

So that we can access the router `history` outside of the `Router` component you need to create your own history object. We can reference this object from the `Auth0Provider`'s `onRedirectCallback`.

We can then use the `withAuthenticationRequired` HOC (Higher Order Component) to create a `ProtectedRoute` component that redirects anonymous users to the login page, before returning them to the protected route:

```
import React from 'react';
import { Router, Route, Switch } from 'react-router-dom';
import { Auth0Provider, withAuthenticationRequired } from '@auth0/auth0-react';
import { createBrowserHistory } from 'history';
import Profile from './Profile';

export const history = createBrowserHistory();

const ProtectedRoute = ({ component, ...args }) => (
  <Route component={withAuthenticationRequired(component)} {...args} />
);

const onRedirectCallback = (appState) => {
  // Use the router's history module to replace the url
  history.replace(appState?.returnTo || window.location.pathname);
};

export default function App() {
  return (
    <Auth0Provider
      domain="YOUR_AUTH0_DOMAIN"
      clientId="YOUR_AUTH0_CLIENT_ID"
      redirectUri={window.location.origin}
      onRedirectCallback={onRedirectCallback}
    >
      { /* Don't forget to add the history to your router */ }
      <Router history={history}>
        <Switch>
          <Route path="/" exact />
          <ProtectedRoute path="/profile" component={Profile} />
        </Switch>
      </Router>
    </Auth0Provider>
  );
}
```

See react-router example app