Welcome to `gatsby@2.31.0` release (January 2021 #2)

Key highlights of this release:

- [Performance improvements]()
- [Support for Gatsby's fragments in GraphiQL]()
- [Use Fast Refresh by default for React 17]()
- [Important gatsby-plugin-image updates]()

Also check out [notable bugfixes]().

Bleeding Edge: Want to try new features as soon as possible? Install `gatsby@next` and let us know if you have any [issues]().

[Previous release notes]()

[Full changelog]()

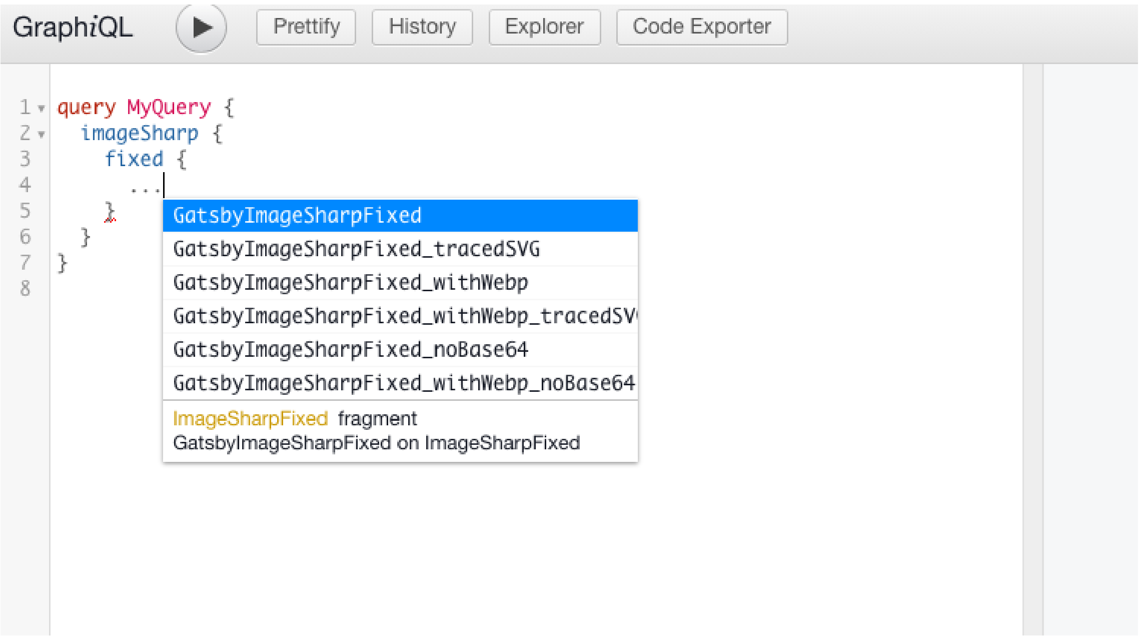
Performance improvements

An improvement to build speed for sites with many (10k+) markdown and images was made in this PR [#28891](). This change especially benefits very large sites—our 16k page markdown/image benchmark site's build time dropped 37% and the 128k pages benchmark dropped 84%.

In [#28957]() an improvement to `gatsby develop` was made by not using the `debug` config for `bluebird`. This can speed your `source nodes`, especially for sites with larger numbers of pages.

Support for Gatsby's fragments in GraphiQL

The underlying `graphiql` package added support for registering fragments and Gatsby now leverages this new feature! You now get auto-completion for image fragments and any fragments that you added yourself inside GraphiQL:


GraphiQL
Prettify
History
Explorer
Code Exporter
GatsbyImageSharpFixed
GatsbyImageSharpFixed_tracedSVG
GatsbyImageSharpFixed_withWebp
GatsbyImageSharpFixed_withWebp_tracedSV
GatsbyImageSharpFixed_noBase64
GatsbyImageSharpFixed_withWebp_noBase64
ImageSharpFixed fragment
GatsbyImageSharpFixed on ImageSharpFixed

```
query MyQuery {
  imageSharp {
    fixed {
      ...
    }
  }
}
```

Fast Refresh enabled by default for React 17

With the PRs #28689 & #28930 merged, Gatsby will now automatically use Fast Refresh if you use React >= 17.0.0 for your site. As you can read in the release notes for 2.28 we've added support for Fast Refresh giving you a better (and custom) error overlay, quicker error recovery and in general a better `gatsby develop` experience.

Two ESLint rules that will warn you against anti-patterns in your code were added now:

- No anonymous default exports
- Page templates must only export one default export (the page) and `query` as a named export

Please give us your feedback about the Fast Refresh integration in the umbrella issue. Thanks!

Important `gatsby-plugin-image` updates

This release includes important changes to the API of the beta image plugin as well as some new features and some proper docs. These are breaking changes, but should be the final API used in the stable release.

We've listened to lots of feedback, we've used the components ourselves, and we've performed lots of tests. From that we've made a few changes.

1. **compat component removed:** The compat component was a temporary fix to allow people to try out the component without needing to change their data. With the new API changes this was going to be hard to maintain, and the release of the codemod makes it easier to migrate instead. This release removes them entirely, so be aware that if you upgrade you will need to update your resolvers to use the new syntax.
2. **`fluid` layout renamed to `fullWidth`:** We're renaming the `fluid` layout to `fullWidth`, to better reflect what it's used for. It was called `fluid` before to match the equivalent layout in the old `gatsby-image` component. However this was confusing, because for many of the cases where people used `fluid` before they should be using the new `constrained` layout. This displays the image at the specified width, but allows it to shrink to fit smaller screens. However `fullWidth` images always expand to the full width of their container, even if that means stretching beyond the maximum size of the source image. The main use for these is hero images and other full-width banners. To make that clearer, we changed the name.
3. **Don't use `maxWidth` for `fullWidth` images:** The old `fluid` resolver allows you to pass a `maxWidth` argument, which we carried-over to the new plugin. Unfortunately it doesn't do what most people think it does. In fact, it's quite hard to explain what it does at all! It doesn't set the actual maximum width of the displayed image: you want constrained for that. It also doesn't set the maximum size of the generated image! It actually sets the size of the `1x` resolution image. The largest image generated would be this value, multiplied by the maximum pixel density! No wonder when we looked at open source projects that use it, almost everyone seems to be misusing the property. Because of this we removed it entirely, along with `maxHeight` (also doesn't do what you'd expect). Its replacement is:
4. **Use breakpoints for `fullWidth` images:** Previously `fluid` layout generated image resolutions that were based on multiples and fractions of the `maxWidth`, similar to how they are done for `fixed` and `constrained`. This makes sense where they are mostly going to be used for higher-pixel density screens, rather than for displaying the image larger. However for a full width image, the size you want is the width of the screen. We can't generate images for every size, but we can choose a good selection of sizes to cover everything from phones to large monitors. We have chosen a default set based on our usage research, but you can pass your own breakpoints in if you'd prefer.
5. **Change `maxWidth/maxHeight` to `width/height`** There was no good reason to have `constrained` take `maxWidth/maxHeight`, while fixed took `width/height`. We've changed it so

they both take `width/height` .

6. **Add `aspectRatio` prop:** We realized that lots of people were using hacks to set the aspect ratio of images, because it was hard to do properly, particularly for `fluid` images. We've added a new `aspectRatio` prop that forces the image to that aspect ratio, cropping if needed. If you pass it on its own, it will default to using the source image width, and then adjusting the height to the correct aspect ratio. Alternatively you can pass your own width or height and it will set the other dimension. This works in all layout types.
7. **Remote static images unflagged:** In the previous version you could pass remote URLs to `StaticImage` , but only if you passed in an environment variable to enable it. In this release we've removed the flag, so try it out! If you use a remote URL it will download that image at build time and then process it in the same way as it would a local one. Bear in mind that because this is done at build time, any updates to the image on the remote server won't show up until you next build. Make sure your server supports ETag headers (most do), otherwise it will need to download them every time.
8. **Automatic `object-fit` polyfill:** This version includes an automatic polyfill for browsers that don't support the CSS `object-fit` and `object-position` properties. It is lazy-loaded so has no overhead for browsers that don't need it.
9. **New docs are live:** Want to know more about the plugin, and dive deeper into the options? There are two new pages in the documentation that are all about this plugin:
    1. A step-by-step how-to guide
    2. A detailed reference guide

This is likely to be the final minor version before the stable release, but please continue to share your feedback. Meanwhile, watch out for @laurieontech sharing the new image experience in Gatsby at Image Ready v2, hosted by Jamstack Toronto, alongside her counterparts from 11ty and NextJS.

Notable bugfixes

- Extract non-CSS-in-JS CSS and add to `<head>` when SSRing in develop mode (with `DEV_SSR` flag), via #28471
- Show stack trace for non-GraphQL errors, via #28888
- Update vulnerable packages and include React 17 in peerDeps, via #28545

Contributors

A big **Thank You** to our community who contributed to this release 💜

- DecliningLotus: Update Fontsource Documentation PR #28750
- saintmalik: chore(docs): fix a broken url PR #28756
- jbampton
    - chore: Fix spelling PR #28761
    - chore(gatsby-transformer-csv): fix whitespace PR #28763
- vse-volod: chore(docs): Fixing broken link PR #28767
- eriklofblad: chore(docs): Updated Twin + Gatsby + Emotion link PR #28782
- yonatanLehman: chore(docs): Update lighthouse instructions in tutorial PR #28785
- sreehari-sj23
    - chore(gatsby): Changed .org to .com PR #28818

  - chore(examples): Remove Broken Link PR #28817
- takapiro99: chore(docs): Fix all `/packages` links to `/plugins` PR #28816
- Taremeh: chore(docs): Update migrate-remark-to-mdx PR #28811
- muescha
  - chore(docs): add http to localhost PR #28896
  - chore(docs): fix code block language to shell PR #28895
  - docs(readme): add file extension, fix code block, fix link, fix code PR #28905
  - fix(gatsby-plugin-netlify): fix typo PR #28907
  - fix(gatsby-plugin-sass): fix brand name, code block around names PR #28908
  - doc(node-creation): fix brand name PR #28912
  - chore(docs): plugin contentful - fix brand name, code blocks for names PR #28915
  - readme(gatsby-plugin-emotion): fix table PR #28906
  - chore(docs): google plugin analytics - upgrade to gtag, fix code PR #28918
  - chore(docs): release notes - code fences, code language PR #28936
  - chore(docs): typography - brand name PR #28931
  - chore(docs): routing - brand name, code language, code block PR #28929
- AlexRAV: Add avif format support to gatsby remark images package PR #28927
- nishantwrp: chore(default-starter): format code using prettier PR #28958
- GrantHair5: chore(docs): Change cd path to "my-gatsby-site" PR #28933
- mrhut10: feat(gatsby-source-shopify): download images option PR #23840
- andrewmcoupe: chore(gatsby): Improve type safety of ShouldUpdateScrollArgs PR #28923
- Vishal19111999: chore(docs): Add Tina in list of Headless CMS PR #28981
- ariadne-github: perf(gatsby-source-contentful): prevent redundant fs/remote fetches for tracedSVG PR #28926
- RookTKO: chore(docs): Updating Chakra UI docs PR #29018
- kaneshin: fix(gatsbu-source-contentful): apply useNameForId when creating the graphql schema PR #28649
- axe312ger
  - Fix/contentful add header PR #29028
  - chore(docs): adjust Contentful Rich Text example codes PR #29029
  - fix(gatsby-source-contentful): Improve base64 placeholders PR #29034
- nategiraudeau: fix(sharp) wrap sharp calls in try/catch to avoid crashing on bad images PR #28645
- hoobdeebla: fix(security): update vulnerable packages, include React 17 in peerDeps PR #28545
- hasparus: fix(gatsby-plugin-typescript): add missing options validations PR #29066