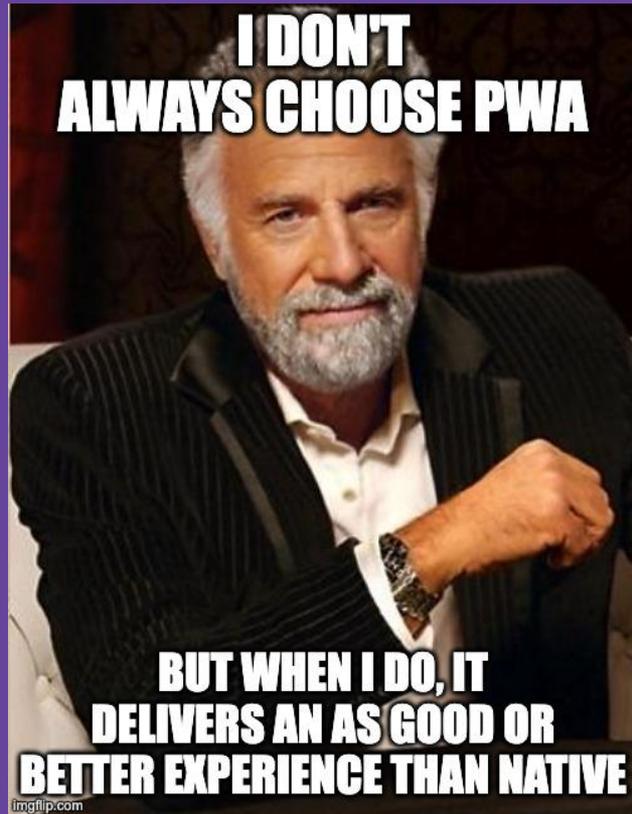




Cross-Platform Apps Powered by Drupal for Everyone

Alex Borsody - Moonraft a UST Company



Obligatory meme from 2010

Disclaimer

- For people familiar with Drupal and working with Drupal's theme layer (there are a lot) for those looking to get the most out of their coupled architecture and also for those who are interested in learning more about decoupled (off the Drupal island)
- Focused on technologies running in the **web browser** Chromium/Webkit (Chrome/Safari).
- PWA is different technologies that come together focused on the web browser to create native app experience.
- This particular talk is from an architect's perspective, choosing and implementing various technologies that come together to make a great PWA.
- This is a survey, any of the slides could be their own talk, there will be code samples and repositories provided as well as a companion blog post.

Where are we going?

- PWA module as a starting point.
- PWA Builder and getting your into the app stores.
- Choosing decoupled or standard Drupal theme layer.
- Everything you need to know about making your PWA truly feel like native app. (which no one will talk about right now but Google)



What is a PWA? What is not a PWA?

PWA - Progressive Web App

Essentially an effort led by Google and Microsoft to make the web browser a runtime environment on par with a native app experience. Truly write code once for Android and iOS, plus get an optimized website for performance and SEO.

By Definition

- Service worker: Offline functionality
- Web manifest: Add to home screen and icons.
- Security: Must use https (service worker and session cookies)

PWAs Can Also

- Be discoverable in app stores via TWA or custom wrapper for iOS
- Have access to native device features
whatwebcando.today
- Web technologies “just work”

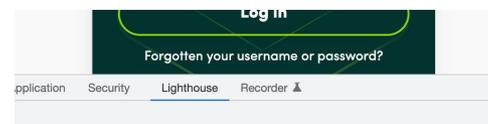
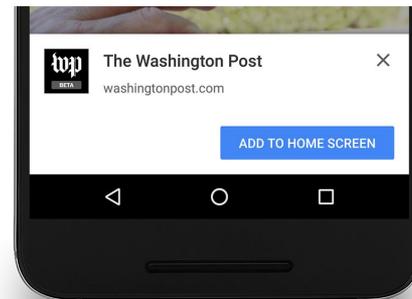
Cross-Platform-App

Cross platform App is a general term that means the app can run on multiple platforms from a single codebase and so could be a PWA or any of these other technologies.

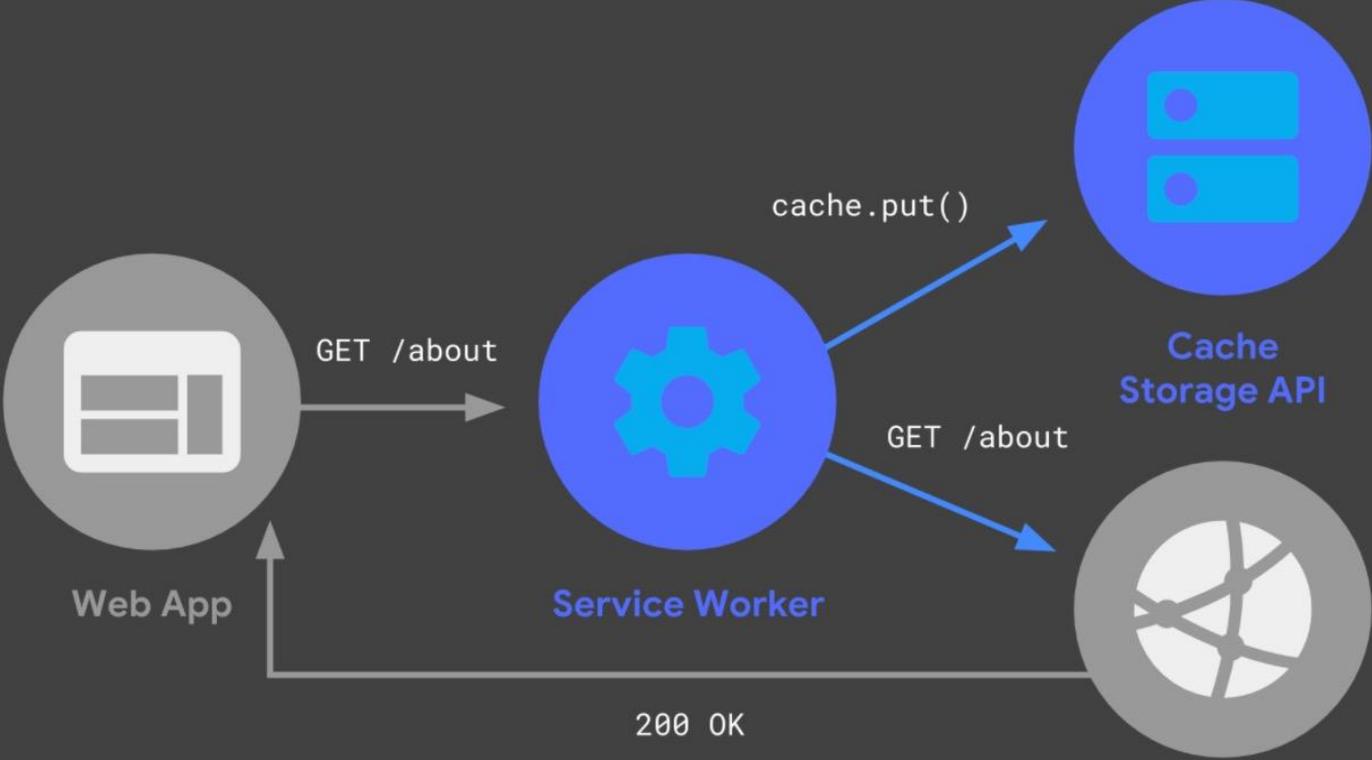
- React Native
- Flutter
- Ionic/Cordova/Phonegap
- In general more technologies involved beyond those familiar with web developers, more labor intensive. Potentially more margin for error.

Drupal PWA Module

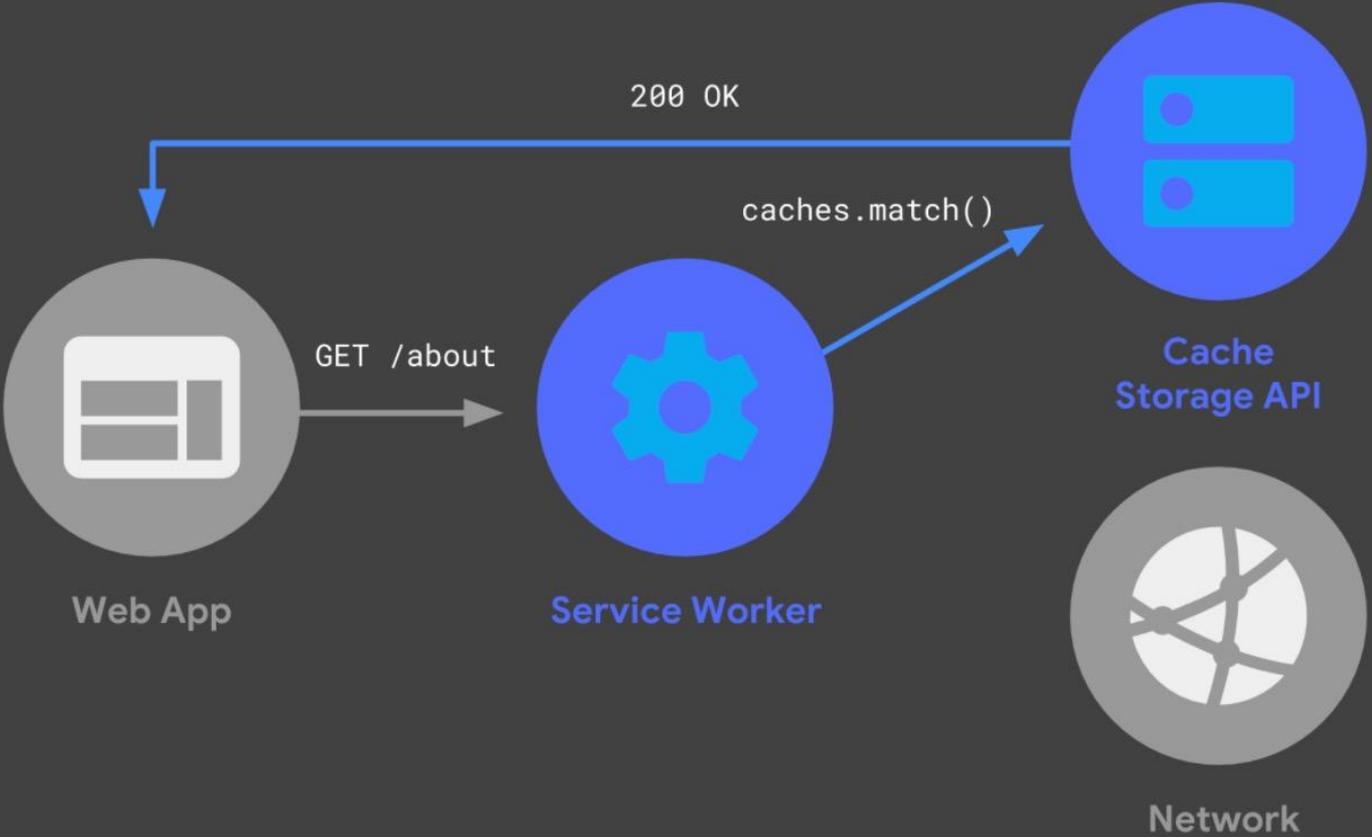
- Offline caching provided by a service worker.
- Manifest + PWA Extras
- Improve lighthouse score, performance and SEO.



Offline caching example in service worker



First Visit



Second Visit

Choose File No file chosen

This image is your application icon. (png files only, format: (512x512))

Current Image:



SERVICE WORKER

URLs to cache on install

```
/offline
```

These will serve the page offline even if they have not been visited, try to limit the amount of URLs here so the user is not downloading too much on their first visit. Make sure the URL is not a 404. Make sure these are relative URLs tokens not supported.

URLs to exclude

```
admin/*  
user/*
```

Takes a regex, these URLs will use network-only, default config should be, admin/* and user/reset/*.

Cache version

1

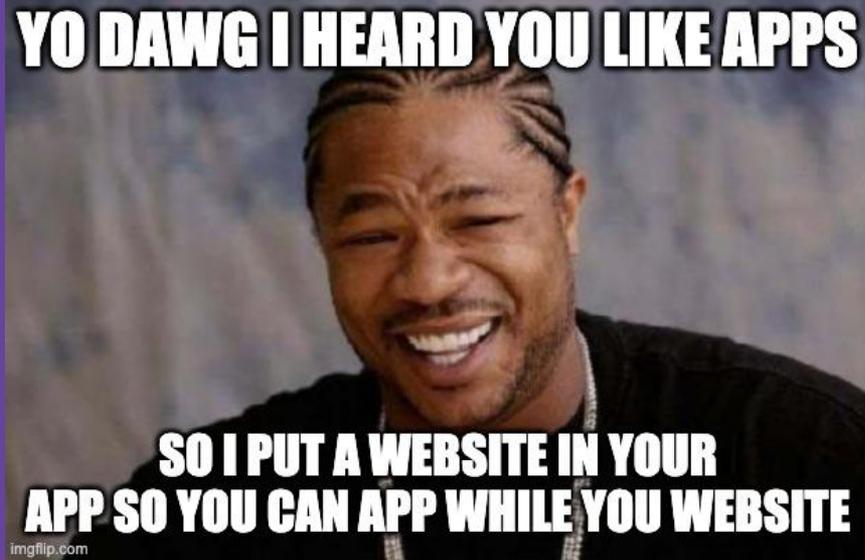
Changing this number will invalidate all Service Worker caches. Use it when assets have significantly changed or if you want to force a cache refresh for all clients.

Workbox + PWA Module

Effort made to port the D8 module to Workbox (now done in D7)

- **Precaching** - Using precache manifest
- **Runtime caching strategies** - cache first, network first, stale-while-revalidate
- **Request routing** - Different strategies for different assets and URLs (cache first for images)
- **Background sync** - Post request fails come back online they retry

Cool... but I want to be
discoverable in the app stores



TWA, WKWebView and PWA Builder

- The PWA module provides everything you need to run through PWA Builder which will...
 - Create a lightweight .apk/.aap to submit to the Play Store 800kb
 - Wrap your website in WKWebView to submit to the App Store
 - Provides you the original source code to make modifications if you want

[Demo](#)

Google TWA

in Chrome Browser

Property	Webview	TWA
Native development effort	High	Low
Mixed Web & Native components	Yes	No
Web/Native bridge	Yes	Limited
Full screen	Yes	Yes
Safe browsing	No	Yes
Data saver	No	Yes
Form auto-fill	No	Yes
Complete Chrome APIs	No	Yes
Native Chrome Performance	No	Yes



iOS Wrapper - WebKit - WKWebView

Functionality possible in custom WKWebView wrapper (GitHub repo to be released)

Mindful of [Apple minimal functionality guidelines](#)

- i. No bounce
- ii. Setting cookie from start_url
- iii. Adding biometric to WKWebView (check URL and don't show on anonymous pages user/login or user/password)
- iv. Adding push to WKWebView iOS workaround with cookies.
- v. [Associated domain](#) to help with login
- vi. Cookie expiration GC and Auto Logout module, extend **session.gc_maxlifetime** and **session.cookie_lifetime**

A close-up image of Morpheus from the movie The Matrix, wearing his signature sunglasses. The image is used as a background for a meme. The text is overlaid in white, bold, sans-serif font with a black drop shadow.

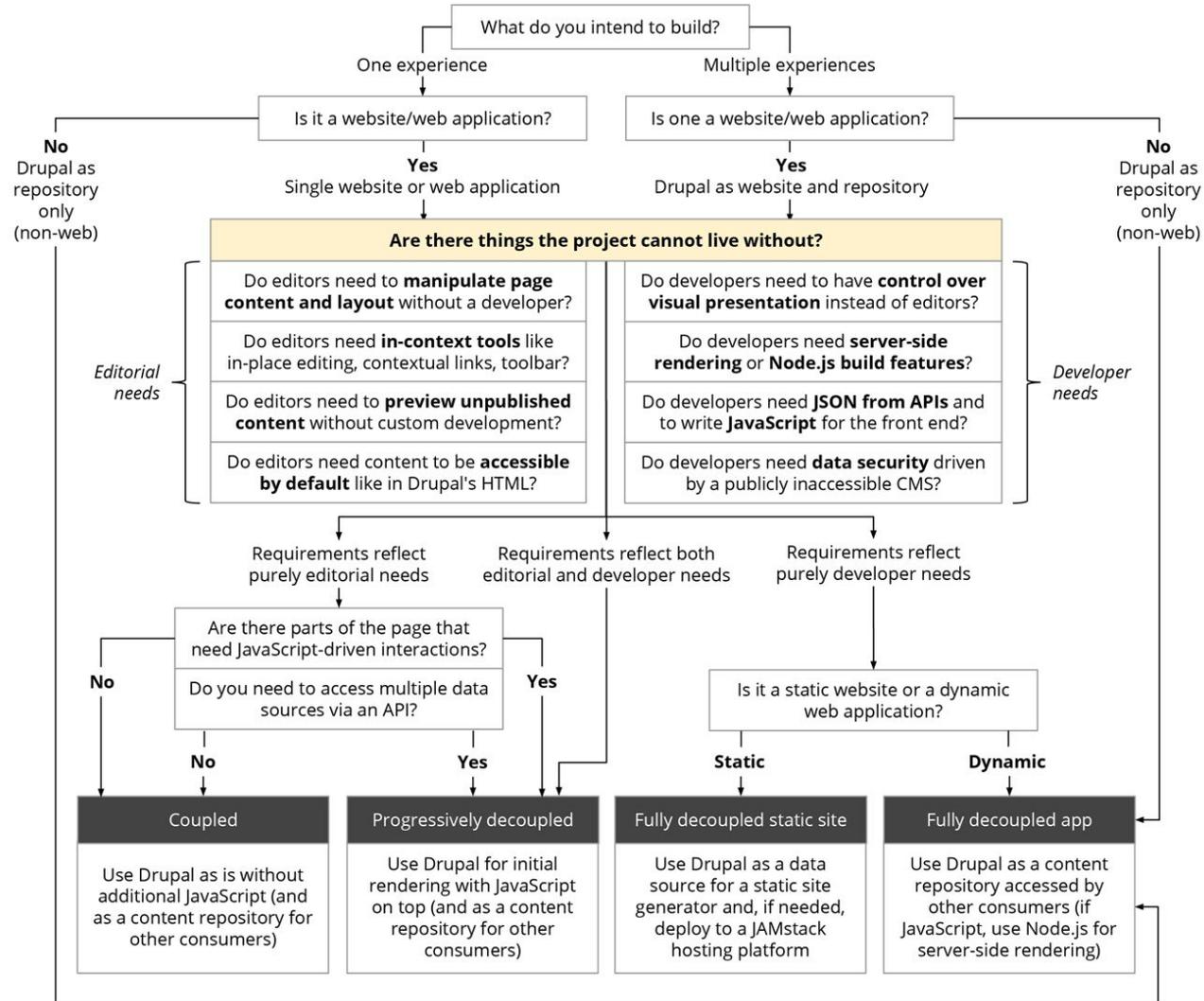
WHAT IF I TOLD YOU

**THAT APP IS JUST A WEB BROWSER IN FULL SCREEN
YOU CAN ADD TO HOMESCREEN WITH A BRANDED OFFLINE PAGE**

imgflip.com

Making your Drupal website “App like”

Decoupled decision making flowchart



- A fully Decoupled SPA architecture when done correctly can improve app-like look and feel through modern UI/UX enhancements and lack of page refreshes to load content - only when done correctly though is the JSON rendering actually faster than the coupled Drupal theme layer rendering + service worker approach. Implementation can take more resources and combination of skill-sets.
- Fully decoupled can limit Drupal's capabilities such as, layout and display management, content previews, user interface (UI) localisation, form display, accessibility, authentication, security features such as XSS & CSRF and performance.
- Dries' flowchart illustrates the decision making process.

Use all web technologies available to feel “app-like”

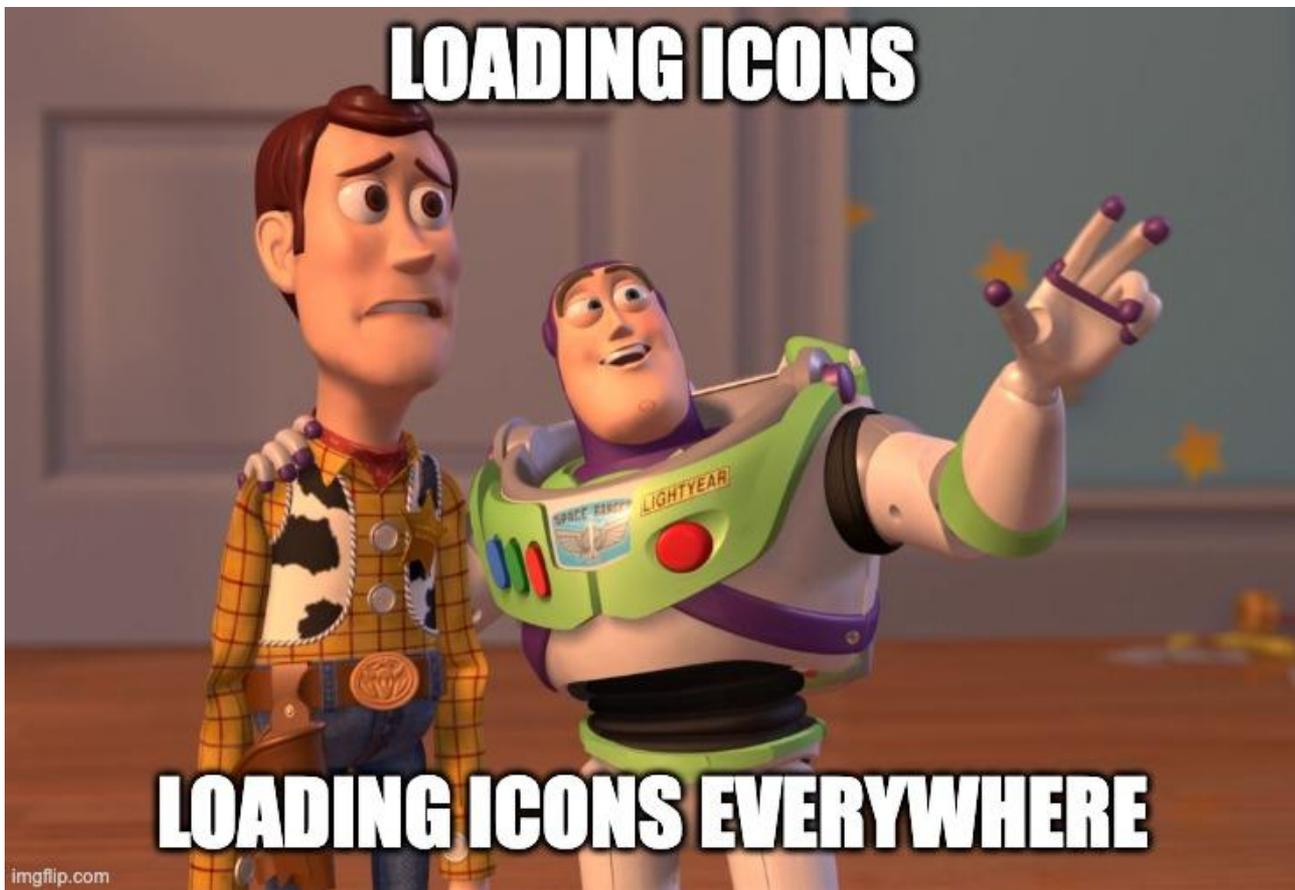
- UX and frontend dev experience is extremely important to making your website feel like an app and is accessible to all *web* developers.
- A lot can be done while only using Drupal’s theme layer to make your web project App-like.
- Keeping mobile/responsive/app-like CSS/HTML/JS updated and compatible with all browsers research but once adjustments are made, changes really only need to happen occasionally when a major version of Safari releases or a new iPhone screen size comes out.
- This is about taking Google’s work on PWAs one step further, considering and combining all web technologies available to make the web look and feel app-like.
- The Chrome team is constantly improving the browser experience, other browsers are following suit. [Project Fugu](#) is the overarching web capabilities project.

CSS, HTML, JS and UX Considerations

- Javascript touch events: disable pinch zoom and swipe gestures
- Viewport meta tag
- CSS:
 - Minify/optimize CSS don't have flashes of unstyled content, Lighthouse
 - "App-like" input/font sizes and making sure everything fits in viewport make it *visually look* like an app
- Preloaders everywhere!
- HTML attributes
 - autocomplete="username"
 - autocomplete="current-password"
 - autocomplete="one-time-code" WebOTP API
 - input type="tel"
- iPhone specific
 - Status bar on iPhone X plus safe area
 - Meta tags and pwa_extras submodule
- Ajax API (Drupal specific)



LOADING ICONS



LOADING ICONS EVERYWHERE