Blink and you’ll miss it
Building a Progressive Web App with HTTP/2
I work for a company called Settled
Instead

➔ Transparent
➔ Online
➔ Everything to help you get “Settled” into your new home
Our Customers

56%
We needed an experience that:

- Is accessible on any device
We needed an experience that:

- Is accessible on any device
- Is fast
We needed an experience that:

- Is accessible on any device
- Is fast
- Works with little or no internet connection
We needed an experience that:

- Is accessible on any device
- Is fast
- Works with little or no internet connection
- Uses one codebase
We needed an experience that:

- Is accessible on any device
- Is fast
- Works with little or no internet connection
- Uses one codebase
- Has an “app-like” feel
Progressive Web Apps
This is our journey
1. Basics of progressive web apps
1. Basics of progressive web apps
2. Fast
1. Basics of progressive web apps
2. Fast
3. Reliable
1. Basics of progressive web apps
2. Fast
3. Reliable
4. Look and feel
The basics of Progressive Web Apps
These apps aren’t packaged and deployed through stores, they’re just websites that took all the right vitamins.

Alex Russell – Google
Building blocks

Website

Service Worker
Think of your web apps requests as planes taking off. ServiceWorker is the air traffic controller that routes the requests.

Jeff Posnick – Google
Behind the scenes
With Service Workers
With Service Workers
With Service Workers
With Service Workers
Service Workers are the key to unlocking the power
The perfect progressive enhancement.
You are already using them
2. Fast
Caching HTTP/2
Caching HTTP/2 Techniques
Caching
With Service Workers
With Service Workers
First Visit
Register
const cacheName = 'pageCache';
const cacheName = 'pageCache';

this.addEventListener('install', event => {
  event.waitUntil(caches.open(cacheName)
    .then(function(cache) { }));
});
const cacheName = 'pageCache';

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

const cacheName = 'pageCache';

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

this.addEventListener('fetch', event => {
    event.respondWith(caches.match(event.request)
        .then()
    );
});
const cacheName = 'pageCache';

this.addEventListener('install', event => {

    event.waitUntil(caches.open(cacheName)
        .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); })
        )
});
Page Load

4.65 secs

2.25 secs

500 ms
HTTP/2
Multiplexing

HTTP 1.1 uses 3 TCP connections to send files:
- `jquery.js`
- `example.css`
- `image.png`

HTTP/2 uses 1 TCP connection to send files:
- `jquery.js`
- `example.css`
- `image.png`
HTTPS
npm install spdy
yarn add

npm install spdy
const spdy = require('spdy');
const express = require('express');
const app = express();
const spdy = require('spdy');
const express = require('express');
const app = express();

app.get('/home', (req, res) => {
  res.status(200).json({message: 'ok'});
});

spdy.createServer({
  key: fs.readFileSync('./privatekey.key'),
  cert: fs.readFileSync('./certificate.crt')
}, app)
  .listen(8012, (err) => {
    if (err) { throw new Error(err); }
    console.log('Listening on port: 8012.'); });
const spdy = require('spdy');
const express = require('express');
const app = express();

app.get('/home', (req, res) => { res.status(200).json({message: 'ok'}); });

spdy.createServer({
  key: fs.readFileSync('./privatekey.key'),
  cert: fs.readFileSync('./certificate.crt')
}, app)
 .listen(8012, (err) => {
  if (err) { throw new Error(err); }
  console.log('Listening on port: 8012.'); });
<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Protocol</th>
<th>Time</th>
<th>Content-Encoding</th>
<th>Timeline – Start Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>hub.settled.co.uk</td>
<td>200</td>
<td>h2</td>
<td>148 ms</td>
<td>gzip</td>
<td>400.00 ms</td>
</tr>
<tr>
<td>css?family=Roboto:regular,bold,italic,thin,light,bold...</td>
<td>200</td>
<td>quic/1+...</td>
<td>17 ms</td>
<td>br</td>
<td></td>
</tr>
<tr>
<td>icon?family=Material+Icons</td>
<td>200</td>
<td>quic/1+...</td>
<td>19 ms</td>
<td>br</td>
<td></td>
</tr>
<tr>
<td>main.css</td>
<td>200</td>
<td>h2</td>
<td>42 ms</td>
<td>gzip</td>
<td></td>
</tr>
<tr>
<td>bece6178363db82a99b8f66ce096f0a2?s=480&amp;...</td>
<td>302</td>
<td>h2</td>
<td>30 ms</td>
<td>br</td>
<td></td>
</tr>
<tr>
<td>analytics.min.js</td>
<td>200</td>
<td>h2</td>
<td>40 ms</td>
<td>gzip</td>
<td></td>
</tr>
<tr>
<td>segmentio_track_links.js</td>
<td>200</td>
<td>h2</td>
<td>53 ms</td>
<td>gzip</td>
<td></td>
</tr>
<tr>
<td>material.min.js</td>
<td>200</td>
<td>h2</td>
<td>56 ms</td>
<td>gzip</td>
<td></td>
</tr>
<tr>
<td>promise.min.js</td>
<td>200</td>
<td>h2</td>
<td>55 ms</td>
<td>gzip</td>
<td></td>
</tr>
<tr>
<td>fetch.js</td>
<td>200</td>
<td>h2</td>
<td>55 ms</td>
<td>gzip</td>
<td></td>
</tr>
<tr>
<td>main.js</td>
<td>200</td>
<td>h2</td>
<td>54 ms</td>
<td>gzip</td>
<td></td>
</tr>
<tr>
<td>eh.png?ssl=1</td>
<td>200</td>
<td>h2</td>
<td>16 ms</td>
<td>gzip</td>
<td></td>
</tr>
<tr>
<td>manifest.json</td>
<td>200</td>
<td>h2</td>
<td>43 ms</td>
<td>gzip</td>
<td></td>
</tr>
<tr>
<td>d-6IYiOFOcCacKzwXSOJ8w1xUtrKptJ_j_Ojans92...</td>
<td>200</td>
<td>quic/1+...</td>
<td>7 ms</td>
<td>br</td>
<td></td>
</tr>
<tr>
<td>oMMgfZMQthOrYQs9n22dcuvwDin1pK8aKteLpeZ...</td>
<td>200</td>
<td>quic/1+...</td>
<td>8 ms</td>
<td>br</td>
<td></td>
</tr>
<tr>
<td>RxZ.Jdnneo3R5zSexge8UUZBw1xUtrKptJ_j_Ojans9...</td>
<td>200</td>
<td>quic/1+...</td>
<td>9 ms</td>
<td>br</td>
<td></td>
</tr>
</tbody>
</table>
spdy: {
    protocols: [ 'h2', 'spdy/3.1', 'http/1.1' ],
    connection: {
        windowSize: 1024 * 1024,
    }
}
spdy: {
  protocols: ['h2', 'spdy/3.1', 'http/1.1'],
  connection: {
    windowSize: 1024 * 1024,
  }
}
spdy: {
  protocols: [ 'h2', 'spdy/3.1', 'http/1.1' ],
  connection: {
    windowSize: 1024 * 1024,
  }
}
Techniques
We used no JavaScript frameworks.
We rendered the page server-side.
We experimented with HTTP/2 Push.
Without H2 Push

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Protocol</th>
<th>Type</th>
<th>Initiator</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>hub.settled.co.uk</td>
<td>200</td>
<td>h2</td>
<td>document</td>
<td>Other</td>
<td>173 ms</td>
</tr>
<tr>
<td>css?family=Roboto:regular,bold,italic,thin...</td>
<td>200</td>
<td>quic/1+sp...</td>
<td>stylesheet</td>
<td>(index):16</td>
<td>149 ms</td>
</tr>
<tr>
<td>icon?family=Material+Icons</td>
<td>200</td>
<td>quic/1+sp...</td>
<td>stylesheet</td>
<td>(index):17</td>
<td>50 ms</td>
</tr>
<tr>
<td>main.css</td>
<td>200</td>
<td>h2</td>
<td>stylesheet</td>
<td>(index):18</td>
<td>370 ms</td>
</tr>
<tr>
<td>d21edb4d07c56dc96de0d594d9c16a...</td>
<td>302</td>
<td>h2</td>
<td>text/html</td>
<td>(index):53</td>
<td>65 ms</td>
</tr>
<tr>
<td>material.min.js</td>
<td>200</td>
<td>h2</td>
<td>script</td>
<td>(index):697</td>
<td>246 ms</td>
</tr>
<tr>
<td>promise.min.js</td>
<td>200</td>
<td>h2</td>
<td>script</td>
<td>(index):698</td>
<td>84 ms</td>
</tr>
<tr>
<td>fetch.js</td>
<td>200</td>
<td>h2</td>
<td>script</td>
<td>(index):699</td>
<td>168 ms</td>
</tr>
<tr>
<td>main.js</td>
<td>200</td>
<td>h2</td>
<td>script</td>
<td>(index):700</td>
<td>99 ms</td>
</tr>
<tr>
<td>be.png?ssl=1</td>
<td>200</td>
<td>h2</td>
<td>webp</td>
<td><a href="https://s.gravatar.co">https://s.gravatar.co</a>...</td>
<td>76 ms</td>
</tr>
<tr>
<td>d-6IYpiOFocCacKzwxXSOJBw1xU1rKp...</td>
<td>200</td>
<td>quic/1+sp...</td>
<td>font</td>
<td>(index):-Infinity</td>
<td>410 ms</td>
</tr>
<tr>
<td>oMMgfZMqthOryQo9n22dcuvvDin1pK...</td>
<td>200</td>
<td>quic/1+sp...</td>
<td>font</td>
<td>(index):-Infinity</td>
<td>272 ms</td>
</tr>
<tr>
<td>2tsd397wLxj96qwHyNlkoxHyhyJsPsvc4...</td>
<td>200</td>
<td>quic/1+sp...</td>
<td>font</td>
<td>(index):-Infinity</td>
<td>417 ms</td>
</tr>
<tr>
<td>RxZJdnzeo3R6zSexge8UUZBw1xU1rK...</td>
<td>200</td>
<td>quic/1+sp...</td>
<td>font</td>
<td>(index):-Infinity</td>
<td>417 ms</td>
</tr>
<tr>
<td>favicon.ico</td>
<td>200</td>
<td>h2</td>
<td>x-icon</td>
<td>Other</td>
<td>49 ms</td>
</tr>
</tbody>
</table>

16 requests | 106 KB transferred | Finish: 1.25 s | DOMContentLoaded: 589 ms | Load: 1.02 s
With H₂ Push

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Protocol</th>
<th>Type</th>
<th>Initiator</th>
<th>Time</th>
<th>Timeline -- Start Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>help</td>
<td>200</td>
<td>h2</td>
<td>document</td>
<td>Other</td>
<td>145 ms</td>
<td></td>
</tr>
<tr>
<td>main.css</td>
<td>200</td>
<td>h2</td>
<td>stylesheet</td>
<td>Push / help:18</td>
<td>12 ms</td>
<td></td>
</tr>
<tr>
<td>d21ed6b4d07c56dc96de0d594d9c16a...</td>
<td>302</td>
<td>h2</td>
<td>text/html</td>
<td>help:54</td>
<td>22 ms</td>
<td></td>
</tr>
<tr>
<td>components.js</td>
<td>200</td>
<td>h2</td>
<td>script</td>
<td>Push / help:211</td>
<td>9 ms</td>
<td></td>
</tr>
<tr>
<td>material.min.js</td>
<td>200</td>
<td>h2</td>
<td>script</td>
<td>Push / help:213</td>
<td>9 ms</td>
<td></td>
</tr>
<tr>
<td>promise.min.js</td>
<td>200</td>
<td>h2</td>
<td>script</td>
<td>Push / help:214</td>
<td>9 ms</td>
<td></td>
</tr>
<tr>
<td>fetch.js</td>
<td>200</td>
<td>h2</td>
<td>script</td>
<td>Push / help:215</td>
<td>9 ms</td>
<td></td>
</tr>
<tr>
<td>main.js</td>
<td>200</td>
<td>h2</td>
<td>script</td>
<td>Push / help:216</td>
<td>9 ms</td>
<td></td>
</tr>
<tr>
<td>be.png?ssl=1</td>
<td>200</td>
<td>h2</td>
<td>webp</td>
<td>1845 ms</td>
<td>9 ms</td>
<td></td>
</tr>
<tr>
<td>e461d1af859b6e6f5d6ce8c46363f5c...</td>
<td>200</td>
<td>quic/1+sp...</td>
<td>font</td>
<td>help:--Infinity</td>
<td>9 ms</td>
<td></td>
</tr>
<tr>
<td>0d5e3a774c26a4b120f5404c5102e40...</td>
<td>200</td>
<td>quic/1+sp...</td>
<td>font</td>
<td>help:--Infinity</td>
<td>12 ms</td>
<td></td>
</tr>
<tr>
<td>Chat / C20 - C22 - C21 - C21 - C21 - C21 - C21...</td>
<td>200</td>
<td></td>
<td>font</td>
<td></td>
<td>10 ms</td>
<td></td>
</tr>
</tbody>
</table>

18 requests | 549 KB transferred | Finish: 317 ms | DOMContentLoaded: 204 ms | Load: 235 ms

Assets are sent with HTML
Assets are already downloaded
HTTP/2

Service Workers
HTTP/2 push is tougher than I thought

"HTTP/2 push will solve that" is something I've heard a lot when it comes to page load performance problems, but I didn't know much about it, so I decided to dig in.

HTTP/2 push is more complicated and low-level than I initially thought, but what really caught me off-guard is how inconsistent it is between browsers – I'd assumed it was a done deal & totally ready for production.

This isn't an "HTTP/2 push is a douchebag" hatchet job – I think HTTP/2 push is really powerful and will improve over time, but I no longer think it's a silver bullet from a golden age of quicker and richer web experiences.
H₂ Push

bit.ly/server-push-http2
We also used prefetch.
Thank you

We'll let you know by email and text once we've heard back from the seller

Settled makes buying a home easier, faster and smarter, with our lovely, friendly app

And the best mortgage rates, with no fees, no paperwork and no tedious hold music
Thank you
We’ll let you know by email and text once we’ve heard back from the seller

Settled makes buying a home easier, faster and smarter, with our lovely, friendly app

And the best mortgage rates, with no fees, no paperwork and no tedious hold music
Brotli = Better compression
Brotli ~ 10%
We tested using Lighthouse.
Audit your Progressive Web App

These audits validate the aspects of a Progressive Web App, as specified by the baseline "PWA Checklist."

- Configured for a custom splash screen
- Failure: Manifest does not have icons at least 512px.

View 10 passed items
Manual checks to verify

Generated by Lighthouse 2.0.0 on Jun 16, 2017, 9:48 AM GMT+1 | File an Issue
15 x less data
15x less data
3x faster
With Service Workers
this.addEventListener('fetch', event => {

});
this.addEventListener('fetch', event => {
    // Check if the user navigated
    if (event.request.method === 'GET' &&
        event.request.headers.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');
      })
    );
  }
});
Hang tight!

It looks like you are offline. You can continue using the app as normal even while you are offline, although you may have limited functionality.
Offline Applications

bit.ly/offline-page
Offline Analytics

github.com/GoogleChrome/sw-helpers
CAUTION

AVAILANCHE DANGER
Test your Service Worker code

bit.ly/sw-testing
3rd Party Scripts
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);
    });
}

self.addEventListener('fetch', event => {
    event.respondWith(Promise.race([[timeout(6000), fetch(event.request.url)]]));
});
bit.ly/sw-toolbox
toolbox.router.get('/(.*)', global.toolbox.cacheFirst, { });
toolbox.router.get('/(.*)', global.toolbox.cacheFirst, {
origin: /\.*\.com$/,
});
toolbox.router.get('/(.*)', global.toolbox.cacheFirst, {
  origin: /\.google\.com$/,
  cache: {
    name: 'javascript',
    networkTimeoutSeconds: 4
  }
});
bit.ly/sw-timeout
Look & Feel
Look & Feel

❖ Manifest file
Look & Feel

- Manifest file
- Add to Home Screen
Manifest

A simple JSON file that allows you to control how your app appears to the user.
Look and feel
{
  "lang": "en",
  "background_color": "#09adec",
  "name": "Settled",
  "short_name": "Settled Dashboard",
  "display": "standalone",
  "icons": [
    {
      "src": "./images/logo-144.png",
      "sizes": "144x144",
      "type": "image/png"
    }
  ],
  "start_url": "/?start=a2hs",
  "orientation": "portrait"
}
{
    "lang": "en",
    "background_color": "#09adec",
    "name": "Settled",
    "short_name": "Settled Dashboard",
    "display": "standalone",
    "icons": [
        {
            "src": "/images/logo-144.png",
            "sizes": "144x144",
            "type": "image/png"
        }
    ],
    "start_url": "/?start=a2hs",
    "orientation": "portrait"
}
<!doctype html>
<html lang="en">
<head>
  <link rel="manifest" href="manifest.json">
</head>
<body>
</body>
</html>
Manifest

bit.ly/manifest-manifest-json
Add to home screen

1. You need a manifest.json file with a start URL and 144x144 PNG icon
Add to home screen

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

1. You need a manifest.json file with a start URL and 144x144 PNG icon
2. Your site is using a Service Worker running over HTTPS
3. The user has visited your site at least twice, with at least five minutes between visits.

{
    lang: "en",
    background_color: "#09adec",
    name: "Settled",
    short_name: "Settled Dashboard",
    display: "standalone",
    icons: [
        {
            src: "./images/logo-144.png",
            sizes: "144x144",
            type: "image/png"
        }
    ],
    start_url: "/?start=a2hs",
    orientation: "portrait"
}
Summary
Summary

❖ Fast
Summary

❖ Fast
❖ Reliable
Summary

❖ Fast
❖ Reliable
❖ Look & Feel
We are still learning
Thank you!

@deanohume