Ghostly Gatsby

Intro to Gatsby.js with Alexx



Overview

- Introduction to Gatsby
 - React
 - GraphQL
- Project Overview



What is Gatsby?

- An open-source static site generator that has performance, scalability, and security built in
- Utilizes React for building the UI and GraphQL to provide a declarative development experience



-

Why did we choose Gatsby

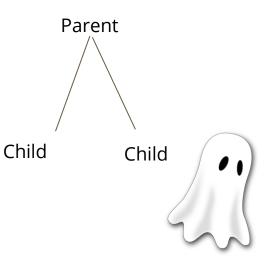
- More than just a static site generator
 - Through the use of React and GraphQL
- The #1 fastest growing framework used by organizations of all shapes and sizes
- Can use popular frameworks and tools
- Easy to get started
- Open source -> Open Source -> Open Source





React in 5 minutes

- A JavaScript framework used for building user interfaces
- Component-based
- Unidirectional (one-way) data flow -- data flows down from parent to child
- Types of data state and props
- Lifecycle methods
- Virtual DOM a diffing tool to check what to update
- Can be used for server-side rendering (SSR)



Gatsby Runtime Datafetching via React

Class Component

```
import React from "react" 8.3K (gzipped: 3.3K)
                                                                                                 import React, { useState, useEffect } from "react" 8.3K (gzipped: 3.3K)
class Example extends React.Component {
                                                                                                 const Example = () => {
  constructor(props) {
                                                                                                  const [data, setData] = useState([])
    super(props);
  this.state = {
                                                                                                    const fetchData = asvnc () => {
      data: []
                                                                                                      const response = await fetch('url');
                                                                                                      setData(response.data);
  async componentDidMount() {
   const response = await fetch('url');
                                                                                                    fetchData();
   this.setState({ data: response.json()});
                                                                                                   }, []);
  render() {
   const { data } = this.state
                                                                                                       {data.map((item, index) => (
                                                                                                         <ExampleItem
     <div>
                                                                                                          key={index}
        {data.map((item, index) => (
                                                                                                          title={item.title}
         <ExampleItem
                                                                                                          description={item.short description}
                                                                                                          url={item.link}
           kev={index}
           title={item.title}
                                                                                                          topics={item.topics}
           description={item.short description}
           url={item.link}
                                                                                                    </div>
           topics={item.topics}
      </div>
                                                                                                 const ExampleItem = ({ title, description, url, topics }) => (
                                                                                                    <a target=" blank" rel="noopener noreferrer" href={url}>{title}</a>
                                                                                                     {description}
const ExampleItem = ({ title, description, url, topics }) => (
                                                                                                    {topics.join(", ")}
                                                                                                   </div>
   <a target="_blank" rel="noopener noreferrer" href={url}>{title}</a>
   {description}
   {topics.join(", ")}
                                                                                                 export default Example
  </div>
export default Example
```

Function Component



GraphQL (briefly)

- A query language that helps you request exactly what you need
- Define the type and structure of your data
- Commonly used for APIs -- as opposed to REST
- In Gatsby -- used at buildtime; optional
 - Static queries
 - Page queries



GraphiQL

Playground that ships with GraphQL for viewing queries

```
Explorer
                                           GraphiQL
                                                                                                   Code Exporter
                                                                   Prettify
                                                                             History
                                                                                       Explorer
                                                                                                                      Docs
query SiteMetaQuery
▶ allContact
                                            1 - query SiteMetaQuery {
▶ allDirectory
                                                                             "data": {
                                                  site {
▶ allFile
                                                    siteMetadata {
                                                                                "site": {
▶ allImageSharp
                                                      title
                                                                                 "siteMetadata": {
▶ allSite
                                                                                   "title": "Women Who Code",
                                                      description
▶ allSiteBuildMetadata
                                                                                   "description": "Women Who Code is a
                                                      subTitle
▶ allSitePage
                                                                           non-profit organization dedicated to inspiring
                                                      author
▶ allSitePlugin
                                                      menuLinks {
                                                                           women to excel in technology careers.",
contact
                                            9
                                                                                    "subTitle": "Cincinnati",
                                                        name
▶ directory
                                            10
                                                        link
                                                                                   "author": "@CincyWWCode",
▶ file
▶ imageSharp
                                                                                    "menuLinks": [
▼ site
                                           12
  ▶ buildTime:
                                           13
                                                                                        "name": "Members",
  ▶ children:
                                           14
                                                                                        "link": "/members"
  ▶ host:
                                           15
  ▶ id:
  ▶ internal:
                                                                                        "name": "Quotes",
  ▶ parent:
                                                                                        "link": "/quotes"
  ▶ pathPrefix:
  ▶ polyfill:
  port:
                                                                                        "name": "Resources",
  ▶ siteMetadata:
                                                                                        "link": "/resources"
 □ buildTime
 children
 host
                                                                                        "name": "Contact".
 id
                                                                                        "link": "/contact"
 ▶ internal
 parent
 pathPrefix
 polyfill
 port
  ▼ siteMetadata
                                                                             "extensions": {}
   author

☑ description

    ▼ menuLinks
      ☑link

✓ name

✓ subTitle

   ☑title
```



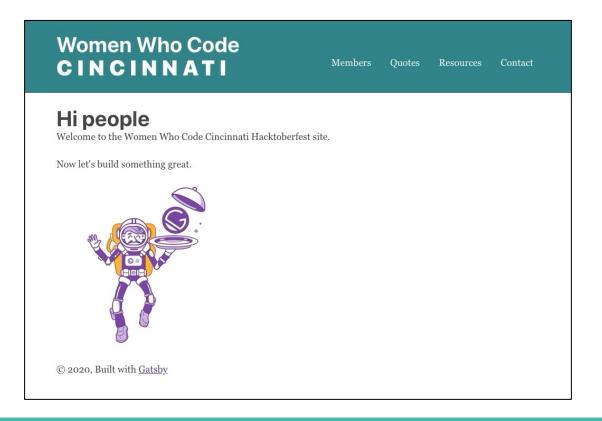
Gatsby, React, GraphQL

```
module.exports = {
 siteMetadata: {
   title: 'Women Who Code',
   subTitle: `Cincinnati`,
   description: `Women Who Code is a non-profit organization dedicated to
   inspiring women to excel in technology careers.,
   author: '@CincyWWCode',
   menuLinks: [{
     name: 'Members',
     link: '/members'
   },{
     name: 'Quotes',
     link: '/quotes'
     name: 'Resources',
     link: '/resources'
     name: 'Contact',
     link: '/contact'
```

```
import React from "react" 8.3K (gzipped: 3.3K)
import PropTypes from "prop-types" 1.7K (gzipped: 869)
import { useStaticQuery, graphql } from "gatsby"
import Header from "./header"
const Layout = ({ children }) => {
 const data = useStaticQuery(graphql`
   query SiteTitleQuery {
     site {
       siteMetadata {
         subTitle
         menuLinks {
           name
 return (
       siteTitle={data.site.siteMetadata.title}
       subTitle={data.site.siteMetadata.subTitle}
       menuLinks={data.site.siteMetadata.menuLinks}
     <div>
       <main>{children}</main>
       <footer>
         © {new Date().getFullYear()}, Built with
         <a href="https://www.gatsbyjs.com">Gatsby</a>
       </footer>
     </div>
Layout propTypes = {
 children: PropTypes.node.isRequired
export default Layout
```

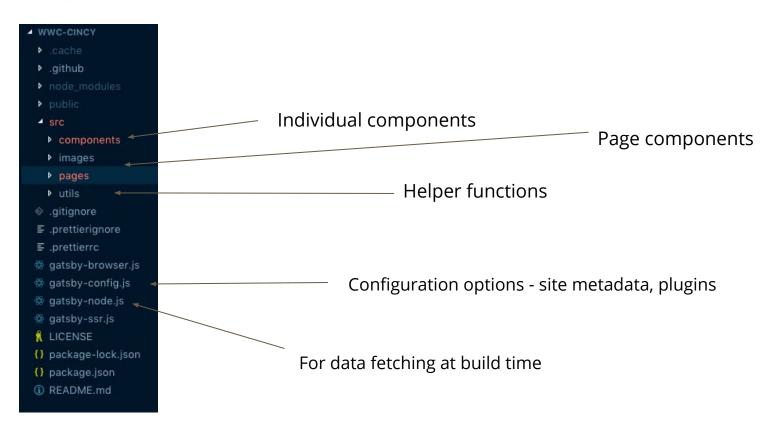


What we're working on





Project Structure





Things to remember

Nothing is set in stone

- This repo is a working repository (things can change)

We are learning together and contributing to open source

