Web Packaging

Jeffrey Yasskin — he/him — Google Chrome
TPAC — 2020-10-26

Meeting Agreements

- This is a shared problem-solving space
- Intent < Impact
- One speaker at a time
- Make space & Take space
- Use I statements
- Anything else?
- Speak slowly for the interpreter

Agenda forming

1. Preliminary agenda on next slide
2. If you have a more important topic, add it to the agenda.
3. We'll approval-vote for the topics you want to discuss, and then discuss the ones that get the most votes.
4. 15 minutes per topic, so we have time for 3?

Agenda

1. Restricting bundles to the same-origin/secure context case (unsigned bundles); 6
2. Subresources with "incorrect" URLs. (#551) 6 votes
3. Ability to send packaged apps from client to client over WebRTC; 6 votes
4. Direct links to bundle subresources. (#26) 5 votes
5. Changing the 7-day bound on signature expiration. (#597) 4 votes
6. Origins for bundle subresources. (#583) 3 votes
7. Letting users limit data usage by blocking subresources. (#594) 2 votes
8. *Your issue here*

Restricting bundles to the same-origin/secure context case (unsigned bundles)

- How do bundles relate to the origin model?

Subresources with "incorrect" URLs

- Package can name subresources with arbitrary URLs.
- Fetching a URL directly might not give the same content as the subresource.
- Content blocker that only blocks based on the claimed URL could be unable to block what it needs to block.
- Can block using fully-qualified name: package$subresource.
- Browser cache might provide incentive for packages to be honest.

Ability to send packaged apps from client to client over WebRTC

7-day bound on signature expiration

- Compromise between safety and usefulness.
- Publishers can set a shorter bound but not a longer one.
- User only has to receive the content within the expiration; content can then install a Service Worker that lives longer.

Letting users limit data usage by blocking subresources

- Pages will compile bundles that serve their intended use, which is likely to leave out users who want to avoid downloading images or scripts.
- Cache-efficiency idea lets users block by name.
- Should we also have the format identify the resource "destination"?

Origins for bundle subresources

- When a subresource in a bundle is used as a top-level page or an iframe, what's its origin?
- Proposed answer: pair of bundle's URL + subresource's name's origin
- URL bar emphasizes the bundle's URL, as if the subresource's name is part of the path?
- When a user saves a bundle locally, what happens to its origin?

Direct links to bundle subresources

- Easier than assigning origins to subresources; could use fragments.
- What use cases do we care about?