Related Website Sets (fka First-Party Sets)

First, some logistics

- Sign-in on <u>goo.gle/fps-meeting-notes</u>
- Scribe volunteers?
- Please use the Queue section in the meeting notes Google Doc

What's new?

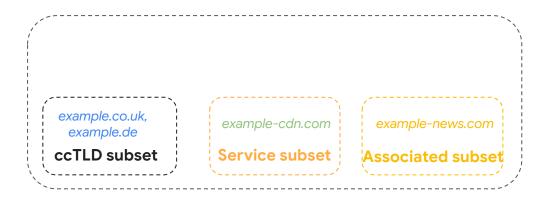
- We're <u>renaming</u> FPS to RWS (Related Website Sets) to better represent the intent of the API
- Numeric limit for domains in the associated subset increased from 3 to 5.
- Currently <u>rolling out</u> RWS in Chrome to 10% of stable population
- Currently <u>rolling out</u> support for Storage Access API (SAA) for sites outside of the same RWS in Chrome 117+.
- Published <u>explainer</u> to extend SAA for access to unpartitioned storage
- Set submission process is operational on <u>GoogleChrome/first-party-sets</u> GitHub repo
- New <u>RWS JSON Generator tool</u> is live

A Refresher

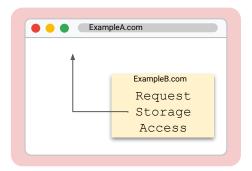
Subset-based definitions

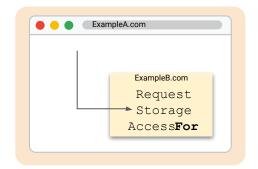
Site owners declare relationships between domains, based on use cases, in subsets. Subsets have technical and policy-based restrictions.

Browsers can use this information to preserve cross-site functionality within a set.



Requesting cookie access





requestStorageAccess:

An iframe ExampleB.com requests storage access *on* ExampleA.com, where A and B are part of the same set

requestStorageAccessFor: A top

level site ExampleA.com requests storage access for ExampleB.com, where A and B are part of the same set

If ExampleB.com and ExampleA.com are in the same set, Chrome will **automatically** grant the cookie access request, without surfacing a prompt to the user.

The process of creating a set

Developer uses SAA and rSAFor

 APIs allow cross-site sharing of info within a set
Chrome checks ingested RWS list on API request

Browser checks set

Owner submits a new set in GitHub

Declare set primary
Add domains to subsets

 Is every domain registrable?
Is any domain present in existing set?
Does the webmaster control the requested domains?
Subset-specific technical checks

Pass

1. Chrome adds set to publicly viewable list 2. Chrome uses list to manage access legitimate requests

Set added to list

Fail

Set <u>not</u> added to list

1. Requester receives a notification of failure through GitHub



Next steps and improvements

Continuing to seek feedback on

- <u>Response header to opt-in to storage access</u>
- Integration with CHIPS
- <u>Ergonomics</u> of requestStorageAccess and requestStorageAccessFor
- Improvements to usability and scalability of <u>set submission process</u>

Potential new adjacent areas of work

- Extending SAA to better serve login use-cases
 - Useful where the associated subset limit restricts critical user journeys like login

Thank you!

Appendix

What to test

1 Enable FirstPartySets & Declare a set locally



Create a local set by following our <u>developer</u> <u>guide</u>. Familiarize yourself with the <u>Submission Guidelines</u> and <u>try submitting</u>

2 Enable StorageAccessAPI & StorageAccessAPIForOriginExtension



Test how these APIs work with sets, and understand the technical constraints

3 Enable UI:

PageInfoCookiesSubpage &PrivacySandboxFirstPartySetsUI



Test how Chrome users interact with RWS