

Problems and Solutions

1. Ensure Session Data is Retrieved Correctly

Problem: If the session data isn't being read correctly on a new session, it could be due to the way the `useEffect` is set up or the state management.

Solution: Use a more reliable way to synchronize session data. Instead of relying solely on `useEffect`, consider moving session management to a context provider.

2. Context-Based State Management

Using a context to manage authentication and session state will help ensure that your app always has the latest session data and can react to changes properly.

Steps:

- Create an Auth Context:

javascript

 Copy code

```
// AuthContext.js import React, { createContext, useState, useContext,
useEffect } from 'react'; import { useNavigate } from 'react-router-dom';
const AuthContext = createContext(); export const AuthProvider = ({ children
}) => { const [token, setToken] = useState(''); const [role, setRole] =
useState(''); const navigate = useNavigate(); useEffect(() => { const
sessionToken = sessionStorage.getItem('token'); const sessionRole =
sessionStorage.getItem('role'); setToken(sessionToken || '');
setRole(sessionRole || ''); }, []); // Empty dependency array ensures this
runs only once on mount useEffect(() => { if (token && role) { if (role ===
'admin') navigate('/admin'); else if (role === 'editor') navigate('/editor');
else if (role === 'user') navigate('/user'); else navigate('/'); } }, [token,
role, navigate]); return ( <AuthContext.Provider value={{ token, role,
setToken, setRole }}> {children} </AuthContext.Provider> ); }; export const
useAuth = () => useContext(AuthContext);
```

- Wrap Your App with AuthProvider:

javascript

 Copy code

```
// index.js or App.js import React from 'react'; import ReactDOM from 'react-
dom'; import { BrowserRouter as Router } from 'react-router-dom'; import App
from './App'; import { AuthProvider } from './AuthContext'; ReactDOM.render(
<Router> <AuthProvider> <App /> </AuthProvider> </Router>,
document.getElementById('root') );
```

3. Handle Login in Context

Update the login logic to use the context for setting the token and role, which will trigger the redirection.

Updated Login Component:

javascript



```
import { useState } from 'react'; import { useNavigate } from 'react-router-dom';
import { LoginApi } from '../Services/API'; import { useAuth } from
'../AuthContext'; function Login() { const [username, setUsername] = useState('');
const [password, setPassword] = useState(''); const navigate = useNavigate();
const { setToken, setRole } = useAuth(); const handleSubmit = async () => { if
(!username || !password) { console.log('empty data'); return; } try { const result
= await LoginApi(username, password); if (result.request.status === 200) { const {
role, token } = result.data; sessionStorage.setItem('role', role);
sessionStorage.setItem('token', token); setToken(token); setRole(role); } else {
console.warn(result.response.data || result.response.message); } } catch (error) {
console.warn(error); } }; return ( <div> <input type="text" placeholder="username"
onChange={(e) => setUsername(e.target.value)} /> <input type="password"
placeholder="password" onChange={(e) => setPassword(e.target.value)} /> <button
onClick={handleSubmit}>Login</button> </div> ); } export default Login;
```

4. Ensure Routes are Protected

Make sure your routes are correctly set up to handle unauthorized access. For instance, if a user tries to access a route they are not authorized for, they should be redirected accordingly.

Summary

By using a context provider for authentication, you ensure that your application has a centralized and consistent way of managing session data. This approach helps in avoiding common pitfalls with state management and session updates.

1. **Use Context for Authentication** to manage session state and navigate based on role.
2. **Wrap Your App with AuthProvider** to ensure the entire application has access to the auth context.
3. **Update Login Logic** to use context setters for token and role, ensuring that changes are reflected across the app.

This should address the issues with session data not being updated or read correctly.