gatsby-transformer-yaml

Parses YAML files. Supports arrays of objects and single objects.

Supported extensions: `.yaml`, `.yml`

Both `.yaml` and `.yml` are treated in the same way. This document uses both of them interchangeably.

Install

`npm install gatsby-transformer-yaml`

Note: You also need to have `gatsby-source-filesystem` installed and configured so it points to your files.

How to use

In your `gatsby-config.js`

```
module.exports = {
  plugins: [
    `gatsby-transformer-yaml`,
    {
      resolve: `gatsby-source-filesystem`,
      options: {
        path: `./src/data/`,
      },
    },
  ],
}
```

Where the *source folder* `./src/data/` contains the `.yaml` files.

Parsing algorithm

You can choose to structure your data as arrays of objects in individual files or as single objects spread across multiple files.

The *source folder* can contain either the following:

- **Array of Objects**: Where each file represents a collection. (*you probably want this one*)
- **Single Object**: Where each *subfolder* represents a collection; each file represents one "record".

Array of Objects

The algorithm for YAML arrays is to convert each item in the array into a node. The type of the node is based on the filename.

So if your project has a `letters.yaml` which looks like:

```yaml
- character: a
- character: b
- character: c
```

Then the following three nodes would be created.

```
[
  {
    "character": "a"
  },
  {
    "character": "b"
  },
  {
    "character": "c"
  }
]
```

Single Object

The algorithm for single YAML objects is to convert the object defined at the root of the file into a node. The type of the node is based on the name of the parent directory.

For example, let's say your project has a data layout like:

```
data/
    letters/
        a.yml
        b.yml
        c.yml
```

Where each of `a.yml`, `b.yml` and `c.yml` look like:

```yaml
character: a
```

```yaml
character: b
```

```yaml
character: c
```

Then the following three nodes would be created.

```
[
  {
    "character": "a"
  },
  {
    "character": "b"
  },
```

```
  {
    "character": "c"
  }
]
```

How to query

You can query the nodes using GraphQL, like from the GraphiQL browser: `http://localhost:8000/___graphql`.

Regardless of whether you choose to structure your data in arrays of objects or single objects, you'd be able to query your letters like:

```
{
  allLettersYaml {
    edges {
      node {
        character
      }
    }
  }
}
```

Which would return:

```
{
  allLettersYaml: {
    edges: [
      {
        node: {
          character: "a",
        },
      },
      {
        node: {
          character: "b",
        },
      },
      {
        node: {
          character: "c",
        },
      },
    ]
  }
}
```

Please do **note** that `allLettersYaml` **will not** show up if you do not have any

`.yaml` files.

Configuration options

`typeName` [string|function][optional]

The default naming convention documented above can be changed with either a static string value (e.g. to be able to query all yaml with a simple query):

```
module.exports = {
  plugins: [
    {
      resolve: `gatsby-transformer-yaml`,
      options: {
        typeName: `Yaml`, // a fixed string
      },
    },
  ],
}
```

```
{
  allYaml {
    edges {
      node {
        value
      }
    }
  }
}
```

or a function that receives the following arguments:

- `node`: the graphql node that is being processed, e.g. a File node with yaml content
- `object`: a single object (either an item from an array or the whole yaml content)
- `isArray`: boolean, true if `object` is part of an array

```
- level: info
  message: hurray
- level: info
  message: it works
- level: warning
  message: look out
```

```
module.exports = {
  plugins: [
    {
      resolve: `gatsby-transformer-yaml`,
```

```
      options: {
        typeName: ({ node, object, isArray }) => object.level,
      },
    },
  ],
}
```

```
{
  allInfo {
    edges {
      node {
        message
      }
    }
  }
}
```

Troubleshooting

`id` and `yamlId` key

If your data contains an `id` key the transformer will automatically convert this key to `yamlId` as `id` is a reserved internal keyword for Gatsby.