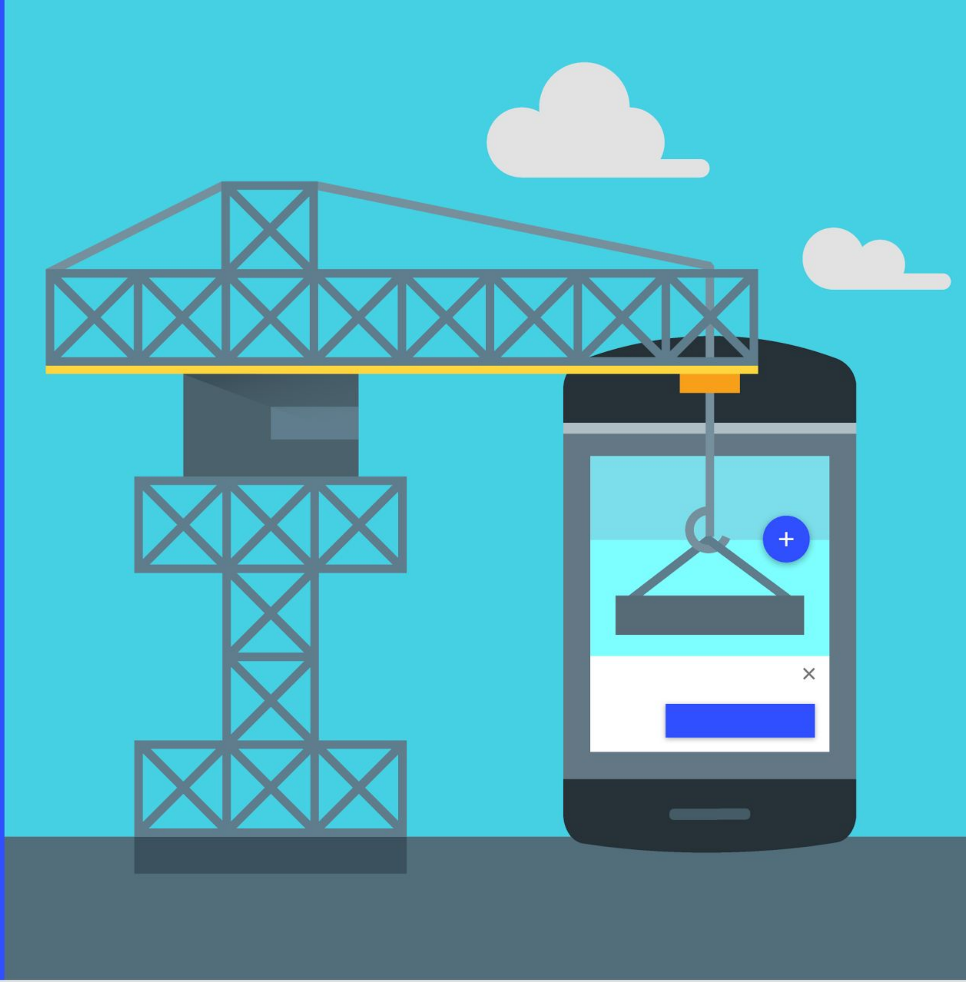
Automating the creation of a service worker

sw-precache and sw-toolbox



Getting Started: Directory Structure

```
$ mkdir my-project

$ cd my-project
```

Getting Started: Directory Structure

```
$ mkdir img

$ mkdir js

$ mkdir css
```

Getting Started: NPM Init

```
$ npm init
```

Getting Started: Installing Gulp

```
$ npm i -g gulp-cli
```

Getting Started: Installing Gulp

```
$ npm install --save-dev gulp
```

Getting Started: installing tools

```
$ npm install --save-dev sw-precache
$ npm install --save-dev sw-toolbox
```

sw-toolbox

Creating routes with sw-toolbox

```
toolbox.router.get('/my-app/index.html',
global.toolbox.networkFirst);
```

sw-toolbox caching strategies

- toolbox.networkFirst
- toolbox.cacheFirst
- toolbox.fastest
- toolbox.cacheOnly
- toolbox.networkOnly

Creating routes with sw-toolbox

Creating routes with sw-toolbox

- Same Origin
- Express like
- Regular expression

sw-toolbox same origin routes

```
toolbox.router.get('my-app/index.html',
global.toolbox.networkFirst);
```

sw-toolbox: Express like Routing

```
toolbox.router.get('/(.*)', global.toolbox.cacheFirst, {
  origin: /\.googleapis\.com$/
});


toolbox.router.get('/.*fly$/', global.toolbox.cacheFirst);
```

sw-toolbox: Regular Expression Routing

```
toolbox.router.get(/index.html$/, function(request) {
  return new Response('Handled a request for ' + request.url);
});

toolbox.router.post(/^https://api.flickr.com\//,
global.toolbox.networkFirst);
```

sw-toolbox: cache control

```
toolbox.router.get('/(.*)', global.toolbox.cacheFirst, {
  cache: {
    name: 'googleapis',
    maxEntries: 10,
    maxAgeSeconds: 86400
  },
  origin: /\.googleapis\.com$/
});
```

sw-precache

sw-precache from command line

```
{  "staticFileGlobs": [
     "app/index.html",
     "app/js/main.js",
     "app/css/main.css",
     "app/img/**/*.{svg,png,jpg,gif}"
  ],
  "stripPrefix": "app/" }
```

Running sw-precache from command line

```
$ sw-precache --config=path/to/sw-precache-config.json --verbose
```

Adding sw-precache to a build system

```
var path = require('path');
var swPrecache = require('sw-precache');


var paths = {
  src: 'app/'
};
```

adding sw-precache into a build system

```
gulp.task('generate-sw', (callback) => {
  swPrecache.write(path.join(paths.src, 'sw.js'),
{
    staticFileGlobs: [    //1
       paths.src + 'index.html',
       paths.src + 'js/main.js',
       paths.src + 'css/main.css',
       paths.src + 'img/**/*.{svg,png,jpg,gif}'
```

```
(function(global) {
  'use strict';
  global.toolbox.router.get('/(.*)',
global.toolbox.networkFirst);


})(self);
```