gatsby-transformer-javascript-frontmatter

Parses JavaScript files to extract frontmatter from exports.

Install

```
npm install gatsby-source-filesystem gatsby-transformer-javascript-frontmatter
```

How to use

To use this plugin you also need gatsby-source-filesystem installed and configured.

```
// In your gatsby-config.js
module.exports = {
  plugins: [
    {
      resolve: `gatsby-source-filesystem`,
      options: {
        name: `pages`,
        path: `${__dirname}/src/pages/`,
      },
    },
    "gatsby-transformer-javascript-frontmatter",
  ],
}
```

Parsing algorithm

This plugin uses @babel/parser and @babel/traverse to statically read the frontmatter exports.

In a `.js` (or `.jsx` / `.ts` / `.tsx`) file, export a frontmatter object to set your metadata variables, like so:

```
import React from "react"

exports.frontmatter = {
  title: "Choropleth on d3v4",
  written: "2017-05-04",
  layoutType: "post",
  path: "choropleth-on-d3v4",
  category: "data science",
  description: "Things about the choropleth.",
}

export default MyComponent
```

You can also use a named export for the frontmatter object:

```
export const frontmatter = {
  title: "Choropleth on d3v4",
  written: "2017-05-04",
  layoutType: "post",
  path: "choropleth-on-d3v4",
  category: "data science",
  description: "Things about the choropleth.",
}
```

How to query

You'd be able to query your frontmatter like:

```
{
  allJavascriptFrontmatter {
    edges {
      node {
        frontmatter {
          error
          path
          title
          written
          category
          description
          updated
        }
      }
    }
  }
}
```

Which would return something like:

```
{
  "data": {
    "allJavascriptFrontmatter": {
      "edges": [
        {
          "node": {
            "frontmatter": {
              "error": false,
              "path": "choropleth-on-d3v4",
              "title": "Choropleth on d3v4",
              "written": "2017-05-04",
              "category": "data science",
              "description": "Things about the choropleth.",
              "updated": null
            }
          }
        }
      ]
    }
  }
}
```

Any attribute on "frontmatter" across your js files will be exported. If a file is missing it, the value will be null.

The `error` field will contain `false` or an object with error information just to give a surface level view of what the query is pulling out.

```
{
  "error": {
    "err": true,
    "message": "we threw an error",
    "stack": "This is a stringified stack trace"
  }
}
```