WEB INSTALL API

Diego González, Amanda Baker

with guests

Dan Murphy, Reilly Grant

Microsoft Edge

Web Install ×

agenda

(logistics)

- Intro to the state of apps.
- Describe the API:
- Give an update on the state of the API.
- Describe how the Web Install API integrates with Acquisition Info API.
- Open discussion.

Why is it like this?

Installed Web Apps - how they are built, function, and struggle today. (Abridged)

Web Install

About “installation”

(or... what constitutes installing?)

What constitutes installing?

- App integrates with OS UX (Edge/Chrome on Windows/macOS/Linux).
- App gets added to a launcher/menu/OS UI (Safari on macOS).
- App gets an icon on the homescreen of a device -A2HS (Add to home screen on Firefox on Android).

There is different ways of across browsers to do “installs”

- UA UX
  - Prompting (Chromium)
- Developer controlled (Imperative)
  - BeforeInstallPrompt (on Chromium devices)

What is Web Install?

The Web Install API provides a way to democratise and decentralise web application acquisition, by enabling "do-it-yourself" developers to have control over the application discovery and distribution process, providing them with the tools they need to allow a web site to install a web app.

💻 Demo

navigator.install

https://aka.ms/**WebInstall**

Use cases

(of navigator.install())

- Web sites installing their web apps.
- Creation of online app catalogues.
- Install apps from a search engine.
- Etc.

Proposed solution

- Promise-based method (navigator.install) that resolves if an app is installed and rejects with a *minimal* set of DOMExceptions to balance privacy and usability.

- Parameters include a manifest_id, an install_url and optional object.
  - Manifest id is the *what* to install
  - Install url is the *where* to find it.

What is an install_url?

- A url meant for installing an app.
- Can be any url in scope of the manifest file that links to the manifest.
- Must not redirect nor contain extra content that is not relevant for installation purposes.
- Ideally it would contain only the information necessary to install the app (manifest, icons, service worker)
- Why? It avoids many of the pitfalls described earlier.

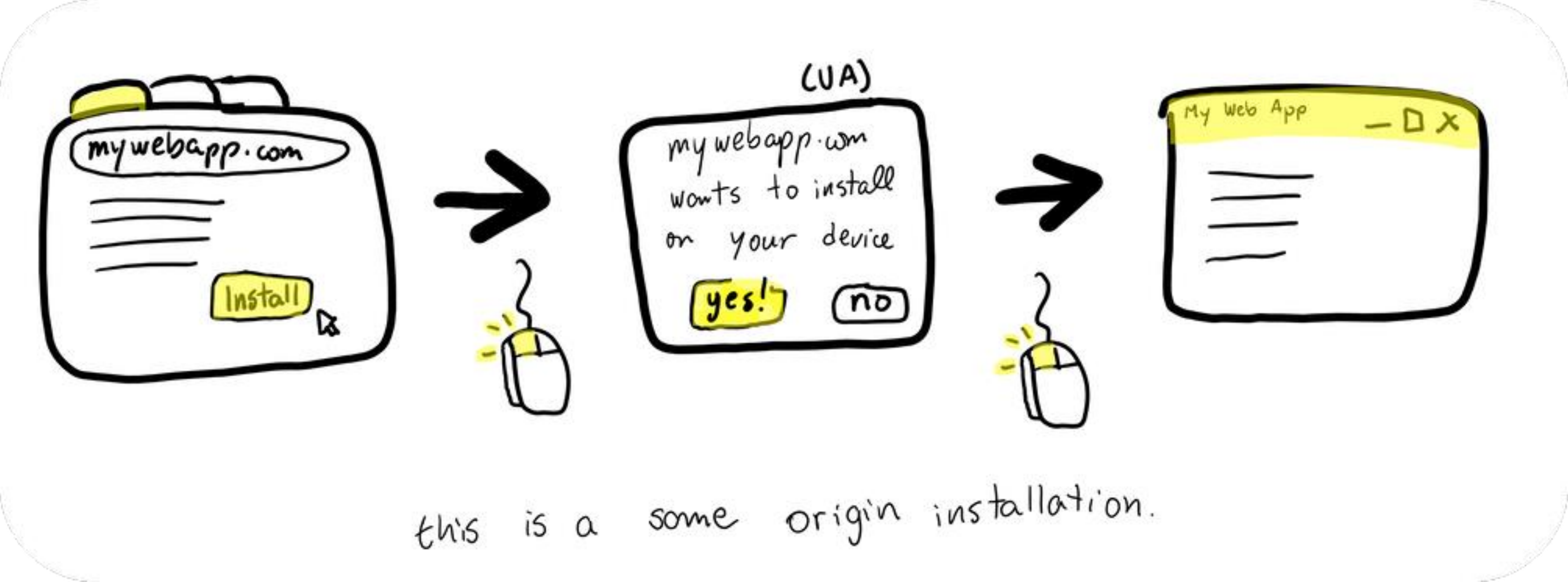
Web Install ×

same-origin install

https://aka.ms/**WebInstall**

navigator.install

(same-origin use case)


mywebapp.com
Install
(UA)
mywebapp.com wants to install on your device
yes!
no
My Web App
this is a same origin installation.

navigator.install

(same-origin code sample)

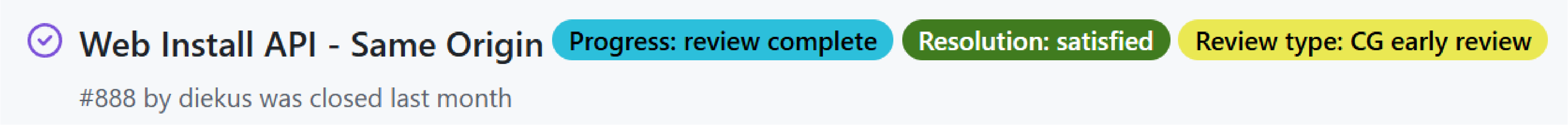
```
/* install current document */
navigator.install(id);

/* or install a different url within the current origin */
navigator.install(id, install_url);
```

Web Install ×

navigator.install

(Status: same-origin)



Web Install API - Same Origin Progress: review complete Resolution: satisfied Review type: CG early review

#888 by diekus was closed last month

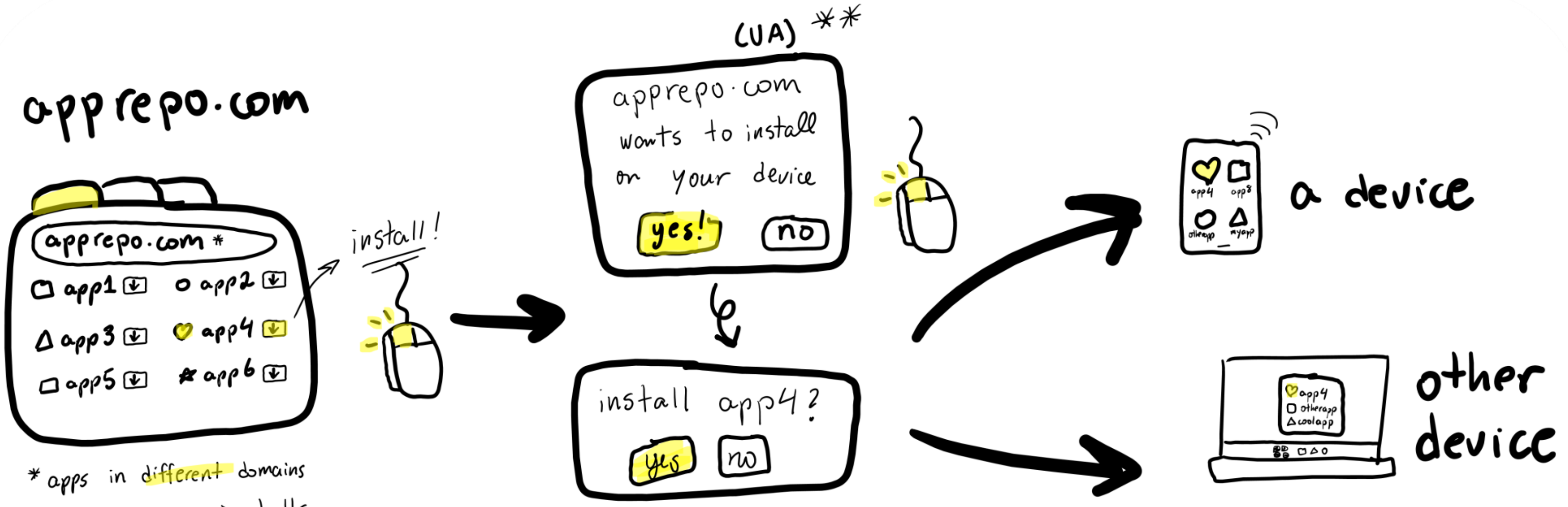
Web Install ×

cross-origin install

https://aka.ms/**WebInstall**

navigator.install

(cross-origin use case)



* apps in different domains (apprepo.com installs app4.com)

** an origin will require permission to install apps. This prompt from the UA appears only if the permission needs to be given.

navigator.install

(cross-origin code sample)

```
/* install_url is required for cross-origin installation*/
navigator.install(manifest_id, install_url);
```

Recent changes

Recent changes

- (Thinking)-> Removed install_sources
  - Addresses “centralization” concerns from TAG?
  - Downside is less control for developers
  - Overall we’re open to the change if it is an issue.

- Updated explainer to:
  - Focus on the value that the feature brings to the web and to web developers.
  - Expand on alternate use cases.

Web Install ×

✓ Respect same-origin policy

✓ User activation

✓ Permissions API integration

✓ Optional custom post-install UX

🦺 MAKE IT SAFE

same-origin policy

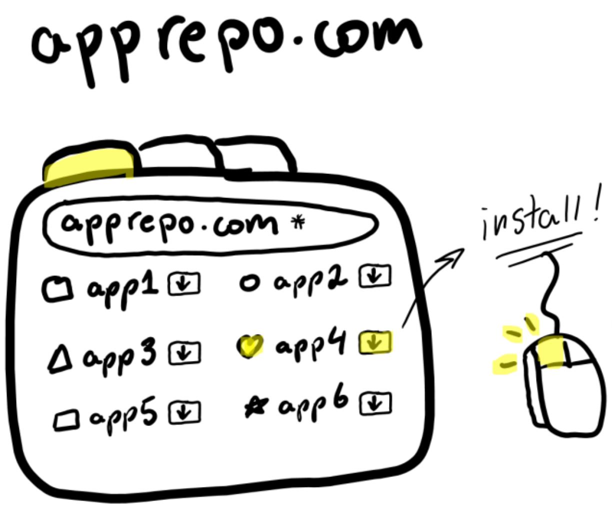
(make it safe)

- Permissions are not auto-granted to installed apps.
- API does not break the same-origin security model of the web.
- Each domain keeps its own permissions controlled by the user.

user activation

(make it safe)

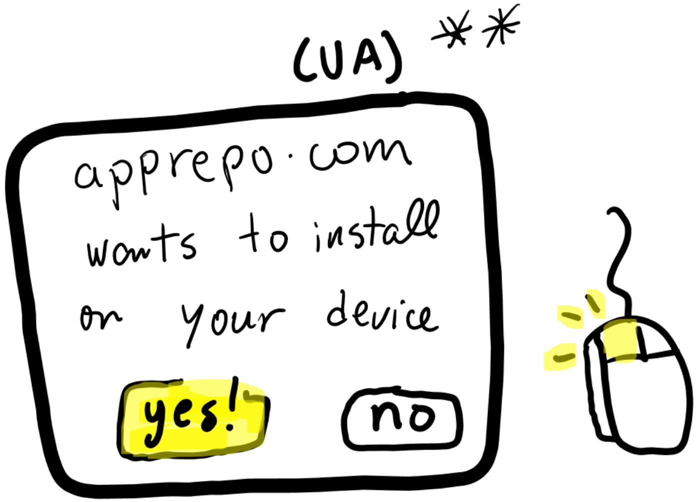
- Installation will be gated behind a user activation (gesture).
- UA should implement rate limiting to further reduce spam risk.



permissions API integration

(make it safe)

- Gate cross-origin installation behind a permission.
- Prevents sites from spamming users who reject the permission.
- Note that same-origin installs do not require this permission.



apprepo.com wants to install on your device

Optional custom post-install UX

- UAs can add additional UX to help the user uninstall the app if it's not what they expected

Web Install ×

📦 Where did my app come from?

navigator.
acquisitionInfoProvider

https://aka.ms/**WebInstallAcquisition**

navigator.acquisitionInfoProvider

- Supplements the Web Install API to let the developer know how the web app was acquired for cross-origin installs.

- Enables retrieval of the install source and attribution id (campaign id).

- Optional parameter of Web Install. (“referralInfo”)
  - *Install source*: (required)
  - *Attribution id*: a proposed standardized dictionary property for the API that keeps track of ad attribution for web app installations, similar to the campaignId for Microsoft Store or Apple Store.
  - Optional data like *install time*, *region*, *referrer*

navigator.acquisitionInfoProvider

```
// Web install with acquisition information
const appInstalled =
await navigator.install("foo.com", {"referral-info": referralInfo});
```

```
details = {
  installSource: "appstore.com",
  attributionId: "adsAug2023",
  httpReferrer: "https://bar.com/",
  installTimestamp: "2023-08-16 10:30:00 UTC"
// referralInfo:
  region: "US"
  installVersion: "1.0.0.0"
}
```

navigator.acquisitionInfoProvider

```
let details = await navigator.acquisitionInfoProvider.getDetails();
let installSource = details['installSource’]; //e.g. apps.microsoft.com
let campaignId = details[‘attributionId]; //e.g. adCampaign2306
```

⁉ Discussion

Topics

- Hot topics
  - Removal of install_sources
  - Declarative approach
  - Same-origin installation
  - (something that we were leaving to UA to decide)
    - Installabilty criteria?
    - Install ui?
    - Feature gated behind web app installation
      - E.g. permissions could be rejected if not a web app
  - GetInstalledApps()
  - Required manifest_id
  - Launch and uninstall

GIVE US FEEDBACK!

GitHub issues on:

- https://aka.ms/**WebInstall**
- https://aka.ms/**WebInstallAcquisition**

Thanks!

Microsoft Edge

Web Install ×