Gatsby Study

SungMin Han

Chapter 1

ZAP

Content

- What is Gatsby?
- Gatsby structure
- Most features

What is Gatsby?

ZAP

About GatsbyJS

What is Gatsby

React 기반 정적 페이지 생성 프레임워크

(즉 React로 작성하면, 정적 컨텐츠를 생성하는 역할을 함)

Gatsby Module

What is Gatsby

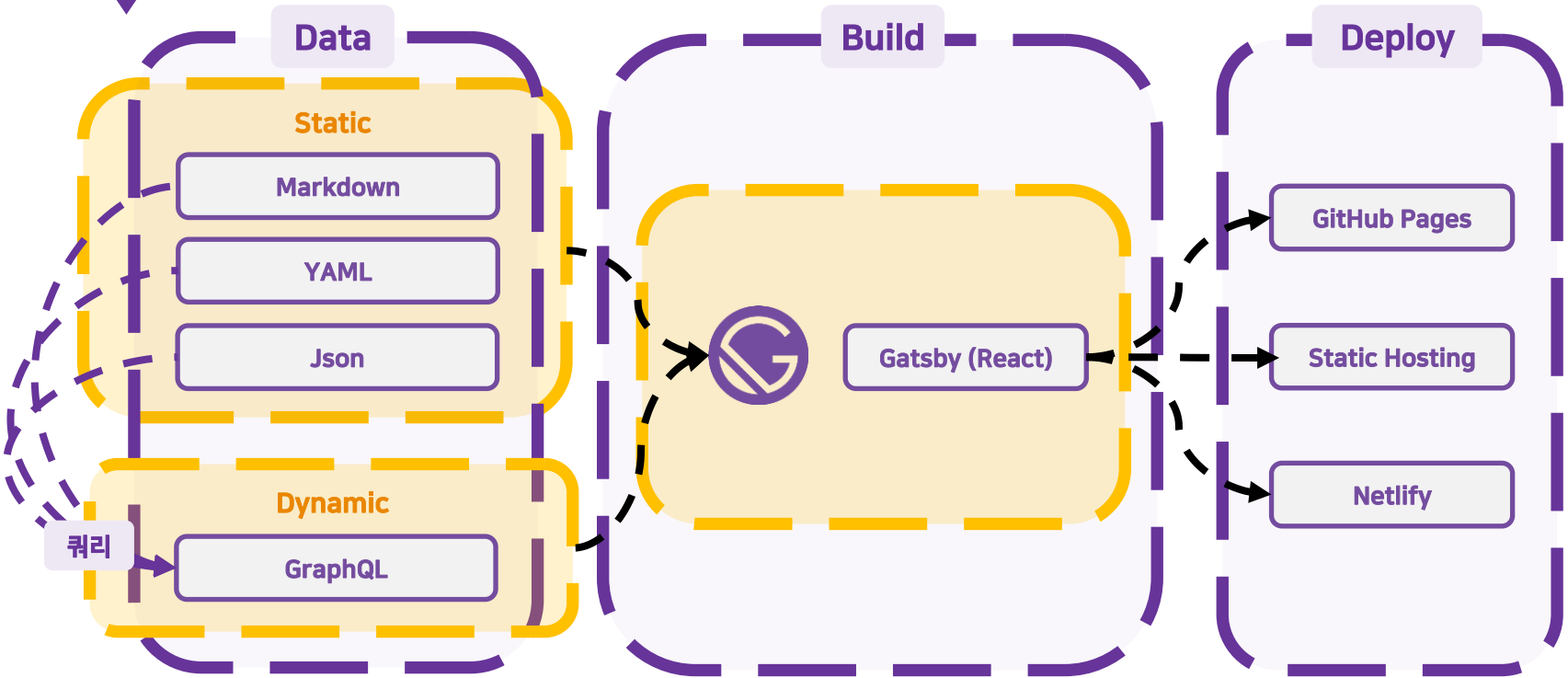
React

GraphQL

Typescript

Gatsby Workflow

What is Gatsby



ZAP

```
/
|-- /.cache
|-- /plugins
|-- /public
|-- /src
    |-- /pages
    |-- /templates
    |-- html.js
|-- /static
|-- gatsby-config.js
|-- gatsby-node.js
|-- gatsby-ssr.js
|-- gatsby-browser.js
```

Gatsby structure

ZAP

Project Structure

Gatsby structure

- `/.cache` — Gatsby 캐시 저장 경로
- `/public` — Gatsby 빌드 저장 경로, 빌드 이후 이곳에 정적 컨텐츠가 생성된다.
- `/plugins` — Gatsby 플러그인 경로, npm 외에 라이브러리나 플러그인 지정 가능.
- `/src`
  - `/pages` — 페이지 컴포넌트를 관리하는 경로
  - `/component` — React 컴포넌트 모음 경로
  - `/images` — 각종 이미지 모음 경로
- `/static` — 정적 컨텐츠 보관 경로, 이곳에 파일을 저장하면 그대로 복사한다.

Configuration Files

Gatsby structure

gatsby-browser.js

브라우저 API를 정의하는 파일.

gatsby-node.js

노드 API를 정의하는 파일.

gatsby-ssr.js

서버 사이드 랜더링(SSR) API를 정의하는 파일.

gatsby-config.js

사이트 및 플러그인의 메타 정보를 기입할 수 있는 파일,
여기서 사이트의 제목 및 각종 정보를 수정할 수 있다.

gatsby-config.js

Gatsby structure

```
module.exports = {
  siteMetadata: {
    title: `Gatsby`,
  },
  plugins: [
    `gatsby-transform-plugin`,
    {
      resolve: `gatsby-plugin-name`,
      options: {
        optionA: true,
        optionB: `Another option`,
      },
    },
  ],
}
```

ZAP

```
/
|-- /.cache
|-- /plugins
|-- /public
|-- /src
    |-- /pages
    |-- /templates
    |-- html.js
|-- /static
|-- gatsby-config.js
|-- gatsby-node.js
|-- gatsby-ssr.js
|-- gatsby-browser.js
```

Most features

ZAP

GraphQL

Most features

pageQuery

페이지 당 제공되는 쿼리로 createPage를 통해 만들어지는 페이지에 대하여
GraphQL 쿼리가 작동하며, 페이지에 채울 수 있는 내용을 Gatsby 리소스를 이용해 조회할 수 있음.

StaticQuery

페이지 당 제공되는 쿼리로 pageQuery와는 다르게
헤더, 네비게이션과 같이 비 페이지 구성 요소를 GraphQL 쿼리를 통해 가져올 수 있음.

useStaticQuery

ReactHook 기능을 통해 사용하는 StaticQuery로 StaticQuery에서 제공하는 기능과 유사하지만,
Props를 이용해 랜더링 하는 방식을 사용할 수 없음.

GraphQL

Most features

```
import React from "react"
import PostLink from "../components/post-link"
const IndexPage = ({
  data: {
    allMarkdownRemark: { edges },
  },
}) => {
  const Posts = edges
    .filter(edge => !!edge.node.frontmatter.date)
    .map(edge => <PostLink key={edge.node.id} post={edge.node} />)
  return <div>{Posts}</div>
}
export default IndexPage
```

ZAP

GraphQL

Most features

```
export const pageQuery = graphql`
  query {
    allMarkdownRemark(sort: { order: DESC, fields: [frontmatter___date] }) {
      edges {
        node {
          id
          excerpt(pruneLength: 250)
          frontmatter {
            date(formatString: "MMMM DD, YYYY")
            slug
            title
          }
        }
      }
    }
  }
```

ZAP

Node API

Most features

```
// Async/await
exports.createPages = async () => {
  // do async work
  const result = await fetchExternalData()
}
// Promise API
exports.createPages = () => {
  return new Promise((resolve, reject) => {
    // do async work
  })
}
// Callback API
exports.createPages = (_, pluginOptions, cb) => {
  // do async work
  cb()
}
```

ZAP

Node API

Most features

```
exports.createPages = ({ graphql, actions }) => {
  const { createPage } = actions
  const blogPostTemplate = path.resolve(`src/templates/blog-post.js`)
  return graphql(`
    query loadPagesQuery ($limit: Int!) {
      allMarkdownRemark(limit: $limit) {
        edges {
          node {
            frontmatter {
              slug
            }
          }
        }
      }
    }
  `, { limit: 1000 }).then(result => {
    if (result.errors) throw result.errors
    // 페이지 생성
    result.data.allMarkdownRemark.edges.forEach(edge => {
      createPage({
        path: `${edge.node.frontmatter.slug}`,
        component: blogPostTemplate,
        context: { /* Page component에 추가적인 정보를 props로 전달 */ },
      })
    })
  })
}
```

ZAP

Browser API

Most features

```
exports.shouldUpdateScroll = ({
  routerProps: { location },
  getSavedScrollPosition
}) => {
  const currentPosition = getSavedScrollPosition(location)
  const queriedPosition = getSavedScrollPosition({ pathname: `/random` })

  window.scrollTo(...(currentPosition || [0, 0]))

  return false
}
```

ZAP

Browser API

Most features

```
exports.onRouteUpdate = ({ location, prevLocation }) => {
  console.log('new pathname', location.pathname)
  console.log('old pathname', prevLocation ? prevLocation.pathname : null)
}
```

ZAP

Browser API

Most features

```
const React = require("react")
const { Provider } = require("react-redux")

const createStore = require("./src/state/createStore")
const store = createStore()

exports.wrapRootElement = ({ element }) => {
  return (
    <Provider store={store}>
      {element}
    </Provider>
  )
}
```

ZAP

Plugins

Most features

shell

```
$ npm install --save gatsby-plugin-react-helmet react-helmet
```

gatsby-config.js

```
plugins: [`gatsby-plugin-react-helmet`]
```

https://www.gatsbyjs.com/plugins/

ZAP

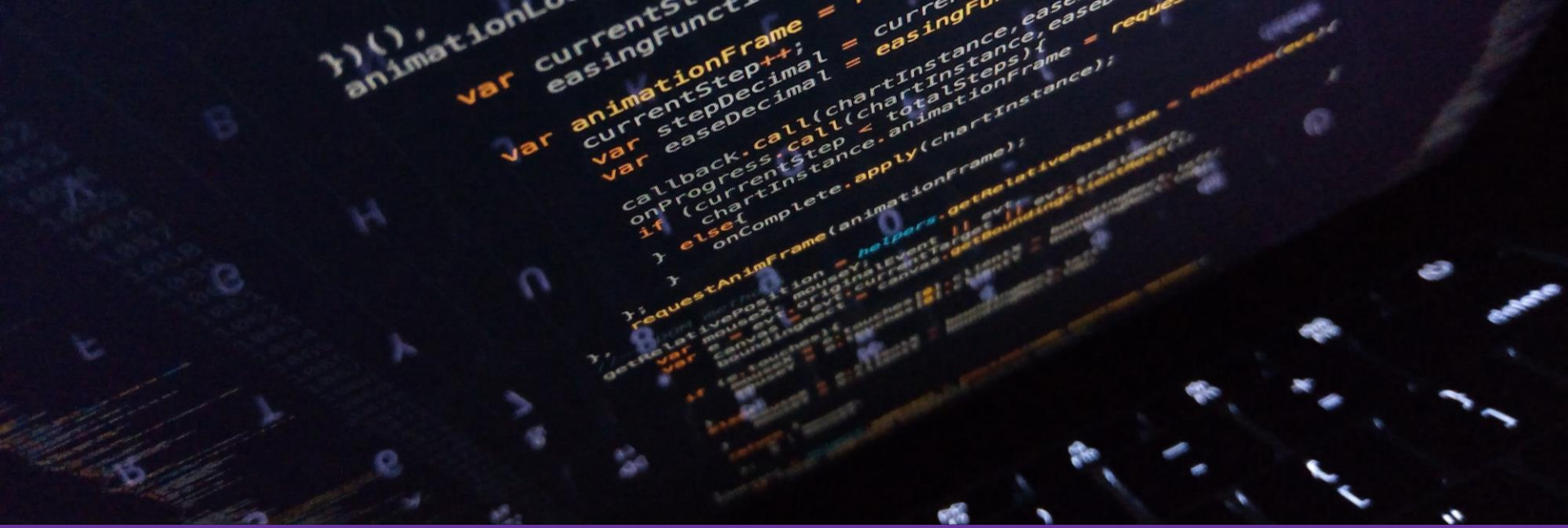
Feature Differences

Most features

vs

https://www.gatsbyjs.com/features/jamstack/

ZAP



tan Q!

ZAP