RUM Community Group

2025-08-08 Edition

Agenda

- Logistics
- Next Meeting
- Agenda
  - Unresponsive crash reports, Self Profiling (Issac Gerges)
  - Chrome Soft Navigation Origin trial (Gilberto Cocchi)

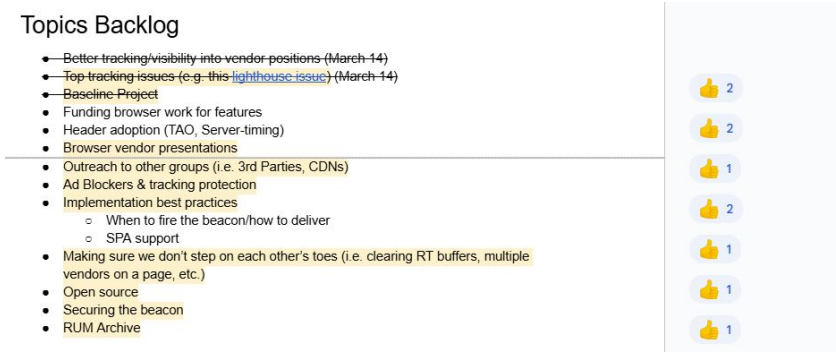
Logistics

- Meeting cadence
  - 2nd Friday of the Month
  - 60 minutes
  - 10am ET
- Agenda Document
  - (bit.ly/rumcg-agenda)
- Google Meet for Meetings
  - sub to rumcg-participants for invites
- Chat on Web Performance Slack
  `#w3c-rum-community-group`
  - (invites)
- public-rumcg@w3.org
  - (for discussions)

- RUM CG Github Repo
  - github.com/w3c-cg/rum
  - Group details
  - Links
  - Meeting minutes?
- RUM CG Tracking Project
  - github.com/orgs/w3c-cg/projects/1
- Meeting Minutes via fireflies.ai
  - AI summary + transcript will be posted after
  - Full video recording also available

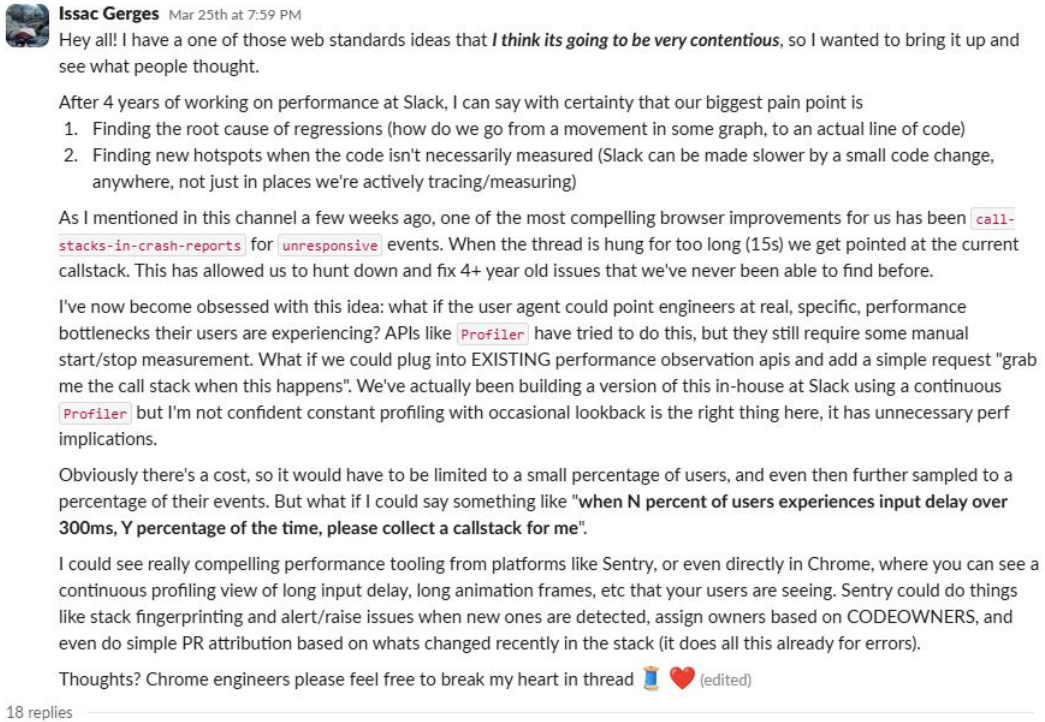
Next Meeting

- Fri **September 12th** @ 10am EST
  - 2nd Friday of each month
- Topics?
  - *(propose your topic here!)*
- Submit & Vote topics @ bit.ly/rumcg-agenda



Unresponsive crash reports + Self Profiling

- https://webperformance.slack.com/archives/C04BK7K1X/p1742947178736999



Issac Gerges Mar 25th at 7:59 PM

Hey all! I have a one of those web standards ideas that ***I think its going to be very contentious***, so I wanted to bring it up and see what people thought.

After 4 years of working on performance at Slack, I can say with certainty that our biggest pain point is

1. Finding the root cause of regressions (how do we go from a movement in some graph, to an actual line of code)
2. Finding new hotspots when the code isn't necessarily measured (Slack can be made slower by a small code change, anywhere, not just in places we're actively tracing/measuring)

As I mentioned in this channel a few weeks ago, one of the most compelling browser improvements for us has been `call-stacks-in-crash-reports` for `unresponsive` events. When the thread is hung for too long (15s) we get pointed at the current callstack. This has allowed us to hunt down and fix 4+ year old issues that we've never been able to find before.

I've now become obsessed with this idea: what if the user agent could point engineers at real, specific, performance bottlenecks their users are experiencing? APIs like `Profiler` have tried to do this, but they still require some manual start/stop measurement. What if we could plug into EXISTING performance observation apis and add a simple request "grab me the call stack when this happens". We've actually been building a version of this in-house at Slack using a continuous `Profiler` but I'm not confident constant profiling with occasional lookback is the right thing here, it has unnecessary perf implications.

Obviously there's a cost, so it would have to be limited to a small percentage of users, and even then further sampled to a percentage of their events. But what if I could say something like "**when N percent of users experiences input delay over 300ms, Y percentage of the time, please collect a callstack for me**".

I could see really compelling performance tooling from platforms like Sentry, or even directly in Chrome, where you can see a continuous profiling view of long input delay, long animation frames, etc that your users are seeing. Sentry could do things like stack fingerprinting and alert/raise issues when new ones are detected, assign owners based on CODEOWNERS, and even do simple PR attribution based on whats changed recently in the stack (it does all this already for errors).

Thoughts? Chrome engineers please feel free to break my heart in thread 🧵 ❤️ (edited)

18 replies