

MỤC TIÊU:

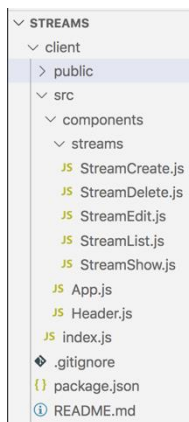
Kết thúc bài thực hành này bạn có khả năng

- ✓ React router
- ✓ Authentication với react

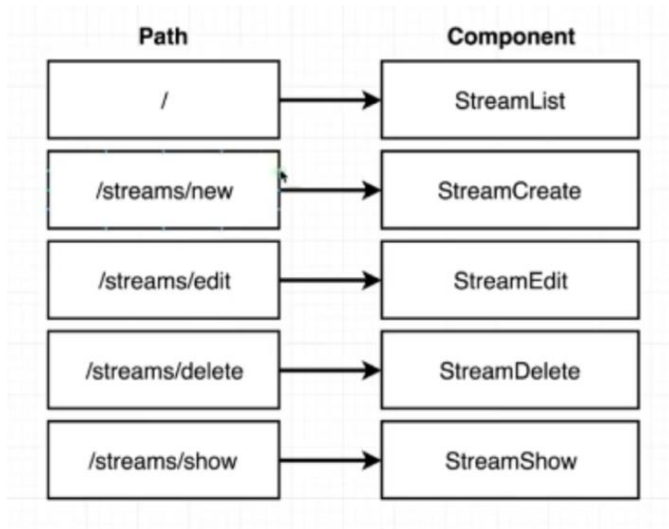
NỘI DUNG

Bài 1: React router

- Tạo project có tên là **streams** có cấu trúc, mô tả và hướng dẫn như sau
- Mô tả
- Cấu trúc



- Cài đặt react-router-dom và tạo route đến các component như sau



- Hướng dẫn

```
const App = () => {
  return (
    <div className="ui container">
      <BrowserRouter>
        <div>
          <Header />
          <Route path="/" exact component={StreamList} />
          <Route path="/streams/new" exact component={StreamCreate} />
          <Route path="/streams/edit" exact component={StreamEdit} />
          <Route path="/streams/delete" exact component={StreamDelete} />
          <Route path="/streams/show" exact component={StreamShow} />
        </div>
      </BrowserRouter>
    </div>
  );
};
```

Bài 2: Google API library

- GoogleAuth component

```
JS GoogleAuth.js X
client > src > components > JS GoogleAuth.js > ...
1 import React from 'react';
2
3 class GoogleAuth extends React.Component {
4   componentDidMount() {
5     window.gapi.load('client:auth2', () => {
6       window.gapi.client.init({
7         clientId:
8           '797401886567-9cumct9mrt3v2va409rasa7fa6fq02hh.apps.googleusercontent.com',
9         scope: 'email'
10       });
11     });
12   }
13
14   render() {
15     return <div>Google Auth</div>;
16   }
17 }
18
19 export default GoogleAuth;
```

- Render Authentication status

```
JS Header.js JS GoogleAuth.js X
client > src > components > JS GoogleAuth.js > ...
1 import React from 'react';
2
3 class GoogleAuth extends React.Component {
4   state = { isSignedIn: null };
5
6 >   componentDidMount() { ...
19   }
20
21   renderAuthButton() {
22     if (this.state.isSignedIn === null) {
23       return <div>I dont know if we are signed in</div>;
24     } else if (this.state.isSignedIn) {
25       return <div>I am signed in!</div>;
26     } else {
27       return <div>I am not signed in</div>;
28     }
29   }
30
31   render() {
32     return <div>{this.renderAuthButton()}</div>;
33   }
34 }
35
36 export default GoogleAuth;
```

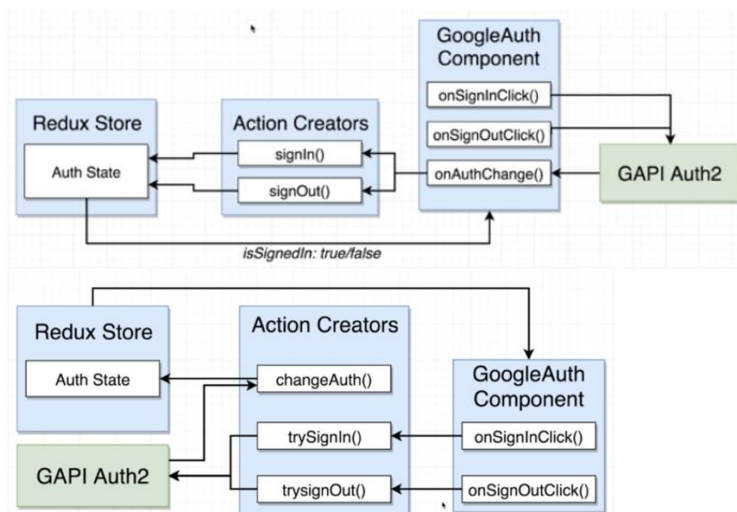
```
JS Header.js • JS GoogleAuth.js
client > src > components > JS Header.js > ...
1 import React from 'react';
2 import { Link } from 'react-router-dom';
3 import GoogleAuth from './GoogleAuth';
4
5 const Header = () => {
6   return (
7     <div className="ui secondary pointing menu">
8 > <Link to="/" className="item"> ...
11 > <div className="right menu"> ...
17 </div>
18 );
19 };
20
21 export default Header;
```

- Update State

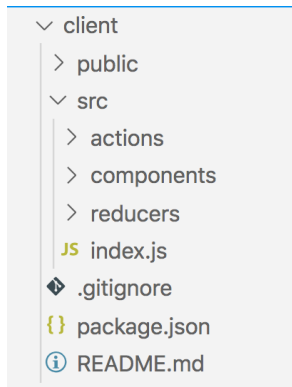
```
JS GoogleAuth.js X
client > src > components > JS GoogleAuth.js > ...
1 import React from 'react';
2
3 class GoogleAuth extends React.Component {
4   state = { isSignedIn: null };
5
6 > componentDidMount() { ...
20 }
21
22 onAuthChange = () => {
23   this.setState({ isSignedIn: this.auth.isSignedIn.get() });
24 };
25
26 > renderAuthButton() { ...
34 }
35
36 > render() { ...
38 }
39 }
40
41 export default GoogleAuth;
```

Bài 3: OAuth Redux Design

- Mô tả



- Cấu trúc



- Hướng dẫn

- Bước 1: Set up

```

client > src > reducers > JS index.js > ...
1 import { combineReducers } from 'redux';
2
3 export default combineReducers({
4   replaceMe: () => 'asldfkj'
5 });
6
  
```

```

JS index.js x
client > src > JS index.js > ...
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import { Provider } from 'react-redux';
4 import { createStore } from 'redux';
5
6 import App from './components/App';
7 import reducers from './reducers';
8
9 const store = createStore(reducers);
10
11 ReactDOM.render(
12   <Provider store={store}>
13     <App />
14   </Provider>,
15   document.querySelector('#root')
16 );
  
```

GoogleAuth.js

```

class GoogleAuth extends React.Component {
  state = { isSignedIn: null };

  componentDidMount() {
    // ...
  }

  onAuthChange = () => {
    this.setState({ isSignedIn: this.auth.isSignedIn.get() });
  };

  onSignInClick = () => {
    this.auth.signIn();
  };

  onSignOutClick = () => {
    this.auth.signOut();
  };

  renderAuthButton() {
    if (this.state.isSignedIn === null) {
      return null;
    } else if (this.state.isSignedIn) {
      return (
        <button onClick={this.onSignOutClick} className="ui red google button">
          <i className="google icon" />
          Sign Out
        </button>
      );
    } else {
      return (
        <button onClick={this.onSignInClick} className="ui red google button">
          <i className="google icon" />
          Sign In with Google
        </button>
      );
    }
  }

  render() {
    return <div>{this.renderAuthButton()}</div>;
  }
}
  
```

- Bước 2: Action creators

JS index.js X

client > src > actions > JS index.js > ...

```

1 export const signIn = () => {
2   return {
3     type: 'SIGN_IN'
4   };
5 };
6
7 export const signOut = () => {
8   return {
9     type: 'SIGN_OUT'
10  };
11 };

```

JS index.js JS GoogleAuth.js X

client > src > components > JS GoogleAuth.js > GoogleAuth

```

23
24
25 onAuthChange = isSignedIn => {
26   if (isSignedIn) {
27     this.props.signIn();
28   } else {
29     this.props.signOut();
30   }
31 };

```

- Bước 3: Auth Reducer

JS authReducer.js X

client > src > reducers > JS authReducer.js > ...

```

1 const INITIAL_STATE = {
2   isSignedIn: null
3 };
4
5 export default (state = INITIAL_STATE, action) => {
6   switch (action.type) {
7     case 'SIGN_IN':
8       return { ...state, isSignedIn: true };
9     case 'SIGN_OUT':
10      return { ...state, isSignedIn: false };
11     default:
12      return state;
13   }
14 };

```

JS index.js X

client > src > reducers > JS index.js > ...

```

1 import { combineReducers } from 'redux';
2 import authReducer from './authReducer';
3
4 export default combineReducers({
5   auth: authReducer
6 });

```

- Bước 4: Xử lý Auth Status

JS index.js JS GoogleAuth.js X

client > src > components > JS GoogleAuth.js >

```

64
65 export default connect(
66   null,
67   { signIn, signOut }
68 )(GoogleAuth);
69

```

Bài 4: Nâng cao

- Action type & UserID

client > src > actions > JS types.js > ...

```

1 export const SIGN_IN = 'SIGN_IN';
2 export const SIGN_OUT = 'SIGN_OUT';
3

```

client > src > actions > JS index.js > ...

```

1 import { SIGN_IN, SIGN_OUT } from './types';
2
3 export const signIn = (userId) => {
4   return {
5     type: SIGN_IN,
6     payload: userId
7   };
8 };
9
10 export const signOut = () => {
11   return {
12     type: SIGN_OUT
13   };
14 };

```

client > src > reducers > JS authReducer.js > ...

```

1 import { SIGN_IN, SIGN_OUT } from '../actions/types';
2
3 const INITIAL_STATE = {
4   isSignedIn: null,
5   userId: null
6 };
7
8 export default (state = INITIAL_STATE, action) => {
9   switch (action.type) {
10     case SIGN_IN:
11       return { ...state, isSignedIn: true, userId: action.payload };
12     case SIGN_OUT:
13       return { ...state, isSignedIn: false, userId: null };
14     default:
15       return state;
16   }
17 };

```

Bài 5:

- Giảng viên cho thêm

***** Yêu cầu nộp bài:**

SV nén file (*hoặc share thư mục google drive*) bao gồm các yêu cầu đã thực hiện trên, nộp LMS đúng thời gian quy định của giảng viên. KHÔNG NỘP BÀI COI NHƯ KHÔNG CÓ ĐIỂM.

--- Hết ---