gatsby-transformer-hjson

Parses raw HJSON strings into JavaScript objects e.g. from HJSON files. Supports arrays of objects and single objects.

Install

```
npm install gatsby-transformer-hjson
```

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-hjson`],
}
```

Parsing algorithm

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

Array of Objects

The algorithm for arrays is to convert each item in the array into a node.

So if your project has a `letters.hjson` with `[{ value: a } { value: b } { value: c }]` then the following three nodes would be created.

```
;[
  { value: "a", type: "Letters" },
  { value: "b", type: "Letters" },
  { value: "c", type: "Letters" },
]
```

Single Object

The algorithm for single JSON 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, lets say your project has a data layout like:

```
data/
    letters/
        a.hjson
```

```
            b.hjson
            c.hjson
```

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

```
value: a
```

```
value: b
```

```
value: c
```

Then the following three nodes would be created.

```
;[
  {
    value: "a",
    type: "Letters",
  },
  {
    value: "b",
    type: "Letters",
  },
  {
    value: "c",
    type: "Letters",
  },
]
```

How to query

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:

```
{
  allLettersJson {
    edges {
      node {
        value
      }
    }
  }
}
```

Which would return:

```
{
  allLettersJson: {
    edges: [
      {
        node: {
          value: "a",
        },
      },
      {
        node: {
          value: "b",
        },
      },
      {
        node: {
          value: "c",
        },
      },
    ]
  }
}
```