

[< Back to Mobile Web Specialist](#)

Restaurant Reviews: Stage 3

REVIEW

CODE REVIEW 3

HISTORY

▶ client/js/restaurant_review.js 2

▼ gulpfile.js 1

```
1 const gulp = require('gulp');
```



AWESOME

Tip ⚡

A popular option to further optimize the performance is to "gzip" the asset to reduce their sizes in production on the concept and examples:

- [Two Quick Ways To Reduce React App's Size In Production](#)

I usually use `webpack`, so this package is my go-to for this approach:

- [compression-webpack-plugin](#)

```
2 const del = require('del');
3 const copy = require('gulp-copy');
4 const rename = require('gulp-rename');
5 const imageMin = require('gulp-imagemin');
6 const htmlMin = require('gulp-htmlmin');
7 const responsive = require('gulp-responsive');
8 const minifyCss = require('gulp-clean-css');
9 const minifyJs = require('gulp-uglify-es').default;
```

```
10 const runSequence = require('run-sequence');
11 const pathRoot = './client';
12 const pathSourceImages = `${pathRoot}/img-src`;
13 const pathDestImages = `${pathRoot}/img`;
14 const pathSourceHtml = `${pathRoot}/html-src`;
15 const pathDestHtml = `${pathRoot}`;
16 const pathCss = `${pathRoot}/css`;
17 const pathJs = `${pathRoot}/js`;
18 const nodeModules = 'node_modules';
19
20
21 /* images */
22
23 gulp.task('clean:images', function () {
24   return del(`${pathDestImages}/**/*`);
25 });
26
27 gulp.task('copy:images:restaurant', function () {
28   return gulp.src([
29     `${pathSourceImages}/restaurant.png`,
30   ])
31     .pipe(responsive({
32       '*.png': [
33         {
34           width: 128,
35           quality: 33,
36           rename: { suffix: '-128' }
37         },
38         {
39           width: 256,
40           quality: 33,
41           rename: { suffix: '-256' }
42         },
43         {
44           width: 512,
45           quality: 33,
46           rename: { suffix: '-512' }
47         },
48         {
49           width: 720,
50           quality: 33,
51           rename: { suffix: '' }
52         }
53       ]
54     }))
55     .pipe(imageMin({ verbose: false }))
56     .pipe(gulp.dest(`${pathDestImages}`));
57 });
58
59 gulp.task('copy:images:buttons', function () {
60   return gulp.src([
61     `${pathSourceImages}/gold-medal.png`,
62     `${pathSourceImages}/write.png`,
63     `${pathSourceImages}/delete.png`
64   ])
65     .pipe(responsive({
66       '*.png': [
67         {
68           width: 32,
69           quality: 33,
70           rename: { suffix: '-32' }
71         },
72         {
73           width: 48,
74           quality: 33,
```

```
75         rename: { suffix: '-48' }
76     },
77     {
78         width: 64,
79         quality: 33,
80         rename: { suffix: '-64' }
81     }
82 ]
83 )))
84 .pipe(imageMin({ verbose: false }))
85 .pipe(gulp.dest(`${pathDestImages}`));
86 });
87
88 gulp.task('copy:images:app', function () {
89     return runSequence(
90         'copy:images:restaurant',
91         'copy:images:buttons'
92     );
93 });
94
95
96 gulp.task('copy:images:data', function () {
97     return gulp.src(`${pathSourceImages}/**/*.jpg`)
98         .pipe(responsive({
99             '*.jpg': [
100                 {
101                     width: 200,
102                     quality: 33,
103                     rename: { suffix: '-200' }
104                 },
105                 {
106                     width: 300,
107                     quality: 33,
108                     rename: { suffix: '-300' }
109                 },
110                 {
111                     width: 400,
112                     quality: 33,
113                     rename: { suffix: '-400' }
114                 },
115                 {
116                     width: 500,
117                     quality: 33,
118                     rename: { suffix: '-500' }
119                 },
120                 {
121                     width: 600,
122                     quality: 33,
123                     rename: { suffix: '-600' }
124                 },
125                 {
126                     width: 800,
127                     quality: 33,
128                     rename: { suffix: '' }
129                 }
130             ]
131         })))
132     .pipe(imageMin({ verbose: false }))
133     .pipe(gulp.dest(`${pathDestImages}`));
134 });
135
136 gulp.task('build:images', function () {
137     return runSequence(
138         'clean:images',
139         'copy:images:app',
```

```
140     'copy:images:data'
141   );
142 });
143
144
145 /* html */
146
147 gulp.task('clean:html', function () {
148   return del(`${pathDestHtml}/*.html`);
149 });
150
151 gulp.task('minify:html', function () {
152   return gulp.src(`${pathSourceHtml}/*.html`)
153     .pipe(htmlMin({collapseWhitespace: true}))
154     .pipe(gulp.dest(pathDestHtml));
155 });
156
157 gulp.task('build:html', function () {
158   return runSequence(
159     'clean:html',
160     'minify:html'
161   );
162 });
163
164
165 /* css */
166
167 gulp.task('clean:css', function () {
168   return del(`${pathCss}/*.min.css`);
169 });
170
171 gulp.task('minify:css', function () {
172   return gulp.src([pathCss + '/*.css', !pathCss + '/*.min.css'])
173     .pipe(minifyCss())
174     .pipe(rename(function (path) {
175       path.extname = '.min.css';
176     }))
177     .pipe(gulp.dest(pathCss));
178 });
179
180 gulp.task('build:css', function () {
181   return runSequence(
182     'clean:css',
183     'minify:css'
184   );
185 });
186
187
188 /* js */
189
190 gulp.task('clean:js', function () {
191   return del(`${pathJs}/*.min.js`);
192 });
193
194 gulp.task('copy:js', function () {
195   return gulp.src(nodeModules + '/idb/lib/idb.js')
196     .pipe(gulp.dest(pathJs));
197 });
198
199 gulp.task('minify:js', function () {
200   return gulp.src([pathJs + '/*.js', !pathJs + '/*.min.js'])
201     .pipe(minifyJs())
202     .pipe(rename(function (path) {
203       path.extname = '.min.js';
204     })))
```

```
205     .pipe(gulp.dest(pathJs));
206   });
207
208   gulp.task('build:js', function () {
209     return runSequence(
210       'clean:js',
211       'copy:js',
212       'minify:js'
213     );
214   });
215
216
217   /* serviceWorker */
218
219   gulp.task('clean:serviceWorker', function () {
220     return del(`${pathRoot}/serviceWorker.min.js`);
221   });
222
223   gulp.task('minify:serviceWorker', function () {
224     return gulp.src(`${pathRoot}/serviceWorker.js`)
225       .pipe(minifyJs())
226       .pipe(rename(function (path) {
227         path.extname = '.min.js';
228       }))
229       .pipe(gulp.dest(`${pathRoot}/`));
230   });
231
232   gulp.task('build:serviceWorker', function () {
233     return runSequence(
234       'clean:serviceWorker',
235       'minify:serviceWorker'
236     );
237   });
238
239
240   /* main build tasks */
241
242   gulp.task('clean', function () {
243     return runSequence(
244       'clean:images',
245       'clean:css',
246       'clean:js',
247       'clean:html',
248       'clean:serviceWorker'
249     );
250   });
251
252   gulp.task('build', function () {
253     return runSequence(
254       'build:images',
255       'build:css',
256       'build:js',
257       'build:html',
258       'build:serviceWorker'
259     );
260   });
261
```

► server.js

► config/views.js