

How should we instrument Vue.js components? How should we monitor and debug errors?

Date: July 20th, 2020

Attendees:

Agenda

1. [How should we implement the Vue.js instrumentation for search?](#)
2. [How should we monitor and debug errors for search?](#)
3. How should we build components in WVUI for instrumentation?

Action items

- [JL] Capacity estimates (and mitigation strats) for logstash backend / error events;
- [JL] Raise priority of error logging documentation task
- [JL] Make ticket about error logging dashboard practices (--> doc)
- [JL] Make ticket about error logging privacy, dedup, etc., chronic vs acute errors
- <Assignee> <Action>
- <Assignee> <Action>

Questions

- Do we still need the Minerva-specific error counter tracking? See [\[Spike, 1hr\]: Analyse Catalan Wikipedia error logs for iOS error](#).
- Do we need a Vue.js or search specific counter?
- What dashboard should we be using to track errors these days? Do we need a project dashboard for Vue.js or search specific errors?
- Are the test wikis covered? See [Enable client error logging for Desktop Improvements Project and Vue.js search](#).
- Where is the documentation on for error tracking?
- Is the error reporting different for desktop and mobile?
- How would this affect the SearchSatisfaction schema/ability to A/B test?

Notes

- Error tracker records when error happened, not stack trace itself

- Example: client-side error logging with stack traces from Catalan Wikipedia https://logstash.wikimedia.org/app/kibana#/dashboard/AXDBY8Qhh3Uj6x1zCF56?_g=h@1251ff0&_a=h@a324cd9
- Want to know what's really happening (e.g. so many errors on a wiki that it's flooding logging)
- Logstash has an ingestion limit, need to look into whether Kafka acts as a buffer between EventGate and Logstash
- Because error logging is not like typical event logging but also isn't like something from statsd that's ops-focused, there aren't best practices or alternative solutions
- It might be that we'll want to add additional fields to the error schema and augment the instrumentation to include that data
- Documentation for error logging is in the works; the need to use it in production came quicker than expected.
- Error logging is not interactive, doesn't require instruments to log errors themselves. Done automatically by listening to errors on the window object. Right now you just enable it and all errors in the client go into logstash, disable it and none of the errors go into logstash.
 - Tagging functionality ([T258406](#)) also in the works
- Given how complex our ecosystem is... A lot of errors come from gadgets, so we don't want to have to guess about where the error came from. Stack trace should be somewhat helpful, although minification makes it harder.
- Vue.js search instrumentation
 - A/B test for moving the widget, using SearchSatisfaction
 - New widget – should use new instrumentation/schema or re-use SearchSatisfaction schema/instrumentation?
 - Will strive to re-use the schema
- When search dropdown appears, plenty of opportunities to decide when search session has started
 - One approach is the sidebar approach where data attributes are added to specific components – downside is if you don't supply the information from the start, you can't add it later
 - Another option is emit specific events
 - Long term – more sophisticated options like what popups instrumentation does
- If you have a button that emits a click event, which a client can do whatever they want with. Less straightforward for more complex situations where there are multiple components.
 - JR: Hard to suggest an approach right now. Would expect the instrumentation code to not be separate from the library itself. If instrumentation lives in WikimediaEvents, how do we hook into the search widget's Vue.js codebase?
 - Note – there's no hard requirement for keeping analytics instrumentation in WikimediaEvents

References

- [\[Spike\] How should we monitor and debug client errors for Vue.js search?](#)
- [Instrument Vue.js search](#)
- [Enable error tracking on a Wikipedia mobile wiki to catch and fix high volume bugs](#)
- [\[Spike\] What should we instrument in the new Vue.js search experience?](#)

- [Enable client error logging for Desktop Improvements Project and Vue.js search](#)
- [\[Spike, 1hr\]: Analyse Catalan Wikipedia error logs for iOS error](#)
- [Adding skin version to search satisfaction](#)