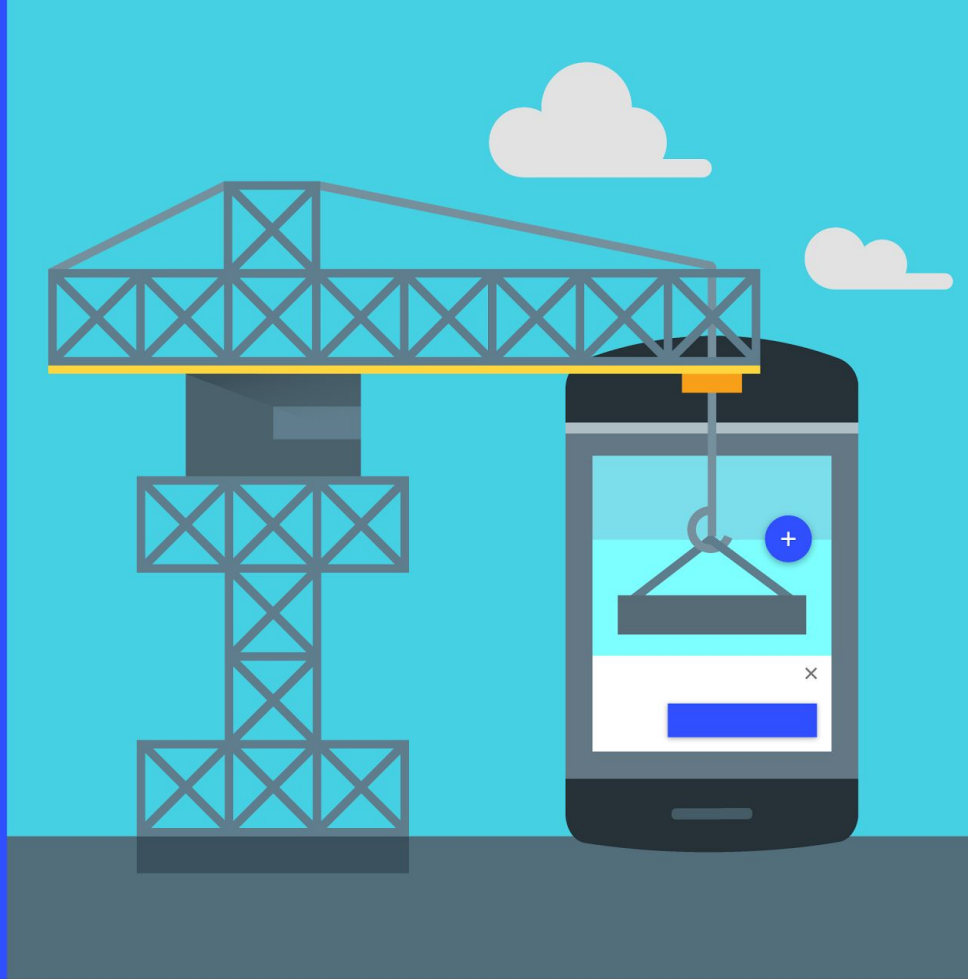


# Caching files with the service worker



# What we will cover

- What is the Cache?
- When to cache resources
  - On install
  - On user interaction
  - On network response
- Serving files from the cache
  - Cache falling back to network
  - Network falling back to cache
  - Cache then network
- Removing outdated caches

# What is the Cache?

- Storage mechanism for request/response object pairs

# When to store resources

# On install - caching the application shell

```
self.addEventListener('install', function(e) {  
  e.waitUntil(  
    caches.open(cacheName).then(function(cache) {  
      return cache.addAll(  
        [ '/css/bootstrap.css',  
          '/css/main.css',  
          '/js/jquery.min.js',  
          '/offline.html' ]  
      );  
    ...});  
});
```

# On user interaction

```
document.querySelector('.article').addEventListener('click',
function(event) {
    event.preventDefault();
    var id = this.dataset.articleId;
    caches.open('mysite-article-' + id).then(function(cache) {
        fetch('/get-article-urls?id=' +
id).then(function(response) {
            return response.json();
        }).then(function(urls) {
            cache.addAll(urls);
        ...});
    });
```

# On network response

```
self.addEventListener('fetch', function(event) {
  event.respondWith(
    caches.open('mysite-dynamic').then(function(cache) {
      return cache.match(event.request)
        .then(function (response) {
          return response ||
            fetch(event.request).then(function(response) {
              cache.put(event.request, response.clone());
              return response;
            });
        });
    });
  ... });
```

# Serving files from the cache



# Cache falling back to the network

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

# Network falling back to the cache

```
self.addEventListener('fetch',  
function(event) {  
    event.respondWith(  
        fetch(event.request).catch(function() {  
            return caches.match(event.request);  
        })  
    );  
});
```

# Cache then network (part 1)

```
var networkDataReceived = false;
startSpinner();
var networkUpdate = fetch('/data.json')
  .then(function(response) {
    return response.json();
  }).then(function(data) {
    networkDataReceived = true;
    updatePage(data);
  });
```

# Cache then network (part 2)

```

caches.match('/data.json').then(function(response) {
  return response.json();
}).then(function(data) {
  if (!networkDataReceived) {
    updatePage(data);
  }
}).catch(function() {
  return networkUpdate;
}).catch(showErrorMessage).then(stopSpinner());

```

# Generic fallback

```
self.addEventListener('fetch', function(event) {  
  event.respondWith(  
    caches.match(event.request).then(function(response)  
    {  
      return response || fetch(event.request);  
    }).catch(function() {  
      return caches.match('/offline.html');  
    ... }));  
});
```

# Removing outdated caches

```
self.addEventListener('activate', function(event)
{
    event.waitUntil(
        caches.keys().then(function(cacheNames) {
            return Promise.all(
                cacheNames.filter(function(cacheName) {
                }).map(function(cacheName) {
                    return caches.delete(cacheName);
                ...}));
        });
    );
```