

142-Setting up Logout to Work with New Routes - Part 2 video

- lets delete these two const in routes.js

```
import { Meteor } from 'meteor/meteor';

import React from 'react';

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

import { Session } from 'meteor/session'

import Signup from '../ui/Signup';

import Dashboard from '../ui/Dashboard';

import NotFound from '../ui/NotFound';

import Login from '../ui/Login'

const unauthenticatedPages = ['/', '/signup'];

const authenticatedPages = ['/dashboard'];

const onEnterPublicPage= ()=>{

  if(Meteor.userId()){

    browserHistory.replace('/dashboard');

  }

}
```

- now over in main.js we are going to be passing our new const currentPagePrivacy to the onAuthChange function

```
import { Meteor } from 'meteor/meteor';

import ReactDOM from 'react-dom';

import { Tracker } from 'meteor/tracker';

import { Session } from 'meteor/session';

import { browserHistory } from 'react-router';

import { routes, onAuthChange } from '../imports/routes/routes';
```

```
import '../imports/startup/simple-schema-configuration.js';
```

```
Tracker.autorun(()=>{  
  const isAuthenticated = !!Meteor.userId();  
  const currentPagePrivacy = Session.get('currentPagePrivacy');  
  console.log('currentPagePrivacy ',currentPagePrivacy);  
  onAuthChange(isAuthenticated, currentPagePrivacy);  
});
```

- lets go back to routes.js and we are going to be making some changes to the onAuthChange function, we will delete the first line and change the second line of code

```
export const onAuthChange = (isAuthenticated) =>{  
  const pathname = browserHistory.getCurrentLocation().pathname;  
  const isUnauthenticatedPage = unauthenticatedPages.includes(pathname);  
  const isAuthenticatedPage = authenticatedPages.includes(pathname);  
  
  if(isUnauthenticatedPage && isAuthenticated ){  
    browserHistory.replace('/dashboard');  
  }  
  else if(isAuthenticatedPage && !isAuthenticated){  
    browserHistory.replace('/');  
  }  
  console.log('isAuthenticated ',isAuthenticated);  
};
```

- we then are going to give isUnauthenticatedPage a different value

```
export const onAuthChange = (isAuthenticated, currentPagePrivacy) =>{  
  const isUnauthenticatedPage = currentPagePrivacy === 'unauth';  
  const isAuthenticatedPage = authenticatedPages.includes(pathname);  
  
  if(isUnauthenticatedPage && isAuthenticated ){
```

```

        browserHistory.replace('/dashboard');
    }
    else if(isAuthenticatedPage && !isAuthenticated){
        browserHistory.replace('/');
    }
    console.log('isAuthenticated ',isAuthenticated);
};

```

- and now to isAuthenticatedPage will give the value if the currentPagePrivacy is auth

```

export const onAuthChange = (isAuthenticated, currentPagePrivacy) =>{
    const isUnauthenticatedPage = currentPagePrivacy === 'unauth';
    const isAuthenticatedPage = currentPagePrivacy === 'auth';

    if(isUnauthenticatedPage && isAuthenticated ){
        browserHistory.replace('/dashboard');
    }
    else if(isAuthenticatedPage && !isAuthenticated){
        browserHistory.replace('/');
    }
    console.log('isAuthenticated ',isAuthenticated);
};

```

- down in the Route component we are going to be removing the onEnter handler to all of them except for the last one

```

export const routes = (
    <Router history={browserHistory}>
        <Route onEnter={globalOnEnter} onChange={globalOnChange}>
            <Route path="/" component = {Login} privacy="unauth" onEnter = {onEnterPublicPage}/>
            <Route path="/signup" component={Signup} privacy="unauth" onEnter={onEnterPublicPage}/>
            <Route path="/dashboard" component={Dashboard} privacy="auth" onEnter=
{onEnterPrivatePage}/>

```

```

    <Route path="/dashboard/:id" component={Dashboard} privacy="auth" onEnter={onEnterNotePage}/>

    <Route path="*" component={NotFound}/>

  </Route>

</Router>

);

```

- and now it should look like this

```

export const routes = (

  <Router history={browserHistory}>

    <Route onEnter={globalOnEnter} onChange={globalOnChange}>

      <Route path="/" component = {Login} privacy="unauth"/>

      <Route path="/signup" component={Signup} privacy="unauth"/>

      <Route path="/dashboard" component={Dashboard} privacy="auth" />

      <Route path="/dashboard/:id" component={Dashboard} privacy="auth" onEnter={onEnterNotePage}/>

      <Route path="*" component={NotFound}/>

    </Route>

  </Router>

);

```

- we are also going to be deleting those function up above

```

import { Meteor } from 'meteor/meteor';

import React from 'react';

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

import { Session } from 'meteor/session'

import Signup from '../ui/Signup';

import Dashboard from '../ui/Dashboard';

import NotFound from '../ui/NotFound';

import Login from '../ui/Login'

```

```
const onEnterPublicPage= ()=>{
  if(Meteor.userId()){
    browserHistory.replace('/dashboard');
  }
}
```

```
const onEnterPrivatePage = ()=>{
  if(!Meteor.userId()){
    browserHistory.replace('/');
  }
}
```

- and in `onEnterNotePage` we no longer `browserHistory.replace` because that is already taking care of so our function should look like this now

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

import Signup from '../ui/Signup';
import Dashboard from '../ui/Dashboard';
import NotFound from '../ui/NotFound';
import Login from '../ui/Login'

const onEnterNotePage = (nextState)=>{
  Session.set('selectedNoteId', nextState.params.id);
}
```

- and now over in the browser when we select a note and logout we kicked out to the home page which is what we want
- but there is one problem when we create a new note and log out and we login again we see the text in the field we just created , and we dont' want that and we also dont see the url to that specifc note, so we need to fix that with another property called `onLeave`

```

export const routes = (
  <Router history={browserHistory}>
    <Route onEnter={globalOnEnter} onChange={globalOnChange}>
      <Route path="/" component = {Login} privacy="unauth"/>
      <Route path="/signup" component={Signup} privacy="unauth"/>
      <Route path="/dashboard" component={Dashboard} privacy="auth" />
      <Route path="/dashboard/:id" component={Dashboard} privacy="auth" onEnter={onEnterNotePage}
    onLeave={onLeaveNotePage}/>
      <Route path="*" component={NotFound}/>
    </Route>
  </Router>
)

```

- lets now create this function up above and all we are going to do in that function is clear the selectedNoteId

```

import { Meteor } from 'meteor/meteor';
import React from 'react';
import { Router, Route, browserHistory } from 'react-router';
import { Session } from 'meteor/session'

import Signup from '../ui/Signup';
import Dashboard from '../ui/Dashboard';
import NotFound from '../ui/NotFound';
import Login from '../ui/Login'

const onEnterNotePage = (nextState)=>{
  Session.set('selectedNoteId', nextState.params.id);
}

const onLeaveNotePage = ()=>{
  Session.set('selectedNoteId', undefined);
}

```

- **now we can add, commit and push our code**

```
joses-MacBook-Pro:notes mendoza$ git add .
```

```
joses-MacBook-Pro:notes mendoza$ git commit -m "Refactor router to allow for dynamic urls"
```

```
joses-MacBook-Pro:notes mendoza$ git push
```