The state of browser storage partitioning

TPAC 2021 Breakout
Logistics

- Please sign-in in the breakout doc
- Queuing will be done in Zoom (next slide for instructions)
- Agenda
- Volunteer scribe(s)
Queuing to speak
W3C Code of Ethics and Professional Conduct

https://www.w3.org/Consortium/cepc/
3.1 Expected Behavior

Treat each other with respect, professionalism, fairness, and sensitivity to our many differences and strengths, including in situations of high pressure and urgency.
3.2 Unacceptable Behavior

W3C strictly prohibits *discrimination*, *intimidation*, *harassment*, and *bullying* of any kind and on any basis.
Group Agreements for this Workshop

- One Mic
- Take Space & Make Space
- Anything else?
Overview of the Session, Introductions (5m)

The state of partitioning in browsers today (in alphabetical order, 20m)
  ○ Chrome (5 minutes)
  ○ Edge (5 minutes)
  ○ Firefox (5 minutes)
  ○ Safari (5 minutes)
  ○ (other browsers not on the agenda?)

Discussion Items (20m)

Wrap-up and next steps (5m)
Session today

In scope

We’re here to talk about Client Side Storage Partitioning, aka “storage and communications APIs above the network stack.”

Out of scope

Literally anything else.
Chrome
Chrome status update

miketaylr@google.com
“Working” partitioned APIs

- ServiceWorker
"Almost working" / in progress APIs

- SharedWorker
- Web Storage
- IndexedDB
- CacheStorage
- FileSystem API(s)
- BroadcastChannel
TODO

- Web Locks API
- Storage Buckets
Web Platform Tests

- **ServiceWorker**: passes in Chromium (when feature is enabled), passes in Safari
- **localStorage**: passes in Safari, passes in Firefox, currently fails in Chromium
Aspirational Milestone

- Would like to have most-used cross-browser storage APIs (IndexedDB, Web Storage, ServiceWorker, CacheStorage) functional behind a flag... soon.
Firefox
State Partitioning in Firefox

Current State

- Network caches are partitioned (by top-level site)
- Cookies and most storage APIs are partitioned
  - Rolled out to Private Browsing + Strict Mode
  - Working on enabling by default
- Storage Access API + Heuristics can “unpartition” cookies and storage
- [Documentation on MDN](#)
Goals:

- Partition Remaining APIs
  - Service Workers WIP, targeting Fx96
  - Blob URLs (WIP, behind a flag)
- Ensure cross-browser compatibility
- Upstream some of our internal tests to WPT
- **Always keep storage partitioned**
  - (Align with Safari’s current behavior)
- Longer term: more keys
thdwb

the hotdog

web browser
Discussion Items

1. “Storage” access + unpartitionining (non-cookie) storage
2. BroadcastChannel partitioning ([link](#))
3. Blob URLs ([link](#))
4. Cross-site ancestor chain bit ([link](#))
5. Web Extensions + Storage Partitioning
Ad hoc discussion

Please queue via Zoom.

(If we have time, we can see if we can close out any open issues).
Thanks!