



## Front End III

# Páginas anidadas y parámetros

Ahora veamos cómo anidar la URL /article dentro de /main. Para esto, debemos modificar nuestro componente **App** para que incluya esta ruta:

```
<Route path="/main" component={Main}>
  <Route path="/main/article/" component={Article}/>
</Route>
```

Cómo podemos ver, es simplemente anidar un componente dentro de otro, al estilo React. Veamos cómo queda **App** con este cambio:

## 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="/main/article/" component={Article}/>
          </Route>
        </Route>
      </Router>
    );
  }
}
```



```
        </Route>
        <Route path="/faq" component={Faq}/>
        <Route path="/about" component={About}/>
      </Route>
    </Router>
  );
}
}
export default App;
```

Lo otro que debemos hacer es incluir la URL en el componente **Home** que provee los enlaces de navegación:

```
    <ul role="nav">
      <li><Link to="/main">Main</Link></li>
      <li><Link to="/about">About</Link></li>
      <li><Link to="/faq">Faq</Link></li>
      <li><Link to="/main/article">Article</Link></li>
    </ul>
```

El componente **Home** ahora queda así:

## 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>
        <li><Link to="/main/article">Article</Link></li>
      </ul>
      <hr/>
      {props.children}
    </div>
  );
};
export default Home;
```

También debemos modificar a **Main** para que renderice al componente **Article** agregándole **{ props.children }**:

```
import React from "react";

const Main = (props) => {
  return (
    <div>
      <h1>Main</h1>
      { props.children }
    </div>
  );
};
export default Main;
```

Por último, veamos cómo pasar parámetros a nuestras rutas. Supongamos que tenemos varios artículos. Podríamos tener una lista de artículos con metainformación (como nombre, ID, descripción, etc.), y podríamos leer esa información desde una base de datos, desde un archivo JSON en el back end, un archivo XML, etc. Lo importante es que supongamos que disponemos de una forma de encontrar un artículo por medio de un ID.

En nuestro ejemplo, queremos que dentro de **Main** se visualice **Article** como página. Es decir, que **Article** tenga su URL y además que acepte un ID como parámetro de la URL para poder conseguir el artículo.

Entonces, dentro de **App**, usaremos **Route** (encerrando al componente **Article**) con el atributo **path** apuntando a la URL anidada y le pasaremos el ID como parámetro:

```
<Route path="/main" component={Main}>
  <Route path="/main/article/:titleId" component={Article}/>
</Route>
```

En el código de arriba, estamos pasando un identificador en el **path** llamado **id** para el artículo. Esa es la forma en que pasamos parámetros a una ruta con React Router v3.

El componente **App** queda finalmente así:

## App.jsx

```
import React from 'react';
import { Home, Main, Article, Faq, About, NotFound } 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="/main/article/:titleId"
component={Article}/>
          </Route>
        </Route>
      </Router>
    );
  }
}
```



```
        <Route path="/faq" component={Faq}/>
        <Route path="/about" component={About}/>
        <Route path="*" component={NotFound}/>
      </Route>
    </Router>
  );
}
}
export default App;
```

Ahora, agreguemos un enlace a la página /article dentro del componente **Home** que tiene la navegación:

```
    <ul role="nav">
      <li><Link to="/main">Main</Link></li>
      <li><Link to="/about">About</Link></li>
      <li><Link to="/faq">Faq</Link></li>
      <li><Link to="/main/article/101">Article</Link></li>
    </ul>
```

En el código de arriba, la línea que pasa el parámetro para la ruta nueva es la última:

```
    <li><Link to="/main/article/101">Article</Link></li>
```

El código de **Home** queda finalmente así:

## 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>  
        <li><Link  
to="/main/article/101">Article</Link></li>  
      </ul>  
      <hr/>  
      {props.children}  
    </div>  
  );  
};  
export default Home;
```

Por último, debemos obtener el parámetro en el componente **Article** que pasamos por la URL:

## Article.jsx

```
import React from "react";  
  
const Article = (props) => {  
  return (  
    <div>  
      <hr/>  
      <h1>Article</h1>
```



```
        <div> { props.params.titleId } </div>
      </div>
    );
  };
};
export default Article;
```

Ahora, cuando entremos a la página /main, se renderizará el componente **Main** y cuando hagamos clic en el enlace /main/article/101, el componente **Article** se renderizará mostrando el parámetro que pasamos en la URL: el ID 101. Por supuesto, en una aplicación real este ID lo usaríamos para algo más complejo que simplemente mostrarlo.

Al hacer clic en cualquiera de los enlaces, veremos el componente respectivo y su URL en la barra de direcciones.