This month I learned about

ServiceWorkers

Remember the HTML5 AppCache?

```
CACHE MANIFEST
# 2010-06-18:v2

CACHE:
index.html
stylesheet.css
images/logo.png
scripts/main.js

# Resources that require the user to be online.
NETWORK:
*

FALLBACK:
/main.py /static.html
```

A **Worker**.

A JavaScript-driven,
Client-side,
Promise-based,
Network-layer
Interceptor or **Proxy**.

```
CACHE MANIFEST
# 2010-06-18:v2

CACHE:
index.html
stylesheet.cs
images/logo
scripts/m    js

# Resources that requir   e us   o be online.
N    RK:
*

FALL
      /static.html
```

Whatever you want it to be?

/js/addWorker.js

```
if ('serviceWorker' in navigator) {
  navigator.serviceWorker.register(
    '/sw.js',
    {scope: '/'}
  ).then(_successFn, _failureFn);
}
```

/sw.js…

```
this.addEventListener('install',  _installFn);
this.addEventListener('fetch',    _fetchFn);
this.addEventListener('activate', _activateFn);
```

le demo…

The basic components of

ServiceWorkers

Cache

Fetch

Promises

Response

Request

What you can do with

ServiceWorkers

Potential Strategies

Offline-First™	Client-Hints
Progressive Caching	Asset Routing
Push notifications	Background Sync

Potential Strategies

Offline-First™

Client-Hints

Progressive Caching

Asset Routing

Push Notifications

Background Sync

and MORE!

Check if you're ready for

ServiceWorkers

Do you have…

HTTPS?

A well-implemented caching policy?

Intention & Interest?

A note on browsers

A note on browsers

Progressive Enhancement

What we learned by using ServiceWorkers

Things you will want:

An array of all resources, routes, assets, etc.

An exclusion policy

ENV-related config switches

Versioning of Caches

Invalidation of Caches

le code…

Gotchas:

Bad scope

Asset-hash on sw.js

Caching of sw.js

Too much pre-cache

Session weirdness

Debugging

ServiceWorkers ServiceWorkers

ServiceWorkers!

Because…

Progressive Enhancement

Imperative and Flexible

Web Performance

Future track