync between processes when end time comes externally?
 - R: we don't have atomic clocks usually, but we recommend customers to sync clocks and it works well

- M: Roch can you add outcome to timestamp issue and close it for now?
- R: sure

Up for grabs:

- Add Tracer API [issues/28](#) - @danielkhan
- Sampling API, Resources API, DistributedContext API.
- No-op implementation of Span API.
- No-op implementation of Tracer API.
- P: in OpenCensus no record spans have empty fn bodies and doesn't hold the same references on this as the normal span, are we planning to do the same here?
- M: basically yes
- R: we plan to call them DefaultSpan, is that okay?
- R: I'd prefer NoOpsSpan as ts describes better what it does
- R: Peter, are you asking it doesn't return the same data tec.
- P: yes...
- R: yes, probably it should but it will be harder to implement
- R: anything that works with normal span, should work as normal span
- R: should have TraceID to NoOps spans?
- R: No record span instance were reused earlier (OC?) and it can be challenging
- M: I'll work on no ops span
- M: I got one approval on my Span Interface, is it safe to merge?
- R: I think at this state of the project it's fine

Tentative goals for the next meeting, feel free to add your own!:

- Review file and folder structure casing and relationship
 - R: files usually in TS have the same name as classes and util functions has some different naming structure.
 - R: have descriptive file names in general
- Should we start work on below packages?
 - opentelemetry/context-base
 - opentelemetry/context-async-hooks
 - opentelemetry/context-zones (if possible).
 - M: is this the right time to start to work on propagation packages?
 - R: I cannot commit yet, but I'd like to work on these
 - R: but don't wait for me if anyone interested
 - R: are these generic?
 - Avoiding double patch would be beneficial with using same package for OT and non OT
 - R: It would mean we cannot assume it is a span
 - R: Maybe one point remove it from OT and live as a separate package
 - M: we had this discussion in OpenCensus at Google as other teams have similar challenges, it didn't happen

How TTL should be used with distributed resources?

- TTL can describe how far specific tags in context should be propagated (next downstream, next two downstream etc.)
- Link OpenCensus description
- R: we have span context shared only locally
- R: we have distributed context which will jump only once
- R: Trace context which will be propagated the whole trace
- R: what is the use-case for multiple limited hops?
- R: I can add more context to the issue and continue conversations
- M: I think it would be nice to define this on the spec level

Spec extension

- Is it possible to deviate from spec like extending it?
- Spec is Java centric today, but we are working on making it more heterogen amongst of languages

2019-06-12

Attendees: Mayur Kale (Google), @tedsuo (Lightstep), Brandon Gonzalez (lightstep), Daniel Khan (Dynatrace), Roch Devost (Datadog), Peter Marton (Netflix)

Tentative goals for the next meeting, feel free to add your own!:

- Follow up on action items from last time?
 - Update on TravisCI vs CircleCI [issues/8](#)
 - Mayur: I checked with the team and looks like we can use CircleCI
 - Can anyone take it?
 - Brandon: I can
 - Node versions [issues/7](#)
 - M: Brandon shared some Node usage and everyone looks like agree on 8
 - Let's start with Node 8 and switch back to 6 if needed
 - Initial skeleton - merged [pull/2](#)
 - M: I added the basic structure
 - We already separated out the types from the code
- Use [Gitter](#) to post news/questions
- Rename to `opentelemetry-js` [issues/20](#)
 - M: I talked with Bogdan they are okay with our decision and okay to rename it
 - Is the plan to use it for browser?
 - P: why do we need to include the JS suffix?
 - Daniel: I think we should include for the repo but not for the package
 - P: is this the decision?
 - Agreement
 - Roch: it can happen that we end up with a separate package for JS and Node
 - Hopefully, one package will be the case
 - D: there are lot's of differences between Node and Browser platforms, so I'm proposing that under OpenTelemetry we have a dedicated package for Node and JS

- R: I think it would be the worst case scenario
- There are few cases that will be different between Node and BRowser like CLS
- API should be the same
- In my experience, the platform is a small part of the code base
- It's especially true because bundlers like Webpack can address platform differences
- D: Instrumentation libraries can be very different between environments, even between general Node and Lambda. I'm concerned about overloading browser
- M: I think our plugin driven package structure covers these use-cases
- M: there will be a lightweight base Tracer package and the user can decide what to use
- R: this challenge is not only browser vs. node, but environments can also be very different just by language features and shims
- If you want smaller package and optimize you have the option to do that but single package that includes everything has a value and will be fine for most users
- R: you can use peer dependencies
- P: Yarn and npm doesn't enforce peer dependencies
- R: that's the concept, if you need it to install, if you don't don't
- R: there can be warnings about was loaded and what was not with a link why
- P: I like it
- M: just to summarize, we plan to have three main packages:
 - Trace all in: comes with all instrumentation
 - Base tracer: no plugins
 - Browser tracer
- Typescript Guideline [issues/22](#)
 - Roch: I started to create guideline but I realized it's more about not use the ?compilation? feature
 - We can close the issue about the guideline and just review TS Config
 - M: all the packages inherits the options from the base package
 - R: The base package is fine just it should have some rules
 - just need to add the specific target
 - target specifies config
 - we don't need to define as the target comes with it
- Lerna usage
 - M: I feel we already agreed on it but just wanted to call out
 - OpenCensus uses it and worked well
 - R: I think the question is the additional dependency
 - R: works until a point but it's getting harder to manage later
 - R: I'm a huge plus one for Lerna
 - T: it doesn't have an overhead on end user
 - All agree to use Lerna
- Milestones <https://github.com/open-telemetry/opentelemetry-node/milestone/1>
 - Start with Tracer API, Span API, SpanContext etc.

- Maybe compare with [Java milestone](#)
- Spec milestones: <https://github.com/open-telemetry/community/pull/106>
- M: we have some specs read for span API
- It makes sense to start working on these and work on a spec in the meantime
- R: The API looks little bit Java driven, do we need to follow it 100%?
- Ted: The idea is the spec that it stands as an initial spec
- The initial spec was created by merging OpenCensus and OpenTracing and resolve conflicts
- This is an initial draft, we will iterate on it.
- The goal is to have a neutral enough spec that everyone is happy to follow that
- We need members from other languages to shape the spec
- We want to say something in a spec about context propagation
- It's important that people show up in spec formation otherwise it will be Java heavy
- M: please contribute to the spec
- T: I'm planning to create a blog post inviting people to contribute
- Creating milestones with tags to help them navigate in the project
- M: should we add tags to APIs
- T: labels are the most important things, milestones are also helpful
- M: should the milestones have a deadline?
- T: I think it would be great to help to hit goals, have visibility and show velocity
- M: what's the deadline?
- T: the goal is that we will hit v1 in September
- We will see if we are stable enough
- Code freezes around November
- And start to migrate people from OpenCensus to OpenTelemetry
- M: the right place to start collaborating on the Tracer and Span would be the types packages
- How to contribute?
 - R: currently things are only available for members
 - R: how would it work? Apply for members?
 - T: I think it should be open to anyone to start to contribute and open PRs
 - T: we should have well-defined processes
 - R: it's possible that multiple people will work on the same thing as we are bootstrapping
 - T: hopefully anyone can be assigned to issues
 - M: nope, you need to be a member for that
 - P: can we create a team like open telemetry-incubator which is people not a member yet but opens up this feature
 - T: I think we can add people here
 - Backlog management proposal (WIP):
<https://github.com/open-telemetry/community/pull/52>
 - T: hopefully we can clean up the membership process in the next couple of days

- T: I expect the project will grow quickly as it's a merge so it would be beneficial to figure it out sooner than later
- Timestamp format
 - D: I've an open PR [pull/111](#)
 - Nanoseconds don't make much sense.....
 - T: I can imagine use-cases where it makes sense like DTrace and system level tracing
 - D: it's a huge topic, let's prepare but it's not significant
 - T: most system cannot produce nanoseconds
 - R: I'd just start with a Number
 - D: the Java implementation uses tuple syntax
 - D: you can always convert from Tuple to Number
 - R: but not on the other way
 - R: it would make sense to use tuple internally as it's more precise and we can always convert it
 - Agreement in the audience
- [Community meeting today](#), 4pm Pacific
 - T: the audience is cross SIG
 - Focus is issues around the community repo
 - It will be too late for European folks
 - OpenTelemetry tries to be more inclusive with Asian timezones

2019-06-05

Attendees: @rochdev(Datadog), @mayurkale(Google), @vmarchaud(OC Contributor), @hekike(Netflix), @bg451(Lightstep)

- Kick-off the meeting: Intro
- Go over issues
- Discuss initial milestone
 - ambitious goal of **September 2019** as the target date for parity in C#, Go, Java, NodeJS, and Python.
- Package management conversation [issues/14](#)
 - Yarn or npm
 - Decided to use Yarn because of performance reasons
 - How about having lockfiles in the repo?
 - Lockfiles are not used by consumer code, they will have their own lockfile
 - Should we have lock files for contributors in the repo?
 - We decided that we will go without lockfiles
 - Exclude from repository
 - @mayurkale needs to confirm that automatic dependency update tooling works without lockfiles
 - Floating semver
 - Decided to go with carets by default, but it's not enforced
- Travis or CircleCI? [issues/8](#)

- Roch: the orchestration of build is easier with CircleCI, TravisCI is mainly scripts
- <https://github.com/cncf/toc/blob/5942d07249dc18aaa555390e3d733964b106da41/proposals/opentelemetry.adoc> -> TravisCI (Infrastructure requirements section)
- Everyone has good experience CircleCI
- @mayurkale will sync with his team
- Node Versions [issues/7](#)
 - Roch: no strong argument to support Node 4 and 6, but it's possible that other APM vendors may need to support older versions of Node
 - Mayur: OpenCensus supports > 8, not much Node usage below 8
 - Brandon (lightstep): supports > 8, will share stats about usage
 - Everyone: Will start with min. 8 version
 - May be open an issue to highlight the version
 - Peter: transpilation is maybe related to Node versions, shims can be less performant than built-ins
- TypeScript or JavaScript [issues/15](#)
 - Roch: I'd advocate to use JS and only use TypeScript for types
 - Mayur: I'd prefer TypeScript
 - Roch: I'm not saying not to use TypeScript, just be smart about it how to use it. We have very small control as a library about the environment.
- Promise usage
 - Peter: I'll open an issue to advocate not using Promises because of several diagnostics reasons, like Core Dumps and stacktraces. Because of the Node versions, we won't be able to use native async await.
 - Roch: callbacks are also faster
- Package structure [issues/6](#)
- Semver
 - Peter: I'd prefer start with 1.0.0 for the first npm publish
 - Mayur: We need to discuss alpha/beta/GM etc.
 - Let's discuss it later in the project
- Continuation Local Storage
 - Peter: JS and Node don't have a native solution for CLS. The existing solutions CLS npm lib and Node's Async Hooks are slow. For such a large scale as Netflix CLS is too much overhead. I understand that APM providers need automatic tracing and CLS, so it's more about making CLS optional.
 - Roch: we typically see 20-30% impact because of CLS but it's acceptable for what it gives. Big companies who prefer non-CLS are the minority but we still should support them.
 - Peter: browser also doesn't have CLS
 - Agreement on making it optional

Agenda:

1. About this meeting
2. High-level plan

3. How to contribute
4. Major steps
5. Questions?

High-level plan

- 4/24** The reference candidate is presented for review.
- 5/8** Teams form, and work begins in all languages.
- 5/20** Project officially launched at Kubecon Barcelona.
- 9/6** C#, Golang, Java, NodeJS, and Python reach parity with existing projects.
- 11/6** OpenTracing and OpenCensus projects are officially sunset.
- 11/20** Sunset party at Observability Summit, Kubecon San Diego.

The migration for each language involves building a production-ready SDK, instrumentation for popular libraries, documentation, CI, and closing down the associated OpenCensus and OpenTracing projects (“sunsetting”). We have set an ambitious goal of **September 2019** as the target date for parity in C#, Go, Java, NodeJS, and Python. If a language is not ready by that time, we will push back the sunset date until it is. But we prefer this not to happen.

How to contribute

1. See: <https://github.com/open-telemetry/opentelemetry-node/blob/master/CONTRIBUTING.md>
 - a. Mark yourself on the issue or assign the issue to yourself.
 - b. I suggest for the initial stage - file issues and keep PRs targeted.
 - c. CLA to sign

Major Steps

We want to

1. Setup Infrastructure
2. Supported Node versions
3. Packages, Interfaces and Implementations
4. Plugins / Frameworks
5. Bridge (OC - OT)
6. Documentation
7. Benchmark

Step 1: Setup Infrastructure

- ~~Create a new repository for Node.js SDK under <https://github.com/open-telemetry>.~~
- ~~Setup CLA ([cla/linuxfoundation](#)).~~
- Setup a TypeScript Project.
- Setup [Lerna](#) to organize codebases into multi-package repositories.
- Configure [Travis CI](#) to build, test and deploy the sdk. Supported node versions 6 (EOL), 8, 10 and 11 and 12 (Current).
- Add/Enable [Renovate Bot](#) to keep source code dependencies up-to-date using automated Pull Requests.
- Add ISSUE_TEMPLATE (bug_report and feature_request) and CODEOWNERS.
- Add AUTHORS.md, CHANGELOG.md, CONTRIBUTING.md and RELEASING.md, CODE_OF_CONDUCT.md files.
- Use [Snyk](#) to continuously find & fix vulnerabilities within your code.
- Setup [Codecov](#) for unit tests, and enforce minimum threshold to 80%.

Step 2: Supported Node versions

- ~~Node 6 (EOL)~~
- Node 8.x
- Node 10.x and 11.x
- Node 12.x (Current)

Step 3: Packages, interfaces and implementations

The plan is to develop the new API, move the current OC implementation as a default implementation for the API and point the current OC public API to the new API.

- Start with what we have in OpenCensus Node.
 - Rename OpenCensus to OpenTelemetry.
- JS Packages
 - **opentelemetry/types**
 - Separate package with only TS interfaces and enums.
 - **opentelemetry or opentelemetry/api or opentelemetry/core**
 - This is the core npm package for OpenTelemetry. It contains the public API which is used by the various plugins. Also contains interfaces for tracing, stats, tags, metrics, etc.
 - **opentelemetry/basic-tracer**
 - With this users will have a full control over instrumentation and span creation. The base package doesn't load Continuation Local Storage (CLS) or any instrumentation plugin by default.
 - **opentelemetry/context-propagation or opentelemetry/context-cls**

- Separate package for CLS (Continuation Local Storage). The primary objective of CLS is to implement a transparent context API, that is, we don't need to pass around a ctx variable everywhere in application code.
- **opentelemetry/cls-tracer or opentelemetry/automatic-tracer**
 - Automatic tracer with CLS: opentelemetry/basic-tracer wrapped with opentelemetry/context-propagation
- **opentelemetry/exporter-***
 - Jaeger, Zipkin, Stackdriver, Instana, Prometheus, oc-agent etc.
- **opentelemetry/instrumentation-***
 - gRPC, HTTP/2/s, mongodb, redis etc.
- **opentelemetry/propagation-***
 - B3, Binary Format, TraceContext
- Interface definitions for tracing, metrics, and context propagation, based on the Cross-Language specification.
- A production-ready SDK, which implements these interfaces and exports Trace-Data. When available, the SDK will be created by porting the existing OpenCensus implementation.

Step 4: Plugins / Frameworks

Instrumentation for popular libraries currently covered by OpenTracing and OpenCensus. Maybe start by copying all plugins from the OC + OT libraries. Also, we should to have meaningful system tests with these frameworks.

- **gRPC**
- **http/s/2**
- **mongodb 1m**
- **redis 1m**
- **express 7m**
- connect 2.4m
- pg 747k
- generic-pool 771k
- mysql 330k
- koa 273k
- hapi 232k
- mysql2 175k
- restify 80k

Step 5: Bridge (OC - OT)

- TODO

Step 6: Documentation

- Add getting started guide and/or step-by-step tutorials.

- Add an example for major features or APIs.

Step 7: Benchmark

- TODO