UDACITY

‹ Back to Mobile Web Specialist

Restaurant Reviews: Stage 3

REVIEW

CODE REVIEW 3

HISTORY

▸ client/js/restaurant_review.js 2

▾ gulpfile.js 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 pro
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

```
const del = require('del');
const copy = require('gulp-copy');
const rename = require('gulp-rename');
const imageMin = require('gulp-imagemin');
const htmlMin = require('gulp-htmlmin');
const responsive = require('gulp-responsive');
const minifyCss = require('gulp-clean-css');
const minifyJs = require('gulp-uglify-es').default;
const runSequence = require('run-sequence');
const pathRoot = './client';
const pathSourceImages = `${pathRoot}/img-src`;
const pathDestImages = `${pathRoot}/img`;
const pathSourceHtml = `${pathRoot}/html-src`;
const pathDestHtml = `${pathRoot}`;
const pathCss = `${pathRoot}/css`;
const pathJs = `${pathRoot}/js`;
const nodeModules = 'node_modules';


/* images */

gulp.task('clean:images', function () {
    return del(`${pathDestImages}/**/*`);
});

gulp.task('copy:images:restaurant', function () {
    return gulp.src([
            `${pathSourceImages}/restaurant.png`,
        ])
        .pipe(responsive({
            '*.png': [
                {
                    width: 128,
                    quality: 33,
                    rename: { suffix: '-128'}
                },
                {
                    width: 256,
                    quality: 33,
                    rename: { suffix: '-256'}
                },
                {
                    width: 512,
                    quality: 33,
                    rename: { suffix: '-512'}
                },
                {
                    width: 720,
                    quality: 33,
                    rename: { suffix: ''}
                }
            ]
        }))
        .pipe(imageMin({ verbose: false }))
        .pipe(gulp.dest(`${pathDestImages}`));
});

gulp.task('copy:images:buttons', function () {
    return gulp.src([
            `${pathSourceImages}/gold-medal.png`,
            `${pathSourceImages}/write.png`,
            `${pathSourceImages}/delete.png`
        ])
        .pipe(responsive({
            '*.png': [
                {
                    width: 32,
                    quality: 33,
                    rename: { suffix: '-32'}
                },
                {
                    width: 48,
                    quality: 33,
                    rename: { suffix: '-48'}
                },
                {
                    width: 64,
                    quality: 33,
                    rename: { suffix: '-64'}
                }
            ]
        }))
        .pipe(imageMin({ verbose: false }))
        .pipe(gulp.dest(`${pathDestImages}`));
});

gulp.task('copy:images:app', function () {
    return runSequence(
        'copy:images:restaurant',
        'copy:images:buttons'
    );
});


gulp.task('copy:images:data', function () {
    return gulp.src(`${pathSourceImages}/**/*.jpg`)
        .pipe(responsive({
            '*.jpg': [
                {
                    width: 200,
                    quality: 33,
                    rename: { suffix: '-200'}
                },
                {
                    width: 300,
                    quality: 33,
                    rename: { suffix: '-300'}
                },
                {
                    width: 400,
                    quality: 33,
                    rename: { suffix: '-400'}
                },
                {
                    width: 500,
                    quality: 33,
                    rename: { suffix: '-500'}
                },
                {
                    width: 600,
                    quality: 33,
                    rename: { suffix: '-600'}
                },
                {
                    width: 800,
                    quality: 33,
                    rename: { suffix: ''}
                }
            ]
        }))
        .pipe(imageMin({ verbose: false }))
        .pipe(gulp.dest(`${pathDestImages}`));
});

gulp.task('build:images', function () {
    return runSequence(
        'clean:images',
        'copy:images:app',
        'copy:images:data'
    );
});


/* html */

gulp.task('clean:html', function () {
    return del(`${pathDestHtml}/*.html`);
});

gulp.task('minify:html', function () {
    return gulp.src(`${pathSourceHtml}/*.html`)
        .pipe(htmlMin({collapseWhitespace: true}))
        .pipe(gulp.dest(pathDestHtml));
});

gulp.task('build:html', function () {
    return runSequence(
        'clean:html',
        'minify:html'
    );
});


/* css */

gulp.task('clean:css', function () {
    return del(`${pathCss}/*.min.css`);
});

gulp.task('minify:css', function () {
    return gulp.src([pathCss + '/*.css', !pathCss + '/*.min.css'])
        .pipe(minifyCss())
        .pipe(rename(function (path) {
            path.extname = '.min.css';
        }))
        .pipe(gulp.dest(pathCss));
});

gulp.task('build:css', function () {
    return runSequence(
        'clean:css',
        'minify:css'
    );
});


/* js */

gulp.task('clean:js', function () {
    return del(`${pathJs}/*.min.js`);
});

gulp.task('copy:js', function () {
    return gulp.src(nodeModules + '/idb/lib/idb.js')
        .pipe(gulp.dest(pathJs));
});

gulp.task('minify:js', function () {
    return gulp.src([pathJs + '/*.js', !pathJs + '/*.min.js'])
        .pipe(minifyJs())
        .pipe(rename(function (path) {
            path.extname = '.min.js';
        }))
        .pipe(gulp.dest(pathJs));
});

gulp.task('build:js', function () {
    return runSequence(
        'clean:js',
        'copy:js',
        'minify:js'
    );
});


/* serviceWorker */

gulp.task('clean:serviceWorker', function () {
    return del(`${pathRoot}/serviceWorker.min.js`);
});

gulp.task('minify:serviceWorker', function () {
    return gulp.src(`${pathRoot}/serviceWorker.js`)
        .pipe(minifyJs())
        .pipe(rename(function (path) {
            path.extname = '.min.js';
        }))
        .pipe(gulp.dest(`${pathRoot}/`));
});

gulp.task('build:serviceWorker', function () {
    return runSequence(
        'clean:serviceWorker',
        'minify:serviceWorker'
    );
});


/* main build tasks */

gulp.task('clean', function () {
    return runSequence(
        'clean:images',
        'clean:css',
        'clean:js',
        'clean:html',
        'clean:serviceWorker'
    );
});

gulp.task('build', function () {
    return runSequence(
        'build:images',
        'build:css',
        'build:js',
        'build:html',
        'build:serviceWorker'
    );
});

```

▸ server.js

▸ config/views.js