

09-Short-Link-App

Refactoring Router Code - [Video](#)

- we are going to be pulling out all the router code from client and server main.js and putting it in its own file
- the new location is going to be in the imports folder, call it routes with a file name routers.js
- lets copy this from client/main.js and paste on the routes.js file

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

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

const unauthenticatedPages = ['/', '/signup'];
const authenticatedPages = ['/links'];
const onEnterPublicPage = () => {
  if (Meteor.userId()) {
    browserHistory.replace('/links');
  }
}

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

const routes = (
  <Router history={browserHistory}>
    <Route path="/" component={Login} onEnter={onEnterPublicPage}/>
    <Route path="/signup" component={Signup} onEnter={onEnterPublicPage}/>
```

```

    <Route path="/links" component={Link} onEnter={onEnterPrivatePage}/>

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

  </Router>

);

```

- now lets change the imports paths to make it look like this

```

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

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

```

- we are now going to change the Tracker.autorun code in the client/main.js
- we are going to cut this following code-only the bold part, leaving only the isAuthenticated variable

```

Tracker.autorun(()=>{
  const isAuthenticated = !!Meteor.userId();

  const pathname = browserHistory.getCurrentLocation().pathname;
  const isUnauthenticatedPage = unauthenticatedPages.includes(pathname);
  const isAuthenticatedPage = authenticatedPages.includes(pathname);

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

});

```

- so client/main.js should look like this now

```
const routes = (  
  <Router history={browserHistory}>  
    <Route path="/" component={Login} onEnter={onEnterPublicPage}/>  
    <Route path="/signup" component={Signup} onEnter={onEnterPublicPage}/>  
    <Route path="/links" component={Link} onEnter={onEnterPrivatePage}/>  
    <Route path="*" component={NotFound}/>  
  </Router>  
)  
);  
  
Tracker.autorun(()=>{  
  const isAuthenticated = !!Meteor.userId();  
});
```

- now in routes.js lets copy that code in a new function

```
import { Meteor } from 'meteor/meteor';  
import React from 'react';  
import ReactDOM from 'react-dom';  
import { Router, Route, browserHistory } from 'react-router';  
import { Tracker } from 'meteor/tracker';  
  
import Signup from '../ui/Signup';  
import Link from '../ui/Link';  
import NotFound from '../ui/NotFound';  
import Login from '../ui/Login';  
  
const unauthenticatedPages = ['/', '/signup'];  
const authenticatedPages = ['/links'];  
const onEnterPublicPage = ()=>{  
  if(Meteor.userId()){  
    browserHistory.replace('/links');  
  }  
}
```

```

}

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

const onAuthChange = () =>{
  const pathname = browserHistory.getCurrentLocation().pathname;
  const isUnauthenticatedPage = unauthenticatedPages.includes(pathname);
  const isAuthenticatedPage = authenticatedPages.includes(pathname);

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

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

```

- lets pass in the isAuthenticated variable

```

const onAuthChange = (isAuthenticated) =>{
  const pathname = browserHistory.getCurrentLocation().pathname;
  const isUnauthenticatedPage = unauthenticatedPages.includes(pathname);
  const isAuthenticatedPage = authenticatedPages.includes(pathname);

```

- lets take out some more stuff from client/main.js, and this is how it should look like now

```
import { Meteor } from 'meteor/meteor';
import ReactDOM from 'react-dom';
import { Tracker } from 'meteor/tracker';

Tracker.autorun(()=>{
  const isAuthenticated = !!Meteor.userId();
});

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

- we can go clean up routes.js as well, lets start with the imports, this is what it should like

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

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

- now we need to export some files from the router.js, so main.js can use them, so lets do that

```
export const onAuthChange = (isAuthenticated) =>{
  const pathname = browserHistory.getCurrentLocation().pathname;
  const isUnauthenticatedPage = unauthenticatedPages.includes(pathname);
  const isAuthenticatedPage = authenticatedPages.includes(pathname);

  if(isUnauthenticatedPage && isAuthenticated ){
    browserHistory.replace('/links');
  }
  else if(isAuthenticatedPage && !isAuthenticated){
    browserHistory.replace('/');
  }
}
```

```

    console.log('isAuthenticated ',isAuthenticated);
  };
  export const routes = (
    <Router history={browserHistory}>
      <Route path="/" component = {Login} onEnter = {onEnterPublicPage}/>
      <Route path="/signup" component={Signup} onEnter={onEnterPublicPage}/>
      <Route path="/links" component={Link} onEnter={onEnterPrivatePage}/>
      <Route path="*" component={NotFound}/>
    </Router>
  );

```

- now lets go to client/main.js and import those two functions
- and also call the onAuthChange method

```

import { Meteor } from 'meteor/meteor';
import ReactDOM from 'react-dom';
import { Tracker } from 'meteor/tracker';

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

Tracker.autorun(()=>{
  const isAuthenticated = !!Meteor.userId();
  onAuthChange(isAuthenticated);
});

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

```

- now we are going to be doing the same with our server/client.js code
- lets first create a new folder in imports called api, and a file called inside the api called users.js
- this file is going to be responsible for making the call to the Accounts method.
- so lets cut out that code and paste in the users.js file

```

Accounts.validateNewUser((user)=>{
  const email = user.emails[0].address

```

```

try{
  new SimpleSchema({
    email:{
      type: String,
      regEx: SimpleSchema.RegEx.Email
    }
  }).validate({ email });
}catch (e) {
  throw new Meteor.Error(400, e.message )
}
return true;
});

```

- now lets add the appropriate imports on top

```

import { Meteor } from 'meteor/meteor';
import SimpleSchema from 'simpl-schema';
import { Accounts } from 'meteor/accounts-base';

```

```

Accounts.validateNewUser((user)=>{
  const email = user.emails[0].address

  try{
    new SimpleSchema({
      email:{
        type: String,
        regEx: SimpleSchema.RegEx.Email
      }
    }).validate({ email });
  }catch (e) {
    throw new Meteor.Error(400, e.message )
  }
  return true;
});

```

- now lets go back to the server/main.js file and clean it up
- this is what it server/main.js should be now

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

```
import '../imports/api/users'
```

```
Meteor.startup(() => {
```

```
});
```

- now test your app and check that everything is working