

# CREATING PROGRESSIVE WEB APPS USING POWER PAGES



Mike Richard B

Architect

#FTRSA

*mgrb.in*

# What is a Progressive Web App?

A progressive web app (PWA) is a type of web application that combines the best features of native apps and websites. A PWA can run on any device and platform and can work offline or on low-quality networks. A PWA can also access device capabilities such as a camera, microphone, push notifications, and geolocation.

# What we are going to do?

- Create a simple powerpages site and then convert it into a Progressive Web App

mgrb.in

# Pre-requisites



PowerPages Access

Environment – Dynamics 365 Apps installation access (Sys Admin)  
Power Pages(Sys Admin)



Android Studio



Open SSL =

[GitHub - openssl/openssl: TLS/SSL and crypto library](#)  
[How To Install OpenSSL on Windows – TecAdmin](#)



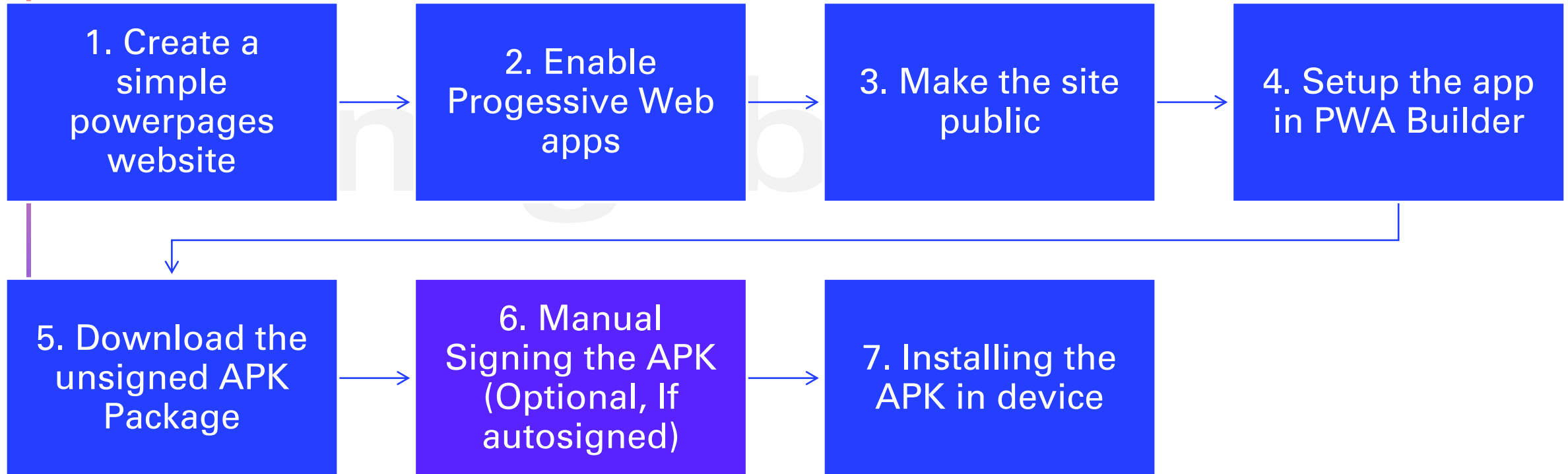
Hexadecimal to Base64 Converter  
(Optional)

[Hex to Base64 converter to convert Hexadecimal Encoded data to Base64 String. \(codebeautify.org\)](#)



Image creator or resizer tools

# High Level Steps





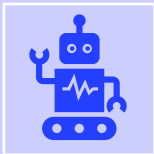
# DETAILED STEPS



# Below steps are considered known and completed..



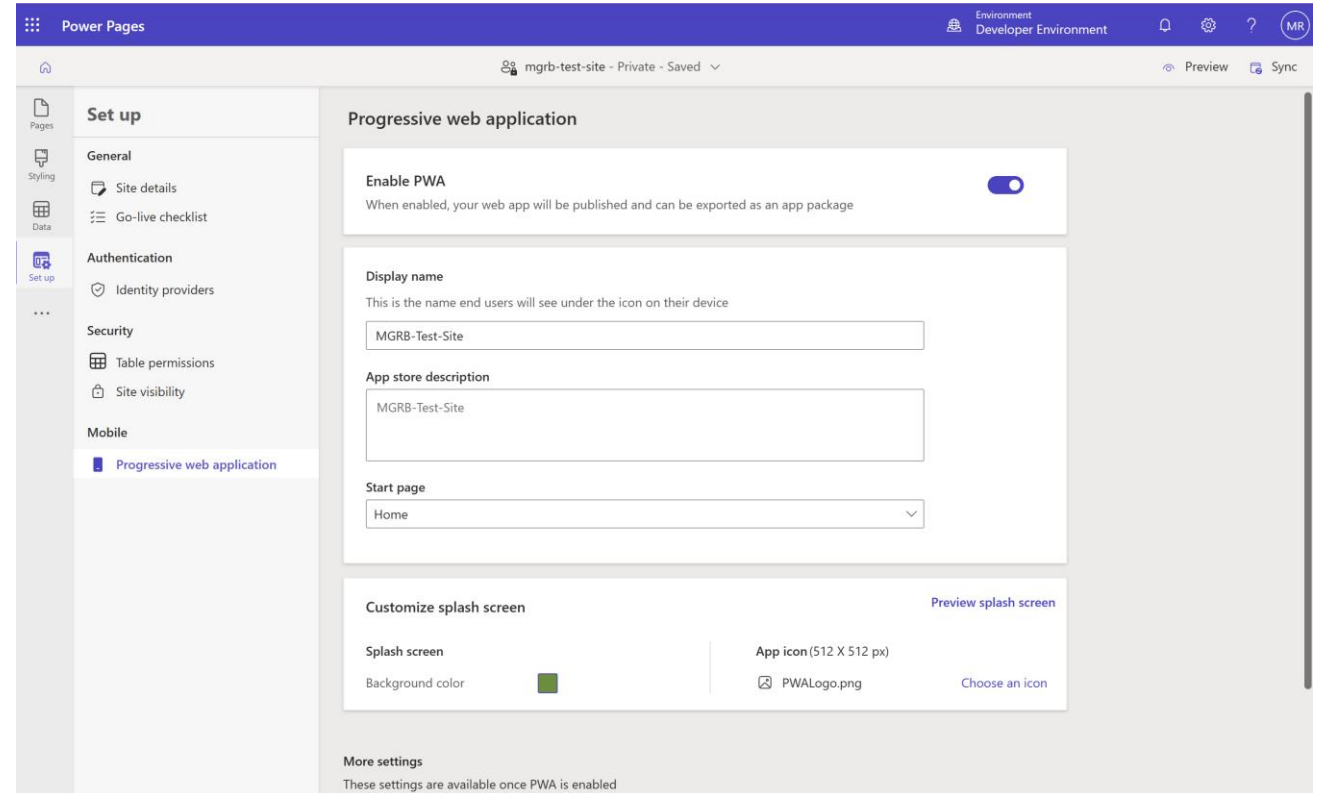
Creating a new Power Pages site



Install Android Studio

# Enable the progressive web application for the Power Pages

1. Go to powerpages designer studio
2. Go to setup
3. Go to Mobile -> Progressive Mobile Application
4. Select the option to 'Enable PWA'

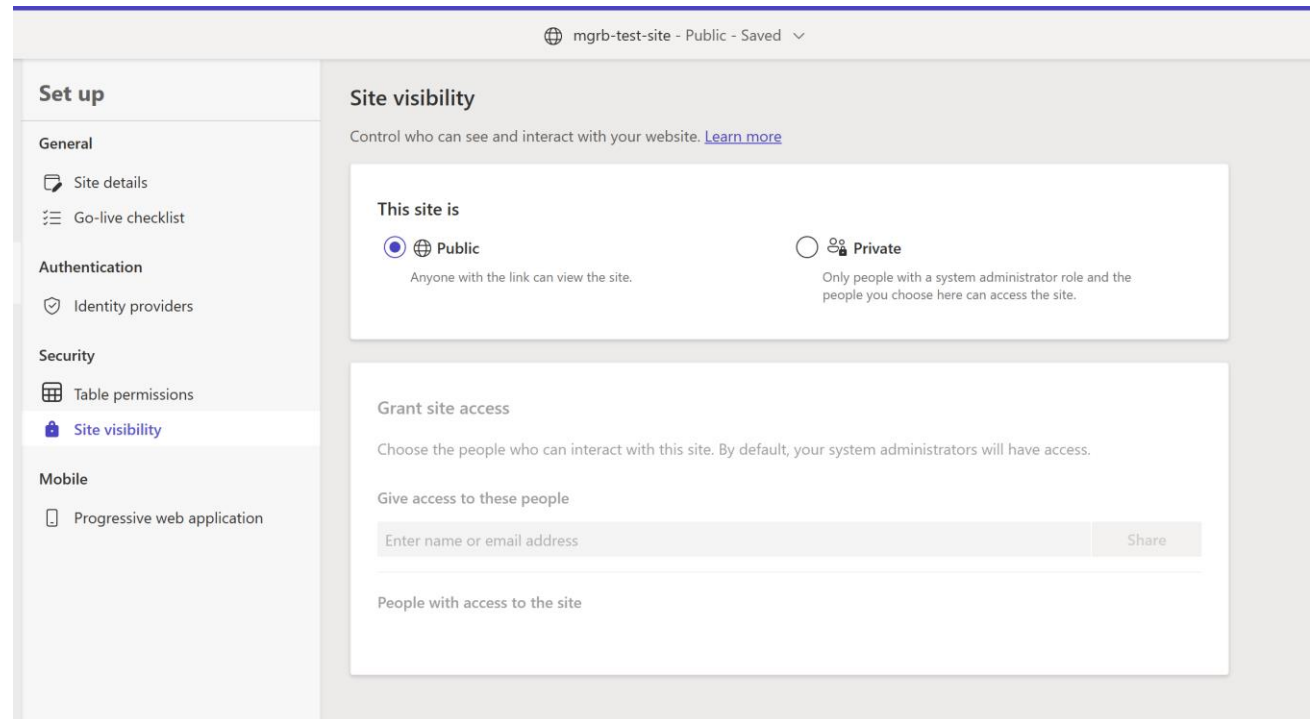




# Make sure your site is public

If your site is not public, the PWA builder cannot access the site to generate the packages

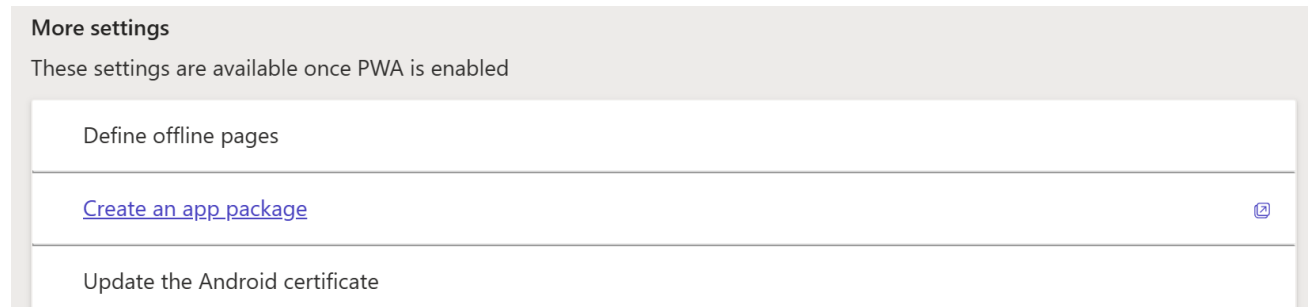
mg



The screenshot shows the 'Site visibility' settings page for a site named 'mgrb-test-site'. The page is divided into a left sidebar and a main content area. The sidebar, titled 'Set up', contains sections for 'General' (Site details, Go-live checklist), 'Authentication' (Identity providers), 'Security' (Table permissions, Site visibility), and 'Mobile' (Progressive web application). The 'Site visibility' section is selected. The main content area has a title 'Site visibility' and a subtitle 'Control who can see and interact with your website. [Learn more](#)'. Below this, there are two radio button options: 'Public' (selected) and 'Private'. The 'Public' option is described as 'Anyone with the link can view the site.' The 'Private' option is described as 'Only people with a system administrator role and the people you choose here can access the site.' Below these options is a section titled 'Grant site access' with the instruction 'Choose the people who can interact with this site. By default, your system administrators will have access.' Underneath, there is a text input field labeled 'Give access to these people' with the placeholder 'Enter name or email address' and a 'Share' button. At the bottom, there is a section titled 'People with access to the site'.

# Creating the app package

1. In the power pages studio setup window, go to the PWA screen
2. Under 'More Settings' click 'Create an app package'
3. A new PWA builder window will open to analyze and share the packages for your website



# Generate the Package

If all works as expected, PWA builder will show this page.

Click 'Generate Package'. For my demo, I have used Android Package

**Awesome! Your PWA is store ready!** ✕

You are now ready to ship your PWA to the app stores. Generate store-ready packages for the Microsoft Store, Google Play, iOS and Meta stores.

### Windows

**Generate Package**

- PWAs can be indistinguishable from native apps on Windows
- PWAs are first class applications
- Collect 100% of revenue generated via third party commerce platforms
- 1B+ store enabled devices

### Android

**Generate Package**

- PWAs are first class applications
- One app store listing for all devices (mobile, tablet, desktop)
- 2.5 billion store enabled devices

### iOS

**Generate Package**

- Leverage same codebase across all platforms
- Large user base
- Premium devices

### Meta Quest

**Generate Package**

- PWAs are first class applications
- Bring your 2D apps to VR's immersive screen
- Build immersive 3D experiences for VR using WebXR
- Currently in developer preview via sideloading

# Give Package Details

I didn't opt for Google Play, instead did manual signing. So chose 'Other Android' and gave the required details

mgrb

Android Package Options  
Customize your Android app below!

Google Play    Other Android ⓘ

**Package ID** ⓘ  
com.powerappsportals.mgrb\_test\_site.twa

**App name** ⓘ  
MGRB-Test-Site

**Short name** ⓘ  
MGRB-Test-Site

**All Settings** ▾

**Version** ⓘ  
1.0.0.0

Click below for instructions on how to submit to other Android stores.

[Packaging Instructions](#) →    [Terms of Use](#)

**Download Package**

# Unsigned APK

Once the package is generated, you can download it as zip and inside it you can see the unsigned package

mg

Name	Type	Compressed size	Password pr...	Size	Ratio	Date modified
MGRB-Test-Site-unsigned.apk	APK File	627 KB	No	676 KB	8%	22/04/2023 10:54
Readme	Microsoft Edge HTML Docum...	1 KB	No	1 KB	28%	20/04/2023 19:27

# To sign the package

- Please refer to my previous article with steps on manually signing the package - <https://mgrb.in/2023/03/25/what-why-how-native-mobile-apps-using-powerapps/>

mgrb.in

# 11. Install APK and Enjoy!



# Useful Links

- [Build progressive web apps - Power Apps | Microsoft Learn](#)
- [Sign your app | Android Developers](#)
  - <https://learn.microsoft.com/en-us/power-apps/maker/common/wrap/code-sign-android#sign-the-apk-package>
  - [Code sign for Android - Power Apps | Microsoft Learn](#)
  - [Sign your app | Android Developers](#)
  - [CloudAPK/Next-steps-unsigned.md at master · pwa-builder/CloudAPK · GitHub](#)
- [Open SSL:](#)
  - [GitHub - openssl/openssl: TLS/SSL and crypto library](#)
  - [How To Install OpenSSL on Windows – TecAdmin](#)



+



o



.



# THANK YOU

[mgrb.in](http://mgrb.in)

# Troubleshooting

If the PWA builder shows this page, it means it cannot access the site. Check if your site is set as private, if yes, change it to public and try again

mg

The screenshot shows the PWA Builder interface. At the top left is the PWA builder logo, and at the top right are links for 'Docs' and 'Community'. The main content area is divided into several sections:

- Missing Name:** A card with a grey square icon containing a white circle with a diagonal slash. The text reads 'Missing Name' and 'mgrb-test-site.powerapp...'. Below this, it says 'Your manifest description is missing.' There is a 'Share score' button and a 'Last tested 6 minutes ago' indicator.
- Package For Stores:** A button labeled 'Package For Stores' and a link 'Download Test Package'.
- Available stores:** A row of icons for Google Play, the App Store, and other platforms.
- Action Items:** A section with a header 'Action Items' and a count of 5 items (5 red dots) and 1 warning (1 yellow triangle). The list includes:
  - Add icons to your manifest
  - Add name to your manifest
  - Edit and download your created manifest (Manifest not found before detection tests timed out)
  - Add short\_name to your manifest
  - Add start\_url to your manifest
- Manifest:** A section with the heading 'Manifest' and a message: 'PWABuilder has analyzed your site and did not find a Web Manifest. Use our Manifest Editor to generate one. You can package for the store once you have a valid manifest.' There is an 'Edit Your Manifest' button, a link to 'Manifest Documentation', and a 'Want to chat? Join us on Discord' button.

# Good to know..

## Advanced configuration

The PWA capability for portals uses the following configurations that you can update by using the Portal Management app.

### Note







We recommend that you use Power Apps portals Studio to work with PWA features instead of customizing them by using the Portal Management app.

Type	Name	Description	Value
Site Setting	PWAFeature	Site setting to enable PWA for the portal	{ "status": "enable" }
			{ "status": "disable" }
Web file	PWAManifestjson.en-US.json	The PWA manifest file is stored as this JSON web file.	This web file contains all customization-related properties of the PWA (such as app name and splash screen color).
Site marker	Default Offline Page	This site marker defines which page to show when the user is offline. By default, it points to the default offline page.	"Default Offline Page"

# Good to know..

**PWAManifest.json** - Saved  
Web File

General Administration Notes Related ▾

Name	* PWAManifest.json				
Website	*  <a href="#">mgrb-test-site - mgrb-test-site</a>				
Parent Page	+  <a href="#">Home</a> ×				
Partial Url	* PWAManifest.json				
Display Date	---		Publishing State	*  <b>Published</b>	
Release Date	---		Expiration Date	---	
Subject	---				
Cloud Blob Address	---				
Content-Disposition	<b>inline</b>				

**Content**

Title	---
-------	-----