

Next.js & Netlify-cms

Blog site build

David Richards - February 2021

jamstack?

- Javascript, API (application program interface), Markup (html, css)
- Uses “someone else’s” compute resources
 - No need to run backend server (AWS, Azure, GCP, DigitalOcean, Heroku)
 - Configure for automatic scaling (up or down demand)

Reactjs

- Reactjs is often referred to as a component library
 - Since it has a set of rules it probably should be considered a framework
 - React apps are built from code components (button, navbar, page)
 - React is very un-opinionated for constructing projects allowing use of many different methods & libraries (e.g. routing: react-router, reach-router)
 - Flexibility can result in complexity of choices
 - Functional components replacing class components
 - Permits different styling methods
 - CSS-modules, styled-components, global styling, css-in-js

Nextjs

- Framework around React
 - Static Site Generation
 - Creates complete code bundle delivered to browser
 - No interaction except navigation
 - Server Side Rendered
 - Builds ahead of browser request - complete page
 - Client Side rendering delivers initial HTML then loads javascript
 - Hybrid
 - Static (product page), Server (Log-In)

Nextjs Benefits

- SEO - Rendered pages actually contain all html
- Code splitting
 - Only loads JS & CSS for viewed page
- HMR - hot module replacement - only reloads changed files
- Built-in routing - just add page
- Built-in CSS support
- API routes for API endpoints with Serverless Functions

SEO Demo

- Open VSCode
- Launch terminal (ctrl + `)
- If using alias go to projects folder/directory otherwise navigate to where you want to create projects (cd folder && mkdir)
- Create working directory ('mkdir cftw')
- >npx create-react-app react_example
 - >cd react_example
 - >npm install
 - Will open web page at localhost:3000

SEO Demo

- `>npx create-next-app next_example`
 - `>cd next_example`
 - `>code .`
 - In new window open terminal (ctrl + `)
 - `>npm install`
 - Will open web page at localhost:3000
 -

Component Styling

- Global styling using stylesheet
- CSS modules (specific to a component)
 - Avoids classname collisions
- Styled components (styling embedded in component) isolates style
- JSX - Javascript as styling (embedded in component) isolates style

Global CSS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title of the document</title>
    <link rel="stylesheet" href="./styles/buttonGlobal.css">
  </head>
  <body>
    <button class="button" type="button">Submit</button>
  </body>
</html>
```

```
.button {
  display: inline-block;
  background-color: #7b38d8;
  border-radius: 10px;
  border: 4px double #cccccc;
  color: #eeeeee;
  text-align: center;
  font-size: 28px;
  padding: 20px;
  width: 200px;
  margin: 5px;
}
```

Nav.module.css

```
import navStyles from ' ../styles/Nav.module.css'
const Nav = () => {
  return (
    <nav className={navStyles.nav}>
      <ul>
        <li>
          <Link href='/'>Home</Link>
        </li>
        <li>
          <Link href='/about'>About</Link>
        </li>
      </ul>
    </nav>
  )
}
```

```
.nav {
  height: 50px;
  padding: 10px;
  background: #000;
  color: #fff;
  display: flex;
  align-items: center;
  justify-content: flex-start;
}
```

```
.nav ul {
  display: flex;
  justify-content: center;
  align-items: center;
  list-style: none;
}
```

```
.nav ul li a {
  margin: 5px 15px;
}
```

Styled Component

Github

```
const Button = styled.a`  
  display: inline-block;  
  padding: 0.5rem 0;  
  margin: 0.5rem 1rem;  
  width: 11rem;  
  background: transparent;  
  color: white;  
`
```

Tagged Template Literal

```
render(  
  <div>  
    <Button  
      href="https://github.com/styled-components/styled-components"  
      primary  
    >  
      GitHub  
    </Button>  
  </div>  
)
```

```
    <Button as={Link} href="/docs">  
      Documentation  
    </Button>  
  </div>  
)
```

JSX Styling

```
function formatName(user) {  
  return user.firstName + ' ' + user.lastName;  
}
```

```
const user = {  
  firstName: 'Harper',  
  lastName: 'Perez'  
};
```

```
const element = (  
  <h1>  
    Hello, {formatName(user)}!  
  </h1>  
);
```

```
ReactDOM.render(  
  element,  
  document.getElementById('root')  
);
```

```
function getGreeting(user) {  
  if (user) {  
    return <h1>Hello, {formatName(user)}!</h1>;  
  }  
  return <h1>Hello, Stranger.</h1>;  
}
```