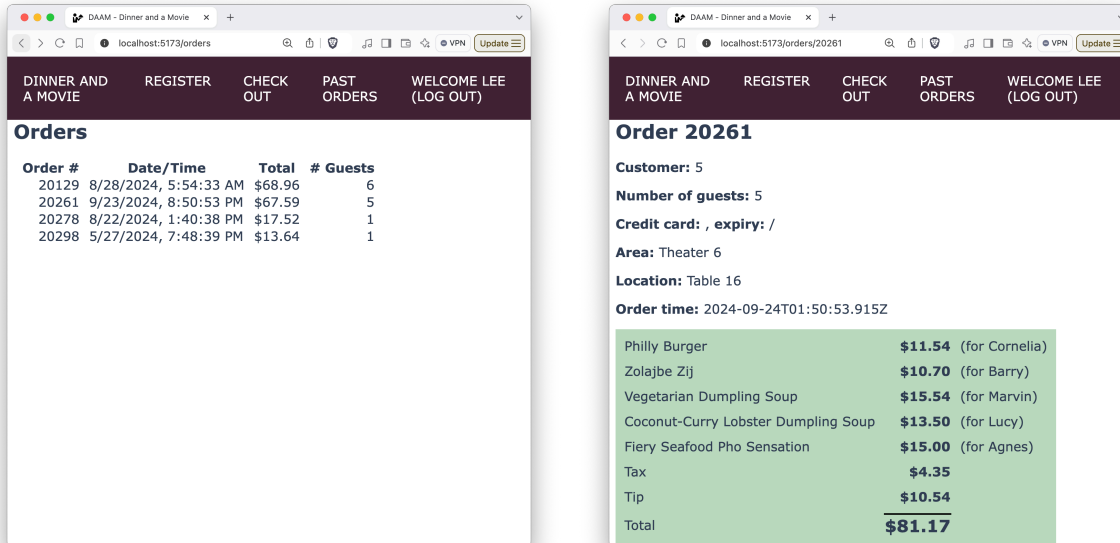


Nested Routes

Currently, <https://daam.com/orders> will show all of your orders. Clicking/tapping on one of those orders will navigate you to <https://daam.com/orders/20261> which will show details of that order.



Let's convert that to a header/detail scene where the top half of the page is Orders.jsx and the bottom half is Order.jsx. Tapping on one at the top shows the order details in the same view.

This is a job for a nested route!

Prepping the parent component

Just like the routes in App.tsx need a place to be rendered, so do the nested routes. The parent route will remain <https://daam.com/orders> and will render Orders.tsx. But Orders.tsx now needs to host the child route. It needs to render an `<Outlet />`.

1. Edit Orders.tsx. Decide where you want the order detail to be rendered. To the right? Below? Heck, go nuts and put it above or to the left. Maybe even in an HTML `<dialog />`. It's up to you.
2. Wherever you decide, add an `<Outlet />` tag to the JSX. (Hint: Outlet will need to be imported from react-router-dom).

Setting up the routing

3. Edit App.jsx. Find your existing routing table.
4. Convert the route for Order into a nested route underneath the Orders route.
5. Run and test. You know you've got it right when you can click/tap an order and see both the list of orders and the order details on the same scene.

If you can, you're finished.