

# 02-Short-Link-App

## Navigating Between Routes

- first we are going to be practicing with BrowserHistory methods.
- we are going to be exploring the browserHistory in our in our window, so what we need to do is go to client/main.js and type the following

```
import { Meteor } from 'meteor/meteor';
import React from 'react';
import ReactDOM from 'react-dom';
import { Router, Route, browserHistory } from 'react-router';

import Signup from '../imports/ui/Signup';
import Link from '../imports/ui/Link';
import NotFound from '../imports/ui/NotFound';
import Login from '../imports/ui/Login'

window.browserHistory = browserHistory;

const routes = (
  <Router history={browserHistory}>
    <Route path="/" component = {Login}/>
    <Route path="/signup" component={Signup}/>
    <Route path="/links" component={Link}/>
    <Route path="*" component={NotFound}/>
  </Router>
);

Meteor.startup(()=>{
  ReactDOM.render(routes, document.getElementById('app'));
});
```

- So now we can go to the console and type browserHistory and you'll see a lot of methods
- so if we explore the method push()
- browserHistory.push('/signup')

- that will take you to our signup route, with no page refresh-all of this happened with the history api, we were able to swap out the content without a refresh.
- we can also use the `getBack()` method, this is like pushing the back button on the browser
- `browserHistory.goBack()` - it'll take us back to the last route/page we were on
- same thing with `goForward()`
- another method is the `go()` method and takes a number for argument, if you want to go 2 pages back you would type `browserHistory.go(-2)`
- if you want to go 1 page forward you would `browserHistory.go(1)`
- what we are interacting here is the history library, and behind the scenes history library uses the history api
- to learn more on history library google [mjackson history](#)
- next we are going to be adding links in our homepage that takes to those other pages without the refresh by using a React component
- first we are going to import a named export in `Login.js`
- right here we use the `to` prop and it is like the href in an anchor tag

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

export default class Login extends React.Component{
  render(){
    return (
      <div>
        <h1>Login to Short Lnk</h1>

        login form here

        <Link to="/signup">Have an account?</Link>
      </div>
    );
  }
}
```

- we then are going to do the something with the `Signup.js` page and add Link to take us the login page

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

export default class Signup extends React.Component{
  render(){
    return (
```

```

        <div>
            <h1>Joint Short Lnk</h1>

            <Link to="/">Already have an account?</Link>
        </div>
    );
}
}

```

- we then are going to be adding a button to Link.js that will take us back to the Login.js page
- so we need an onClick event, this event takes function that we are going to call onLogout

```

import React from 'react';
import { browserHistory } from 'react-router';

export default class Link extends React.Component{
    onLogout(){
        browserHistory.push('/');
    }
    render(){
        return (
            <div>
                <h1>Your Links</h1>
                <button onClick={this.onLogout.bind(this)}>Logout</button>
            </div>
        );
    }
}

```

•