Web Debuggability - Push & Notifications

rayankans@, knollr@
2019-04-29

Background

There's an ongoing effort to make web features, specifically features that involve background behaviour (via Service Workers), more debuggable. Web features are encouraged to notify the `DevToolsBackgroundServicesContext` when key events that might be of interest to a developer happen.

An event the background web feature provides must include the following:

- Origin + Service Worker ID
  To provide the source of the event.
- Instance ID
  An identifier for grouping related events together, e.g. a tag.
- Event Name
  A short description of the event being shown
- Metadata
  Event-specific information that might be useful (key-value pairs).

This has been implemented for Background Fetch & Background Sync. We plan to extend this functionality to two other commonly used web background features - Push & Notifications.

This doc will go over the events we plan to log for each feature, describing when it would happen, and the extra metadata we plan to include.

Push

There is no concept of Instance Id here, we will have to provide some unique identifier for events related to one push.

Event Name	Description	Metadata
Received	A push message has been received. This is would be useful for capturing config errors. The push message is being delivered but something unexpected happened in the pipeline.	success, was_encrypted, failure_reason - encryption related - other (e.g. permission changed)
Event Dispatched	When the Service Worker `push` event is called. Useful for giving a full picture of the web push flow, and the current state an instance is in.	
Event Completed	The result of the Service Worker execution Not exposed, can help developers differentiate between the error types that are different in nature (timeout vs failure).	Service Worker Status (success/rejected/timeout/ other)
Unregistered (not by user)	Sometimes the subscription gets unregistered if certain errors are encountered (code)	reason
Generic Notification Shown	If the origin failed to display a notification the browser shows a generic one	
Subscription Changed [In the Future]	There are plans to periodically update the subscription in the future.	registration/subscription related metadata

Subscribe/Unsubscribe events are not included here since a lot of details are already exposed to developers. These events also don't fit in nicely with the concept of instance IDs (tags), and debugging a workflow.

Notifications

Instance Id could be the developer provided "tag" for a notification.

Event Name	Description	Metadata
Show	Notification is shown to the user.	
Close	Notification is closed by the user.	
Activate	User clicked on the notification.	success, error code
Schedule	Notification is scheduled to be shown.	showTriggerTimestamp, failure reason (e.g. too far in the future, quota exceeded)

Appendix

How to try it out on DevTools

How to enable

- Make sure you are on Chrome Canary (not available on gLinux)
- Enable the `enable-devtools-experiments` flag in chrome://flags
- Open DevTools and go to Settings > Experiments
- Click shift repeatedly until the hidden experiments show up
- Enable the `Background Service events` experiment
- Restart DevTools

How to try it out

- Go to https://backgroundfetch.com
- Open DevTools > Application > Background Fetch
- Toggle the record button
- Start a fetch and see the events flowing!

Alternative domains to try things out on

- https://www2.backgroundfetch.com, it's a clone of backgroundfetch.com to try out events from multiple domains.