gatsby-remark-images-contentful

Processes images in markdown so they can be used in the production build using Contentful's Image API

In the processing, it makes images responsive by:

- Adding an elastic container to hold the size of the image while it loads to avoid layout jumps.
- Generating multiple versions of images at different widths and sets the `srcset` and `sizes` of the `img` element so regardless of the width of the device, the correct image is downloaded.
- Using the "blur up" technique popularized by Medium and Facebook where a small 20px wide version of the image is shown as a placeholder until the actual image is downloaded.

Install

```
npm install gatsby-remark-images-contentful
```

How to use

```
// In your gatsby-config.js
plugins: [
  `gatsby-plugin-sharp`,
  {
    resolve: `gatsby-transformer-remark`,
    options: {
      plugins: [
        {
          resolve: `gatsby-remark-images-contentful`,
          options: {
            // It's important to specify the maxWidth (in pixels) of
            // the content container as this plugin uses this as the
            // base for generating different widths of each image.
            maxWidth: 590,
          },
        },
      ],
    },
  },
]
```

Options

Name	Default	Description
`maxWidth`	`650`	The `maxWidth` in pixels of the `div` where the markdown will be displayed. This value is used when deciding what the width of the various responsive thumbnails should be.
`linkImagesToOriginal`	`true`	Add a link to each image to the original image. Sometimes people want to see a full-sized version of an image e.g. to see extra detail on a part of the image and this is a convenient and common pattern for enabling this. Set this option to false to disable this behavior.
`showCaptions`	`false`	Add a caption to each image with the contents of the title attribute, when this is not empty. Set this option to true to enable this behavior.
`wrapperStyle`		Add custom styles to the div wrapping the responsive images. Use regular CSS syntax, e.g. `margin-bottom:10px; background: red;`
`backgroundColor`	`white`	Set the background color of the image to match the background of your design
`withWebp`	`false`	Additionally generate WebP versions alongside your chosen file format. They are added as a srcset with the appropriate mimetype and will be loaded in browsers that support the format.
`loading`	`lazy`	Set the browser's native lazy loading attribute. One of `lazy`, `eager` or `auto`.

Troubleshooting

Incompatible library version: sharp.node requires version X or later, but Z provides version Y

This means that there are multiple incompatible versions of the `sharp` package installed in `node_modules`. The complete error typically looks like this:

```
Something went wrong installing the "sharp" module

dlopen(/Users/misiek/dev/gatsby-starter-blog/node_modules/sharp/build/Release/sharp.node, 1)
  Referenced from: /Users/misiek/dev/gatsby-starter-blog/node_modules/sharp/build/Release/sh
  Reason: Incompatible library version: sharp.node requires version 6001.0.0 or later, but l
```

To fix this, you'll need to update all Gatsby plugins in the current project that depend on the `sharp` package. Here's a list of official plugins that you might need to update in case your projects uses them:

- `gatsby-plugin-sharp`
- `gatsby-plugin-manifest`
- `gatsby-remark-images-contentful`
- `gatsby-source-contentful`
- `gatsby-transformer-sharp`
- `gatsby-transformer-sqip`

To update these packages, run:

```
npm install gatsby-plugin-sharp gatsby-plugin-manifest gatsby-remark-images-contentful gatsb
```

If updating these doesn't fix the issue, your project probably uses other plugins from the community that depend on a different version of `sharp`. Try running `npm list sharp` or `yarn why sharp` to see all packages in the current project that use `sharp` and try updating them as well.