

Cookies



Future

Notes:

<https://pad.w3.org/p/tpac-cookie-layering>

IRC: #cookie-layering

2022 @ TPAC:

- Discussed and agreed on the problem
 - We can't standardize 3PC semantics, SAA, etc. without fundamental layering changes to how 6265bis and browser specs integrate

2023 @ TPAC:

- Agreed on overall goals and infrastructure
- Started drafting changes to RFC 6265bis (Cookies)
- Started sketching out how Fetch / HTML would have to be changed

This year:

- 6265bis is moving out of HTTPWG to become an RFC
- Wrote an updated Cookies spec that addresses layering
- Reviewed all integration points with the cookie spec in browser land, filed issues
 - Primary concern is the Cookie Store API
- Wrote a draft Fetch PR

Recap: Why 6265bis (Cookies) needs a revision

- Contains many layering violations
- Needs hooks to standardize “third-party cookie blocking”
- Is hard to build on for:
 - Cookie Store API
 - Fetch
 - HTML
 - Storage Access API
- Has many deferred issues
- Partitioned attribute is getting traction

What did we change?

- Have a shared concept of a “cookie store”
- Define a set of shared operations around this “cookie store”
- Define a set of expectations around how these operations are to be implemented across all user agents
 - I.e., that a Set-Cookie header has to be parsed and the resulting cookie is stored.
- Provide a concrete implementation for non-browser user agents
- Leave the implementation for browser user agents up to Fetch et al.
 - We don't want Cookies leaving crumbs all over the DOM!

Our draft (feedback welcome!)

<https://github.com/johannhof/draft-annevk-johannhof-httpbis-cookies>

Next steps / open questions

- Get our Cookies draft adopted by IETF HTTP WG
- Refine Cookies draft
 - Cookie Store!
 - Make events work
 - Incorporate CHIPS / Partitioning
 - Find some way forward on SameSite=Lax default behavior
- Finish Fetch PR
- Create HTML PR
- Create Cookie Store API PR
- Integrate Storage Access and 3PC Blocking semantics
- Write Web Platform Tests!!!