Adding Lazy Loading to the Bank App

In this lesson, we'll add lazy loading to our bank app to increase its performance.

WE'LL COVER THE FOLLOWING

- Lazy Loading the App Component

We can add some lazy loading to the bank application we saw in the last chapter.

Consider the `Root` component of the application:

```
const Root = () => (
  <UserProvider>
    <UserConsumer>
      {({ user, handleLogin }) =>
        user ? <App /> : <Login handleLogin={handleLogin} />
      }
    </UserConsumer>
  </UserProvider>
)
```

Lazy Loading the App Component

When a user isn't logged in, we display the login page. The `App` component is displayed only when the user is logged in.

We could lazy load the `App` component, right?

This is very easy. You use the dynamic import syntax with `React.lazy` and wrap the `App` component in a `Suspense` component.

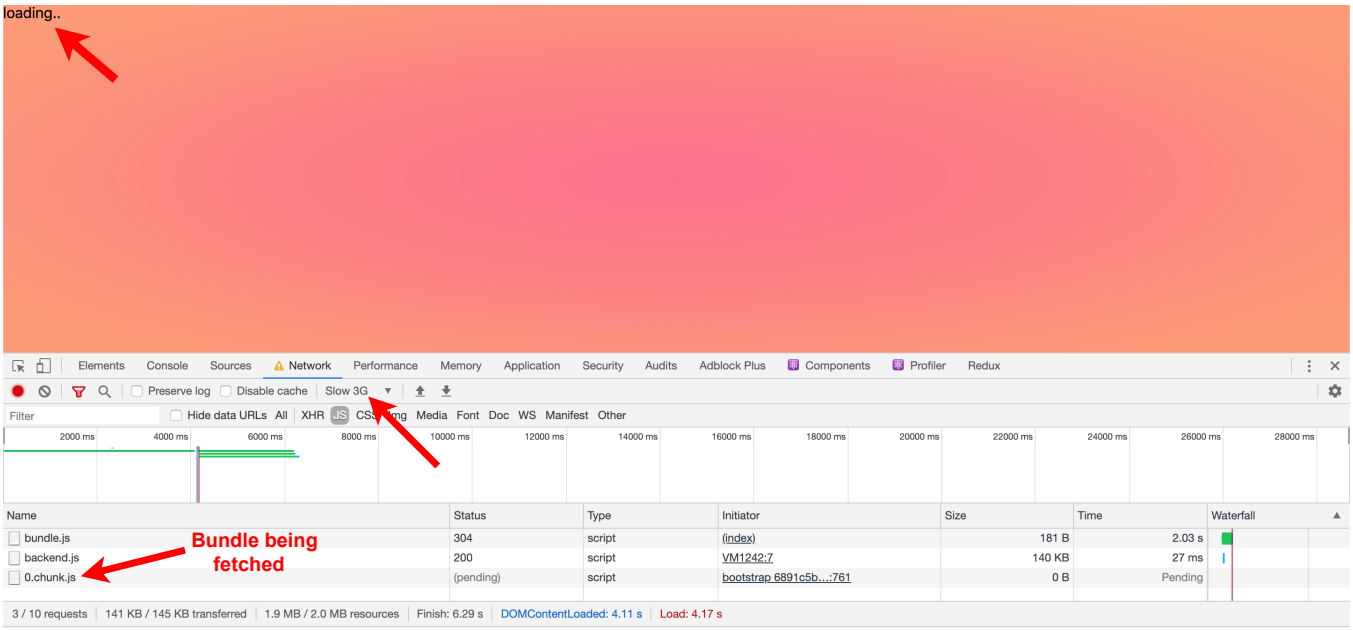
Here's how:

```
...
const App = React.lazy(() => import('./containers/App'))
const Root = () => (
```

```
  <Suspense fallback='loading...'>
     <App />

  </Suspense>
)
```

Now, if you throttle your network connection to simulate Slow 3G, you'll see the intermediate *"loading..."* text after logging in.



Let's look at the implementation of the bank app after using lazy loading.

```
export const USER = {
  name: 'June',
  totalAmount: 2500701
}
```

Let's move on to the conclusion of this chapter in the next lesson.