[EXTERNAL] Web Install API – TPAC discussion

Link to explainer: https://github.com/MicrosoftEdge/MSEdgeExplainers/blob/main/WebInstall/explainer.md

Present:

- Diego González (Microsoft)
- Dan Appelquist (TAG)
- Thomas Steiner (Google Chrome)
- Simon Pieters (Mozilla)
- Sangwhan Moon (TAG, Google not Chrome)
- Reilly Grant (Google Chrom
- Matt Giuca (Google ChromeOS)
- Brian Kardell (Igalia)
- Javier Fernandez (Igalia)
- Patrick Brosset (Microsoft Edge)
- Vincent Scheib (Chrome)
- Penelope McLachlan (Google Chrome)
- Yves Lafon (W3C, TAG)
- John Riviello (Comcast)
- Andreas Bovens (Whereby)
- Christian Liebel (Thinktecture)
- Fuqiao Xue (W3C)
- Kagami Rosylight (Mozilla)
- Matt Reynolds (Google Chrome)
- Lu Huang (Microsoft Edge)

Why?

- Allow PWAs and web apps to install from the web platform.
- Enable store-front scenarios to work on the web without workarounds like proprietary protocols

Use cases

- Installing an app from the current domain.
- Installing an app from an associated (cross-domain)
- Creation of online catalogs

Goals

- Enable installation of web apps (same and cross-domain).

- Replace beforeinstallprompt or associated behaviour (current way to install apps from the same-domain).
- Allow a vetted installation origin to know if the web app is installed (see install_sources new manifest field).
- Allow the web app to report to the installation origin the outcome of the installation.
- Enable UAs to supress potential installation-prompt spailam.
- Track campaign IDs for marketing campaigns.

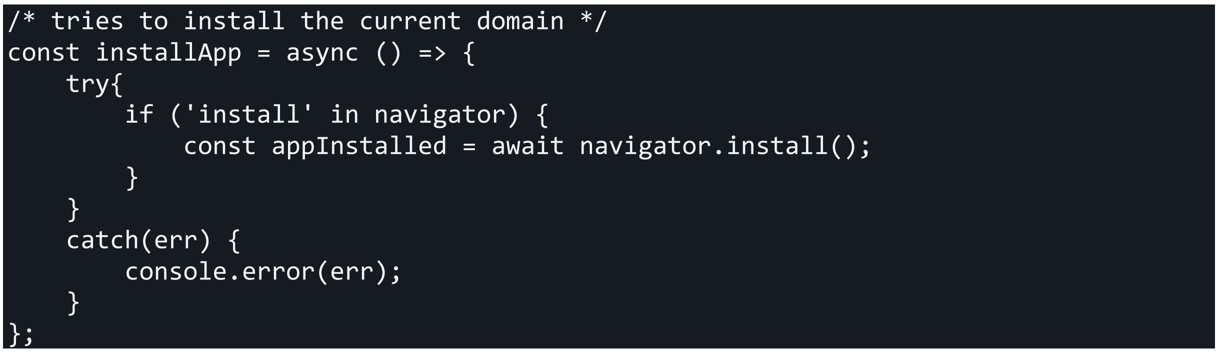
Non-goals

- Change the way the UA currently prompts for installation of a PWA.
- Associate ratings and reviews with the installed app (see Ratings and Reviews API explainer).
- Process payments for installation of PWAs (see Payment Request API).
- List purchased/installed goods from a store (see Digital Goods API).
- Installing non-PWAs or apps that do not pass the installability criteria (for security reasons).
- Enumerate if the app/related apps are installed (see getInstalledRelatedApps).

What?

JS API. Consists of a method that takes a manifest ID and optional parameters to install the application.

```
navigator.install([<manifestID>[, <params>]]);
```



```
/* tries to install the current domain */
const installApp = async () => {
    try{
        if ('install' in navigator) {
            const appInstalled = await navigator.install();
        }
    }
    catch(err) {
        console.error(err);
    }
};
```

Promise-based method that resolves if the app has being installed or rejects if not.

- Resolves:
  - Object -> where was it installed
- Rejects:
  - There is no Permissions Policy for the origin to install sites
  - The target url/id does not comply with installability criteria
  - There hasn’t been enough engagement with the app
  - The user cancels the installation process
  - Other

Privacy and Security

Main concern is to avoid installation prompt spam.

- API can only be involved in a top level secure context, and in the main scope of the application.
- Require a user gesture
  - Enough engagement with the app. (same threshold that oBIP current uses)
- Integrated with the Permissions API (new `installation` permission type)
  - Time-limited
- New manifest field (install_sources) (application allowing the origin – most restrictive scenario is the same one that we current have for apps – only being installable from the same domain)
  - Optional `inquire` key/value that may allow the origin to know if the app is installed, ONLY with prior permission from the app and ALWAYS in control from the user allowing to turn this off.

```
  {
    "name": "Awesome Application",
    "display": "standalone",
    "start_url": "/index.html",
    "install_sources": [
            {"origin": "apps.microsoft.com", "inquire": true},
            {"origin": "store.app", "inquire": false}
    ]
}
```

Other

- A method to test if an app can be installed (likely to be removed, theres no more need for this)
- Can be used with prefers_related_applications to try to handoff the installation to the native store platform

Open Questions

- Try to buy scenario
- navigator.isInstalled(manifestID) ??
- alternatives for app install check??

Discussion notes

Diego presents the above feature.

Kenneth: what about iframes.

Diego: you can’t do this, It's only top-level.

Penelope McLachlan : why not specify passing preferred payment handling (digital goods API endpoint?) via params? Could be fully optional. Agree that UA should have no part in enforcing this, but it could be a useful hint for catalogs

Diego: parameters object is made in a way that if you wanted to pass more info, then you could. Was thinking of using for attribution/campaign id.

Marcos: feels like this will be a windows-solution. Deviating from the manifest start-url, etc. Be mindful about the manifest. The side-panel thing seems weird, feels like it should be in the manifest.

Diego: if it makes sense to align with a new type of display mode. But, if there is a new surface (sidebar), that multiple browsers have, this could be used via the navigator.install() method.

Marcos: navigator.install is a big decision. We thought about this API a long time ago. Should be a thoughtful decision.

Dmurphy: The options object could be seen as a hint. Then checking the manifest, to make sure that the sidebar is supported. The app will support a certain number of display mode. Standalone or sidepanel. This could be seen as a hint to let the user choose (in a button) where to install.

Marcos: there might not be a store. Feature should be designed in a way that the store isn't required.

Tess: but the mode is already in the manifest. This is repeating ourselves here. Also it's called display in the manifest, and mode here.

Marcos: what does this mean in terms of data sharing between the install_sources sites, regarding the inquire property?

Diego: goal is to establish different levels of trust.

Penny: If you want a catalog to behave like other existing catalogs, which know if apps have already been installed or not. Could be worked around. User might be attempting to install the same app multiple times.

Matt: what is the mechanism for inquire?

Diego: there's currently an issue on the repo about this. There might be a need for a method instead. This is just in the explainer for now.

Matt: makes a lot of sense for a store. There is the non-standard getInstalledRelatedApps. But you have the have a bi-di relationship between the sites. Which this proposal is doing. Could reuse this.

Marcos: why integrating it with the permissions API?

Tess: seems redundant. Can't install it unless the user agrees to.

Diego: this would be for the first time the user visits the app repo. You'll get prompted the first time. Once granted, you will just be prompted to install a site.

Tess: why have N+1 prompt.

Marcos: am I installing the store app before I can install other apps? IF you want a repo to have this capability, then it should be built-in to the manifest. Why would bank.com need this? Why does mumandpop.com need this? Should only be app stores, that should need to be installed.

Reilly: there's a split that needs to be made between sites that install themselves, and sites that install other apps.

Penny: Any sites wouldn't need to install any other sites. There wouldn't be a circumstance for a random site to install other sites anyway.

Marcos: It's a big deal to have this power though.

Tess: a website that doesn't have a manifest can't know if it's installed? (regarding navigator.isInstalled).

Dmurphy: it can, using the manifest ID.

DanA: comment: I wish we saw more explainers that had great diagrams! Thank you.

Penny: Generally like this proposal.

- Penny: To disambiguate from prior discussion, this is just about self-install permission. Interested in making Installing a site (self install, not x-origin) a Permission, could be benefits to making install just like any other capability, interested in reactions from the room on this. Would make it PEPC compatible too, at least for same origin (possible for x-origin too but need to think about it)!
- Penny: -1 to a permission for a catalog site to install other sites, doesn't feel necessary given the allowlist restriction for target apps, we discussed this Tue though and don't want to flog it further unless there's interest in reopening the discussion.
    - Reilly: I do like the idea of using the embedded permission control as a button. But push back on trying to fit it into the permissions system. Difference between permissions and settings. There are things that are more user control than permissions.
    - Matt: I think it's neither a permission nor a setting. Doing an app install creates a new object. It doesn't change the mode that you are in currently. PEPC.
- Howard: Interested in the room's thoughts/concerns with the ability to pass in string data (in the form of the ReferralInfo optional property) to allow an installed site the ability to possibly learn what ad campaign might have led a user to install the web app from a different source.
- Matt: as a non-privacy person, it seems reasonable. It's like one of those utm things.
- Reilly: it feels like a referrer header of some kind.
- Tess: we shouldn't <couldn't get the gist of this, please help scribe>
- Howard: for stores, some stores have the desire to track per user

MS Store has this concept of campaignID. Recording for eternity since the first time a user installed an app. The app can then query this. That is a per user tracked event. Curious if there is concern continuing to do this.

Tess: yes, there is. If we're going to persist that ID. If it's recorded the first time and never goes away, that's not going to fly.

Matt: nothing prevents you from building your own one. Like add your utm thing in the destination manifest file. You'd be able to forever look at this to see who installed you.

Marcos: chances are, the user agent would strip that. Like Mail for example strips things like that that are easy to recognize. A user agent could decide to do the same.

- François: Isn't "install_sources" a bit restrictive? How may I write an HTML page that would list applications that I installed on my computer and may want to re-install somewhere else? Said differently, this seems to encode a business contract between the app and some catalog site which is not necessarily beneficial to users.
    - Matt: +1 to that.
    - Diego: use case would be if you get the app from the apple store.
    - Matt: this is authorizing stores to install you. Analogy: every page had a list of sites that are allowed to link to me. That's not how the web works. You don't get to control what installs you. Where does come from?
    - Reilly: not entirely true. There are headers to prevents embedding things on webpages for example. There's a correlation.

  - Matt: is there a reason for an app not to be installed by another site?
  - Reilly:
  - .
- Dan Murphy: (we've chatted already) but I'd love for us to always specify the manifest_id first, for all of these method calls.
  - Github issue: https://github.com/MicrosoftEdge/MSEdgeExplainers/issues/669
  -
  -
  - .
- Reilly: Mostly a question for Apple, for sites installing other sites: Is that fundamentally abhorrent or would some combination of the app store site being installed or requesting an "install other sites" permission be reasonable?
  - Vincent: let's use a real use case. GMail, GChat are on different origins. But they also have integrated experiences. And they can cross-promote. They want to promote the install of an app on another origin.
  - Tess: can they just link to the other site?
  - Vincent: could just link, yes, and the other site would self-install.
  - Tess: I like that.
  - Vincent: more clicks though. Additional steps.
  - Marcos: manifest was defined the same way as CSS. Can't use it without its originated document. It often needs the document as well.
  - Matt: we fixed that though.
  - Tess:
  -
  -
- Brian: Is the idea of a site that can install other sites an example that everyone can agree is a "powerful feature" inline with the other session today? It seems to me it is, but I'd like to see if that is others perception too?