

Web Install API

peconn@chromium.org

Contents

- Intro
- Our thoughts so far
 - Motivation
 - Concerns
 - Example user flows
 - Open questions
- Discussion



Who am I?

- Peter Conn
- Software Engineer at Google.
- Work on Chrome on Android.
- Part of the Mobile Web Install team.

Me



What's the purpose of this session?

- To see what other browsers think of a Web Install API.
 - Is there a need for it?
 - How would we make it?
 - What are your concerns?
- We have an incredibly rough draft, to help discussion.



What is this API about?

An API that allows a website to request the installation of another website.

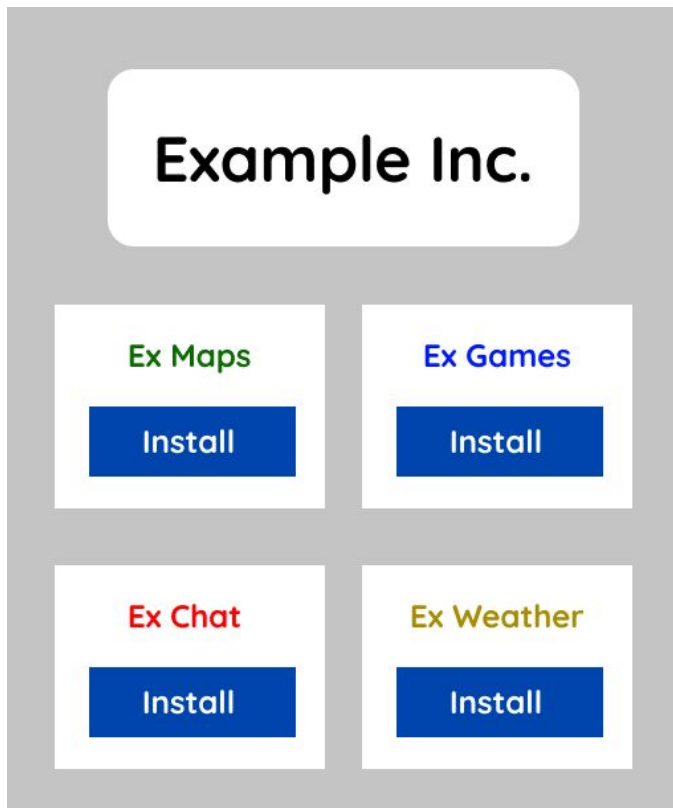


Our thoughts so far

(a summary of [our draft explainer](#))

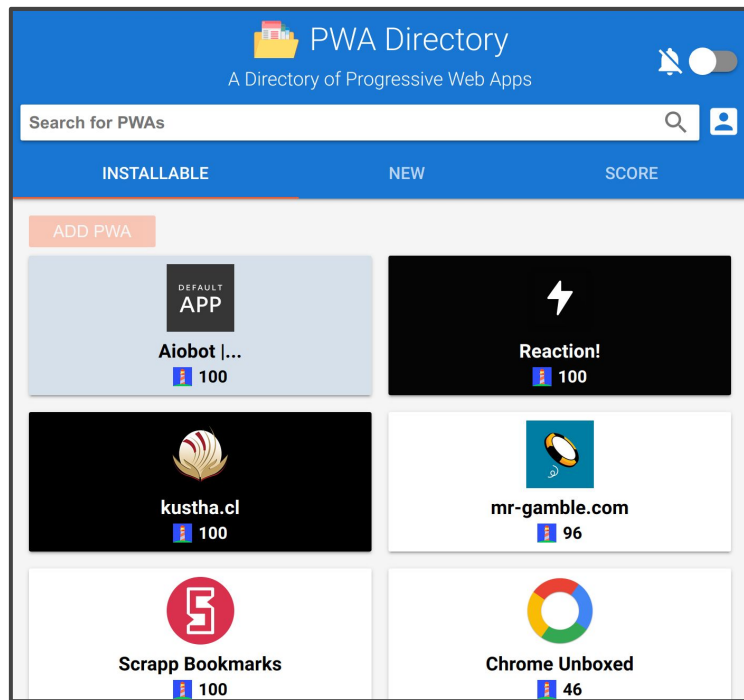
Use cases

- A web app directory, such as [PWA Directory](#), [findPWA](#) or [Appscope](#).
- A unified installation page for a vendor of multiple installable web apps.



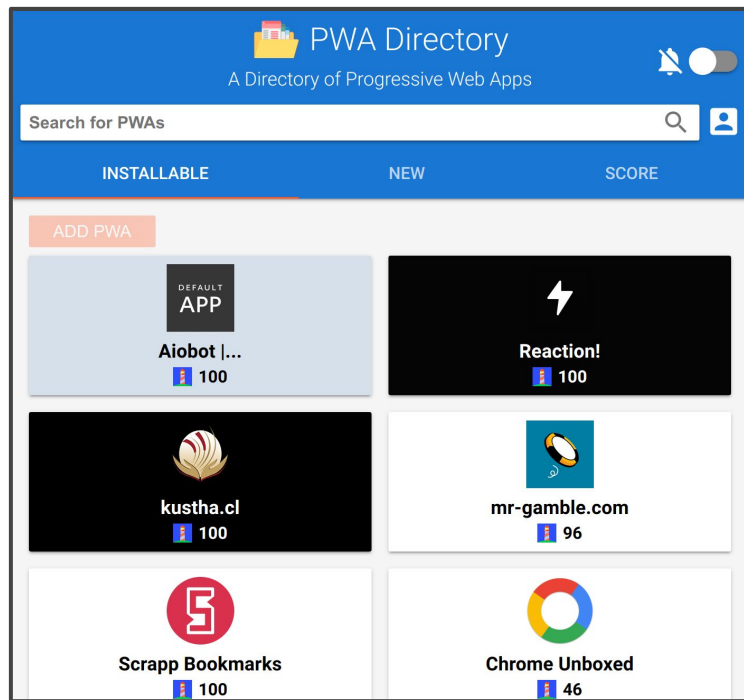
Why app directories?

- Better discoverability strengthens web apps.
- Provide ratings and reviews for users.
- Leads to better monetization for developers.



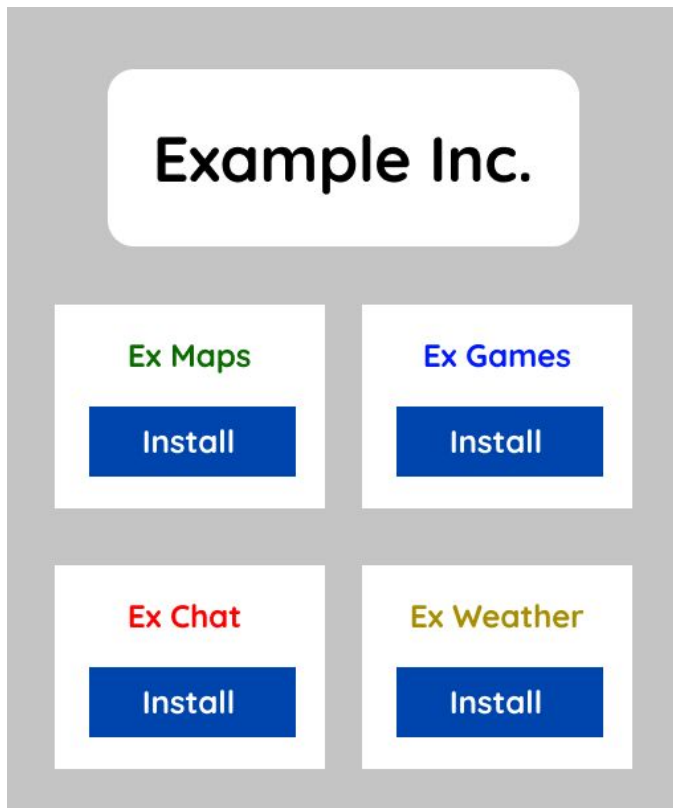
Why app directories?

- Will allow users to choose where to get their installed apps from - democratising installation.
- Users are trained to get their apps from stores - without this the web could lose out.



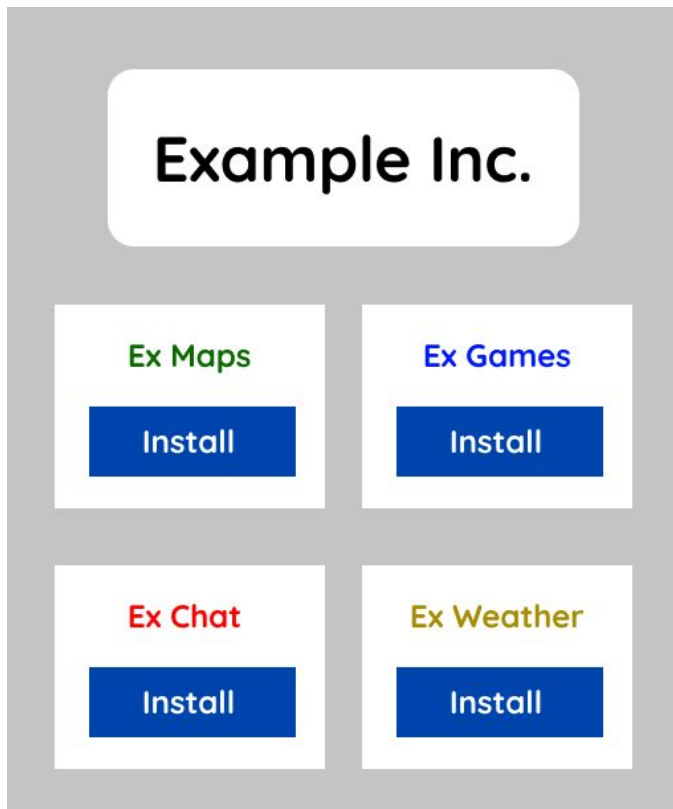
Why unified install pages?

- A place for a single publisher to list a suite of installable apps.
- Easier for users to get the whole suite.
- A place for users to check their entitlements and subscriptions.



Why unified install pages?

- Encourages splitting functionality into separate apps.
- Stronger origin isolation.
- More granular permissions.



Concerns

Concerns

We should protect:

- The **user** from the **catalogue**.
- The **user** from the **target website**.
- The **target website** from the **catalogue**?



Concerns

We should protect:

- The **user** from the **catalogue**.
- The **user** from the **target website**.
- The **target website** from the **catalogue**?

What do we lock this capability behind?

- Website permission?
- Browser prompt?
- User gesture?
- Website trust?



Concerns

We should protect:

- The **user** from the **catalogue**.
- The **user** from the **target website**.
- The **target website** from the **catalogue**?

We want to apply our normal installability criteria.



Concerns

We should protect:

- The **user** from the **catalogue**.
- The **user** from the **target website**.
- The **target website** from the **catalogue**?

Proxx was copied and sold within hours of being uploaded.

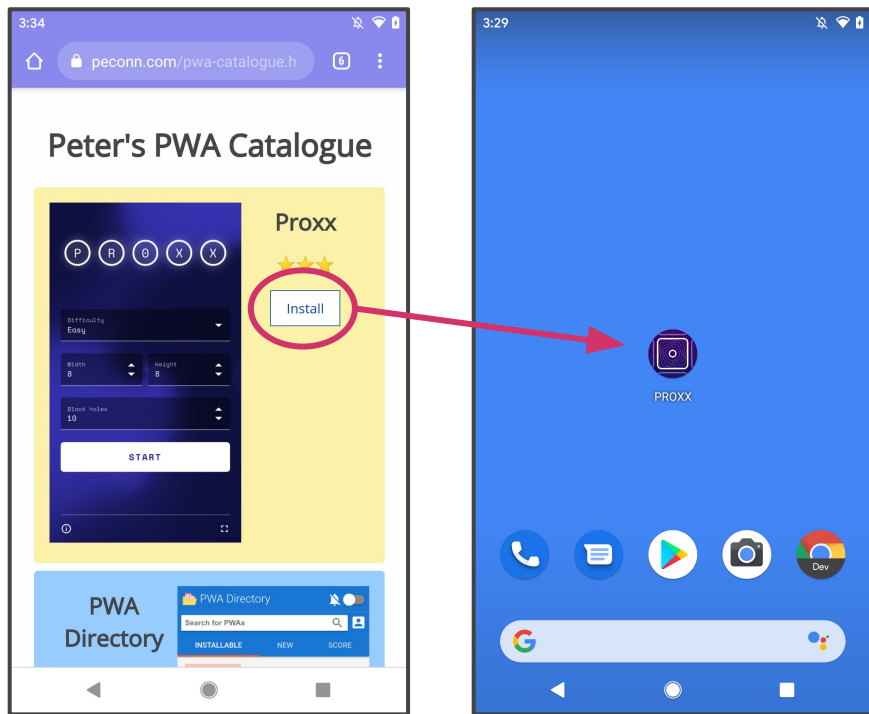
Should a web app have control over who can list/install it?



UX examples

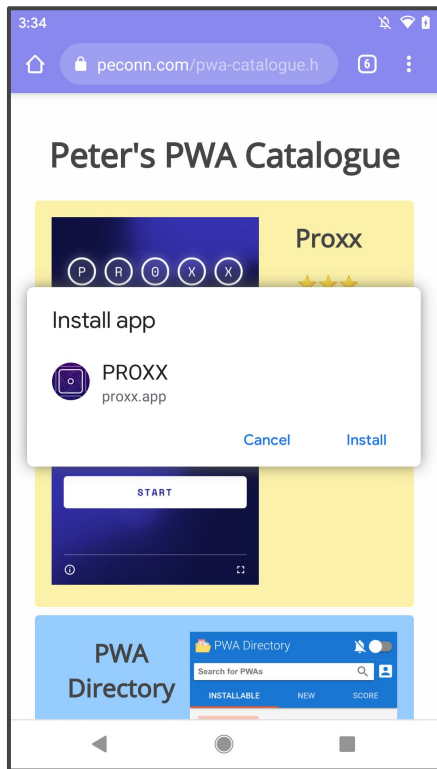
A few ideas for how users could
install an app.

Seamless installation



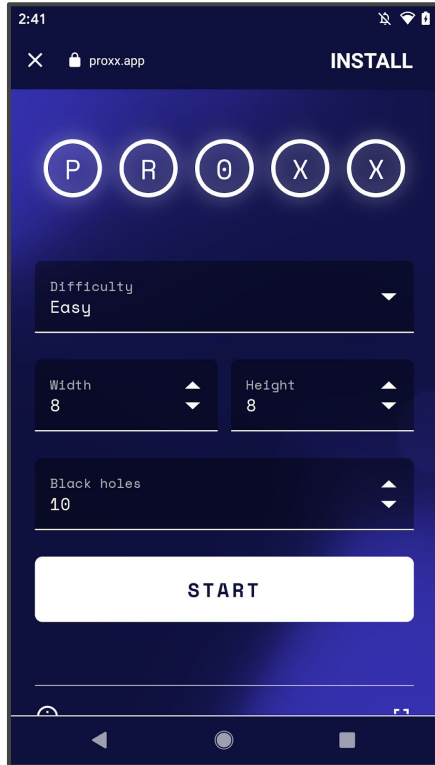
- Easiest to abuse.
- The user doesn't even see the URL of the installed app.

Browser Prompt



- Similar to beforeInstallPrompt.
- Gated behind a user gesture?

Try Before You Buy



- Launch a link in a specialized browser UI.
- Install is featured prominently.
- User can return to the catalogue easily.

Open Questions

Install source attribution

- Should the website know where the installer came from?
- It would help app catalogue monetization.
- Could just use the referrer?

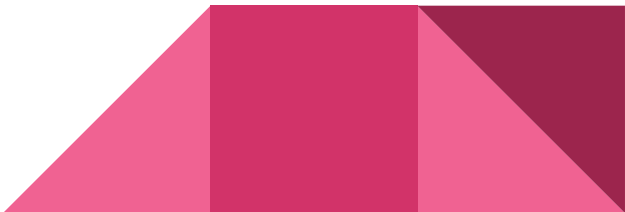


Informing the directory of the outcome

- Should the directory know if the user went through with the installation?
- Could be used as a signal for web app quality.
- Could be used to customize directory UI.



Checking for existing installs

- Should the directory know if a web app is already installed?
 - Allowing a website to see a list of the user's installed apps would allow fingerprinting.
 - Maybe just apps that the directory itself installed?
 - Could be used to customize directory UI.
- 



And now for your thoughts?

Discuss!

- Should we do this?
- Have we missed any concerns?
- Can you think of any alternative user flows?
- What about the open questions?



Thanks for listening and engaging!

- [Link to explainer draft.](#)

