

gulp-responsive generates images at different sizes <https://npmjs.com/gulp-responsive>

[#responsive-images](#) [#responsive](#) [#gulp](#) [#sharp](#)

🕒 147 commits

🌿 9 branches

📦 24 releases

👤 10 contributors

📄 MIT

Branch: master ▾


New pull request

Create new file

Upload files

Find file

Clone or download ▾

 mahnunchik 2.9.0

Latest commit 75b2e4d 2 days ago

📁 examples	added examples to the readme	2 years ago
📁 lib	Replace deprecated gulp-util package	6 days ago
📁 test	Replace deprecated gulp-util package	6 days ago
📄 .editorconfig	file structure	3 years ago
📄 .gitattributes	initial commit	3 years ago
📄 .gitignore	added npm-debug.log to gitignore	3 years ago
📄 .jshintrc	file structure	3 years ago
📄 .npmignore	added brunch to travis config	3 years ago
📄 .travis.yml	updated node versions for travis	2 days ago
📄 History.md	updated changelog	a year ago
📄 LICENSE	added nodejs 5 to travis	2 years ago
📄 README.md	implemented `trim` option	6 months ago
📄 package.json	2.9.0	2 days ago

📖 README.md

gulp-responsive

build passing

Greenkeeper enabled

Generates images at different sizes

Installation

gulp-responsive depends on [sharp](#). Sharp is one of the fastest Node.js modules for resizing JPEG, PNG, WebP and TIFF images.

If you are using Mac OS then before installing `gulp-responsive` you should install the [libvips](#) library. Further information and instructions can be found in the [sharp installation guide](#).

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

Usage

```
var gulp = require('gulp');
var responsive = require('gulp-responsive');

gulp.task('default', function () {
  return gulp.src('src/*.png,jpg')
    .pipe(responsive({
      'background-*.jpg': {
        width: 700,
        quality: 50
      }
    }));
});
```

```

    },
    'cover.png': {
      width: '50%',
      // convert to jpeg format
      format: 'jpeg',
      rename: 'cover.jpg'
    },
    // produce multiple images from one source
    'logo.png': [
      {
        width: 200
      }, {
        width: 200 * 2,
        rename: 'logo@2x.png'
      }
    ]
  })
  .pipe(gulp.dest('dist'));
});

```

Examples

- [Simple example](#)
- [Multiple resolutions](#)
- [Advanced example](#)

Integration

- [HTML srcset attribute](#)
- [HTML <picture> element](#)
- [CSS image-set method](#)

All together 🌟

- [gulp-responsive config generation example](#)

API

responsive(**config**, **options**)

config

Configuration can be provided in one of the following formats:

1. Array of unique configurations

```

[ {
  name: 'logo.png',
  width: 200,
  height: 100
}, {
  name: 'banner.png',
  width: 500
} ]

```

2. Object of unique configurations. Keys are filenames.

```

{
  'logo.png': {
    width: 300,
    height: 200,
    rename: 'logo@2x.png'
  },
  'background-*.png': {
    width: 1400,
    withoutEnlargement: true
  }
}

```

3. Object of arrays of unique configurations. Keys are filenames.

```
{
  'logo.png': [{
    width: 200,
    rename: 'logo@1x.png'
  }, {
    width: 400,
    rename: 'logo@2x.png'
  }],
  'background-*': [{
    height: 400
  }]
}
```

Configuration unit

Configuration unit is an object:

- **name**: *String* — filename glob pattern.
- **width**: *Number* or *String* — width in pixels or percentage of the original, not set by default.
- **height**: *Number* or *String* — height in pixels or percentage of the original, not set by default.
- **withoutEnlargement**: *Boolean* — do not enlarge the output image, default `true`.
- **skipOnEnlargement**: *Boolean* — do not write an output image at all if the original image is smaller than the configured width or height, default `false`.
- **min**: *Boolean* — preserving aspect ratio, resize the image to be as small as possible while ensuring its dimensions are greater than or equal to the `width` and `height` specified.
- **max**: *Boolean* — resize to the max width or height the preserving aspect ratio (both `width` and `height` have to be defined), default `false`.
- **quality**: *Number* — output quality for JPEG, WebP and TIFF, default `80`.
- **progressive**: *Boolean* — progressive (interlace) scan for JPEG and PNG output, default `false`.
- **withMetadata**: *Boolean* — include image metadata, default `false`.
- **compressionLevel**: *Number* — zlib compression level for PNG, default `6`.
- **rename**: *String*, *Object* or *Function* — renaming options, file will not be renamed by default. When `extname` is specified, output format is parsed from extension. You can override this autodetection with `format` option.
- **format**: *String* — output format `jpeg`, `png`, `webp` or `raw`, default is `null`.
- **crop**: Crop the resized image to the exact size specified, default is `false`.
- **embed**: Preserving aspect ratio, resize the image to the maximum `width` or `height` specified then embed on a background of the exact width and height specified, default is `false`.
- **ignoreAspectRatio**: *Boolean* — Ignoring the aspect ratio of the input, stretch the image to the exact `width` and/or `height` provided via `resize`, default is `false`.
- **interpolator**: *String* — The interpolator to use for image **enlargement**, defaulting to `bicubic`.
- **kernel**: *String* — The kernel to use for image **reduction**, defaulting to `lanczos3`.
- **background**: *Color* — Set the background for the embed and flatten operations, '#default is `fff`'.
- **flatten**: *Boolean* — Merge alpha transparency channel, if any, with `background`, default is `false`.
- **negate**: *Boolean* — Produces the "negative" of the image, default is `false`.
- **rotate**: *Boolean* — Rotate the output image by either an explicit angle or auto-orient based on the EXIF `orientation` tag, default is `false`.
- **flip**: *Boolean* — Flip the image about the vertical Y axis. This always occurs after rotation, if any. The use of `flip` implies the removal of the EXIF `orientation` tag, if any. Default is `false`.
- **flop**: *Boolean* — Flop the image about the horizontal X axis. This always occurs after rotation, if any. The use of `flop` implies the removal of the EXIF `orientation` tag, if any. Default is `false`.
- **blur**: *Boolean* — When used without parameters, performs a fast, mild blur of the output image. This typically reduces performance by 10%. Default is `false`.
- **sharpen**: *Boolean* — When used without parameters, performs a fast, mild sharpen of the output image. This typically reduces performance by 10%. Default is `false`.
- **threshold**: *Number* or *Boolean* — Converts all pixels in the image to greyscale white or black, default is `false`.

- **gamma**: *Boolean* — Apply a gamma correction by reducing the encoding (darken) pre-resize at a factor of `1/gamma` then increasing the encoding (brighten) post-resize at a factor of `gamma`. Default is `false`.
- **grayscale**: *Boolean* — Convert to 8-bit greyscale; 256 shades of grey, default is `false`.
- **normalize**: *Boolean* — Enhance output image contrast by stretching its luminance to cover the full dynamic range. This typically reduces performance by 30%. Default is `false`.
- **trim**: *Boolean or Number* — Trim "boring" pixels from all edges that contain values within a percentage similarity of the top-left pixel. Default is `false`.
- **tile**: *Boolean or Object* — The size and overlap, in pixels, of square Deep Zoom image pyramid tiles, default is `false`.
- **withoutChromaSubsampling**: *Boolean* — Disable the use of [chroma subsampling](#) with JPEG output (4:4:4), default is `false`.

Detailed description of each option can be found in the [sharp API documentation](#).

Renaming

Renaming is implemented by the [rename](#) module. Options correspond with options of [gulp-rename](#).

options

errorOnUnusedConfig

Type: `Boolean`

Default: `true`

Emit the error when configuration is not used.

errorOnUnusedImage

Type: `Boolean`

Default: `true`

Emit the error when image is not used.

passThroughUnused

Type: `Boolean`

Default: `false`

Keep unmatched images in the stream. To use this option `errorOnUnusedImage` should be `false`.

errorOnEnlargement

Type: `Boolean`

Default: `true`

Emit the error when image is enlarged.

stats

Type: `Boolean`

Default: `true`

Show statistics after the run — how many images were created, how many were matched and how many were in the run in total.

silent

Type: `Boolean`

Default: `false`

Silence messages and stats if 0 images were created. If you wish to suppress all messages and stats, set the `options.stats` to `false` as well.

You can specify **global default value** for any of the [configuration options](#).

```
gulp.task('default', function () {
  return gulp.src('src/*.png')
    .pipe(responsive(config, {
```

```
// global quality for all images
quality: 50,
errorOnUnusedImage: false
}))
.pipe(gulp.dest('dist'));
});
```

License

MIT © [Evgeny Vlasenko](#)