



Front End III

Enrutando una aplicación con React Router v3

Para entender mejor cómo funciona el enrutamiento estático en React, escribiremos una aplicación muy simple con cuatro páginas para empezar (/home, /main, /about y /faq). Luego, anidaremos una página dentro de la página /main, así: /main/article. Después, veremos cómo trabajar con parámetros y pasaremos un parámetro a la página /main/article, así: /main/article/title.

Cada una de las páginas será un componente de React. Es decir que para las URLs: /home, /main, /article, /about y /faq; tendremos <Home/>, <Main/>, <Article /> <About/> y <Faq/> respectivamente.

Utilizaremos React Router v3 para manejar nuestras páginas. Empezaremos por definir nuestros componentes. Serán todos componentes funcionales, no hará falta crear componentes de clase.

El primer componente que crearemos es el que contendrá todas las rutas. En este componente importaremos tres componentes desde react-router-3: **Router**, **Route** y **browserHistory**. Existe una alternativa a browserHistory llamada **hashHistory**, pero esta administra el historial de enrutamiento con el hash de la URL. Nosotros usaremos browserHistory. Veamos cómo se ve el componente App.jsx en principio:

App.jsx

```
import React from 'react';
import { Home, Main, Article, Faq, About } from './index.jsx';
import { Router, Route, browserHistory } from 'react-router-3';

class App extends React.Component {
  render() {
    return (
      <Router history={browserHistory}>
        {null}
      </Router>
    );
  }
}

export default App;
```

Ahora, veamos cómo declaramos las rutas. Dentro del par `<Router></Router>`, usaremos el componente **Route** con dos atributos llamados **path** y **component** para manejar cada ruta. Cuando el atributo `path` coincida con la URL entonces renderizará al componente referenciado por su atributo `component`.

Lo otro que haremos será meter todas las rutas dentro de la ruta raíz (`/`). Esto es importante para nuestro ejemplo porque queremos usar enlaces a las URLs y queremos que estos enlaces estén disponibles en todas las páginas que se carguen. De manera que el componente que se mostrará al cargarse la ruta raíz estará siempre presente y tendrá estos enlaces.

Esto es lo que pondremos dentro del Router de App por ahora:



```
<Route path="/" component={Home}>
  <Route path="/main" component={Main}/>
  <Route path="/faq" component={Faq}/>
  <Route path="/about" component={About}/>
</Route>
```

Nuestro componente App por ahora luce así:

App.jsx

```
import React from 'react';
import { Home, Main, Article, Faq, About } from './index.jsx';
import { Router, Route, browserHistory } from "react-router-3";

class App extends React.Component {
  render() {
    return (
      <Router history={browserHistory}>
        <Route path="/" component={Home}>
          <Route path="/main" component={Main}/>
          <Route path="/faq" component={Faq}/>
          <Route path="/about" component={About}/>
        </Route>
      </Router>
    );
  }
}

export default App;
```

El componente **Home** que se renderiza en la ruta raíz tendrá los enlaces de navegación,

y como además es el que contiene al resto de los componentes, es esencial que incluyamos sus componentes **children**: { props: children }.

Los enlaces (links) los crearemos usando un componente importado desde react-router-3 llamado **Link** (y su atributo **to**). Link es similar a la etiqueta <a/> excepto que es consciente del enrutador en el que se procesó y no dispara una recarga de página como hacen las anclas (<a>).

Creemos una lista desordenada (), ítems de listas (), y usaremos al componente Link para crear los vínculos a las páginas.

Home.jsx

```
import React from 'react';
import { Link } from 'react-router-3';

const Home = (props) => {
  return (
    <div>
      <ul role="nav">
        <li><Link to="/main">Main</Link></li>
        <li><Link to="/about">About</Link></li>
        <li><Link to="/faq">Faq</Link></li>
      </ul>
      <hr/>
      { props.children }
    </div>
  );
};

export default Home;
```

Ahora veamos al componente **Main**:

Main.jsx

```
import React from "react";

const Main = () => {
  return (
    <div>
      <h1>Main</h1>
    </div>
  );
};

export default Main;
```

<Main /> por ahora será muy simple. Solo retornará un elemento de encabezado de sección **h1** que dice "Main", todo dentro de un div. Más adelante trabajaremos con este componente para que anide al componente <Article /> definido más abajo.

Ahora veamos al componente **Article**:

Article.jsx

```
import React from "react";

const Article = () => {
  return (
    <div>
      <h1>Article</h1>
    </div>
  );
};
```

```
export default Article;
```

El componente `<Article />` también es sencillo. Retorna un `div` con un **h1** adentro que dice "Article". Este componente estará anidado dentro del componente `<Main />`.

About.jsx

```
import React from "react";

const About = () => {
  return (
    <div>
      <h1>About</h1>
    </div>
  );
};

export default About;
```

`<About />` es también simple, en el texto dice "About".

Faq.jsx

```
import React from "react";

const Faq = () => {
  return (
    <div>
      <h1>FAQ</h1>
    </div>
  );
};
```

```
};  
  
export default Faq;
```

<Faq /> es similar a todos los demás. En el texto dice "FAQ".

Hasta acá tenemos cinco componentes que representan cinco páginas: el componente **Home** que se mostrará en la ruta raíz y que contiene a los otros cuatro; el componente **Main** que se mostrará en la página /main; el componente **Article** que se muestra en la página /main/article; el componente **About** que se mostrará en la página /about y el componente **Faq** que se mostrará en la página /faq:

- /home
- /main/article
- /about
- /faq