

第 8 章

メルカリ カウルと SketchTool

8.1 はじめに

こんにちは。メルカリ カウルのデザイナー akey (@urakey^{*1}) です。

この章では、メルカリ カウルにおいて SketchTool をどのように取り入れているか紹介したいと思います。Sketch アプリ^{*2}の使い方や SketchTool^{*3}の細かい説明などには触れていませんので予めご了承ください。

本章が『Sketch ってコマンドラインから使えるんだな。デザイナーと相談して取り入れてみようかな。』『コマンドラインはよくわからないけど、書き出し設定はわかるし、エンジニアに相談してみようかな?』など、そういったほんの少しのきっかけになれば幸いです。

8.2 ところで SketchTool ってなに？

SketchTool とは、Sketch にバンドルされているコマンドラインユーティリティです。Sketch アプリを起動せずとも Sketch ドキュメントのレイヤー構造を確認できたり (JSON 形式)、アセットの書き出しができたりします。

Sketch ドキュメントのガイドラインを決めている場合、命名ルールに則った記述をしているかどうかチェックするような使い方もできますし、GitHub での操作をトリガーに CI ツール上でアセットの書き出しコマンドを実行するなどといったことが可能になります。

^{*1} <https://twitter.com/urakey>

^{*2} <https://www.sketchapp.com/>

^{*3} <https://developer.sketchapp.com/guides/sketchtool/>

SketchTool をインストールする

SketchTool は Sketch にバンドルされているのでインストールが不要です。Sketch.app/Contents/Resources/sketchtool/bin/sketchtoolにあります。

Sketch 48 まではこれを/usr/local/bin/配下にコピーする install.shも同梱されていたのですが、Sketch 49 でなぜかなくなってしまいました。そこで、筆者は以下のようにシンボリックリンクを貼って使用しています。

```
$ ln -s /Applications/Sketch.app/Contents/Resources/sketchtool/bin/sketchtool \
/usr/local/bin/sketchtool
```

SketchTool が使えるかは、以下のコマンドで確認できます。

```
$ sketchtool -v
sketchtool Version 49.2 (51160)
```

SketchTool で利用できるコマンドは、sketchtool helpで確認できます。

```
$ sketchtool help
Commands:
  dump                Dump out the structure of a document as JSON.

  export artboards   Export one or more artboards
  export layers      Export one or more layers
  export pages       Export an area from one or more pages
  export preview     Export a preview image for a document
  export slices      Export one or more slices

  help               Show this help message.

  list artboards     List information on the document's artboards.
  list formats       List the supported export formats.
  list layers        List information on all of the document's layers.
  list pages         List information on the document's pages.
  list slices        List information on the document's slices.

  metadata           List the metadata for a document.
  run                Run a command from a plugin, inside Sketch.
  show               Show the location of the various sketch folders.
```

8.3 メルカリ カウルでの使い方

メルカリ カウルではアセットの書き出しに SketchTool を取り入れています。以下のコマンドで、書き出し設定済みのアートボードが一括で書き出されます*4。

```
$ sketchtool export artboards path/to/document.sketch
```

SketchTool の書き出しコマンドを毎回打つのが面倒なときは、`package.json`の `npm scripts`*5で管理するのがおすすめです:D

メルカリ カウルのデザインデータ運用

メルカリ カウルではデザインデータ運用の最低限のルールとして次のようなものを定めています。

- UI のデザインは主に Sketch を使用
- デザインデータのバージョン管理は GitHub で行う
- デザインレビューは GitHub でプルリクエストする形で依頼する
- Sketch ドキュメントはガイドラインに従ってメンテナンスする

実際に運用してみて「冗長だな」と感じるものは、デザイナー同士で相談しながら都度ルールから外しています。

メルカリ カウルで基本としている Sketch ドキュメントのガイドラインは「[itachoco Sketch Guideline^a](#)」です。メルカリ カウルと一緒に作っているデザイナーの@wataame^bと 2 人でつくりました。概要は「[Sketch の命名ルールなどをまとめた Guideline をつくりました - itachoco Sketch Guideline v0.1^c](#)」を参照ください。

^a <https://github.com/itachoco-labs/ita-sketch-guideline>

^b <https://twitter.com/wataame>

^c <https://note.mu/wataame/n/n8a929d010e90>

*4 オプションで書き出し形式を指定することも可能です。

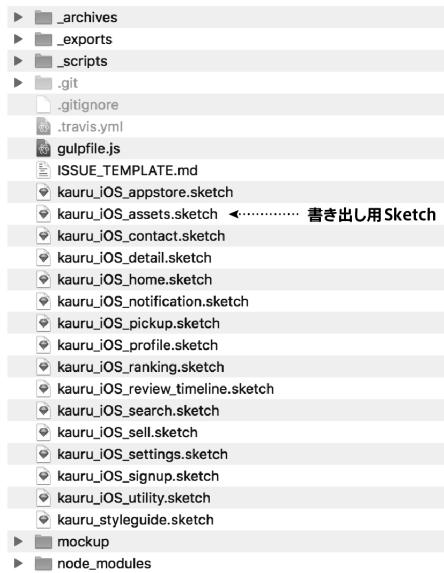
*5 <https://docs.npmjs.com/misc/scripts>

Sketch ドキュメントの管理

メルカリ カウルでの Sketch ドキュメントの管理について触れたいと思います。

デザインデータのバージョン管理を GitHub で行うにあたり、プラットフォームごとにリポジトリを分けて運用しています。こうすることで、それぞれのプラットフォームのエンジニアが必要なデザインデータだけをプルできるようにしています。また、Sketch ドキュメントは画面単位で分割しています。アセットとして書き出すものはそれ専用の Sketch ドキュメントとして用意しています (図 8.1)。

iOS



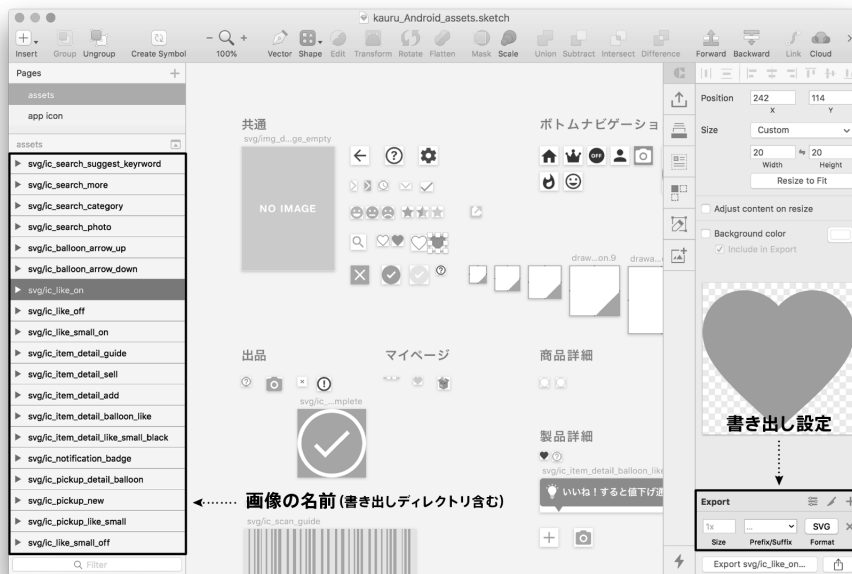
Android



▲ 図 8.1 Sketch ドキュメントの分割

書き出し用 Sketch ドキュメント

書き出したいアセットごとにアートボードを用意し、それぞれに書き出し設定を行います。書き出し先のディレクトリを細かく指定したい場合はアートボード名に設定します (図 8.2)。



▲ 図 8.2 Sketch ドキュメント: Android のイメージ

アイコンの追加や削除といったさまざまな変更を1番理解しているのはデザイナーです。デザイナーは書き出しの設定を含め、Sketch ドキュメントの最新の状態を担保することに責任を持ちます。(と、メルカリ カウルのチームメンバーは相談して決めました)

書き出しの実行と利用

gulp-sketch^{*6}を利用して書き出した画像を軽量化するタスクを追加しています。軽量化タスクは書き出し時点ではなく実装サイドに持つこともできますので、これはプロジェクトやチームによってどちらで持つべきか相談して決めましょう。

▼リスト 8.1 メルカリ カウルの gulpfile.js

```
var gulp = require('gulp');
var $ = require('gulp-load-plugins')({
  pattern: ['gulp-*', 'gulp.*'],
  replaceString: /\bgulp[^\-\.]/
});
```

^{*6} <https://github.com/cognitom/gulp-sketch>

```
    });  
    var runSequence = require('run-sequence');  
    var minimist = require('minimist');  
    var PATHS = {  
      src: './kauru_Android_assets.sketch',  
      dest: './_exports',  
    };  
  
    var argv = minimist(process.argv.slice(2));  
  
    // Asset 以外を書き出したいときは以下のオプションを渡す  
    // * --file 'ファイル名'  
    gulp.task('sketch', function() {  
      var src = PATHS.src;  
      if (argv.file) src = './' + argv.file;  
      var options = {  
        export: 'slices',  
      };  
      return gulp.src(src)  
        .pipe($.sketch(options))  
        .pipe(gulp.dest(PATHS.dest))  
        .pipe($.filelog());  
    });  
  
    gulp.task('imageoptim', function() {  
      return gulp.src(PATHS.dest + '/*.png')  
        .pipe($.imageoptim.optimize())  
        .pipe(gulp.dest(PATHS.dest));  
    });  
  
    gulp.task('svgmin', function () {  
      return gulp.src(PATHS.dest + '/*.svg')  
        .pipe($.svgmin())  
        .pipe(gulp.dest(PATHS.dest));  
    });  
  
    gulp.task('export', function() {  
      runSequence('sketch', ['imageoptim', 'svgmin']);  
    });
```

どのデザインリポジトリでも、以下のコマンドでアセットの一括書き出しと画像軽量化のタスクが実行されます。

```
$ gulp sketch export
```

リスト 8.1 の設定をそのまま利用する場合、次のパッケージをインストールしてください。

- `gulp`^{*7}

^{*7} <https://gulpjs.com/>

- [gulp-load-plugins](#)^{*8}
- [gulp-filelog](#)^{*9}
- [gulp-imageoptim](#)^{*10}
- [gulp-svgmin](#)^{*11}
- [run-sequence](#)^{*12}
- [minimist](#)^{*13}

メルカリ カウルでは、デザインデータが `master` にマージされると CI ツールが動き、開発用のリポジトリに変更のあったアセットをプルリクエストするような仕組みも導入しています。詳しくは @yudoufu^{*14} が以前発表した「**Project 開始後に導入する Assets の自動共有**^{*15}」を見てみてくださいね。

8.4 おわりに

今回紹介した内容は、メルカリ カウルにおいて SketchTool をどう使っているかという点に集約しました。インターネットにはもっと詳しく書かれた記事がたくさんありますので、Sketch を使うことがあったら、ぜひ探してみてくださいね。本章で気になる点がありましたら、筆者である akey (@urakey^{*16}) まで、いつでもお気軽にご連絡ください。

最後になりますが「二ノ国 II レヴァナントキングダム^{*17}」クリアしました！プロダクトを作るときと同じ気持ちで二ノ国を愛でながら大きくしてきました。これからゆっくりとクリア後の世界を楽しみます♪

*8 <https://github.com/jackfranklin/gulp-load-plugins>

*9 <https://github.com/adamayres/gulp-filelog>

*10 <https://github.com/allbitsnbytes/gulp-imageoptim>

*11 <https://github.com/ben-eb/gulp-svgmin>

*12 <https://github.com/OverZealous/run-sequence>

*13 <https://github.com/substack/minimist>

*14 <https://twitter.com/yudoufu>

*15 <https://speakerdeck.com/yudoufu/projectkai-shi-hou-nidao-ru-suruassetsfalsezi-dong-gong-you>

*16 <https://twitter.com/urakey>

*17 <http://www.ninokuni.jp/rk/>