

What is a Progressive Web App?

"Progressive web app" (PWA) is both a general term for a new philosophy toward building websites and a specific term with an established set of three explicit, testable, baseline requirements.

As a general term, the PWA approach is characterized by striving to satisfy the [following set of attributes](#):

1. Responsive
2. Connectivity independent
3. App-like-interactions
4. Fresh
5. Safe
6. Discoverable
7. Re-engageable
8. Installable
9. Linkable

As a specific term, websites can be tested against the following [three baseline criteria](#) to qualify as a PWA:

1. It must run under HTTPS.
2. It must include a Web App Manifest.
3. It must implement a service worker.

PWAs are apps delivered through the web (as opposed to native apps, which are packaged and deployed through stores). As Alex Russell, who together with Frances Berriman [coined the term PWA](#), said:

they're just websites that took all the right vitamins.

How is a Gatsby site a Progressive Web App?

Gatsby is designed to provide top-notch performance out of the box. It handles code splitting, code minification, and optimizations like pre-loading in the background, image processing, etc., so that the site you build is highly performant, without any kind of manual tuning. These performance features are a big part of supporting the progressive web app approach.

Then there are the three baseline criteria for a site to qualify as a PWA.

It must run under HTTPS

Running your site under HTTPS is a highly recommended security practice, no matter the content of your site. Specifically concerning progressive web apps, running under HTTPS is a criterion for many new browser features that are required for Progressive Web Apps to work.

This one's all you!

It must include a Web App Manifest

A [Web App manifest](#) is a JSON file that provides the browser with information about your web app, and makes it possible for users to save to their home screen.

It includes information like the Web App's `name`, `icons`, `start_url`, `background-color` and [more](#).

Gatsby provides a plugin interface to add support for shipping a manifest with your site -- [gatsby-plugin-manifest](#).

It must implement a service worker

A [service worker](#) provides support for an offline experience for your site, and makes your site more resilient to bad network connections.

It's a script that runs separately in the background, supporting features like push notifications and background sync.

Gatsby also provides a plugin interface to create and load a service worker into your site -- [gatsby-plugin-offline](#).

We recommend using this plugin together with the [manifest plugin](#). (Don't forget to list the `offline` plugin after the `manifest` plugin so that the manifest file can be included in the service worker).