> *This documentation isn't up to date with the latest version of Gatsby.*
>
> *Outdated areas are:*
>
> - `data.json` *doesn't exist anymore*
> - *Add* `page-data.json`
> - *Add* `match-paths.json`
>
> *You can help by making a PR to update this documentation.*

Throughout the Gatsby code, you'll see the below object fields and variables mentioned. Their definitions and reason for existence are defined below.

Page

Page Object

created by calls to createPage (see Page Creation).

path

The publicly accessible path in the web URL to access the page in question. E.g

`/blog/2018-07-17-announcing-gatsby-preview/` .

It is created when the page object is created (see Page Creation)

updatedAt

Last updated time.

Redux `pages` namespace

Contains a map of Page path -> Page object.

matchPath

Think of this instead as `client matchPath` . It is ignored when creating pages during the build. But on the frontend, when resolving the page from the path (find-path.js), it is used (via reach router) to find the matching page. Note that the pages are sorted so that those with matchPaths are at the end, so that explicit paths are matched first.

It is also used by gatsby-plugin-netlify when creating `_redirects` .

jsonName

The logical name for the page's query JSON result. The name is constructed during createPage using a kebabHash of page path. E.g. For the above page path, it is:

`blog-2018-07-17-announcing-gatsby-preview-995`

The actual JSON file is written to disk after Query Execution.

component

The path on disk to the JavaScript file containing the React component. E.g

`/src/templates/template-blog-post.js`

Think of this as `componentPath` instead.

Redux `components` namespace

Mapping from `component` (path on disk) to its Page object. It is created every time a page is created (by listening to `CREATE_PAGE`).

```
{
  `/src/templates/template-blog-post.js`: {
    query: ``,
    path: `/blog/2018-07-17-announcing-gatsby-preview/`,
    jsonName: `blog-2018-07-17-announcing-gatsby-preview-995`,
    componentPath: `/src/templates/template-blog-post.js`,
    ...restOfPage
  }
}
```

Query starts off as empty, but is set during the extractQueries phase by query-watcher/handleQuery, once the query has compiled by relay (see Query Extraction).

componentChunkName

The `[name]` portion of the webpack chunkFilename (`[name]-[contenthash].js`) (see Production App webpack config). Its name is the concatenation of `component---` and the `component` name passed through kebab-hash. E.g, the componentChunkName for component

`/src/blog/2.js`

is

`component---src-blog-2-js`

This is used extensively throughout Gatsby, but especially during Code Splitting.

internalComponentName

If the path is `/` , internalComponentName = `ComponentIndex` . Otherwise, for a path of `/blog/foo` , it would be `ComponentBlogFoo` .

Created as part of page, but currently unused.

page.context

This is merged with the page itself and then is passed to GraphQL queries as the `context` parameter.

Query

dataPath

Path to the page's query result. Relative to `/public/static/d/{modInt}` . Name is kebab hash on `path--${jsonName}` - `result->sha1->base64` . E.g

`621/path---blog-2018-07-17-announcing-gatsby-preview-995-a74-dwfQIanOJGe2gi27a9CLKHjamc`

Set after Query Execution has finished.

Redux `jsonDataPaths` namespace (dataPaths)

Map of page jsonName to dataPath. Updated after Query Execution. E.g

```
{
  // jsonName -> dataPath
  "blog-2018-07-17-announcing-gatsby-preview-995": "621/path---blog-2018-07-17-announcing-gatsby-preview-995-a74-dwfQIanOJGe2gi27a9CLKHjamc"
}
```

This is also known via the `dataPaths` variable.

Query result file

`/public/static/d/621/${dataPath}`

E.g

`/public/static/d/621/path---blog-2018-07-17-announcing-gatsby-preview-995-a74-dwfQIanOJGe2gi27a9CLKHjamc.json`

This is the actual result of the GraphQL query that was run for the page `/blog/2018-07-17-announcing-gatsby-preview/`. The contents would look something like:

```
{
  "data": {
    "markdownRemark": {
      "html": "<p>Today we....",
      "timeToRead": 2,
      "fields": {
        "slug": "/blog/2018-07-17-announcing-gatsby-preview/"
      },
      "frontmatter": {
        "title": "Announcing Gatsby Preview",
        "date": "July 17th 2018",
        ...
      },
      ...
    }
  },
  "pageContext": {
    "slug": "/blog/2018-07-17-announcing-gatsby-preview/",
    "prev": {
      ...
    },
    "next": null
  }
}
```

For a query such as:

```
export const pageQuery = graphql`
  query($slug: String!) {
    markdownRemark(fields: { slug: { eq: $slug } }) {
      html
      timeToRead
      fields {
        slug
      }
      frontmatter {
        title
        date(formatString: "MMMM Do YYYY")
        ...
      }
      ...
    }
  }
`
```

webpack stuff

/.cache/async-requires.js

See Write Out Pages.

.cache/data.json

See Write Out Pages.