

[Open in app](#)[Get started](#)

Alexander Ivantsov

[Follow](#)Nov 23, 2016 · 2 min read · [Listen](#)

Save



Using React Router and History

This post is not a tutorial but describes one particular use-case and the problems that I met along the way.

In a nutshell, we use React Router in our app and we want to listen to location changes to apply some specific actions in response. There are a few ways to do that:

- [onUpdate](#) (in this case you won't get any information about current location)
- [react-redux-router](#) (it's one more package that you might not need at all and it just increases the bundle size)
- [History](#) (React Router uses this package internally, so using it explicitly doesn't add anything new to the bundle)

As you can see the last option is the best way to solve the problem: you don't add a new package to your app and it provides you with all information about current location. However, there are a couple of things you should be aware of.

First of all, the latest stable version of React Router is version 3 and for History package is version 4. However, with that version of React Router **we should use History v3**.

To see another one caveat let's try to solve our initial problem. One of the solutions could be this code:

```
import {browserHistory, Router, Route} from 'react-router';

browserHistory.listen(location => {
  browserHistory.push('/super/url');
});

const router = (
  <Router history={browserHistory}>
```



[Open in app](#)[Get started](#)

As you can see we don't use History package explicitly (but React Router uses it internally) and we listen to location changes as required. But what if our app has *base name*? Now it's getting more complicated because we use already created instance of History and we can't add base name on the fly. Then let's fix it by creating our own history:

```
import {createBrowserHistory} from 'history';
import {useRouterHistory, Router, Route} from 'react-router';

const history = useRouterHistory(createBrowserHistory)({
  basename: '/basename'
});

history.listen(location => {
  history.push('/super/url');
});

const router = (
  <Router history={history}>
    <Route path="/" component={App}>
      ...
    </Route>
  </Router>
);
```

And we're done! The only thing that you should know is that now we pass our own History instance to React Router and when you want to call e.g. *push* method you have to use exactly this instance but not *browserHistory* from React Router.

[About](#) [Help](#) [Terms](#) [Privacy](#)[Get the Medium app](#)



Open in app

Get started

