

Welcome to `gatsby@2.30.0` release (January 2021 #1)

Key highlights of this release:

- [Query on Demand and Lazy Images: Generally available](#) - improves `gatsby develop` bootup time
- [Server Side Rendering \(SSR\) in development](#) — helps you find and fix many build errors in development. We're starting a partial release of this feature to 5% of users.
- [gatsby-plugin-sass v3](#) - use the latest `sass-loader` and `Dart Sass` by default
- [Image transformations up to 30% faster](#)

And several impactful updates in the new [gatsby-plugin-image](#) (beta):

- [AVIF support](#) - enjoy next-gen image format
- [Support remote static images](#) - download, transform and optimize remote images with a single line

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](#)

## Query on Demand and Lazy Images: Generally Available

Those two features change the way how `gatsby develop` works:

- We no longer transform all images eagerly on dev server start
- We no longer run all queries

Instead, it happens lazily when the browser requests a page or a specific image variant. This significantly shortens the boot-up time for `gatsby develop`.

*Note that it does not affect `gatsby build` in any way, only `gatsby develop`*

We've been gradually rolling out those features:

- "Query on Demand" was introduced behind the flag in [v2.27](#).
- "Lazy images" were introduced behind the flag in [gatsby-plugin-sharp@2.10.0](#).
- In [v2.29](#) 10% of our users were automatically opted-in

This allowed us to collect the feedback and fix bugs as they were discovered.

Now we feel confident enough to publish it as generally available. You don't need to add flags or any environment variables to enable it. Just upgrade to `gatsby@2.30.0` (and `gatsby-plugin-sharp@2.12.0` for lazy images).

In v2.29 we improved the UX around long-running queries by adding a loading indicator and message to the browser console (only in `gatsby develop`). If you want or need to de-activate this indicator, you can! For more details please see the [umbrella discussion](#).

## Server Side Rendering (SSR) in development

There are certain types of build errors that haven't been detectable while developing. The most common is code that tries to access browser globals (like `window`) that don't exist in Node.js when SSRing your Gatsby site.

This is frustrating as you might develop for a while before building and only then discover the error. Then actually fixing the problems is painful as the feedback cycle is slow as you have to run a build after each code change.

We've been working to add SSR support to the develop server so that you can immediately see SSR bugs and get quick feedback as you fix them. With this change, whenever you do a full reload, the Gatsby dev server will deliver a SSRed HTML file along with your React code, mimicking how production Gatsby sites work.

This is related to our general efforts to make the develop and build environment identical.

Like the recent Query on Demand and Lazy Images changes, we released this change first behind a flag for early testing and now, we're rolling this out to a small percentage of users (5%) for more real-world testing before the final release to 100% of users.

We'll let you know after upgrading if your site has SSR enabled. If it's not enabled and you'd like to start using it immediately, simply add the `DEV_SSR` flag to your `gatsby-config.js`.

```
module.exports = {  
  flags: { DEV_SSR: true },  
  plugins: [...],  
}
```

### **gatsby-plugin-sass@3.0.0**

Now that [LibSass and Node Sass are deprecated](#), we've upgraded `sass-loader` to `10.1.0` and thus switched sass implementation to [Dart Sass](#).

The plugin itself is compatible with the previous version. For the majority of projects the upgrade won't require any special actions.

But keep in mind that **Dart Sass** may still have subtle differences in some edge cases comparing to **Node Sass**, so if you encounter any issues make sure to check out ["How do I migrate section"](#) in sass docs.

See also:

- <https://github.com/gatsbyjs/gatsby/issues/27754>
- <https://github.com/gatsbyjs/gatsby/pull/27991>

### **gatsby-plugin-sharp@2.12.0**

New approach to concurrency speeds up image transformations up to 30%.

Previously we were applying several transformations to a single image concurrently. With the new approach we transform multiple images in parallel but apply transformations to each image serially.

Another major change is [lazy images in develop](#).

### **gatsby-plugin-image@0.5.0 (beta)**

#### **AVIF support**

This release adds beta support for generating and displaying [AVIF images](#). AVIF is a brand-new image format, which offers considerably better file sizes and quality than JPG and even WebP. It is currently [only supported in Chrome](#),

with Firefox support coming soon. However it is safe to use today, because unsupported browsers will fall back to using a supported format.

AVIF is not currently enabled by default, so to use it in your site you need to add it to the `formats` array. You should also include auto and WebP to support other browsers. Ensure that you have upgraded to the latest version of `gatsby-plugin-sharp`, `gatsby-transformer-sharp` and `gatsby-plugin-image`, and then use the following syntax:

In `StaticImage` :

```
<StaticImage
  src="./cornwall.jpg"
  formats={["auto", "webp", "avif"]}
  alt="Cornwall"
/>
```

...or in GraphQL:

```
query {
  file(relativePath: { eq: "cornwall.jpg" }) {
    childImageSharp {
      gatsbyImageData(maxWidth: 720, formats: [AUTO, WEBP, AVIF])
    }
  }
}
```

This does not currently work in Gatsby Cloud, but will be enabled later this week.

This is possible thanks to the incredible work by the [sharp](#) and [libvips](#) contributors.

## Support remote image URLs (experimental)

This release enables experimental support for remote URLs in the `StaticImage` component. This means that you can pass an absolute URL in the `src` prop and Gatsby will download the file and process it at build time. It is currently experimental, but you can try it out today and let us know how you find it.

To enable, set the `GATSBY_EXPERIMENTAL_REMOTE_IMAGES` environment variable to `1` when building:

```
GATSBY_EXPERIMENTAL_REMOTE_IMAGES=1 gatsby develop
```

or...

```
GATSBY_EXPERIMENTAL_REMOTE_IMAGES=1 gatsby build
```

You can then pass absolute URLs to `StaticImage` :

```
<StaticImage src="https://placekitten.com/400/400" alt="Kittin" />
```

Please note that this is only supported in `StaticImage` .

## Notable bugfixes

- Display meaningful message on errors when writing base64 images, via [#28614](#)
- Fix broken VS Code debugger, via [#28643](#)
- Always add both `childField` and `childrenField` in GraphQL to avoid schema inference errors, via [#28656](#)
- Fix obscure `undefined failed` message for webpack compilation errors, via [#28701](#)
- `gatsby-plugin-mdx` : performance improvements during sourcing, via [#27974](#)
- `gatsby-source-graphql` : add HTTP error reporting, via [#28786](#)
- `gatsby-source-filesystem` : retry a download in `createRemoteFileNode` if a file is incomplete, via [#28547](#)
- `gatsby-plugin-schema-snapshot` : fix warning `The type Foo does not explicitly define the field childBar`, via [#28483](#)
- `gatsby-source-contentful` : dont re-create nodes unnecessarily, via [#28642](#)

## Contributors

A big **Thank You** to [our community who contributed](#) to this release ❤️

- [mdb571](#): docs: update styling-css.md [PR #28581](#)
- [erezrokah](#): chore(plugin-netlify-cms): update Slack link [PR #28529](#)
- [ax-vasquez](#): Fix broken import statement (#28611) [PR #28611](#)
- [runningdeveloper](#): Fixed typo brackets in file-system-route-api.md [PR #28612](#)
- [herecydev](#): chore: update ink to v3 [PR #26190](#)
- [gugu](#): feat(gatsby): Add AVIF file support to image loader in webpack config [PR #28638](#)
- [angristan](#): fix(docs): dead links sourcing-data -> querying-data [PR #28651](#)
- [jbampton](#): docs: fix spelling [PR #28665](#)
- [connor4312](#): fix(gatsby): Do not activate inspect if already active [PR #28643](#)
- [antoinerousseau](#): feat(gatsby-source-graphql): Default Apollo Link fetch wrapper to show better API errors [PR #28786](#)