



**HOC**

Aqib Rehman

```
// withLoading.jsx
```

```
import React from 'react';
```

```
const withLoading = (WrappedComponent) => {  
  return function WithLoadingComponent({ isLoading, ...props }) {  
    if (isLoading) {  
      return <h2>Loading...</h2>;  
    }  
    return <WrappedComponent {...props} />;  
  };  
};
```

```
export default withLoading;
```

```
// UserList.jsx
```

```
import React from 'react';
```

```
const UserList = ({ users }) => {
```

```
  return (
```

```
    <ul>
```

```
      {users.map((user, idx) => (
```

```
        <li key={idx}>{user}</li>
```

```
      ))}
```

```
    </ul>
```

```
  );
```

```
};
```

```
export default UserList;
```

```
// App.jsx

import React, { useState, useEffect } from 'react';
import UserList from './UserList';
import withLoading from './withLoading';

const EnhancedUserList = withLoading(UserList);

const App = () => {
  const [loading, setLoading] = useState(true);
  const [users, setUsers] = useState([]);
```

```
  useEffect(() => {
    setTimeout(() => {
      setUsers(['Alice', 'Bob', 'Charlie']);
      setLoading(false);
    }, 2000);
  }, []);

  return (
    <div>
      <h1>Users</h1>
      <EnhancedUserList isLoading={loading} users={users}
    />
    </div>
  );
};

export default App;
```

# Another Example

```
// withAuth.jsx
import React from 'react';

const withAuth = (WrappedComponent) => {
  return function WithAuthComponent({ isLoggedIn, ...props }) {
    if (!isLoggedIn) {
      return <h2>You must be logged in to view this content.</h2>;
    }
    return <WrappedComponent {...props} />;
  };
};

export default withAuth;
```

```
// Dashboard.jsx
import React from 'react';

const Dashboard = ({ username }) => {
  return <h2>Welcome to your dashboard, {username}!</h2>;
};

export default Dashboard;
```

```
// App.js
import React, { useState } from 'react';
import Dashboard from './Dashboard';
import withAuth from './withAuth';
```

```
const ProtectedDashboard =
withAuth(Dashboard);
```

```
const App = () => {
  const [isLoggedIn, setIsLoggedIn] =
useState(false);
```

```
  const handleLogin = () => {
    setIsLoggedIn(true); // login logic
  };
```

```
  return (
    <div>
      <h1>My App</h1>
      {!isLoggedIn && <button
onClick={handleLogin}>Login</button>}
      <ProtectedDashboard
isLoggedIn={isLoggedIn} username="Ali" />
    </div>
  );
};

export default App;
```