moz://a

Intro to Progressive Web Apps

IWD Celebration – GDG Abu Dhabi – NYUAD | 9th March 2018

Overview

Hi.

We're Mozilla, the proudly non-profit champions of the Internet, helping to keep it healthy, open and accessible to all.

Tweet at Us!

#mozilla

#moztechspeakers

TAKE 3 MIN to Tell Us What you think!

mzl.la/devsurvey

Me.

Alaa Shaheen

@FloweryCoder https://www.linkedin.com/in/alaashaheen1/

Email: alaa.shaheen2012@gmail.com

Software Product Manager at Bilbareed.com

Mozilla Tech Speaker

What is PWA

- Web apps are the websites, that are using web technologies, and they have the capabilities to act like a mobile app.
- Top level in task switcher
- Top level in home screen
- Top level in the notification tray

- ➔ "A progressive web application is basically a website built using modern web technologies but acts and feels like a mobile app"

Why we need to go into Progressive web apps?

- Recent studies shows that progressive web apps, increases business revenues and web stands for the companies.

Using the mobile apps,,

- Needs to install app from the app store
- Some apps are not available in our countries
- Limited access to app stores
- Users install or buy apps when buying phone, then less apps are being installed.

FlipKart

- Largest online shopping site in India, called FlipKart, launched their light application using progressive web apps, and they found a huge increase in the number of visitors.
- **62%** from the users accessed the website from **2G network.**
- Uses **three times less mobile data** to access the website and get the items they want.

Progressive Web Apps

- ➔**Progressive:** must work on any device and enhance progressively.
- ➔**Discoverable:** in search engines.
- ➔**Linkable:** should use the URI to indicate the current state of the application.
- ➔**Responsive:** must fit the device’s form factor and screen size.
- ➔**App-like:** like a native app and be built on the application shell model, with minimal page refreshes.

Progressive Web Apps

- **Connectivity-independent:** low connectivity or offline.
- **Re-engageable:** push notifications.
- **Installable:** installed on the device’s home screen.
- **Fresh:** new content should be made available in the app.
- **Safe:** hosted over HTTPS to prevent man-in-the-middle attacks.

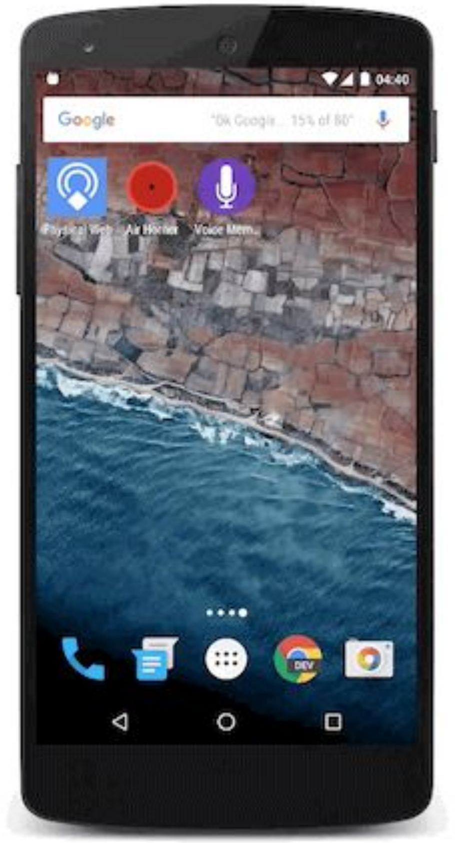
Progressive Web Apps

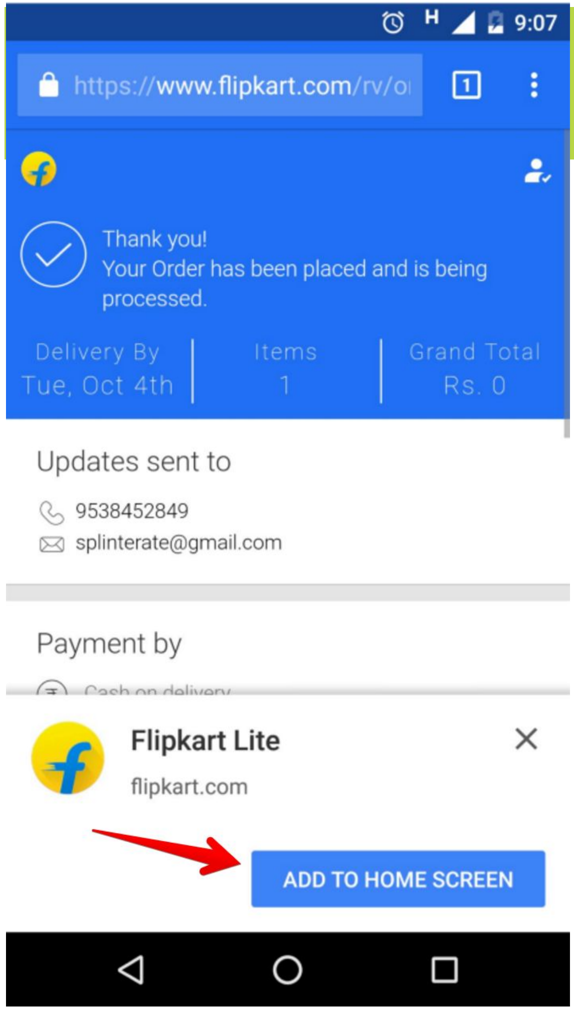
- **Connectivity-independent:** low connectivity or offline.
- **Re-engageable:** push notifications.
- **Installable:** installed on the device’s home screen.
- **Fresh:** new content should be made available in the app.
- **Safe:** hosted over HTTPS to prevent man-in-the-middle attacks.

moz://a

Characteristics:

- Add to home
- Splash Screen
- App Shell





9:07

https://www.flipkart.com/rv/o

Thank you!
Your Order has been placed and is being processed.

Delivery By	Items	Grand Total
Tue, Oct 4th	1	Rs. 0

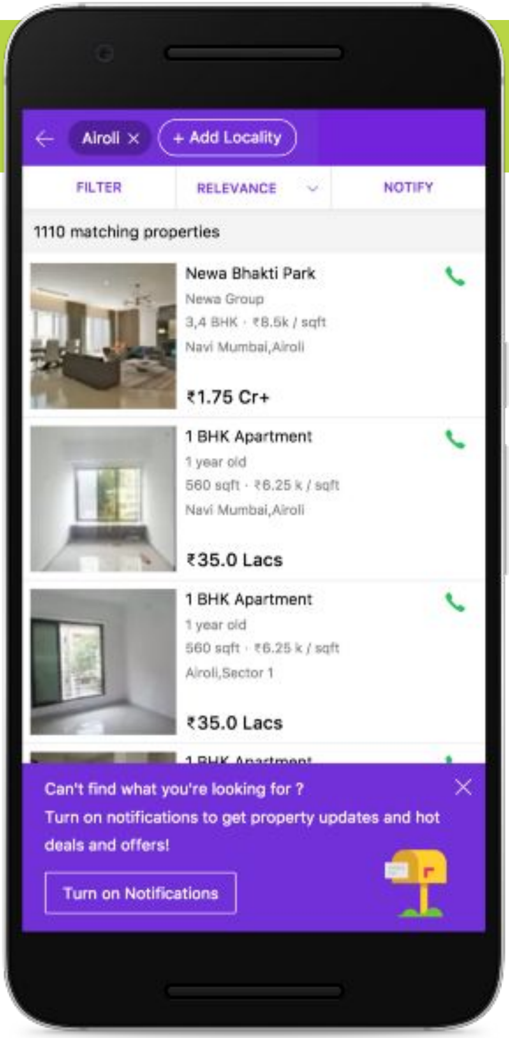
Updates sent to

9538452849
splinterate@gmail.com

Payment by

Cash on delivery

Flipkart Lite
flipkart.com



1110 matching properties

Newa Bhakti Park
Newa Group
3,4 BHK · ₹8.5k / sqft
Navi Mumbai,Airoli
₹1.75 Cr+

1 BHK Apartment
1 year old
560 sqft · ₹6.25 k / sqft
Navi Mumbai,Airoli
₹35.0 Lacs

1 BHK Apartment
1 year old
560 sqft · ₹6.25 k / sqft
Airoli,Sector 1
₹35.0 Lacs

1 BHK Apartment

Can't find what you're looking for ?
Turn on notifications to get property updates and hot deals and offers!

Technologies behind it?

- Service Workers
- Application Shell
- Web App Manifest File

Service Workers

- Web Apps are being built on top of:
- Server
- Client

Service Worker sit between client and server to enhance network connectivity to the app.

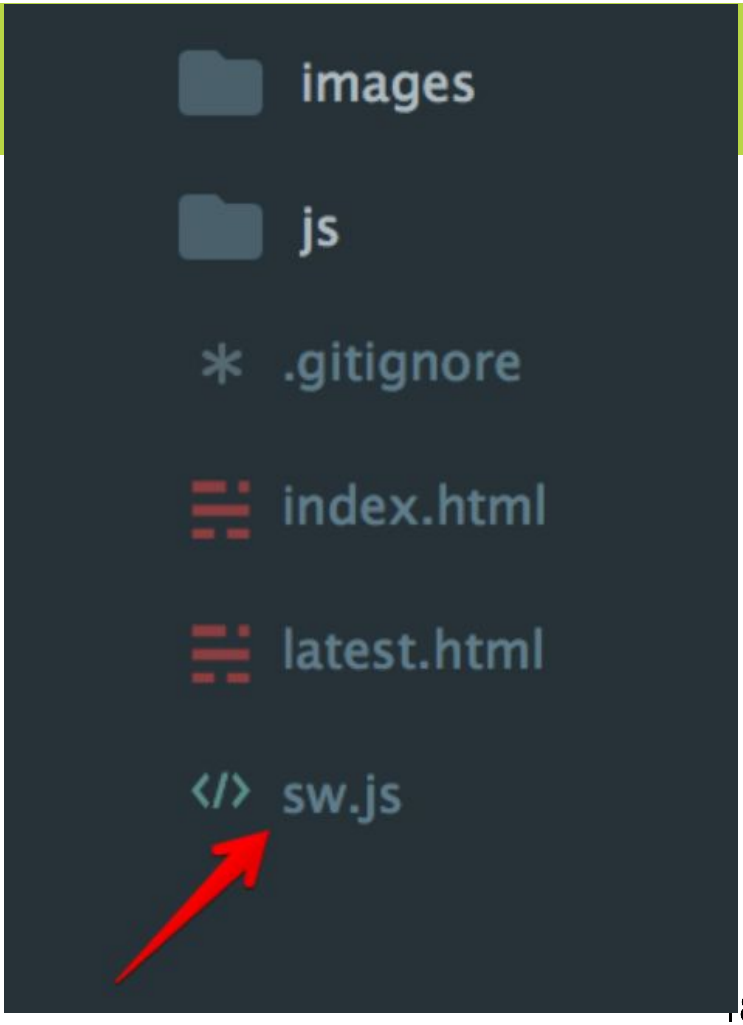
moz://a

Service Workers

- a script, that your browser runs in the background.
- handle http requests and push notifications.
- cache all static resources.
- can be used to display the application shell.
- inform users that they are disconnected from the internet.

Code Example

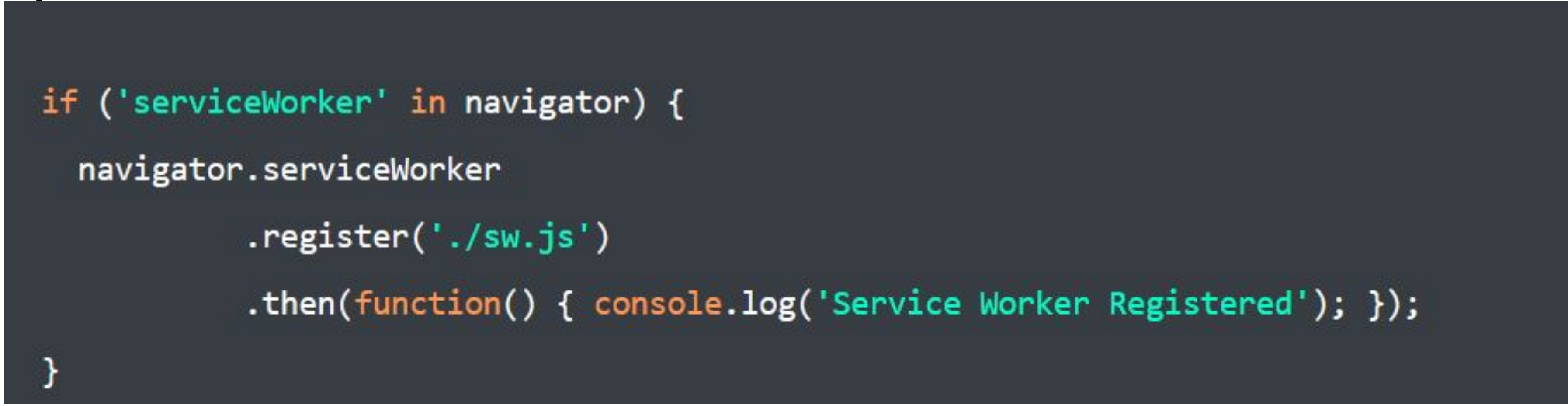
- Service Worker's Life Cycle:
- Register
- Install
- Activate
- Fetch



Code Example

→ Register the service worker in your app's js file

app.js

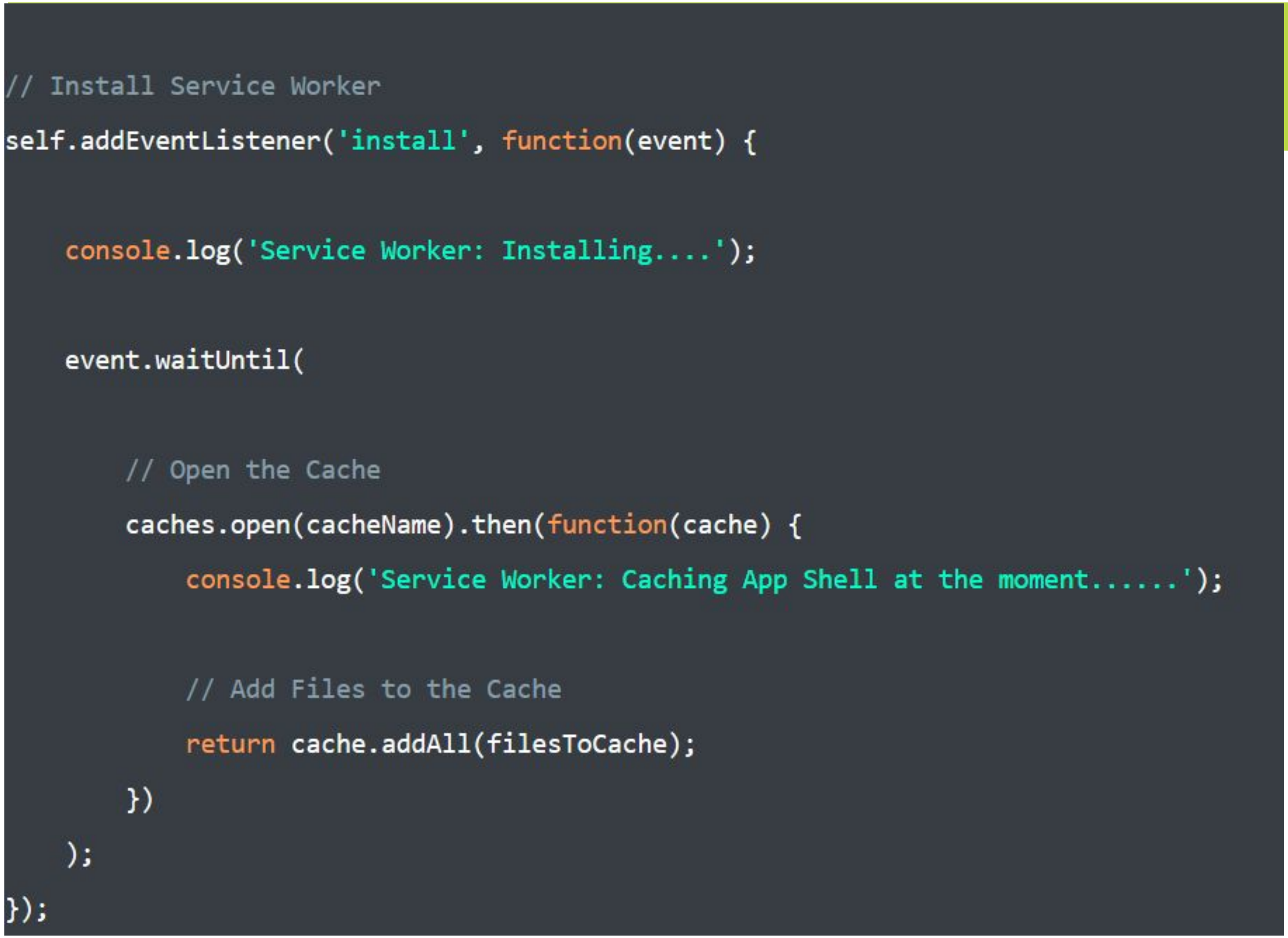


```
if ('serviceWorker' in navigator) {
  navigator.serviceWorker
           .register('./sw.js')
           .then(function() { console.log('Service Worker Registered'); });
}
```

Code Example

- An install event is triggered the first time a user visits the page.
- the service worker is installed in the browser.
- you can cache all the static assets in your web app.

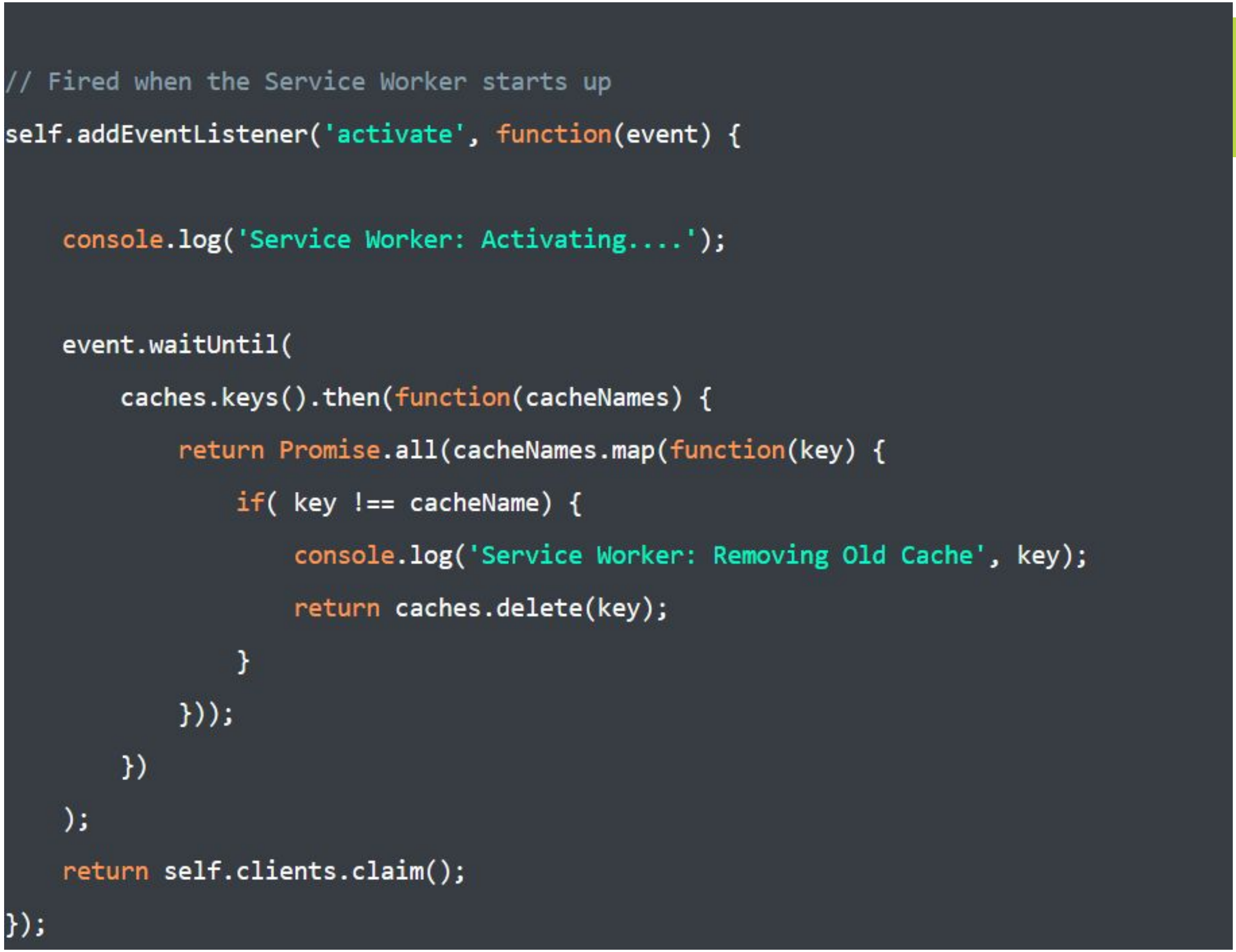
moz://a



```javascript
// Install Service Worker
self.addEventListener('install', function(event) {

    console.log('Service Worker: Installing....');

    event.waitUntil(

        // Open the Cache
        caches.open(cacheName).then(function(cache) {
            console.log('Service Worker: Caching App Shell at the moment......');

            // Add Files to the Cache
            return cache.addAll(filesToCache);
        })
    );
});
```

Code Example

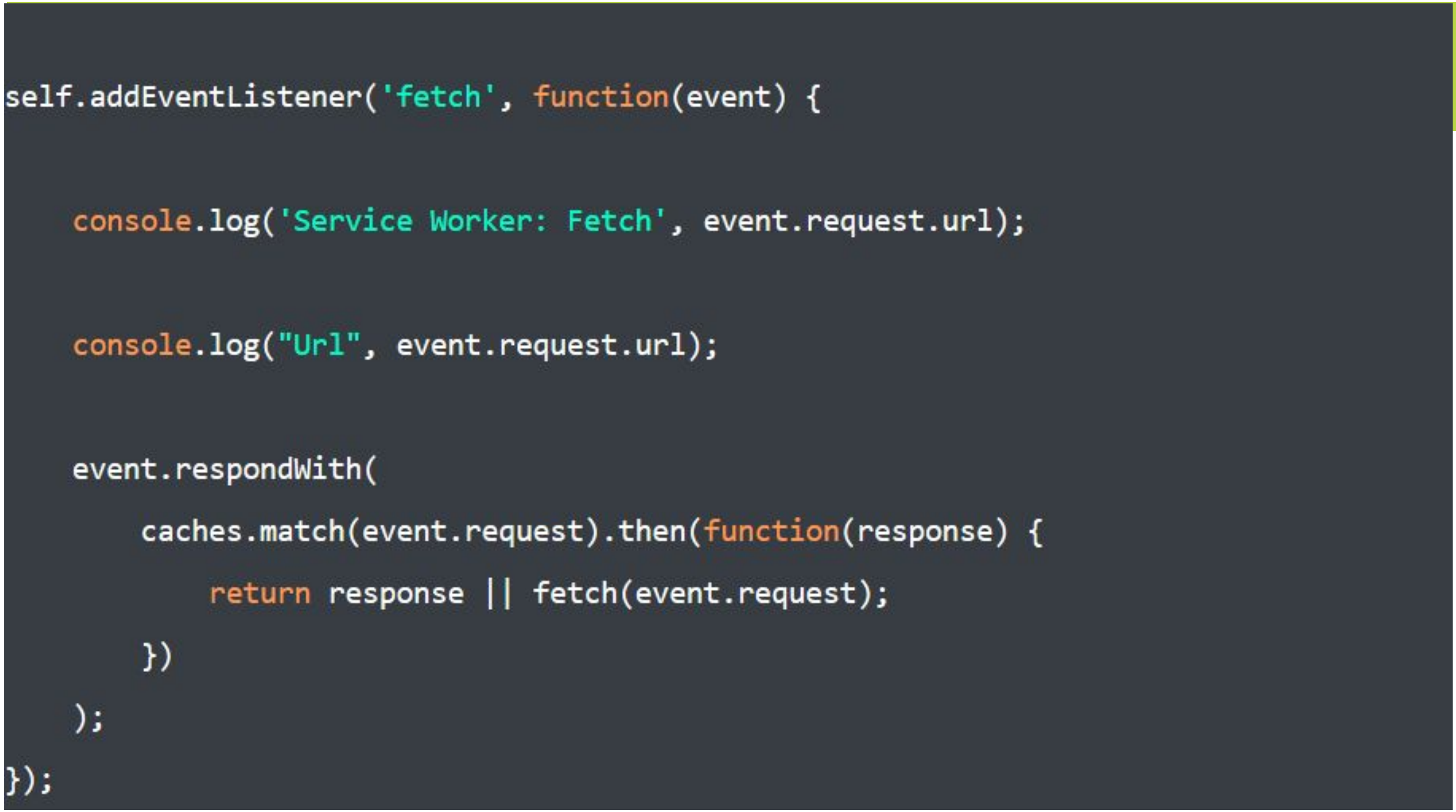
- **Activate**: This event is fired when the service worker starts up.
- service worker updates its cache whenever any of the app shell files change.



```
// Fired when the Service Worker starts up
self.addEventListener('activate', function(event) {

    console.log('Service Worker: Activating....');

    event.waitUntil(
        caches.keys().then(function(cacheNames) {
            return Promise.all(cacheNames.map(function(key) {
                if( key !== cacheName) {
                    console.log('Service Worker: Removing Old Cache', key);
                    return caches.delete(key);
                }
            }));
        })
    );
    return self.clients.claim();
});
```

Code Example

- **Fetch**: This event helps serve the app shell from the cache.
- It then either responds with the cached version, or uses fetch to get a copy from the network.



```
self.addEventListener('fetch', function(event) {

    console.log('Service Worker: Fetch', event.request.url);

    console.log("Url", event.request.url);

    event.respondWith(
        caches.match(event.request).then(function(response) {
            return response || fetch(event.request);
        })
    );
});
```

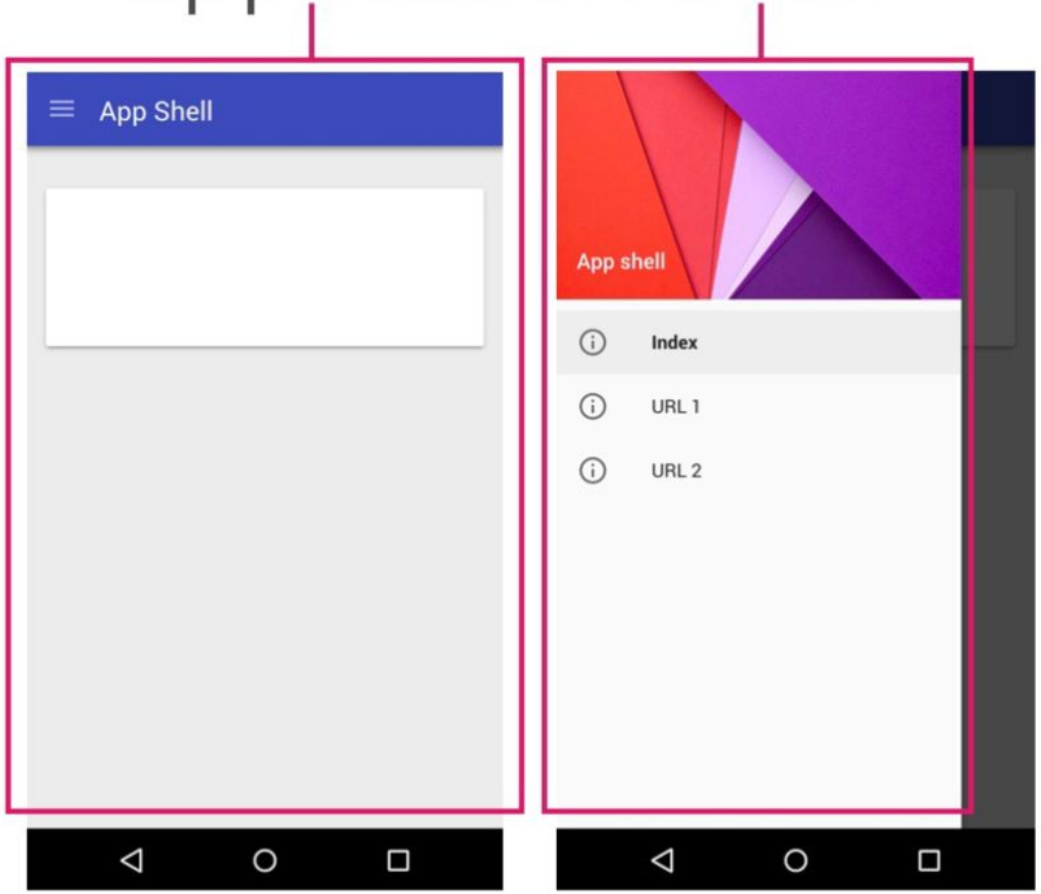
Web app Manifest File

- It controls how your app should appear to the user in mobile phone.
- Where to find the app in the phone and how to launch it.

Application Shell

- Usually the app assembles the page content in one place.
- App Shell, separates the content of the app that does not change often.
- It helps to boot the app when it starts, and power the user interface of the web app.
- It is written in HTML , CSS and JS.

application shell



App Shell

- Service Worker can save locally the content of the app, and the application shell can load the main app interface.
- App Shell , with caching mechanism and using the Service Workers, allows developers to focus on performance and speed.

Offline Mode

Using PWA , developer can cache the App Shell, and load it offline, by saving content locally.

App Shell

- Break design to main components:
- Main design on the screen
- Other UI components key to the app
- Supporting resources to App Shell, JS , Styles , etc

App Shell

- Should contain all the necessary resources to launch the app:
- HTML
- CSS
- JS
- Images

moz://a

How to Add Data to the App?

- We have three methods to display data to our PWA:
- Server Side Rendering: fastest
- Get Data Via Ajax Request: slowest method
- Combination of server side and Ajax request: server inject data into the app JS.

- Eliminate the need for HTML request
- But we need JS to run the data
- We can cache data after loading for further use

Storage data

- Local Storage: easiest , and available to everywhere, but the it is Synchronous and may cause bad performance.

moz://a

Storage data

- Cache:
- Ready to use
- Asynchronous
- Fast
- Not available to all browsers

Storage data

- Indexed DB:
- Fast
- Asynchronous
- Supported on all browsers
- Check Mozilla develop website for more info.

Appendix

Resources

MDN Web Docs

https://developer.mozilla.org/en-US/Apps/Progressive/Introduction

https://developer.mozilla.org/en-US/Apps/Progressive

The Firefox Frontier

https://blog.mozilla.org/firefox/progressive-web-apps-whats-big-deal/

Auth0.com

https://auth0.com/blog/introduction-to-progressive-apps-part-one/

Google Developers

https://developers.google.com/web/progressive-web-apps/

moz://a

Thanks.

Alaa Shaheen

@FloweryCoder https://www.linkedin.com/in/alaashaheen1/

Email: alaa.shaheen2012@gmail.com

Software Product Manager at Bilbareed.com

Mozilla Tech Speaker

Q&A