

Service Workers and Offline Support



Matthew Soucoup

Principal – Code Mill Technologies

@codemillmatt | www.codemillmatt.com

Service Workers

“Just” a JavaScript file

Event-driven

Introduces new capabilities

Intercept, modify, and respond to network requests



Why Service Workers Are Important

Make app offline capable and resilient

Run in separate thread

Enable push notifications

Enable background sync



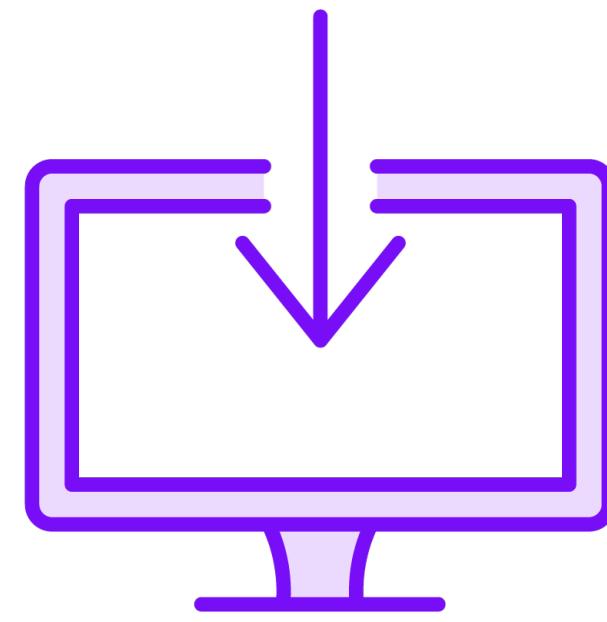


Getting ready for offline use

- Prep app for offline
- Verify everything is working

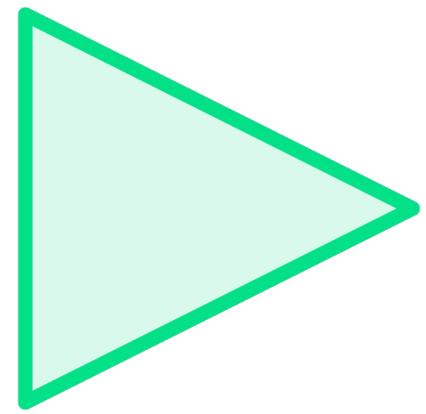


Events for Offline and Data Caching



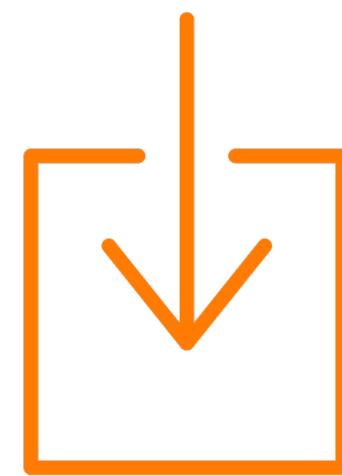
Install

Service worker installed for first time



Activate

Service worker is functional

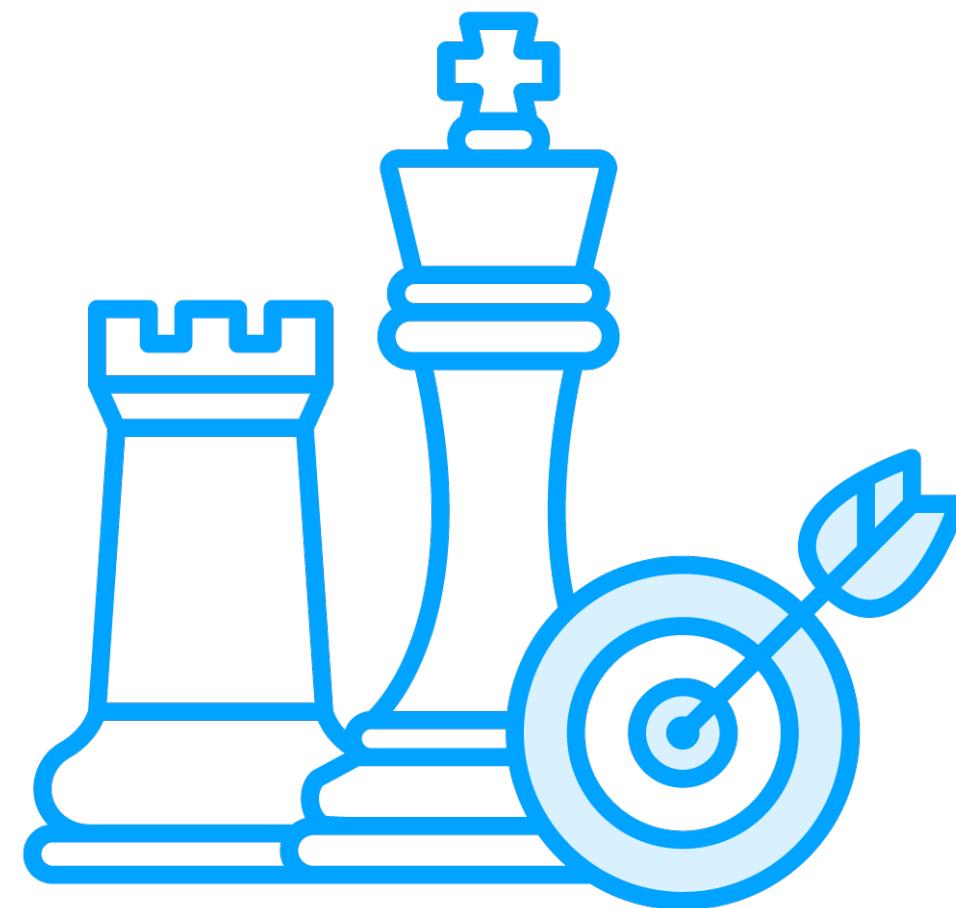


Fetch

App requests something from network



Caching Strategies



Put enough in cache so app can display something immediately

Download just enough dynamic data

Cache-first pattern

Stale-while-invalidate pattern

Great offline page





Page and asset caching

- Specify assets to include
- Specify assets to never include





Caching data

- Frequently used data



Online and Offline Detection

client.js

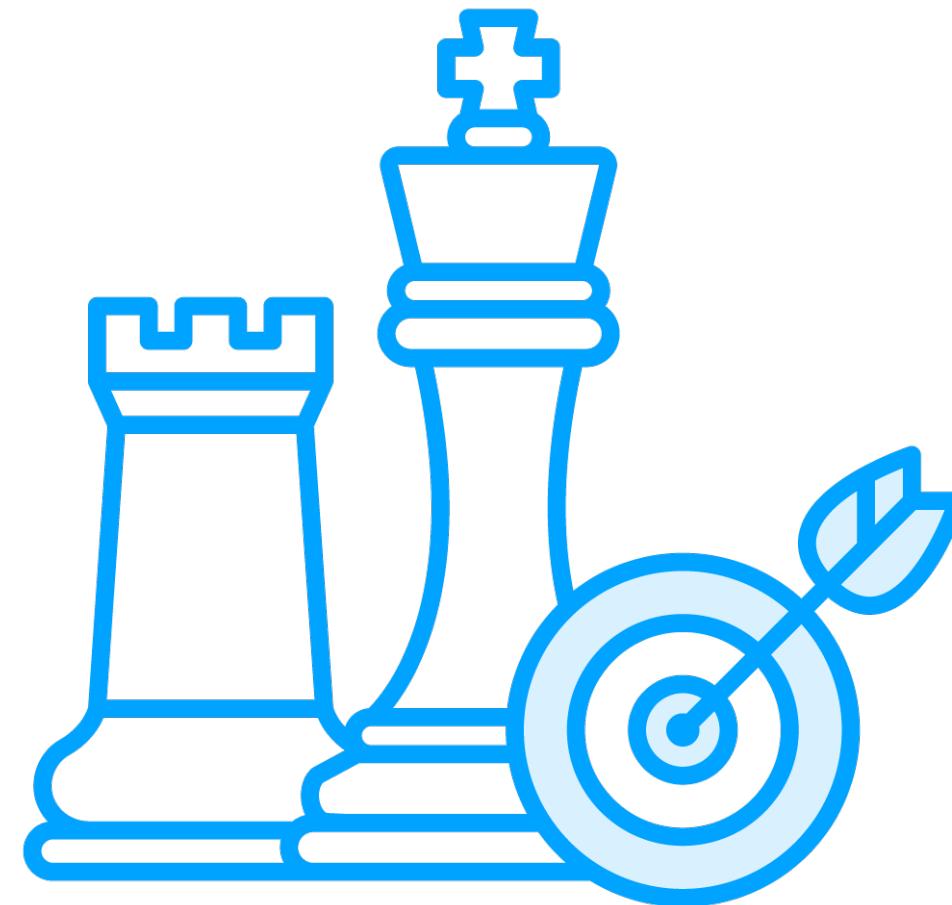
```
window.addEventListener('offline',
  function () =>
    navigator.serviceWorker
      .controller.postMessage(
        {
          type: 'IS_OFFLINE'
          // more properties if needed
        }
      )
);
```

Service-worker.published.js

```
self.addEventListener('message',
  (event) => {
    if (event.data &&
        event.data.type === 'IS_OFFLINE')
    {
      // do something
    }
});
```



Offline Considerations



Not all apps benefit from offline support

- Weigh benefits against complexity

Support beneficial when:

- Primary data local to browser
- App caches data for offline navigation
- App must load immediately



Summary



Service workers key to enabling offline capabilities

Offline adds benefits with complexity

Caching strategies

Page and data caching

Offline considerations



Up Next:

Push Notifications

