

PWA as URL Handlers - UX Review

luhua@microsoft.com, mandy.chen@microsoft.com

THIS DOCUMENT IS PUBLIC

[crbug/1072058](#)

[Design doc](#)

[Explainer](#)

This feature is similar to other web app manifest features in that it has both developer facing and user facing experiences. It introduces these new user experiences:

1. When a user clicks on a URL link and as a result the Chromium browser is launched by the OS to handle the navigation, an installed web app can be launched instead of a regular browser window (current behavior).
2. An intent picker dialog is displayed with choices of installed web apps that are able to handle the link navigation. The user is able to save their choices.
3. New options in `chrome://settings/content` allow the user to remove saved choices.

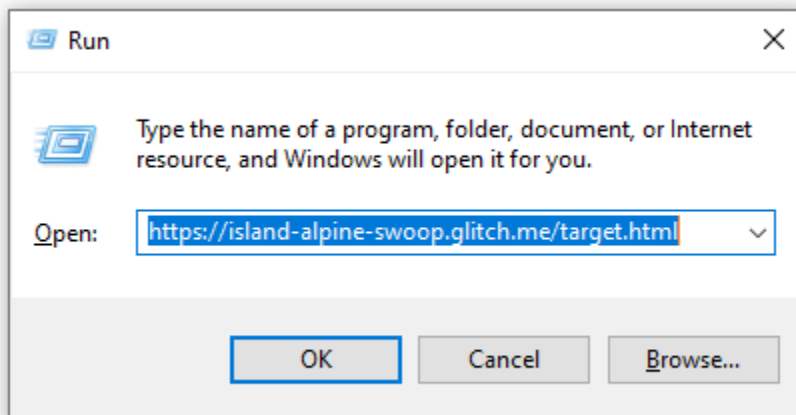
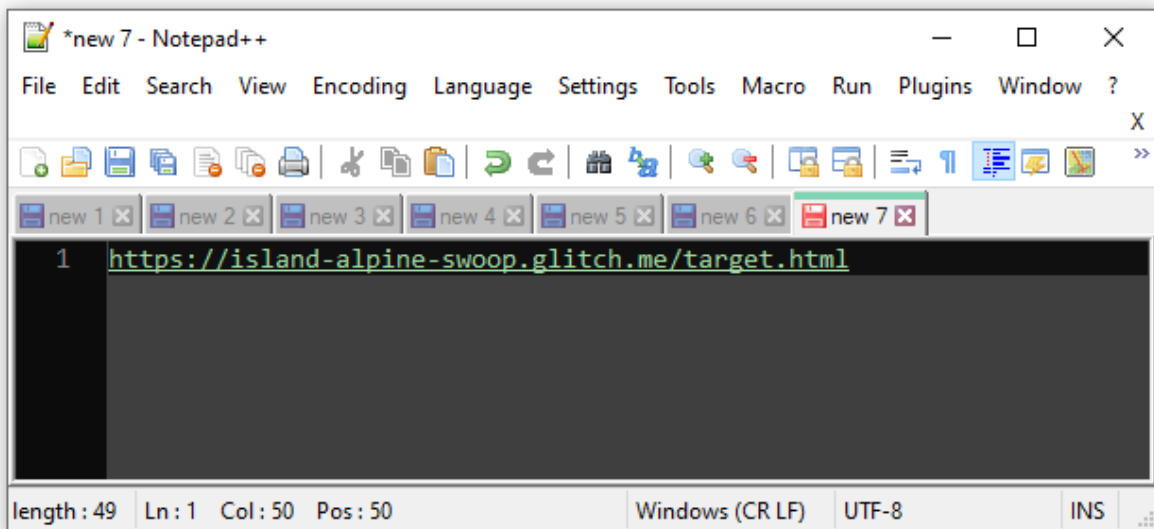
This feature does not currently integrate with OS features and APIs so only browser UI/UX is in scope.

Web App Install, Uninstall, Manifest Update

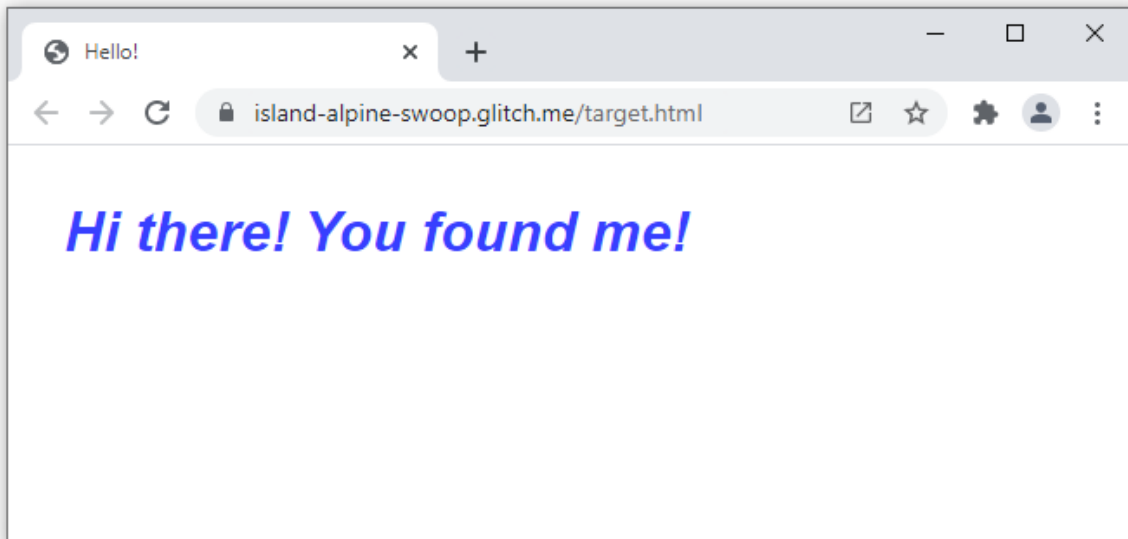
The install, uninstall, and update user experiences are unchanged. There are no additional prompts or options presented to the user during those actions. “url_handlers” information from the web app manifest will be registered silently.

Launch from URL Activation

When activated, URL links in the system are often handled by launching the default browser. This can be tested in a native app with clickable links like Notepad++ or with a `run.exe` dialog in Windows. This can be done from the command line using “`xdg-open [url]`” in Linux, and “`open [URL]`” in MacOS.

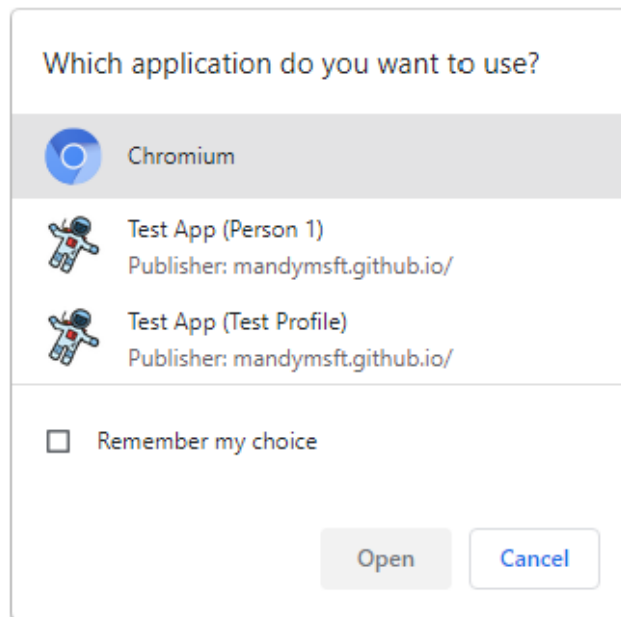


Normally this would cause a new browser window or browser tab to be opened and navigated to the target URL.

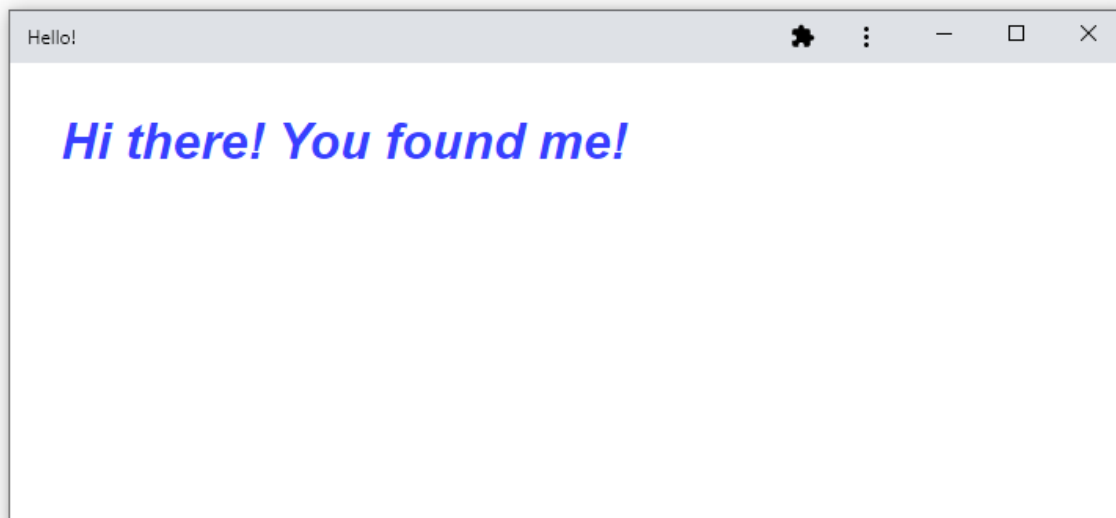


When this feature is enabled, if the link URL matches the “url_handlers” manifest field of an installed web app, the user is prompted to launch the app through the use of an intent picker dialog. This intent dialog is being developed for the [PWA Protocol Handling](#) feature and should undergo UX review in that launch process. See [crbug/1105257](#).

The intent picker dialog is shown every time unless the user has previously launched an app and asked the browser to remember their choice. There should always be a choice to open the URL in the browser. I.e. The browser should appear as an option in the app list, and it should be possible for the user to save this as their choice. The browser option should appear top-most in the list. No option should be pre-selected in the list. The ‘Open’ button should be disabled until a selection is made.



When an app is selected and accepted, the app window will launch and navigate to the target URL. When "Cancel" is selected, the process will exit and nothing will happen.



Note that the intent picker dialog could be shown where there are no other browser windows or UI visible. The browser as the topmost option is the most important clue to the user that this dialog is browser UI.

App client behavior

URL Launches will always open a new app window by default. This prevents loss of current web contents.

This is true even if:

- PWA tabstrip feature is enabled
(chrome://flags/#enable-desktop-pwas-tab-strip-link-capturing)
- There is an existing app window open

This is false if:

- The web app is configured to open in browser window

Future work could apply the manifest's "[capture_links](#)" preference.

Launch to out-of-scope URLs

When handling a URL that is outside of a web app's app scope, the new app window will navigate directly to that URL. This could cause a [CCT UI bar](#) to appear at the top of the web contents.

Future work will allow the launched app to handle out-of-scope URLs in more sophisticated ways (with document events or service worker events).

User Profiles

Web apps can be installed from different user profiles. The user will be able to choose from the same web app installed on 2 or more different user profiles using a suitable UI treatment (under development) in the dialog or be allowed to change profiles in the app window UI.

"Remember My choice"

The intent picker dialog has a "Remember my choice" checkbox. When checked and accepted, all paths that matched the input URL will be marked. The accepted and launched app will be the default app for those paths.

Here is an example:

There are 2 apps installed:

App1: <https://app-1.com/>

- Registered to handle <https://contoso.com> for paths:

- /*

App2: <https://app-2.com/>

- Registered to handle <https://contoso.com/> for paths:
- /A/*

Input URL: <https://contoso.com/index.html>

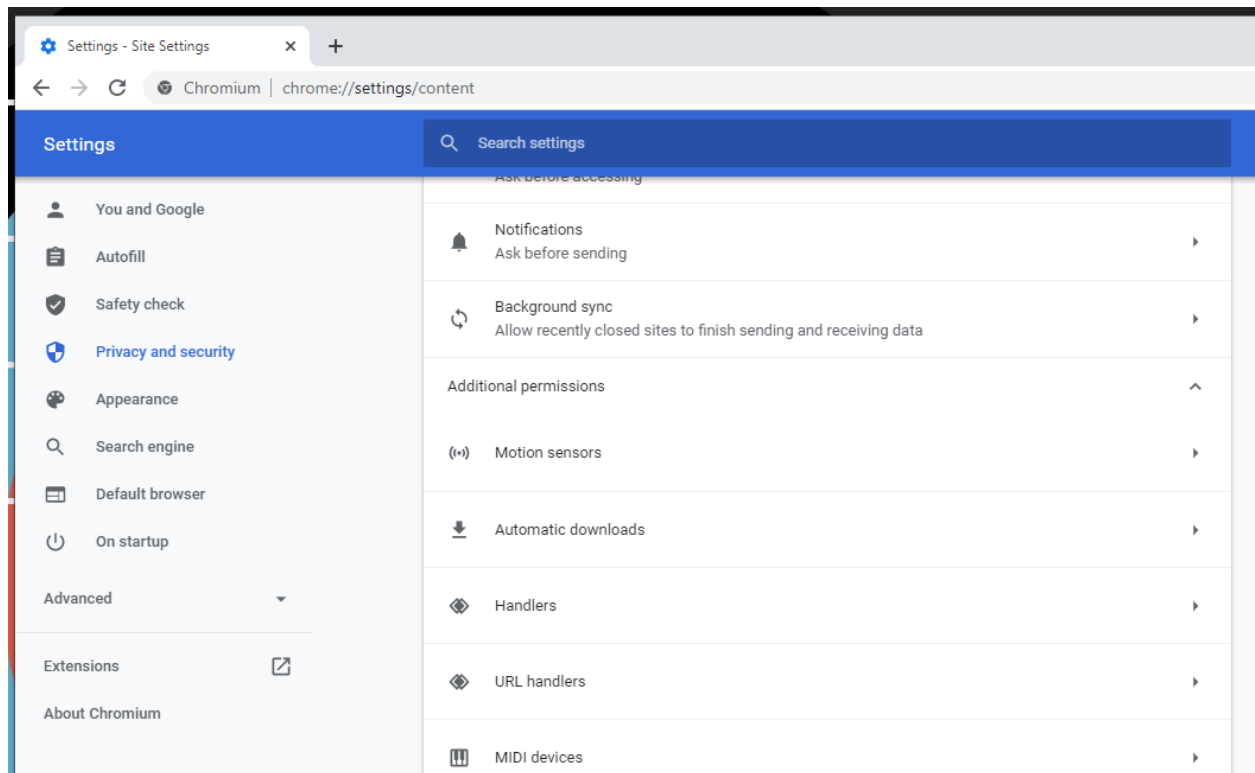
The input matches App1 but not App2. App1 is the only choice in the intent picker. The user checks "Save my choice" and accepts. This causes the /* entry of App1, for <https://app-1.com> to be marked in preferences.

When the same input is seen again, App1 matches again with the /* path. Since this path is marked and is the only marked option, App1 is launched by default with no intent picker dialog.

Input URL: <https://contoso.com/A/index.html>

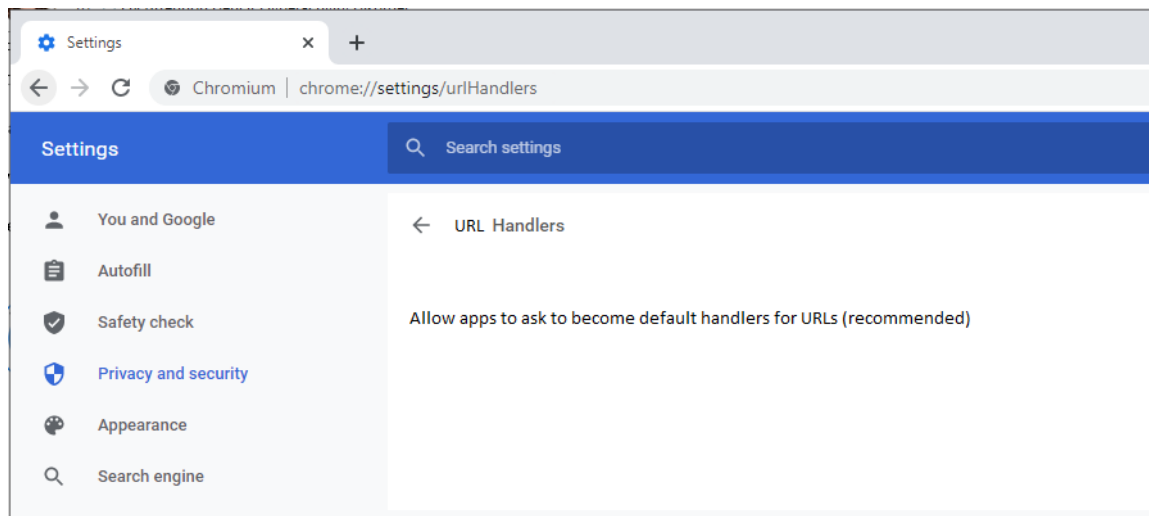
The input matches both App1 and App2. The /* path is marked and is the only marked option, App1 is launched by default even though App2 also matches the input. The user can remove the saved default from the `chrome://settings/urlHandlers` page to access the intent picker dialog again.

User Settings



We will add a “URL Handlers” sub-page to the “Additional permissions” list at “chrome://settings/content”.

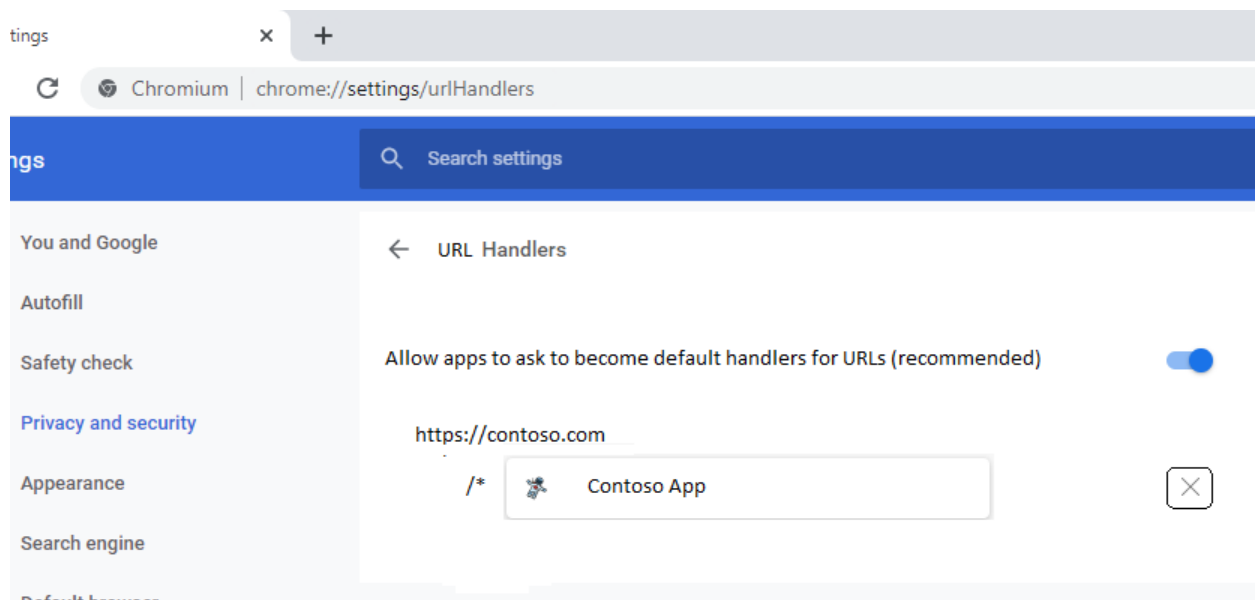
This sub-page will have a main toggle switch the user can use to disable all PWA URL handling behavior. The sub-page URL will be “chrome://settings/urlHandlers”.



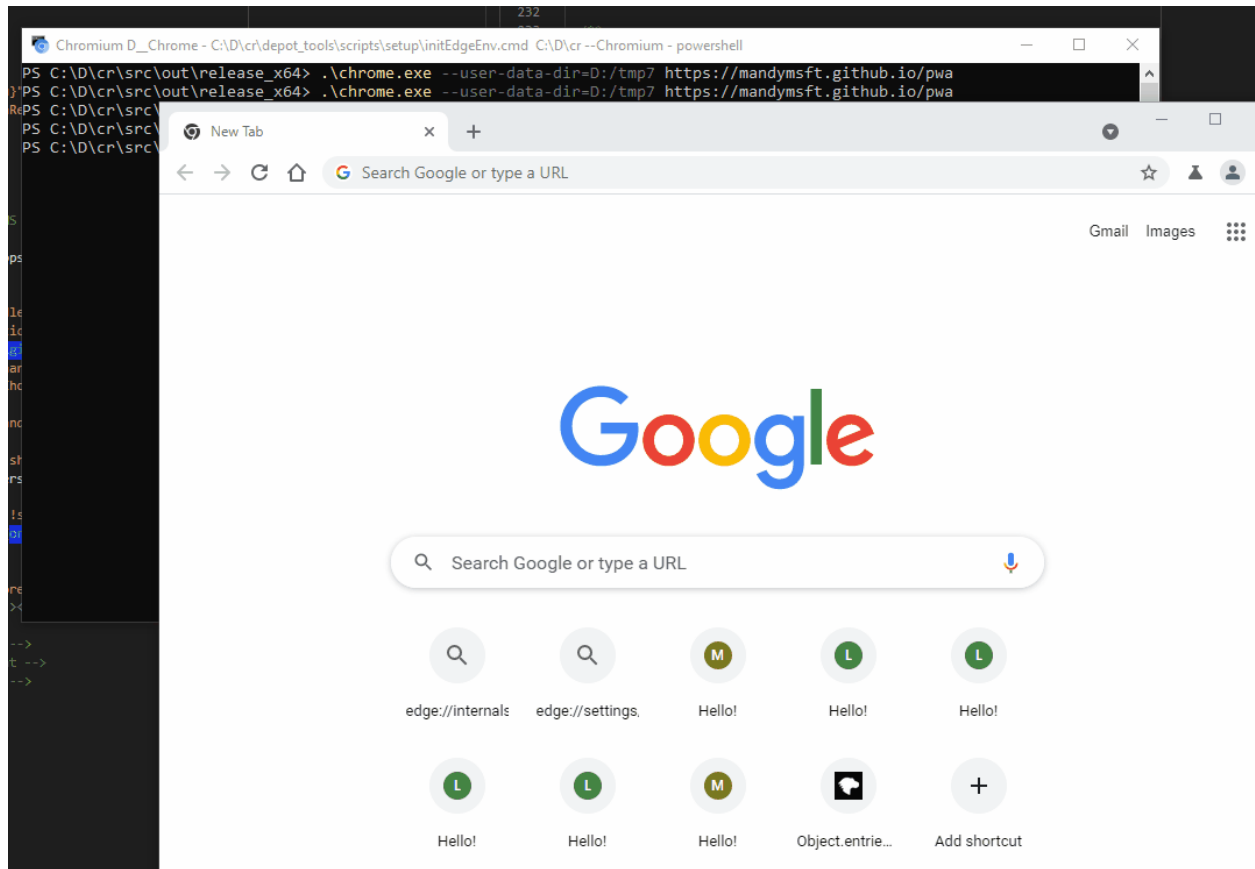
Furthermore we will list groups of saved choices from the “Remember my choice” checkbox, grouped by origins. The user is able to remove these defaults from chrome://settings/urlHandlers or when clearing “Site Settings” from chrome://settings/clearBrowserData. After relevant saved defaults are removed, the intent picker dialog will be presented again so the user will be able to save a different choice as default.

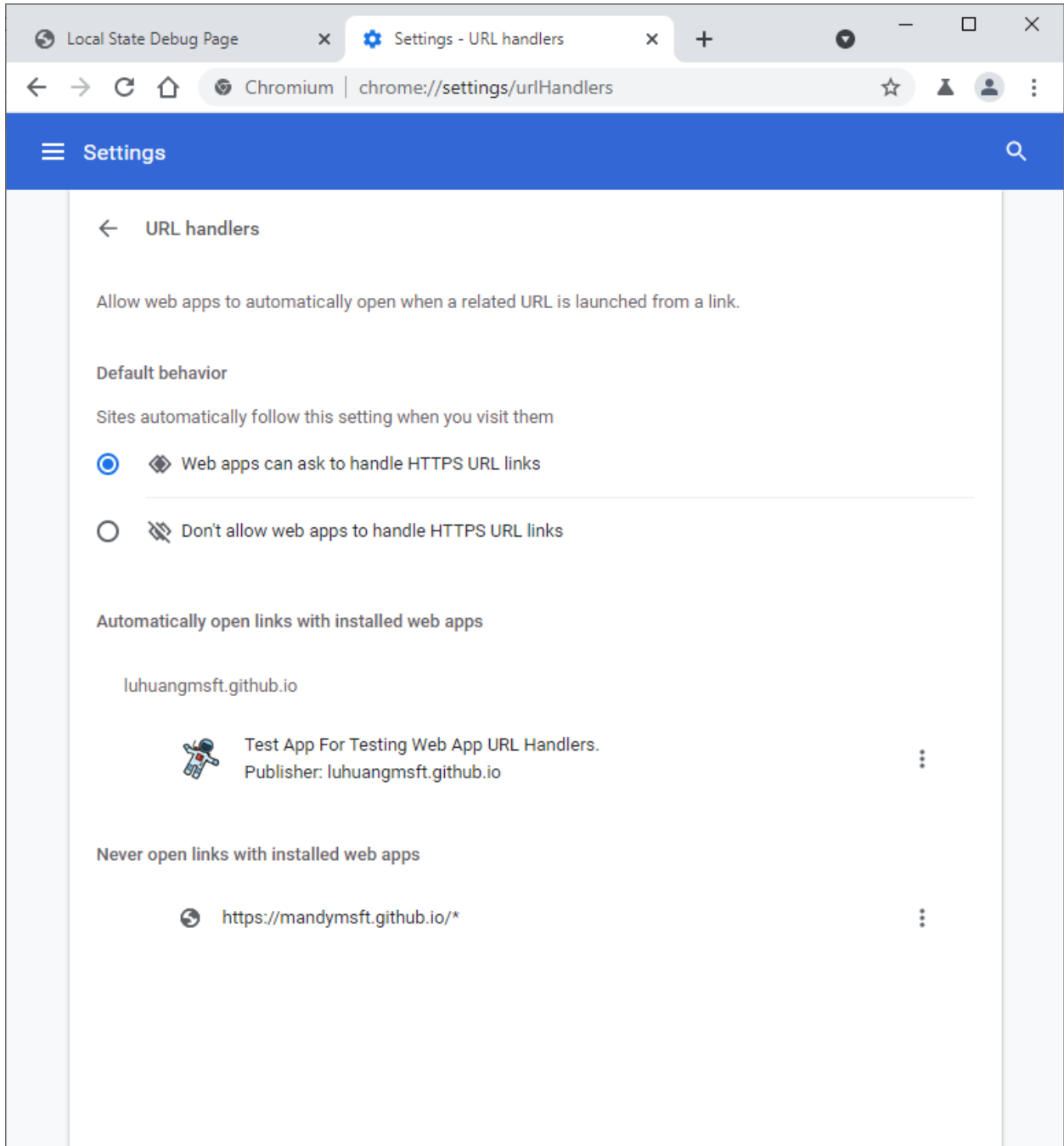
Mock up

(Out of date. see current implementation below.)



Current Implementation





Automatically open links with installed web apps

luhuangmsft.github.io



Test App For Testing Web App URL Handlers.
Publisher: luhuangmsft.github.io

Remove

[Screen capture video of settings page usage](#)

Site Details (Site Permissions)

chrome://settings/content/siteDetails will include a “URL handlers” permission entry for origins that include at least 1 installed web app. This entry will be hidden for origins that do not have any installed web apps.

Chromium D_Chrome - C:\D\cr\depot_tools\scripts\setup\initEdgeEnv.cmd C:\D\cr --Chromium - powershell
PS C:\D\cr\src\out\release_x64>

```
Chromium D_Chrome - C:\D\cr\depot_tools\scripts\setup\initEdgeEnv.cmd C:\D\cr --Chromium - powershell
PS C:\D\cr\src\out\release_x64> .\chrome.exe --user-data-dir=D:/tmp7 https://luhuangmsft.github.io/pwa
PS C:\D\cr\src\out\release_x64> .\chrome.exe --user-data-dir=D:/tmp7 https://luhuangmsft.github.io/pwa
PS C:\D\cr\src\out\release_x64> .\chrome.exe --user-data-dir=D:/tmp7 https://luhuangmsft.github.io/pwa
PS C:\D\cr\src\out\release_x64> .\chrome.exe --user-data-dir=D:/tmp7 https://luhuangmsft.github.io/pwa
PS C:\D\cr\src\out\release_x64> .\chrome.exe --user-data-dir=D:/tmp7 https://luhuangmsft.github.io/pwa
PS C:\D\cr\src\out\release_x64>
```

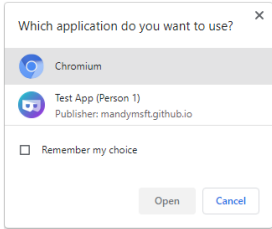
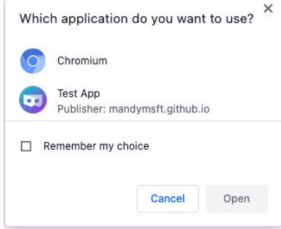
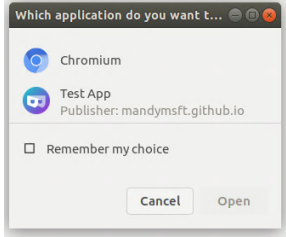
The permission options are [Allow (default), Allow, Block, Manage]. This follows the pattern established by other permissions. Manage is a completely new option not seen with any other previous permission.

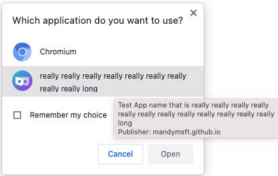
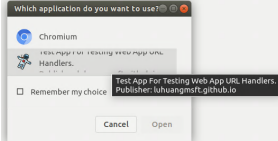
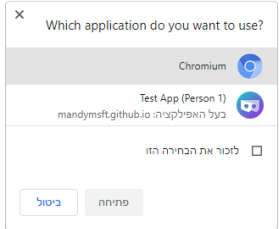
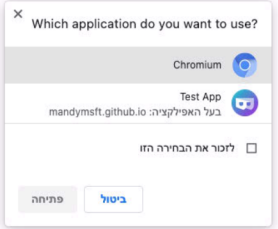
- Allow (default) - this is the default option for all origins with installed web apps.
- Allow - this is the allow option that the user can explicitly choose.
- Block - the user must explicitly set this option to prevent any installed web apps from the origin from asking to handle URL launches
- Manage - this option is not persisted. If the user selects this option in WebUI, the page will navigate to the <chrome://settings/urlHandlers> page

Accessibility

All the new strings are localizable.

	Windows	Mac	Linux
--	---------	-----	-------

Normal			 Dialog is resizable and can show the full title after being resized.
Screen reader	Which application do you want to use? window, Chromium, button - Should each app option be recognized as a "button"? - Should the number of options be announced? E.g. "Chromium, 1 of 2" - Should the option status be announced? E.g. "Selected"	Chromium, Which application do you want to use? window, Chromium, button Same problems with Windows	Not sure how to test
Keyboard nav	"Chromium" is highlighted by default. Tab: "Test App" highlighted Enter: Option selected Tab: Checkbox for "Remember my choice" highlighted Tab: "Open" highlighted Tab: "Cancel" highlighted Tab: x button highlighted	Tab only works for "Chromium" and "Test App" Enter and Escape work When VoiceOver is on, keyboard nav works on all buttons The behavior above is comparable to other dialogs, such as web app install and uninstall dialogs.	Tab: "Chromium" highlighted Enter: Option selected Tab: "Test App" highlighted Tab: Checkbox for "Remember my choice" highlighted Tab: "Cancel" highlighted Tab: "Open" highlighted Escape: Dialog closed

	Escape: Dialog closed		
Long strings	 • Cut off or have it scroll vertically		
RTL			Unsure how to test. --lang=he didn't work.

Questions

- Dialog is not modal, but it doesn't show input URL to open. There can be multiple dialogs which would not be distinguishable. Should we present the URL in the dialog?
- Are there guidelines for dialog size and placement? Can it be resizable?

Meeting Notes

2021-7-6:

<https://docs.google.com/document/d/1BajvM7oUUXHBDTGOGerRG9DEOAXAc5XKoOXf0rYxhUU/edit?usp=sharing>