

# Progressive Web ~~Apps~~ Maps (PWA)

Malcolm Meyer

GIS Specialist | [City of Zanesville](#)

# Introductions

Your GIS origin story in two minutes or less...

# Goals

- ☐ Quick Overview of PWAs
- ☐ Create a Basic Web Map
- ☐ Turn this Map into a PWA
- ☐ Make the PWA Installable
- ☐ Allow the PWA to be used Offline
- ☐ Host & Install the PWA

# Time Allowing

- ☐ Add a Vector Tiles Basemap
- ☐ Add Overlay Layers

# What is a PWA

*Set of components to allow a website to behave more like a native application*

# Benefits

Native App Behaviors

Custom App Colors

Caching & Offline Support

\*Push Notifications

# Multi-Platform

Android

Chrome

Chromebooks

iOS

Windows Store



A community-driven list of stats and news related to Progressive Web Apps  
brought to you by Cloud Four

After launching its PWA, **Thomas Kent** lowered its bounce rate by **57%** and increased the revenue from organic traffic by **79%**.



#E-commerce #Revenue #Performance



**Butcher of Blue**'s PWA increased mobile users by **154%**, monthly active users by **154%**, and conversion by **169%**. Pages load **85%** faster.



#Conversions #E-commerce #Fashion #Performance



**Kubota** launched their e-commerce PWA which



Fashion brand **Rooted Objects** saw an increase in conversions of **162%** after launching their PWA.





# Petlove PWA

## Progressive Web App

These checks validate the aspects of a Progressive Web App. [Learn more](#).



### Fast and reliable

1	Page load is fast enough on mobile networks	✓	▼
2	Current page responds with a 200 when offline	✓	▼
3	start_url responds with a 200 when offline	✓	▼



### Installable

4	Uses HTTPS	✓	▼
5	Registers a service worker that controls page and start_url	✓	▼
6	Web app manifest meets the installability requirements	✓	▼



### PWA Optimized

7	Redirects HTTP traffic to HTTPS	✓	▼
8	Configured for a custom splash screen	✓	▼
9	Sets an address-bar theme color	✓	▼
10	Content is sized correctly for the viewport	✓	▼
11	Has a <meta name="viewport"> tag with width or initial-scale	✓	▼
12	Contains some content when JavaScript is not available	✓	▼



### Additional items to manually check

3 audits ▼

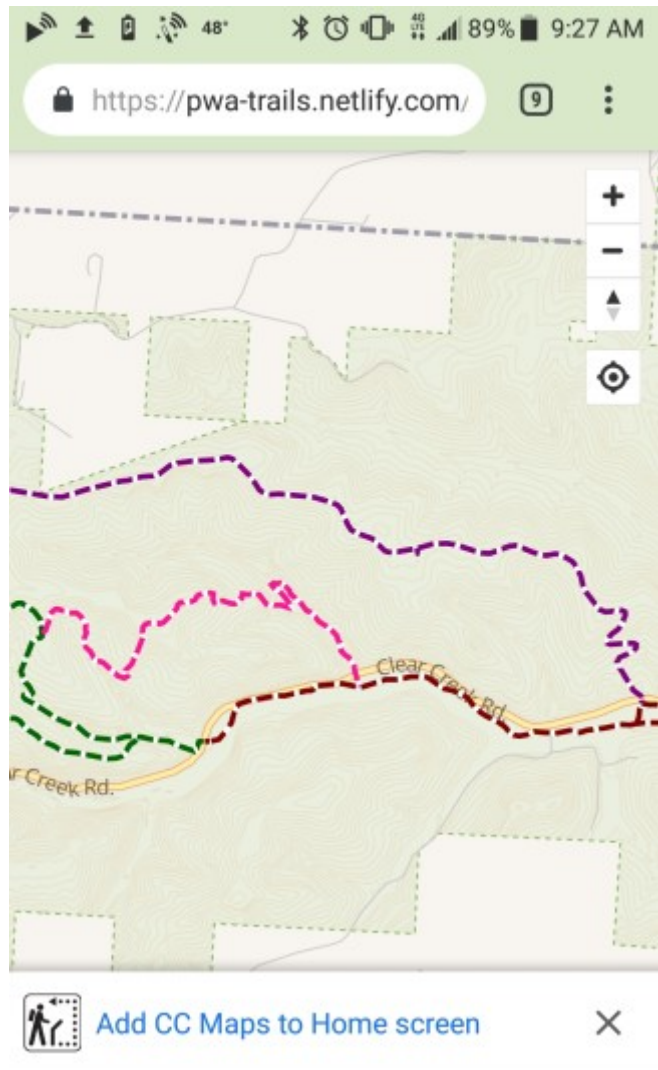
# Why a PWA Map?

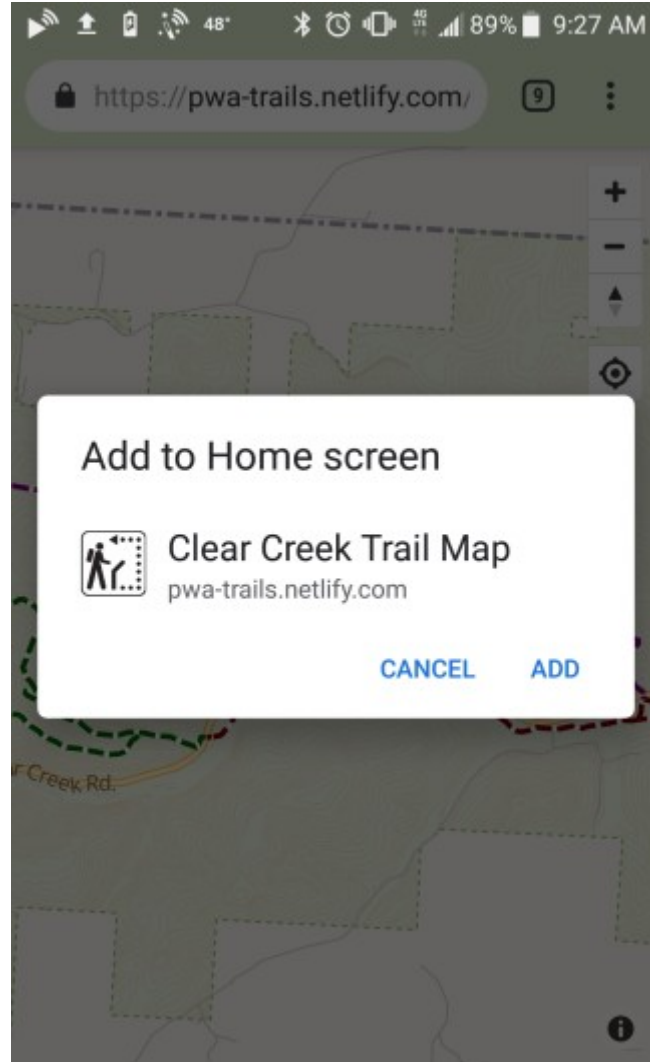
- Cached Assets for Landing Pages & Map UI
- Offline Maps for Field Use - No App Required
- User Convenience - App Drawer/Homescreen

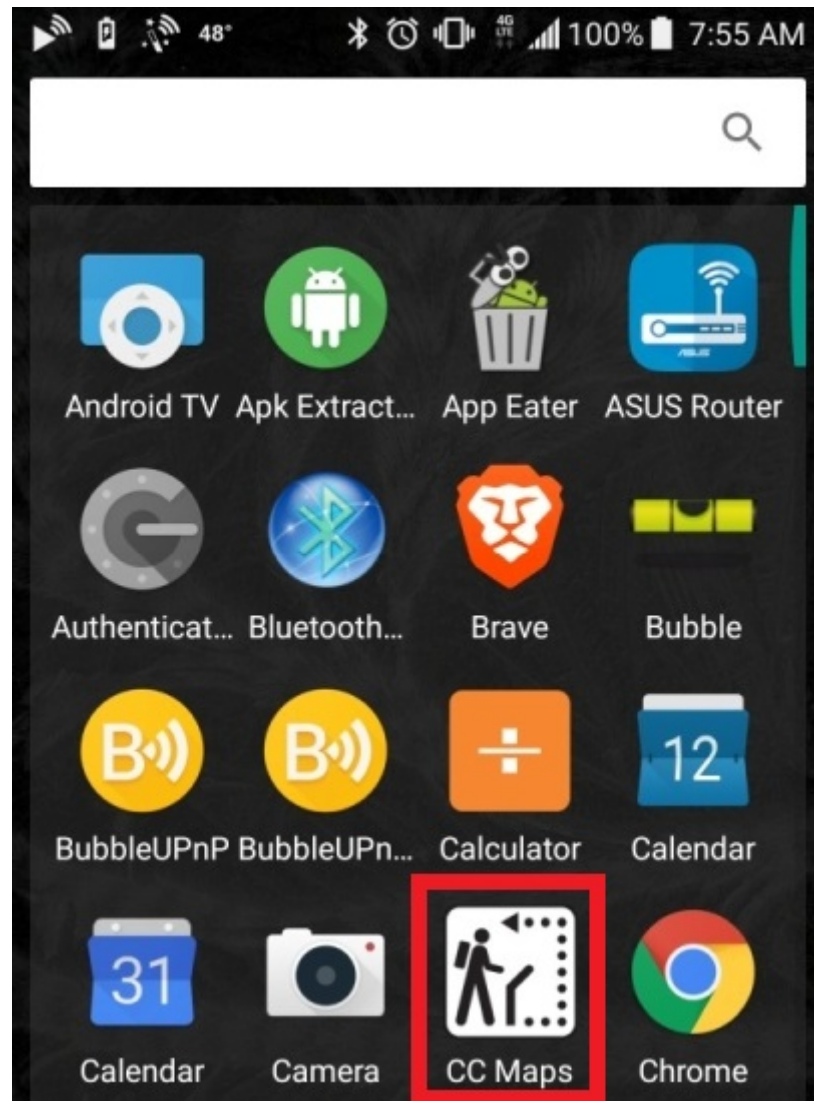
# Example PWA Map

<https://pwa-trails.netlify.com>

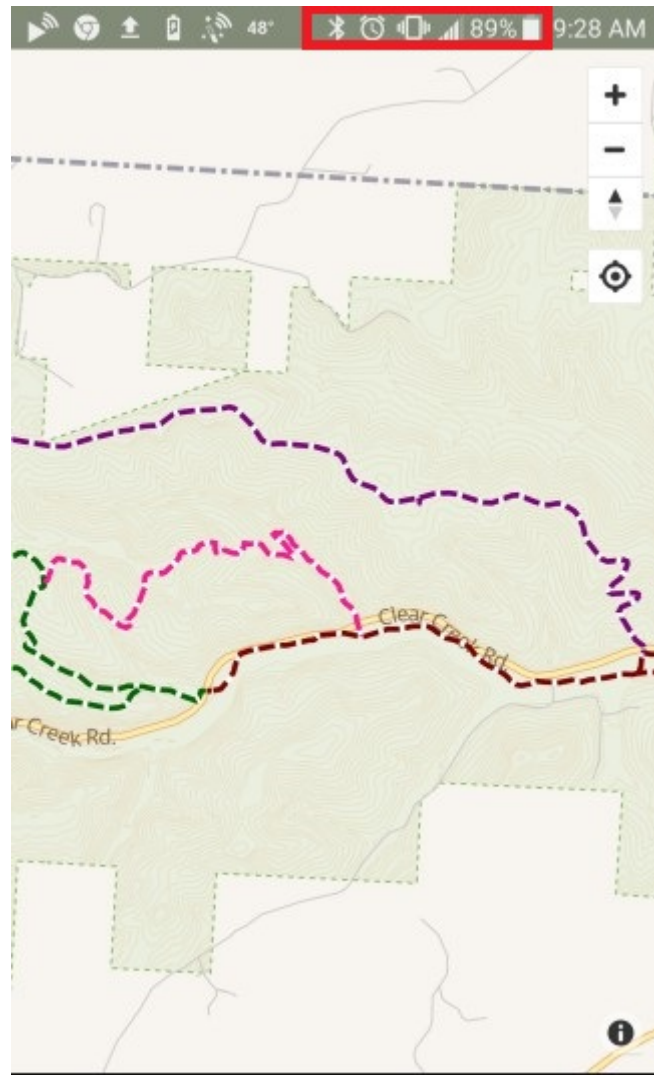




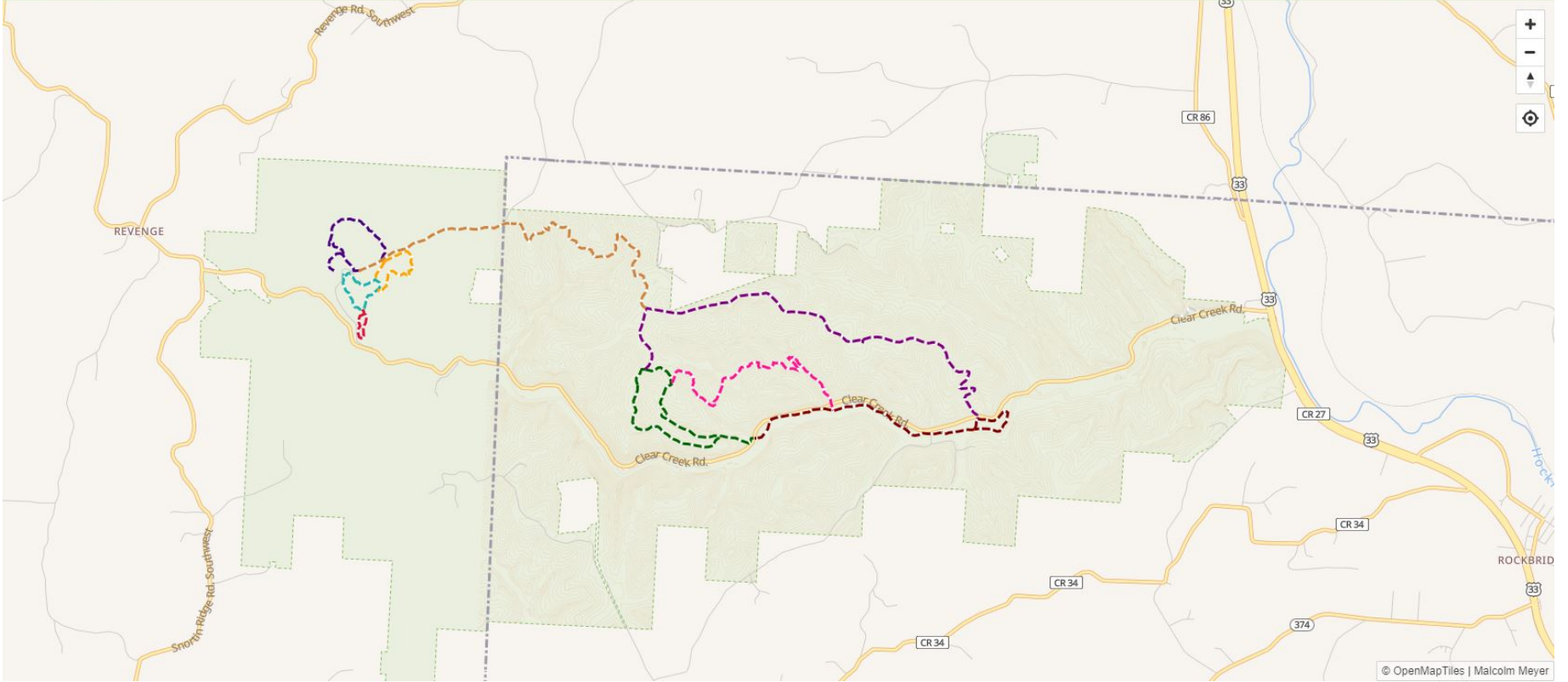




New tab	Ctrl+T
New window	Ctrl+N
New incognito window	Ctrl+Shift+N
History	▶
Downloads	Ctrl+J
Bookmarks	▶
Zoom	- 100% +
Print...	Ctrl+P
Cast...	
Find...	Ctrl+F
Install Clear Creek Trail Map...	
More tools	▶
Edit	Cut Copy Paste
Settings	
Help	▶
Exit	







# Code Break

# Step 1. Install

```
/*  
https://github.com/reyentm/pwa-maps/archive/master.zip  
unzip and then open this folder in with VS Code  
CTRL + '`' to open the terminal  
*/  
npm install  
npm run build  
npm start
```



# Step 2. Test

//CTRL + SHIFT + i then Audit Tab

- Progressive Web App Audit in Chrome
- Check 'Offline' in the Application Tab

# Errors!!

- No manifest
- No service worker
- App does not work offline

# Goals

- ☒ Quick Overview of PWAs
- ☒ Create a Basic Web Map
- ☐ Turn this Map into a PWA
- ☐ Make the PWA Installable
- ☐ Allow the PWA to be used Offline
- ☐ Host & Install the PWA

# Let's Look at the Elements of a PWA

- `manifest.json`
- `service-worker.js`
- Mobile First Design
- Progressive Enhancement



# manifest.json

```
{  
  "name": "Clear Creek Trail Map",  
  "short_name": "CC Maps",  
  "scope": "/",  
  "start_url": "/",  
  "icons": [  
    {  
      "src": "/img/trails512.png",  
      "type": "image/png",  
      "sizes": "512x512"  
    }  
  ],  
  "background_color": "#d8e8c8",  
  "theme_color": "#d8e8c8",  
  "display": "standalone"
```

# Test Again

- Passes manifest tests
- No service-worker
- No offline support

# Service workers

*...intercept and handle network requests, including programmatically managing a cache of responses.*

# sw.js

```
/*
```

Create an empty sw.js file in the root of 'public'.  
In your index.html you will already see the following  
code in the head of the document.

```
*/
```

```
<!--register the service worker-->
```

```
if ('serviceWorker' in navigator) {  
  navigator.serviceWorker  
    .register('/sw.js')  
    .then(function () {  
      console.log("Service Worker Registered");  
    });  
}
```

# Success!

- Installable
- PWA optimized

# Success!

- Installable
- PWA optimized
- Still no offline support

# Success!

- Installable
- PWA optimized
- Still no offline support
- No service worker caching

# Adding Offline Cache

<https://css-tricks.com/serviceworker-for-offline/>

```
npm run cache
/*
this will simply copy a prepared sw.js file to the public folder
*/
```



# Examining the Events

```
/* the cache is added after the install event */
6 self.addEventListener("install", function(event) {

/* requests are fulfilled with the cache, then the network */
38 self.addEventListener("fetch", function(event) {

/* when a new sw.js is activated the cache is refreshed */
141 self.addEventListener("activate", function(event) {
```

# Service Worker Test

- Installable
- PWA optimized
- Offline support

# Service Worker Test

- Installable
- PWA optimized
- Offline support
- But that's not much of a map...

# Service Worker Test

- Installable
- PWA optimized
- Offline support
- But that's not much of a map...
- And that's a lot of code

# Workbox

Automate the creation of the service worker

```
npm install workbox-cli # this is already installed
```

```
npm run workbox-wizard # workbox wizard
```

```
npm run workbox-cache # workbox generateSW workbox-config.js
```

```
#you could use the native cli commands if installed globally
```

# Creating the Basemap

- Use raw vector tile files
- Use GeoJSON for any additional map layers

# OpenMapTiles

1. Download a prepared OpenMapTiles extract

# OpenMapTiles

1. Download a prepared OpenMapTiles extract
2. Create a GeoJSON to clip the extract



# OpenMapTiles

1. Download a prepared OpenMapTiles extract
2. Create a GeoJSON to clip the extract
3. Install the mbtiles-extracts Node JS Package

# OpenMapTiles

1. Download a prepared OpenMapTiles extract
2. Create a GeoJSON to clip the extract
3. Install the mbtiles-extracts Node JS Package
4. Edit it to allow for not passing in a property n

# OpenMapTiles

1. Download a prepared OpenMapTiles extract
2. Create a GeoJSON to clip the extract
3. Install the mbtiles-extracts Node JS Package
4. Edit it to allow for not passing in a property n
5. Unpack raw vector tiles from mbtiles
  - <https://www.npmjs.com/package/mbtiles>

# OpenMapTiles

1. Download a prepared OpenMapTiles extract
2. Create a GeoJSON to clip the extract
3. Install the mbtiles-extracts Node JS Package
4. Edit it to allow for not passing in a property n
5. Unpack raw vector tiles from mbtiles
  - <https://www.npmjs.com/package/mbtiles-extracts>
6. Result is hundreds of small files which must be cached

# Extract Sizes

Cleveland Metro Area - 17 MB

Cuyahoga County - 6 MB

City of Columbus - 7.5 MB

OVRDC Region (12 Counties) - 16 MB

# Copy the Basemap

```
npm run copy
```

```
/*
```

```
this will copy the OpenMapTiles data, the basemap style and the  
trails GeoJSON data to the public folder
```

```
*/
```

# Edit index.html

Replace the blank style with "bright.json"

```
var map = new mapboxgl.Map({  
  container: "map",  
  style: "bright.json",  
  hash: true,  
  center: [-82.58844, 39.5933],  
  zoom: 6  
});
```

```
/* This will change your basemap, but will not add the trails  
layer. If we have time we will do that manually, otherwise you  
can work on this on your own.*/
```

# Publish to Netlify

Delete the `.netlify` folder!!

```
npm run deploy
```

```
/*
```

```
login
```

```
choose the public folder when asked
```

```
*/
```



# Final Notes

Once your site is live, replace `http://127.0.0.1` in your `bright.json` file with the secure url of your new site

```
npm run workbox-cache
```

```
npm run deploy
```

# Goals

- ☑ Quick Overview of PWAs
- ☑ Create a Basic Web Map
- ☑ Turn this Map into a PWA
- ☑ Make the PWA Installable
- ☑ Allow the PWA to be used Offline
- ☑ Host & Install the PWA

# Thanks

Malcolm Meyer

[@getbounds](#)