Gatsby v2 introduces `StaticQuery`, a new API that allows components to retrieve data via a GraphQL query.

In this guide, you'll see an example using `StaticQuery`, and learn about the difference between a StaticQuery and a page query.

## How to use `StaticQuery` in components

### Basic example

Here is an example of a `Header` component using `StaticQuery`:

```
import React from "react"
import { StaticQuery, graphql } from "gatsby"

export default function Header() {
  return (
    <StaticQuery
      query={graphql`
        query HeadingQuery {
          site {
            siteMetadata {
              title
            }
          }
        }
      `}
      render={data => (
        <header>
          <h1>{data.site.siteMetadata.title}</h1>
        </header>
      )}
    />
  )
}
```

By using `StaticQuery`, you can colocate a component with its data. It is no longer required to, say, pass data down from `Layout` to `Header`.

### useStaticQuery

There's also a React hooks version of StaticQuery: check out the documentation on `useStaticQuery`

### Typechecking

With the above pattern, you lose the ability to typecheck with PropTypes. To regain typechecking while achieving the same result, you can change the component to:

```
import React from "react"
import { StaticQuery, graphql } from "gatsby"
import PropTypes from "prop-types"
```

```
const Header = ({ data }) => (
  <header>
    <h1>{data.site.siteMetadata.title}</h1>
  </header>
)

export default function MyHeader(props) {
  return (
    <StaticQuery
      query={graphql`
        query {
          site {
            siteMetadata {
              title
            }
          }
        }
      `}
      render={data => <Header data={data} {...props} />}
    />
  )
}

Header.propTypes = {
  data: PropTypes.shape({
    site: PropTypes.shape({
      siteMetadata: PropTypes.shape({
        title: PropTypes.string.isRequired,
      }).isRequired,
    }).isRequired,
  }).isRequired,
}
```

## How StaticQuery differs from page query

StaticQuery can do most of the things that page query can, including fragments. The main differences are:

- page queries can accept variables (via `pageContext`) but can only be added to *page* components
- StaticQuery does not accept variables (hence the name "static"), but can be used in *any* component, including pages
- StaticQuery does not work with raw React.createElement calls; please use JSX, e.g. `<StaticQuery />`