

Showing the detail of selected user

29 August 2017 02:44

Now we want to display the selected user detail.

- Create user-detail.js file in containers folder
- Write below code

```
import React, {Component} from 'react';
import {connect} from 'react-redux';
class UserDetails extends Component{
  render(){
    // add this code once the complete component is created, show the error
    if(!this.props.user){
      return (<h3>Select a user...</h3>)
    }
    return(
      <div>
        <p>
          First Name : {this.props.user.first} <br/>
          Last Name : {this.props.user.last} <br/>
          Age : {this.props.user.age} <br/>
          Description : {this.props.user.description}
        </p>
      </div>
    )
  }
}
function mapStateToProps(state){
  return{
    user:state.activeUser
  }
}
export default connect(mapStateToProps)(UserDetails);
```

- Make changes in the component/app.js
- Import the UserDetails
- Call the <UserDetail/> in the appropriate place.

```
import React from 'react';
import UserList from '../containers/user-list';

import UserDetails from '../containers/user-detail';

const App=()=>(
  <div>
    <h1>React-Redux Application Demo</h1>
    <hr/>
    <h2>User List</h2>
    <UserList/>
    <hr/>
    <h2>User Detail</h2>

    <UserDetail/>

    <hr/>
  </div>
);
export default App;
```

Check the application.

It will show the error as :

TypeError: Cannot read property 'first' of null

This is because in the reducer-active-user.js file we have created function and this function takes the state, this is state is null initially, and when UserDetails component renders the user detail, then we are trying to show the property of null value like user.first, user.last, user.description etc. and this is exactly what error is saying.

To avoid this error for the time being make below changes in the reducer-active-user.js file

Instead of state=null, write state={}

Follow the code.

```
export default function (state = {}, action) {
  switch (action.type) {
    case "USER_SELECTED":
      return action.payload;
    break;
  }
}
```

```
return state;
}
```

But now if you see the result initially in the user detail you will just get the labels without data, because once the application loads, there is no user selected. But if you select the user, user detail section will be updated.

But ideally it should not display the user detail section if there is no user, rather it should prompt user to select the user. So first of all undo the changes, make state=null from state={}

Now before returning user detail from render method of the UserDetails component you need to check if user is null it should display the message as "select user....".

You need to add below code in render method before return statement.

```
if(!this.props.user){
return <h2>Select User.....</h2>
}
```

Follow the complete code of containers/ user-detail.js below.

```
import React, { Component } from 'react';
import { connect } from 'react-redux';
class UserDetails extends Component {
render() {

if(!this.props.user){
return <h2>Select User.....</h2>
}

return (
<div>
<p>
First Name : {this.props.user.first} <br />
Last Name : {this.props.user.last} <br />
Age : {this.props.user.age} <br />
Description : {this.props.user.description}
</p>
</div>
)
}
}
function mapStateToProps(state){
return{
user:state.ActiveUser
}
}
export default connect(mapStateToProps)(UserDetails);
```