PROGRESSIVE WEB APPS in the real world
"A Progressive Web App uses modern web capabilities to deliver an *app-like* user experience"
CORE DRIVERS

URLS & links
CORE DRIVERS

URLS & links

System Capabilities

Accessible
CORE DRIVERS

URLS & links

Accessible

System Capabilities

Free!
These apps aren’t packaged and deployed through stores, they’re just websites that took all the right vitamins.

Alex Russell
Awesome
Awesome

NO INSTALL NEEDED
Awesome

NO INSTALL NEEDED
NO DOWNLOAD
Awesome

NO INSTALL NEEDED
NO DOWNLOAD

BUILT ON THE FABRIC OF THE WEB
You are already using them
What does this actually mean for our users?
What does this actually mean for our users?

★ More engaging
What does this actually mean for our users?

★ More engaging
★ Faster
What does this actually mean for our users?

★ More engaging
★ Faster
★ Reliable
1. Understanding Progressive Web Apps
1. Understanding Progressive Web Apps
2. Engage your users
1. Understanding Progressive Web Apps
2. Engage your users
3. Speed up your site
This Talk

1. Understanding Progressive Web Apps
2. Engage your users
3. Speed up your site
4. Improve reliability
1 Understanding progressive web apps
Service workers
Think of your web apps requests as planes taking off. ServiceWorker is the air traffic controller that routes the requests.

Jeff Posnick
“The perfect progressive enhancement.”
Behind the scenes
With Service Workers
With Service Workers
With Service Workers
With Service Workers
HTTPS Only
Free SSL
Service Workers are the key to unlocking the power
Support
The Approach
App. Shell Architecture

The Approach
Pick & Choose
App. Shell Architecture

The Approach
Application Shell Architecture

Cached shell loads **instantly** on repeat visits.

Dynamic content then populates the view.
Cached shell loads **instantly** on repeat visits.

Dynamic content then populates the view.
bit.ly/app-shell
Or pick & choose the features you like!
Progressive Web Apps

operasoftware.github.io/pwa-list
Summary

- Service Workers
- SSL only
- Architecture
2. Engage your users
2.1 Look and feel
2.1 Look and feel

2.2 Push notifications
2.1 Look and feel

2.2 Push notifications

2.3 Add to home screen
Look and feel
Look and feel

<!-- Chrome, Firefox OS and Opera -->
<meta name="theme-color" content="#09adec">

<!-- Windows Phone -->
<meta name="msapplication-navbutton-color" content="#09adec">

<!-- iOS Safari -->
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="black-translucent"
look and feel

bit.ly/color-custom
Manifest

A simple JSON file that allows you to control how your app appears to the user
Customize
<!doctype html>
<html lang="en">
<head>
<link rel="manifest" href="manifest.json">
</head>
<body>
</body>
</html>
Manifest

bit.ly/manifest-json
Manifest Generator
tomitm.github.io/appmanifest

Web App Manifest Generator

Web App Manifests are one of the key pieces to making your web app look and feel like a native app. Learn more

While everything is optional, once certain criteria is met, some browsers will automatically display an install banner for your app.

Name

Web App Manifest Generator
Name of your web app

Short Name

Manifest Generator
This will be used when there is insufficient space to display the full name, such as the homescreen

manifest.json

```json
{
    "lang": "en-GB",
    "start_url": "/",
    "display": "standalone"
}
```

More...
Three Steps

1. Subscribe User
Three Steps

1. Subscribe User
2. Save Subscription
Three Steps

1. Subscribe User
2. Save Subscription
3. Send from Google Cloud
if ('serviceWorker' in navigator) {
if ('serviceWorker' in navigator) {
    navigator.serviceWorker.register('service-worker.js').then(function (registration) {
        // Registration was successful
    }).catch(function (err) {
        // registration failed :(
        console.log('ServiceWorker registration failed: ', err);
    });
}
if ('serviceWorker' in navigator) {
    navigator.serviceWorker.register('service-worker.js').then(function (registration) {

        // Registration was successful
        registration.pushManager.subscribe({userVisibleOnly: true}).then(function (subscription) {

        });
    }).catch(function (err) {

        // registration failed :(
        console.log('ServiceWorker registration failed: ', err);
    });
}
Save Subscription
if ('serviceWorker' in navigator) {
  navigator.serviceWorker.register('service-worker.js').then(function (registration) {
    // Registration was successful
    registration.pushManager.subscribe({userVisibleOnly: true}).then(function (subscription) {
      // Send to Server
      return sendSubscriptionToServer(subscription);
    });
  }).catch(function (err) {
    // registration failed :(
    console.log('ServiceWorker registration failed: ', err);
  });
}
Send from Google Cloud
Manifest
{ "name": "Push Demo",
 "short_name": "Push Demo",
 "icons": [{
  "src": "icon.png",
  "sizes": "192x192",
  "type": "image/png"
 }],
 "start_url": "/index.html",
 "display": "standalone",
 "gcm_sender_id": "57874135625436178",
 "gcm_user_visible_only": true,
 "permissions": [
  "gcm"
 ]}
Further reading

bit.ly/push-notifications-google

bit.ly/push-notifications-dean
Web Typography

Find resources that help you build beautiful websites using Web Fonts.

Articles

A pick of some of the best typography articles from around the web.

Font Performance

Introducing fonts to a web page can have an impact on the page load times. Find out how to reduce this overhead.
Add to home screen
Add to home screen

1. You need a manifest.json file
Add to home screen

1. You need a manifest.json file
2. Your manifest file needs a start URL
Add to home screen

1. You need a manifest.json file
2. Your manifest file needs a start URL
3. You need a 144x144 PNG icon
Add to home screen

1. You need a manifest.json file
2. Your manifest file needs a start URL
3. You need a 144x144 PNG icon
4. Your site is using a Service Worker running over HTTPS
Add to home screen

1. You need a manifest.json file
2. Your manifest file needs a start URL
3. You need a 144x144 PNG icon
4. Your site is using a Service Worker running over HTTPS
5. The user has visited your site at least twice, with at least five minutes between visits.
{  
  "lang": "en",
  "background_color": "#09adec",
  "name": "Building Great Startup Teams",
  "short_name": "Building Great Startup Teams",
  "display": "standalone",
  "icons": [
    {
      "src": "./images/logo-144.png",
      "sizes": "144x144",
      "type": "image/png"
    }
  ],
  "start_url": "./index.html",
  "orientation": "portrait"
}
Record something!
Summary

Look and feel

Push notifications

Add to home screen
3. Speed up your site
3.1 Caching
3.1 Caching

3.2 Save-Data Header
3.1 Caching

3.2 Save-Data Header

3.3 Background Sync
Caching
With Service Workers
With Service Workers
```javascript
var cacheVersion = 1;
var currentCache = { assetCache: 'cache' + cacheVersion };```
var cacheVersion = 1;
var currentCache = { assetCache: 'cache' + cacheVersion };

this.addEventListener('install', event => {
    event.waitUntil(caches.open(currentCache.assetCache)
        .then(function(cache) {

        })
    );
});
Caching

```javascript
var cacheVersion = 1;
var currentCache = { assetCache: 'cache' + cacheVersion };

this.addEventListener('install', event => {
  event.waitUntil(caches.open(currentCache.assetCache)
    .then(function(cache) {
      return cache.addAll(['result.min.css', './js/material.min.js']);
    })
  );
});
```
var cacheVersion = 1;
var currentCache = { assetCache: 'cache' + cacheVersion }

this.addEventListener('install', event => {
    event.waitUntil(caches.open(currentCache.assetCache)
        .then(function(cache) {
            return cache.addAll(['result.min.css', './js/material.min.js']);
        })
    );
});

this.addEventListener('fetch', event => {
    event.respondWith(caches.match(event.request)
        .then(function (response) {
            return response || fetch(event.request);
        })
    );
});
Service Worker Toolbox

github.com/GoogleChrome/sw-toolbox

bit.ly/sw-toolbox
Save-Data Header
Using Service Workers
Using Service Workers
this.addEventListener('fetch', event => {
  if(event.request.headers.get('save-data')){
    if (/\.(jpg$|gif$|png$)/.test(event.request.url)) {
      // We can do something
      event.respondWith(
        fetch('low-quality-' + event.request.url, {
          mode: 'no-cors'
        })
      )
    }
  }
});
this.addEventListener('fetch', event => {
  if (event.request.headers.get('save-data')) {
    if (/\(\.jpg\)|\.gif\)|\.png\)/.test(event.request.url)) {
      // We can do something
      event.respondWith(
        fetch(`low-quality-` + event.request.url, {
          mode: 'no-cors'
        })
      )
    }
  }
});
this.addEventListener('fetch', event => {
  if (event.request.headers.get('save-data')) {
    if (/\..*jpg$|\..*gif$|\..*png$/.test(event.request.url)) {
      // We can do something
      event.respondWith(
        fetch('low-quality-' + event.request.url, {
            mode: 'no-cors'
        })
      )
    }
  }
});
this.addEventListener('fetch', event => {
  if (event.request.headers.get('save-data')) {
    if (/\.(jpg$|gif$|png$)/.test(event.request.url)) {
      // We can do something
      event.respondWith(
        fetch(`low-quality-` + event.request.url, {
          mode: 'no-cors'
        })
      )
    }
  }
});
If you only read one book about hiring and building teams...

Building Great Startup Teams

Don't just build teams. Build great teams.

Building Great Startup Teams - A new book by Dean Hume.
Building Great Startup Teams

Don't just build teams. Build great teams.

Building Great Startup Teams - A new book by Dean Hume.
Saved 900 KB!
Background sync
Using Service Workers
Using Service Workers
With Service Workers
Try later....
// Request a one-off sync:

navigator.serviceWorker.ready.then(
    function(swRegistration) {
        return swRegistration.sync.register('myFirstSync');
    });
// Request a one-off sync:

navigator.serviceWorker.ready.then(
    function(swRegistration) {
        return swRegistration.sync.register('myFirstSync');
    });

self.addEventListener('sync', function(event) {
    if (event.tag == 'myFirstSync') {
        event.waitUntil(expensiveOperation());
    }
});
// Request a one-off sync:

navigator.serviceWorker.ready.then(
function(swRegistration) {

    return swRegistration.sync.register('myFirstSync');
});

self.addEventListener('sync', function(event) {

    if (event.tag == 'myFirstSync') {

        event.waitUntil(expensiveOperation());
    }
});
Background sync

bit.ly/background-sync
4. Reliable
4.1 Offline Browsing

4.2 3rd Party Scripts
With Service Workers
this.addEventListener('fetch', event => {
});
this.addEventListener('fetch', event => {
  // Check if the user navigated
  if (event.request.method === 'GET' && event.request.requestHeaders.get('accept').includes('text/html')) {
    // Respond appropriately
  }
});
this.addEventListener('fetch', event => {
  // Check if the user navigated
  if (event.request.method === 'GET' && event.request.headers.get('accept').includes('text/html')) {
    // Respond appropriately
    event.respondWith(
      fetch(event.request.url).catch(error => {
        // Return the offline page
        return caches.match('the-offline-page');
      })
    );
  }
});
Offline Applications

bit.ly/offline-apps

bit.ly/offline-page
Server Fails
Using Service Workers
Using Service Workers
With Service Workers
function timeout(delay) {
    return new Promise(function(resolve, reject) {
        setTimeout(function(){
            resolve(new Response('', { status: 408, statusText: 'Request timed out.' }));
        }, delay);
    });
}
function timeout(delay) {
  return new Promise(function(resolve, reject) {
    setTimeout(function() {
      resolve(new Response('', { status: 408, statusText: 'Request timed out.' }));
    }, delay);
  });
}

self.addEventListener('fetch', event => {
});
function timeout(delay) {
    return new Promise(function(resolve, reject) {
        setTimeout(function() {
            resolve(new Response('', { status: 408, statusText: 'Request timed out.' }));
        }, delay);
    }, delay);
}

self.addEventListener('fetch', event => {
    event.respondWith(Promise.race([[timeout(6000), fetch(event.request.url)]])
});
Summary

1. Understanding Progressive Web Apps
2. Engage your users
3. Speed up your site
4. Improve reliability
Performance is a feature
Further reading

❖ Shell Architecture - bit.ly/instant-shell
❖ Getting Started - bit.ly/starting-pwas
❖ Service Worker - bit.ly/pwa-swers
❖ Code Labs - bit.ly/code-labs-pwa
❖ Recipes
  ➢ bit.ly/mozilla-service-workers
  ➢ bit.ly/chrome-service-workers
What's still to come?
What's still to come?

**Web Bluetooth**
webbluetoothcg.github.io/web-bluetooth

**Streams API**
jakearchibald.com/2016/streams-ftw

**Web NFC**
w3c.github.io/web-nfc

**Geofencing**
w3c.github.io/geofencing-api
Thank you!

@deanohume