
What is
PWA?

What is Progressive Web App?

- A web Application + additional Power
  - Responsive
  - Work in Offline mode(without Internet)
  - Look Like native mobile app
  - Push Notification
  - Update to date in background
  - Can use GPS and access hardware component

Each devices have their own language to implement

All in One :
PWA will work on all devices

PRIMARY PROGRAMMING LANGUAGE

iOS	Android	PWA
iOS Apps	Android Apps	PWA Websites
Objective-C	JAVA	HTML, CSS, Javascript

Progressive means ?

- **PWAs are web applications. Here, web application simply means an app running as a website like, for example, Twitter. The first part of the term – Progressive – means that the user experience is enhanced gradually based on the browser's capabilities.**

PWA uses which technologies ?

PWA based on any library like React, Angular etc ?

PWA and modern UI frameworks

Few people think that a PWA is coupled with the latest UI frameworks like ReactJs, Angular 6 or Vue.js. Well, not necessarily. PWA has nothing to do with the framework you are using, it only needs the required components.

WHO USED PWAS?

Why PWA ?

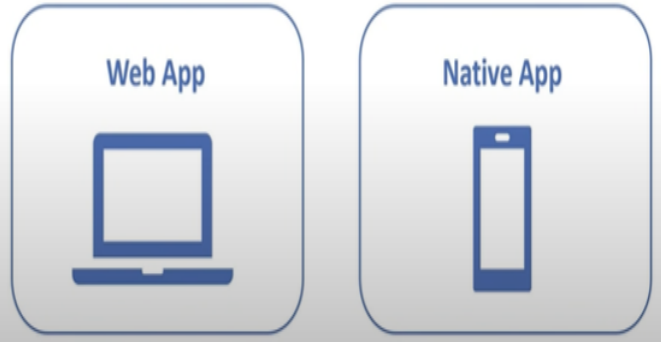
PWA

Provides functionality of native apps

Works on all mobile platforms

Can run in it's own window outside of a web browser

Can run on any device that supports web



PWA Capabilities

PWA

Load Instantly

Capable of working offline

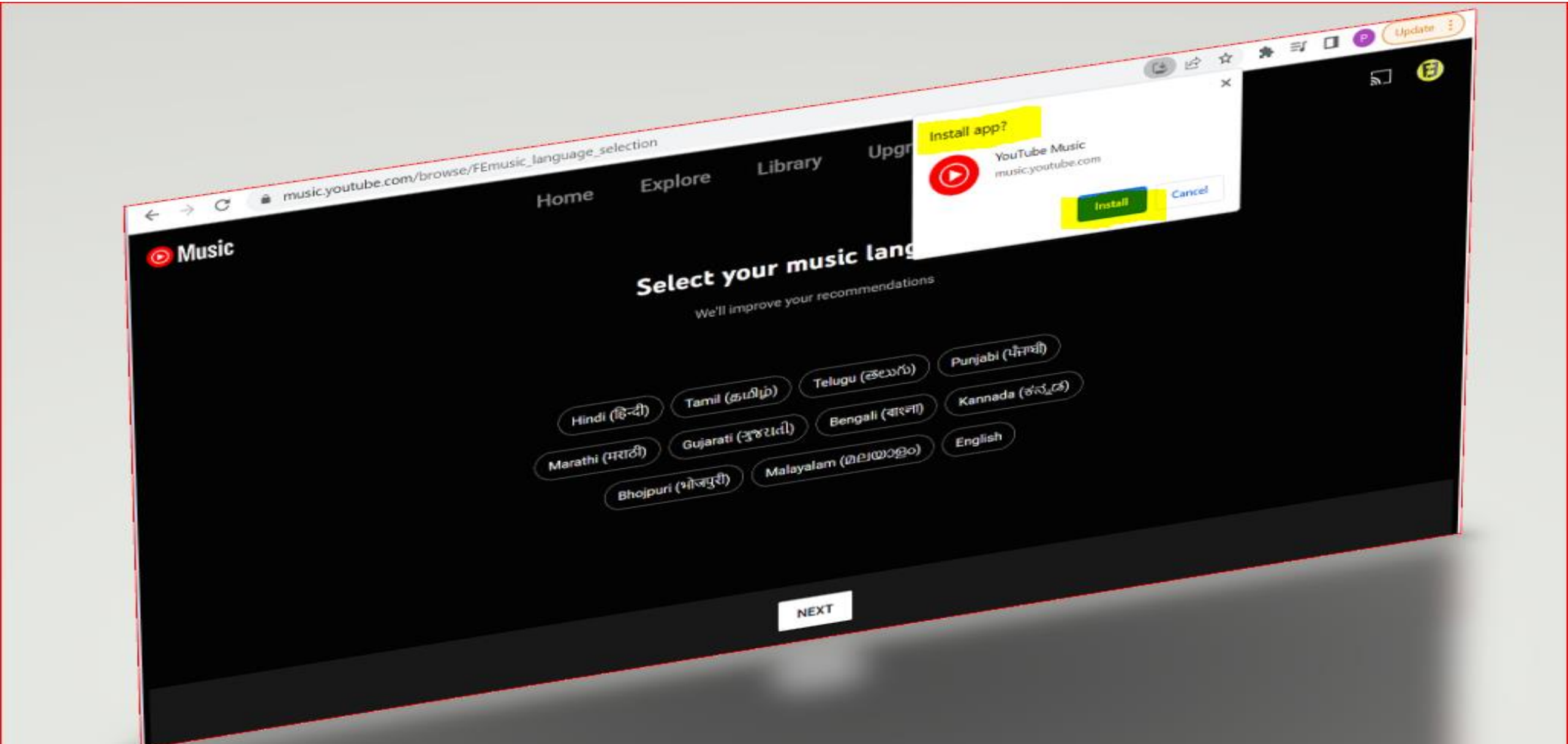
Support for push notifications

Automatic Self-Updates

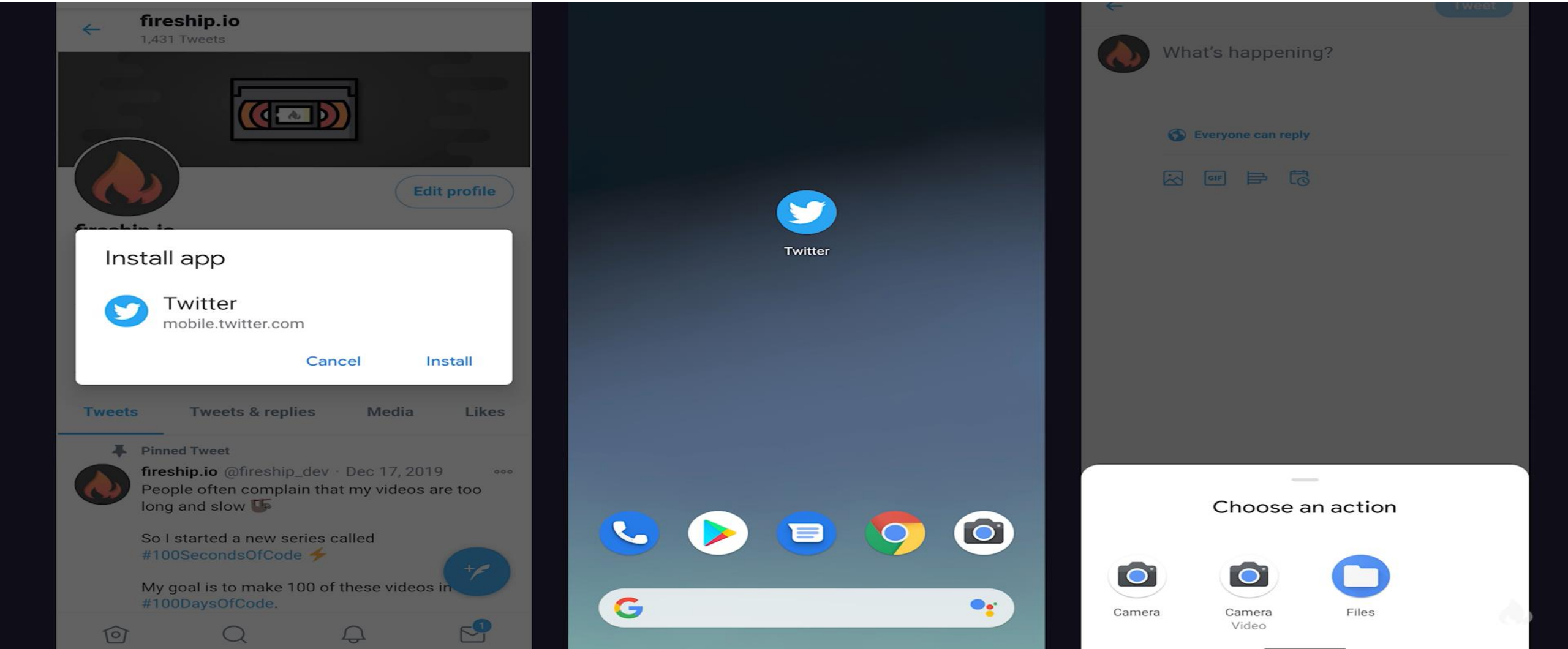
App launch options

Own app window

PWA example if we use web application on laptop



PWA example if we use web application at mobile:
Installable, Reliable and capable



Why we need PWA even after web application and native app ?

Why we need PWAs and what are their features?

Let's discuss some of the core attributes which are missing from common non-progressive web applications.

1. **Reliable**
   The app should be lightning fast when loading, it should be close to instantaneous and should also open when there is **no network** or fairly **low-speed** network like 2G. Google found that 53% of the users abandon the website if the page took longer than 3 seconds to load.
2. **Fast**
   The scrolls and page transitions should be buttery smooth when the user is interacting with the web app. Everyone hates crappy scrolls.
3. **Responsive**
   The app should fit in all the different sizes of devices. The perfect web app should be like liquid, which takes the shape of its vessel.
4. **Installable**
   If we want to make web apps closer to the native apps, they have to be installable and should reside in the home screen along with other native apps, so that the user can access the PWA in one click.
5. **Splash Screen**
   PWA adds a splash screen during the startup of the app. This makes the PWA feel more like a native app 😉
6. **Highly engage-able**
   The app should keep the users engaged. A PWA provides features like push notification, home screen icon, full-screen and *offline first* app to glorify user engagement.

All the above characteristics are possible in PWA.

Real Time Example

Meet progressive web applications

Let's take the example of Whatsapp on your phone. When there is no network, you can still open the app, check past messages and even reply to someone. When the phone gets the internet connection, the messages are being automatically sent in the background.

This is what PWA promises to provide in web apps. It enables web apps to load when there is no network, sync in the background and seamlessly do things while providing a native-like experience for your users.

How does it possible to use in offline mode ?

How does a PWA makes websites available offline?

That was exactly my first thought about PWA. How exactly can web apps open without the internet?

We all know native apps can open without an internet connection because when we download and install them, critical resources like UI components and some data are stored in the device…. well, this is exactly what happens in a PWA as well 😀

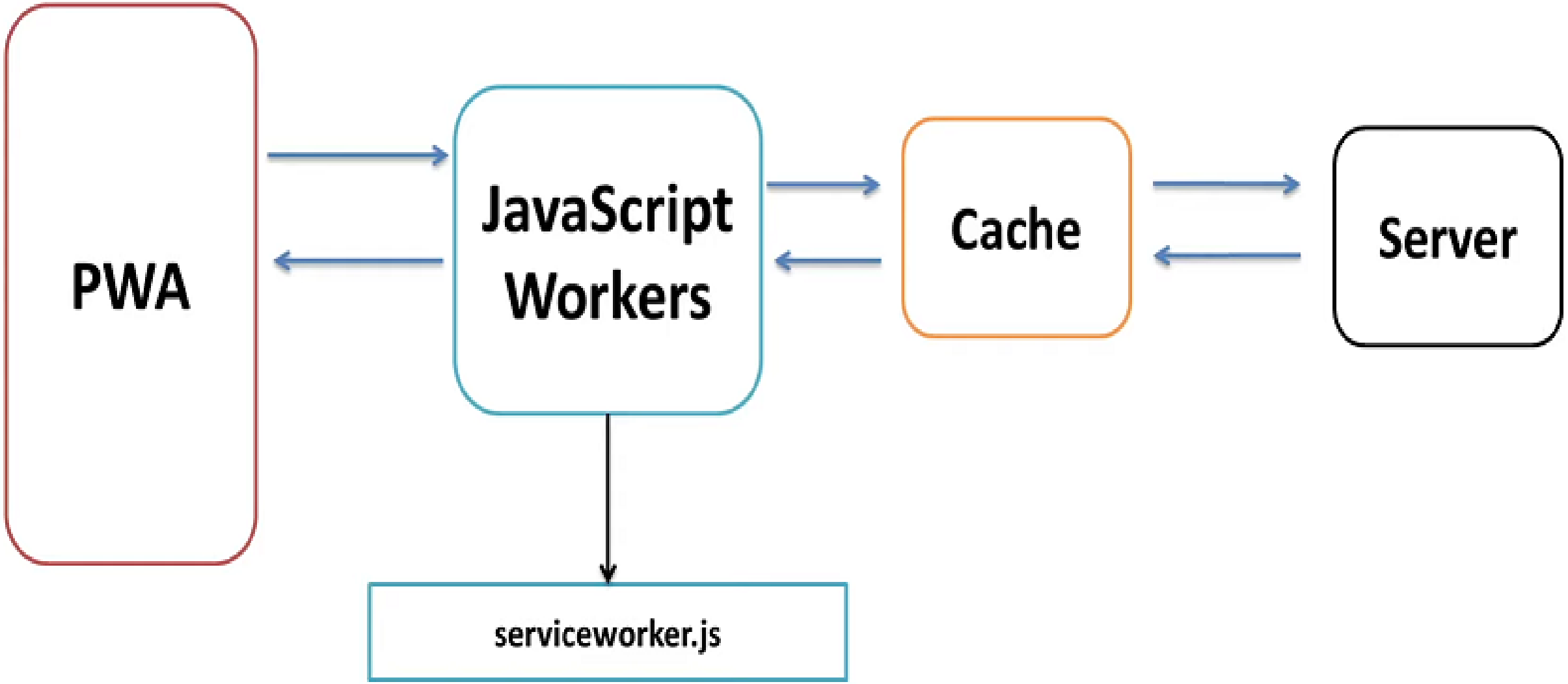
The PWA stores HTML files, CSS files and images in the browser cache and the developers can fully control the network call. All of these are being achieved by **Service Workers**.

Example

PWA Flipkart

Flipkart Website

How does PWA works?



Component of PWA

What are the Technical components of a PWA?

PWA has some important technical components which work together and energizes the regular web app. The following components are required to develop a good PWA.

Service Worker

Manifest.json

HTTPS

Service Worker

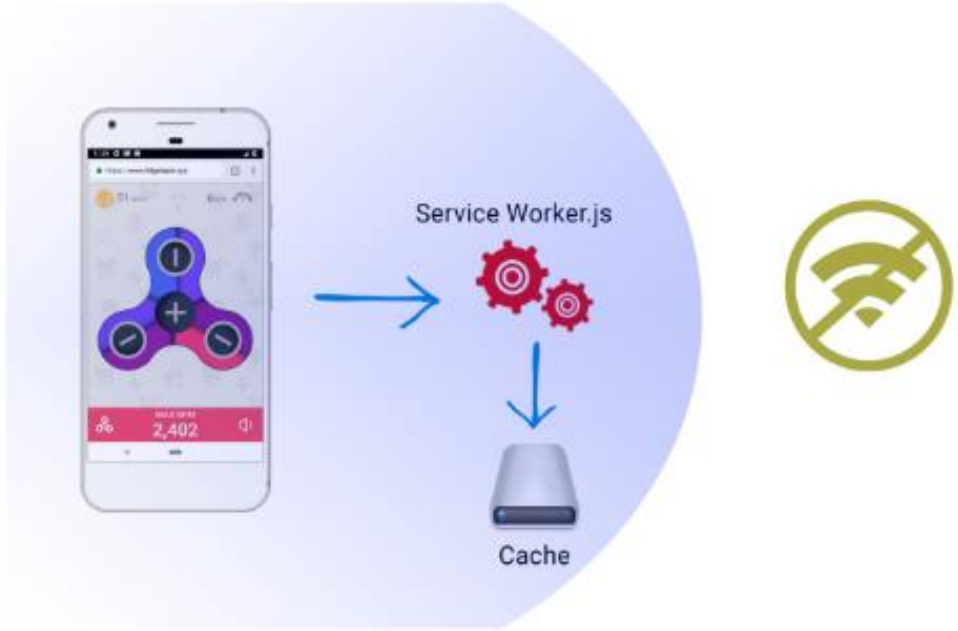
1. **Service Worker**

Our web apps talk to the network directly and if there is no network, the screen shows the famous dinosaur.

Screen when our normal web cannot connect to Internet

Here is an opportunity to optimize this process. For the first-time load, the service worker stores the required resources in the browser cache. And when the user visits the app next time, the service workers check the cache and returns the response to the user before even checking the network.

A **Service worker** is just a component of JavaScript code which works as a proxy between the browser and the network. A Service worker manages the push notifications and helps to build the *offline first* web application using the browser's cache API.



This is how the service worker performs its responsibility to empower the regular websites

This can speed up the performance of your app, wheter the device is connected or not connected to the internet.

The manifest file

2. The manifest file

The manifest file is a config JSON file which contains the information of your application, like the icon to be displayed on the home screen when installed, the short name of the application, background color, or theme.

If the manifest file is present, the Chrome browser will automatically trigger the web app install banner, and if the user agrees, this adds the icon to the home screen and the PWA is installed. Isn't this bossy? 😎

The following is the sample manifest file:

```
{
  "short_name": "Spinner",
  "name": "Fidget spinner",
  "icons": [
    {
      "src": "/images/icons-192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "/images/icons-512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": "/home/?source=pwa",
  "background_color": "#3367D6",
  "display": "standalone",
  "scope": "/home/",
  "theme_color": "#3367D6"
}
```

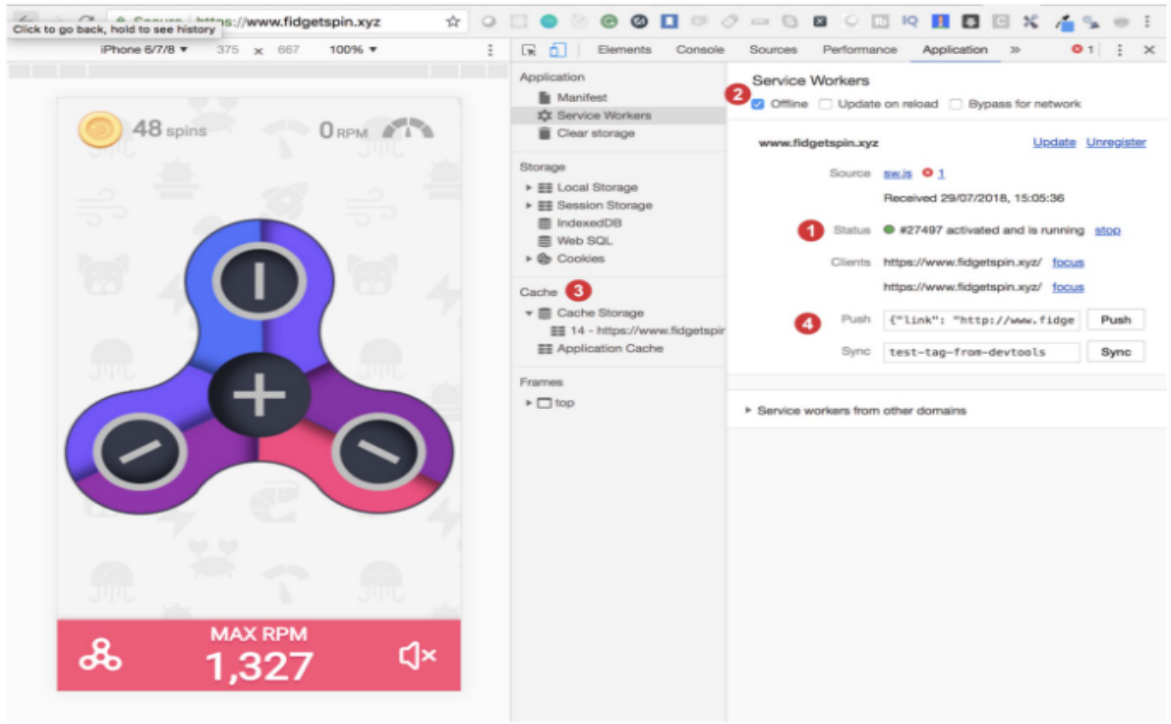
HTTPS

3. HTTPS

Service workers have the ability to intercept the network requests and can modify the responses. Service workers perform all the actions on the client side. Hence, PWA requires secure protocol HTTPS.

The service worker has the ability to receive push notifications and background sync, which definitely increases the user experience and keeps the customer engaged. Push notification and background sync are optional, but are recommended to provide a more native-like experience.

If you want to play around with a PWA and want to see service workers in action



1. **Status**: Tells us that the service worker is activated and running.
2. **Offline**: By checking this option, chrome treats the app is if it's offline. Refresh the tab and it will emulate how the PWA will respond when there is no network. You can also switch off the wifi or data to test the PWA.
3. **Cache**: This section shows what all files are stored in the cache by the service worker.
4. **Push & Sync**: These sections are used during development to test push notification and background sync.

How to implement PWA ?

```
<script>
    if ('serviceWorker' in navigator) {
        navigator.serviceWorker.register('/sw.js');
    }
</script>
```

Tools and Libraries

Tools and libraries

There are few open source tools available which enhance and make it easy to develop PWA.

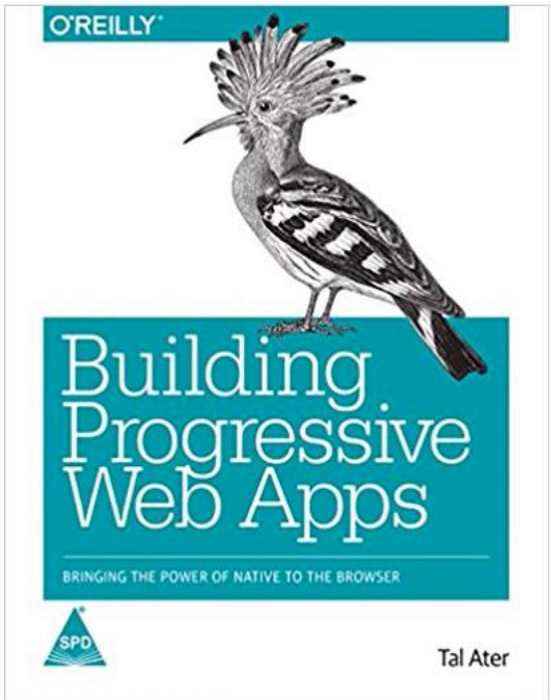
1. Lighthouse is an audit tool which can run against any web page, public or personal, and generates one report with the checklist required for PWA. This can be used during the development of your PWA to crosscheck and get recommendations to further improve the experience.
2. Workbox is a collection of libraries, open sourced by Google and can be used to generate the service worker file. Workbox also comes along with various caching strategies of images and other resources.

Advantages of PWA

- You don't have to go through the process to get into different app stores.
- You can build PWAs with common web technologies.
- They are often cheaper to build.
- You'll have fewer code-bases to maintain.
- PWAs are responsive and work with many different screen sizes.
- PWAs are smooth, fast and lightweight.
- They work offline, unlike your regular site.

Where to Learn about PWA?

Google Developers



Follow me for such info

https://www.linkedin.com/in/priya-bagde/

GitHub

https://github.com/priya42bagde

YouTube

https://www.youtube.com/channel/UCK1_Op30_pZ1zBs9l3HNyBw/videos (Priya Frontend Vlogs)