REACT ROUTING

Prof. Andrew Sheehan

Boston University/MET
Computer Science Dept.

REACT ROUTER

WHAT IS IT?

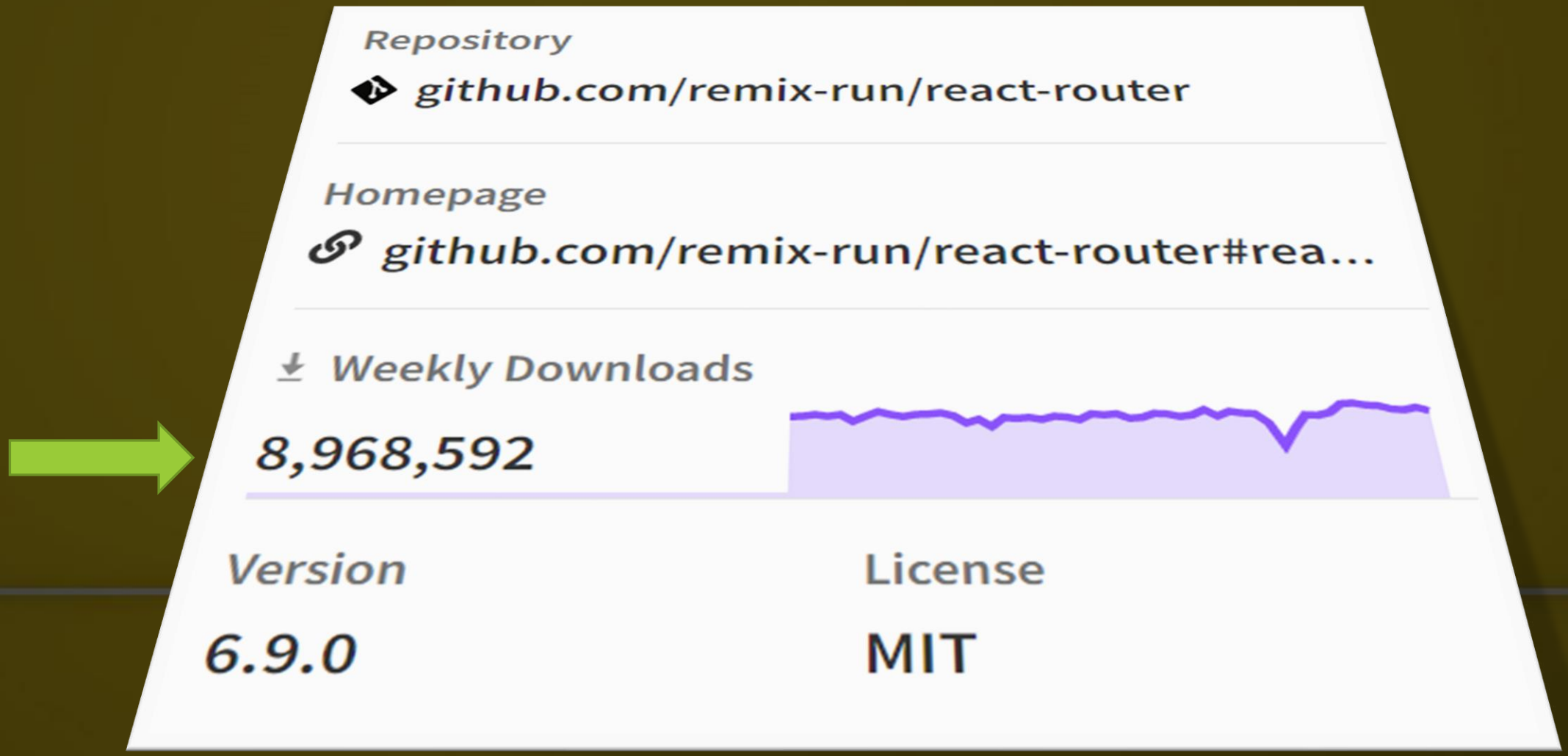
Allows you to configure client-side routing for react-based applications

Route allows you to map your app's location to different React components

SAID ANOTHER WAY:
WHAT IS REACT ROUTER??

It keeps the application in sync as your client navigates within your application (on the client).

NPMJS DETAILS



Last Updated: 25-Mar-2023 @9:11 PM EST

(1) Client makes request through an URL

(2) Server sends HTML, CSS and Javascript

TRADITIONAL CLIENT/SERVER RELATIONSHIP

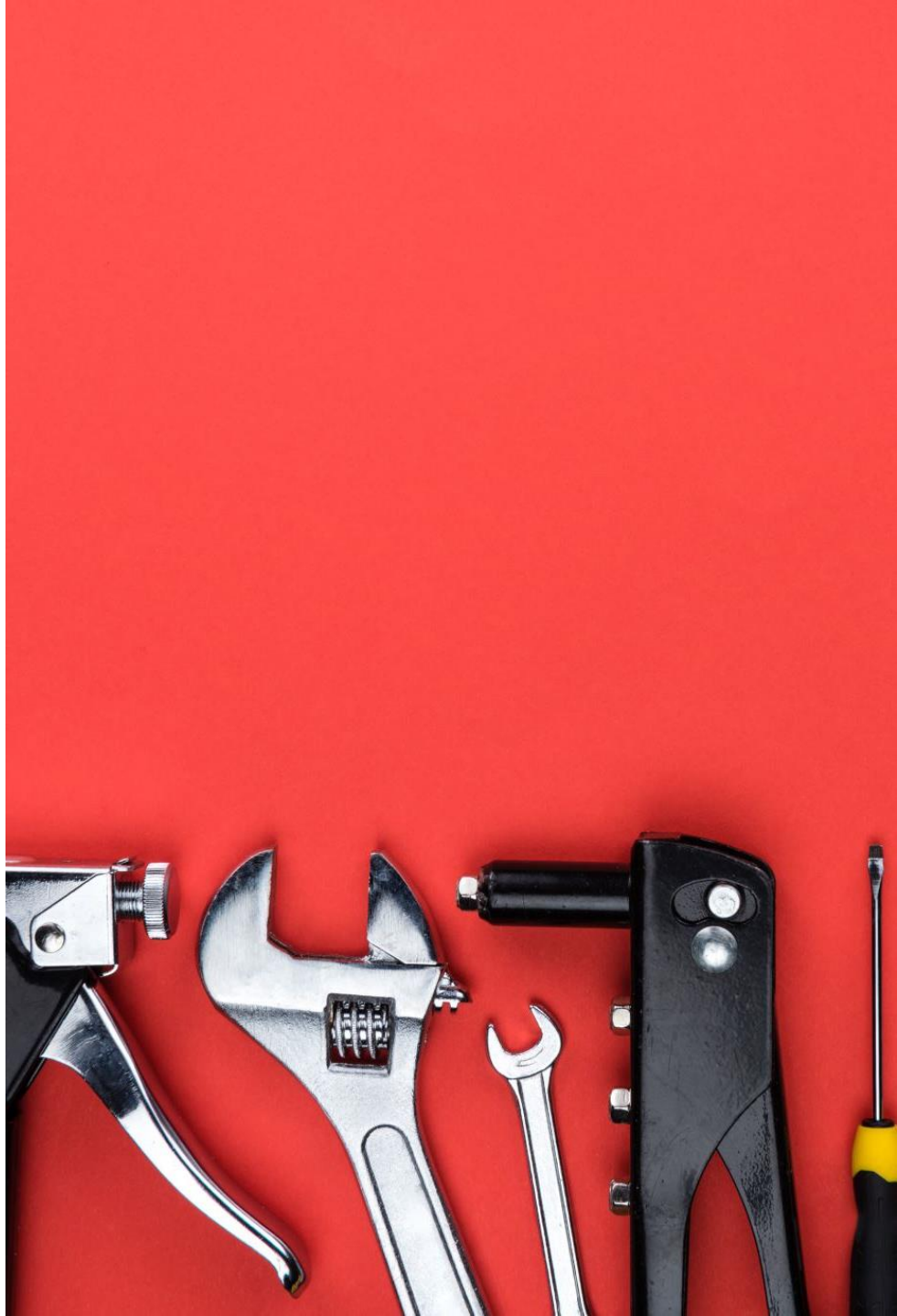
Better UX:
Faster rendering

Client side routing allows you to request a partial segment of HTML *(or data)*

The user does not need to leave the application or load another page

3-THINGS TODO

1. *Setup your router*
2. *Define your routes*
3. *Have app logic*



INTRO

VITE

i. A tool for building
ii. Optimized bundles
iii. A dev server

react-router-dom vs react-router-dom

`react-router` is a core package containing standard components and functionalities to implement routing.

`react-router-dom` is a module that you can use only in **browser-based** applications

CHOOSING A ROUTER

createBrowserRouter()

createMemoryRouter()

Useful in non-browser based applications; testing purposes

createHashRouter()

https://reactrouter.com/en/main/routers/create-hash-router

> ⚠ IMPORTANT
> Using hash URLs is not recommended.

BROWSER ROUTER

The recommended router for all React Router web projects

It uses the DOM History API to update the URL and manage the history stack

HISTORY API

developer.mozilla.org/en-US/docs/Web/API/History_API

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

const router = createBrowserRouter()
```

ABBREVIATED BROWSER ROUTER EXAMPLE

```jsx
import React from "react";
import ReactDOM from "react-dom/client";
import {
  createBrowserRouter,
  RouterProvider,
} from "react-router-dom";
import "./index.css";

const router = createBrowserRouter([
  {
    path: "/",
    element: <div>Hello world!</div>,
  },
]);

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <RouterProvider router={router} />
  </React.StrictMode>
);
```

EXAMPLE

WHAT IS A ROUTE?

Likely the most-important aspect of Routing

Routes are objects that are passed to the router creation function

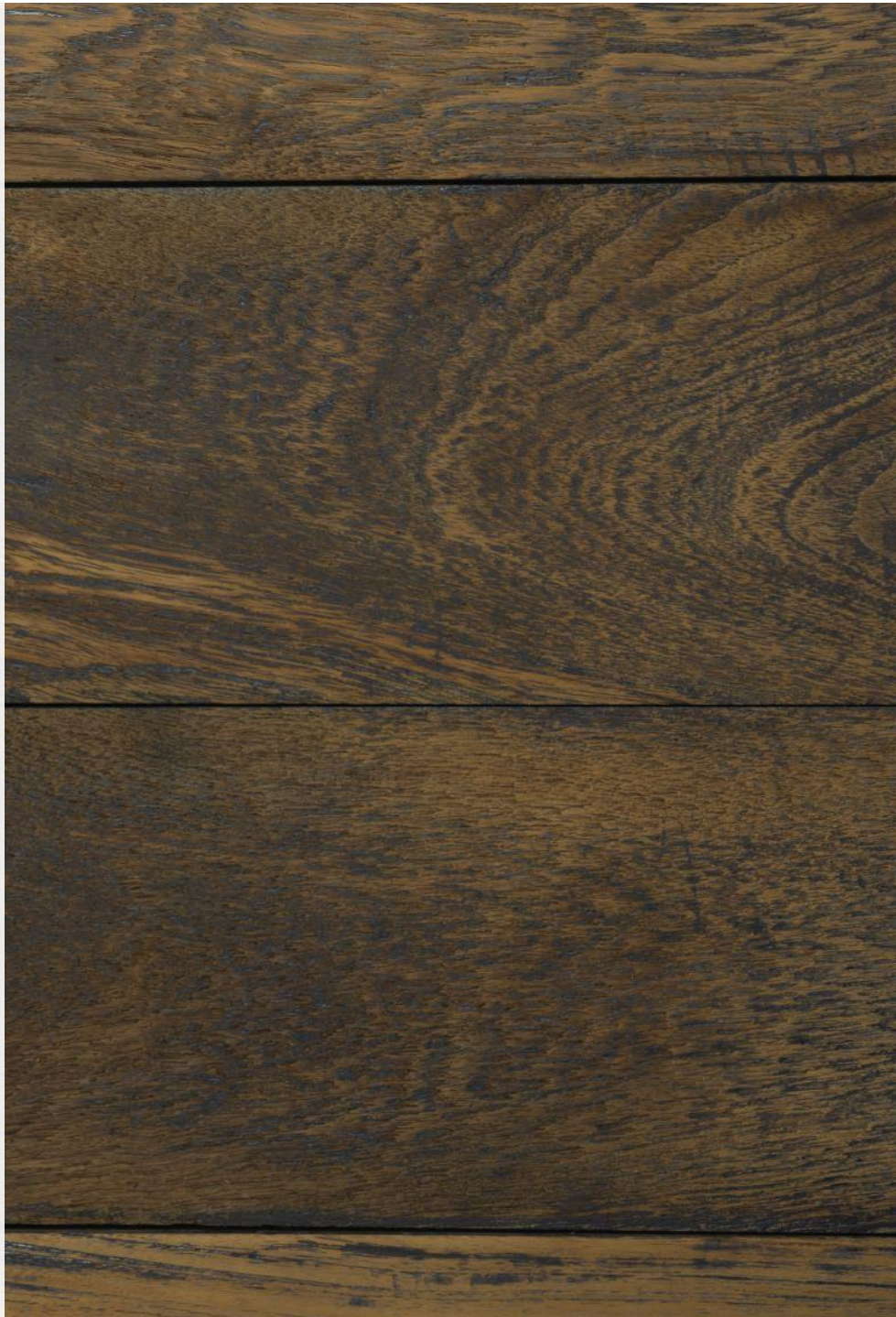
THE INDEX ROUTE

```
import Index from "./routes/index";

const router = createBrowserRouter([
  {
    path: "/",
    element: <Root />,
    errorElement: <ErrorPage />,
    loader: rootLoader,
    action: rootAction,
    children: [
      { index: true, element: <Index /> },
      /* existing routes */
    ],
  },
]);
```

CHILD ROUTES

```
const router = createBrowserRouter([
  {
    path: "/",
    element: <Root />,
    errorElement: <ErrorPage />,
    children: [
      {
        path: "contacts/:contactId",
        element: <Contact />,
      },
    ],
  },
]);
```

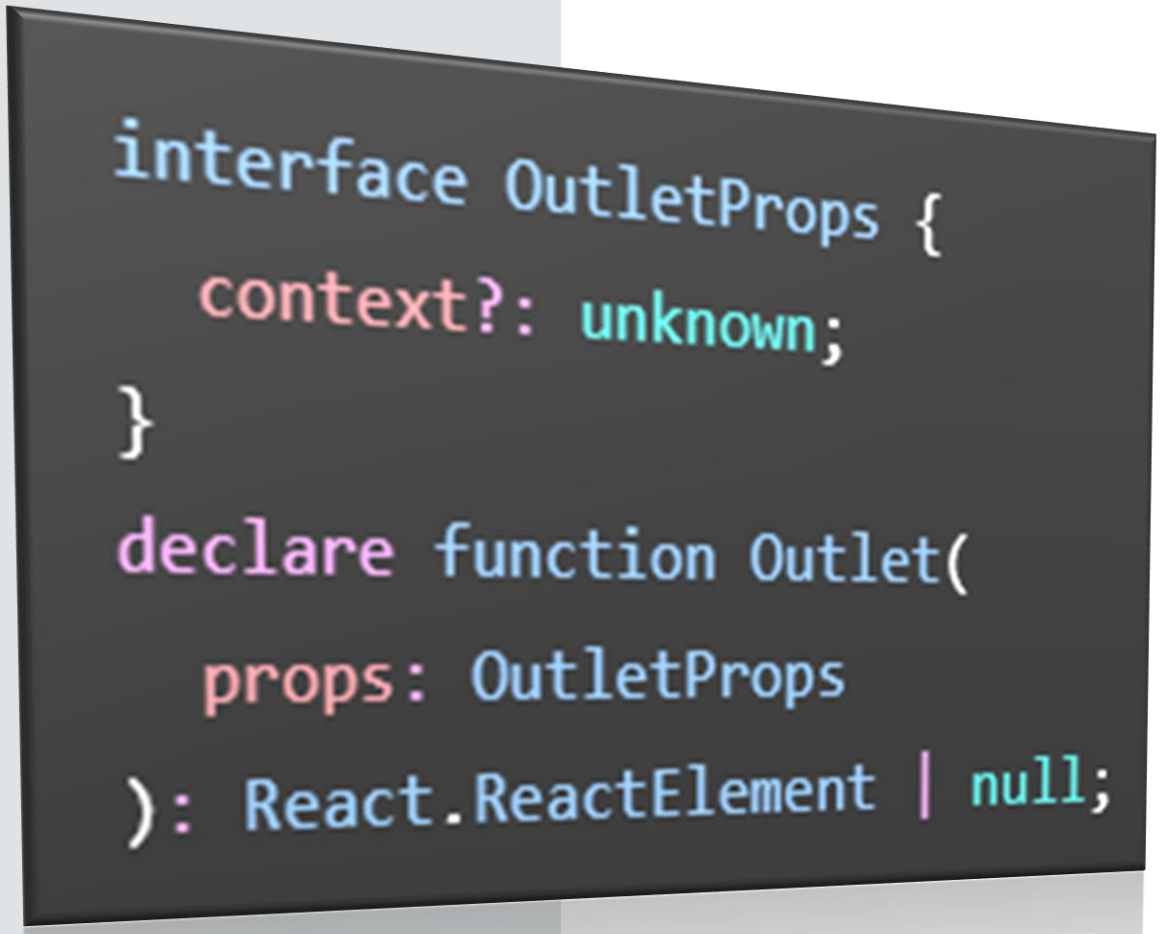


TYPE DEFINITION ROUTE

```ts
interface RouteObject {
  path?: string;
  index?: boolean;
  children?: React.ReactNode;
  caseSensitive?: boolean;
  id?: string;
  loader?: LoaderFunction;
  action?: ActionFunction;
  element?: React.ReactNode | null;
  errorElement?: React.ReactNode | null;
  handle?: RouteObject["handle"];
  shouldRevalidate?: ShouldRevalidateFunction;
}
```

reactrouter.com/en/main/route/route

<OUTLET/> SYNTAX



<Outlet/>'s are defined in parent elements. They are placeholders for child routes.

```
import {
  BrowserRouter,
  Routes,
  Route,
  Link,
  Outlet
} from 'react-router-dom';

function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="users" element={<Users />}>
          <Route path="/" element={<UsersIndex />} />
          <Route path=":id" element={<UserProfile />} />
          <Route path="me" element={<OwnUserProfile />} />
        </Route>
      </Routes>
    </BrowserRouter>
  );
}

function Users() {
  return (
    <div>
      <nav>
        <Link to="me">My Profile</Link>
      </nav>

      <Outlet />
    </div>
  );
}
```

The `<Outlet>` element is used as a placeholder. In this case an `<Outlet>` enables the `Users` component to render its child routes. Thus the `<Outlet>` element will render either a `<UserProfile>` or `<OwnUserProfile>` element depending on the current location.

<OUTLET/> SYNTAX

Tell the root route where we want it to render its child routes

```
const router = createBrowserRouter([
  {
    path: "/",
    element: <Root />,
    errorElement: <ErrorPage />,
    children: [
      {
        path: "contacts/:contactId",
        element: <Contact />,
      },
    ],
  },
]);
```

URL Params

/CONTACT/:CONTACT

INDICATES A DYNAMIC URL SEGMENT

A link allows the user to navigate to another view or page in your application

```
9           <nav>
10            <ul>
11              <li>
12                <Link to={`contacts/1`}>Your Name</Link>
13              </li>
14              <li>
15                <Link to={`contacts/2`}>Your Friend</Link>
16              </li>
17            </ul>
18          </nav>
```

<LINK>