

THIS PACKAGE HAS BEEN DEPRECATED IN FAVOR OF

[GATSBY-TRANSFORMER-JAVASCRIPT-FRONTMATTER](#)

Parses JavaScript files to extract data from exports.

Install

```
npm install gatsby-transformer-javascript-static-exports
```

How to use

```
// In your gatsby-config.js
plugins: [`gatsby-transformer-javascript-static-exports`]
```

Parsing algorithm

The algorithm uses babylon and traverse (from the babel family of code) to statically read the data exports.

In a .js file, export a data object to set your metadata variables, like so:

```
import * as React from 'react'

exports.data = {
  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 data object:

```
export const data = {
  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 data like:

```

{
  allJsFrontmatter {
    edges {
      node {
        data {
          error
          path
          title
          written
          category
          description
          updated
        }
      }
    }
  }
}

```

Which would return something like:

```

{
  "data": {
    "allJsFrontmatter": {
      "edges": [
        {
          "node": {
            "data": {
              "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 "data" 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"  
  },
```