gatsby-transformer-xml

Parses XML files. It also supports attributes

Install

```
npm install gatsby-transformer-xml
```

How to use

```
// In your gatsby-config.js
plugins: [`gatsby-transformer-xml`]
```

Parsing algorithm

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

So if your project has a `books.xml` with

```
<?xml version="1.0"?>
<catalog>
  <book id="bk101">
      <author>Gambardella, Matthew</author>
      <title>XML Developer's Guide</title>
      <genre>Computer</genre>
      <price>44.95</price>
      <publish_date>2000-10-01</publish_date>
      <description>An in-depth look at creating applications
      with XML.</description>
   </book>
   <book id="bk102">
      <author>Ralls, Kim</author>
      <title>Midnight Rain</title>
      <genre>Fantasy</genre>
      <price>5.95</price>
      <publish_date>2000-12-16</publish_date>
      <description>A former architect battles corporate zombies,
      an evil sorceress, and her own childhood to become queen
      of the world.</description>
   </book>
</catalog>
```

The plugin uses xml-parser to convert it to json

```
{
  "declaration": {
    "attributes": {
      "version": "1.0"
    }
  },
  "root": {
    "name": "catalog",
    "attributes": {},
    "children": [
      {
        "name": "book",
        "attributes": {
          "id": "bk101"
        },
        "children": [
          {
            "name": "author",
            "attributes": {},
            "children": [],
            "content": "Gambardella, Matthew"
          },
          {
            "name": "title",
            "attributes": {},
            "children": [],
            "content": "XML Developer's Guide"
          },
          {
            "name": "genre",
            "attributes": {},
            "children": [],
            "content": "Computer"
          },
          {
            "name": "price",
            "attributes": {},
            "children": [],
            "content": "44.95"
          },
          {
            "name": "publish_date",
            "attributes": {},
            "children": [],
            "content": "2000-10-01"
          },
          {
            "name": "description",
            "attributes": {},
            "children": [],
            "content": "An in-depth look at creating applications\n      with XML."
          }
        ],
        "content": ""
      },
      {
        "name": "book",
        "attributes": {
          "id": "bk102"
        },
        "children": [
          {
            "name": "author",
            "attributes": {},
            "children": [],
            "content": "Ralls, Kim"
          },
          {
            "name": "title",
            "attributes": {},
            "children": [],
            "content": "Midnight Rain"
          },
          {
            "name": "genre",
            "attributes": {},
            "children": [],
            "content": "Fantasy"
          },
          {
            "name": "price",
            "attributes": {},
            "children": [],
            "content": "5.95"
          },
          {
            "name": "publish_date",
            "attributes": {},
            "children": [],
            "content": "2000-12-16"
          },
          {
            "name": "description",
            "attributes": {},
            "children": [],
            "content": "A former architect battles corporate zombies,\n      an evil sorceress, and her own childhood to become queen\n      of the world."
          }
        ],
        "content": ""
      }
    ],
    "content": ""
  }
}
```

Which then is used to create the nodes.

How to query

You'd be able to query your books like:

```
{
  allBooksXml {
    edges {
      node {
        name
        xmlChildren {
          name
          content
        }
      }
    }
  }
}
```

Which would return:

```
{
  "data": {
    "allBooksXml": {
      "edges": [
        {
          "node": {
            "name": "book",
            "xmlChildren": [
              {
                "name": "author",
                "content": "Gambardella, Matthew"
              },
              {
                "name": "title",
                "content": "XML Developer's Guide"
              },
              {
                "name": "genre",
                "content": "Computer"
              },
              {
                "name": "price",
                "content": "44.95"
              },
              {
                "name": "publish_date",
                "content": "2000-10-01"
              },
              {
                "name": "description",
        "content": "An in-depth look at creating applications\n   with XML."
       }
      ]
     }
    },
    {
     "node": {
      "name": "book",
      "xmlChildren": [
       {
        "name": "author",
        "content": "Ralls, Kim"
       },
       {
        "name": "title",
        "content": "Midnight Rain"
       },
       {
        "name": "genre",
        "content": "Fantasy"
       },
       {
        "name": "price",
        "content": "5.95"
       },
       {
        "name": "publish_date",
        "content": "2000-12-16"
       },
       {
        "name": "description",
        "content": "A former architect battles corporate zombies,\n   an evil sorceress, and her own childhood to become queen\n   of the world."
       }
      ]
     }
    }
   ]
  }
 }
}
```

Note that the root element "catalog" is ignored, and nodes are created with the children elements.