Related Website Sets (fka First-Party Sets)

TPAC 2023

First, some logistics

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

What's new?

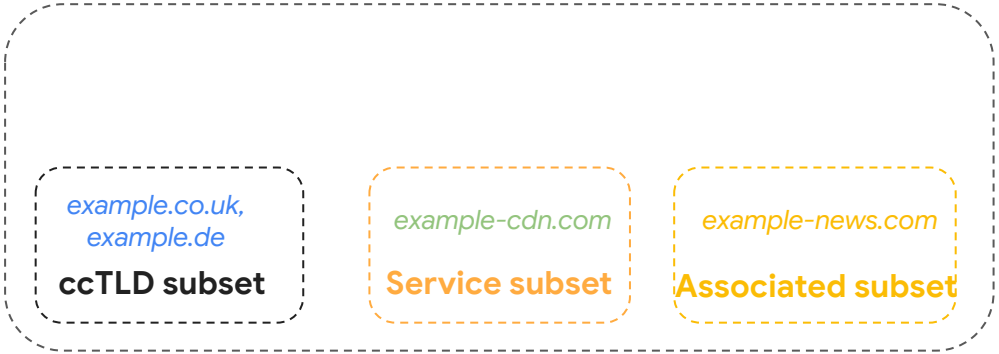
- We're renaming FPS to RWS (Related Website Sets) to better represent the intent of the API
- Numeric limit for domains in the associated subset increased from 3 to 5.
- Currently rolling out RWS in Chrome to 10% of stable population
- Currently rolling out support for Storage Access API (SAA) for sites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

A Refresher

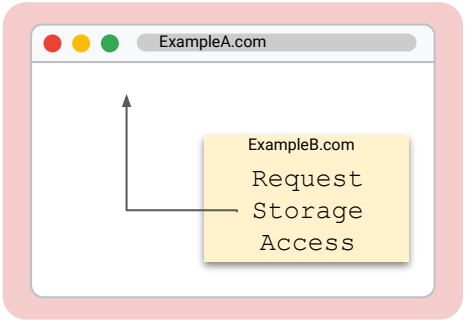
Subset-based definitions

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

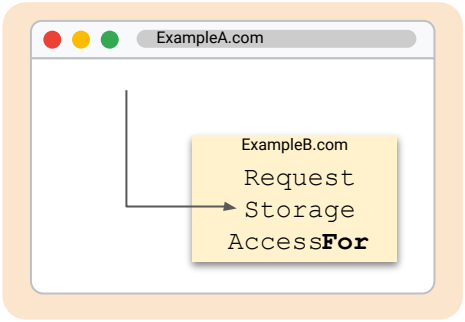
Browsers can use this information to preserve cross-site functionality within a set.



Requesting cookie access



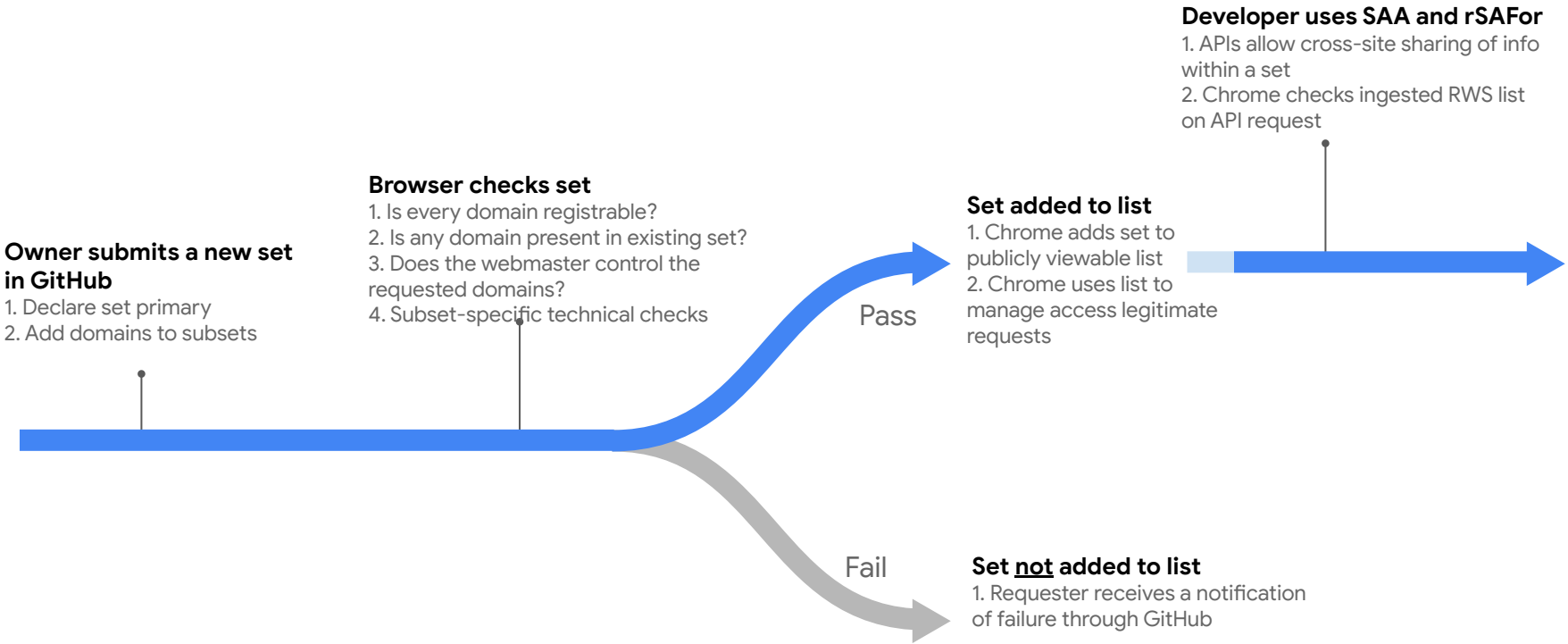
requestStorageAccess: An iframe ExampleB.com requests storage access *on* ExampleA.com, where A and B are part of the same set



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



[Submission Guidelines]

Demo

Next steps and improvements

Continuing to seek feedback on

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

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 developer guide. Familiarize yourself with the Submission Guidelines and try submitting

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