

import { useOutletContext } from "react-router-dom";

```
import { useOutletContext } from "react-router-dom";
```

imports a React Router hook that lets child route components access data (like state or functions) passed down from a parent route using the `<Outlet context={...} />` prop, avoiding prop drilling and simplifying data sharing in nested routes. This hook provides a convenient way to share dynamic data between parent layouts and their nested content, making it ideal for managing user info, theme settings, or data fetching states across different views.

How it works

1. **Parent Route (e.g., `DashboardLayout.js`)**: Renders the `<Outlet>` component and passes data to its `context` prop.

```
import { Outlet } from 'react-router-dom';

function DashboardLayout() {
  const [user, setUser] = React.useState({ name: 'Alice' });
  return (
    <div>
      <h1>Dashboard</h1>
      /* Pass data (or state/setters) through context */
      <Outlet context={{ user, setUser }} />
    </div>
  );
}
```

2. **Child Route (e.g., `Profile.js`)**: Uses `useOutletContext()` to get the passed data.

```
import { useOutletContext } from 'react-router-dom';

function Profile() {
  // Destructure data from the context
  const { user, setUser } = useOutletContext();
  return (
    <div>
```

```
    <p>Welcome, {user.name}</p>
    <button onClick={() => setUser({ name: 'Bob' })}>Change User</button>
  </div>
);
}
```

Key benefits

- **Avoids Prop Drilling:** Data is accessible to any nested component without passing it through every intermediate component.
- **Centralized Data:** Parent routes manage state or data needed by multiple children.
- **TypeScript Friendly:** Can be easily typed for better development experience

Revision #3

Created 29 October 2025 02:43:39 by AI API

Updated 9 December 2025 11:41:57 by AI Channel