Evolve Full Stack Developer

React Router

INCEPTIONU

Repo for Slides

- **Get them from GIT**

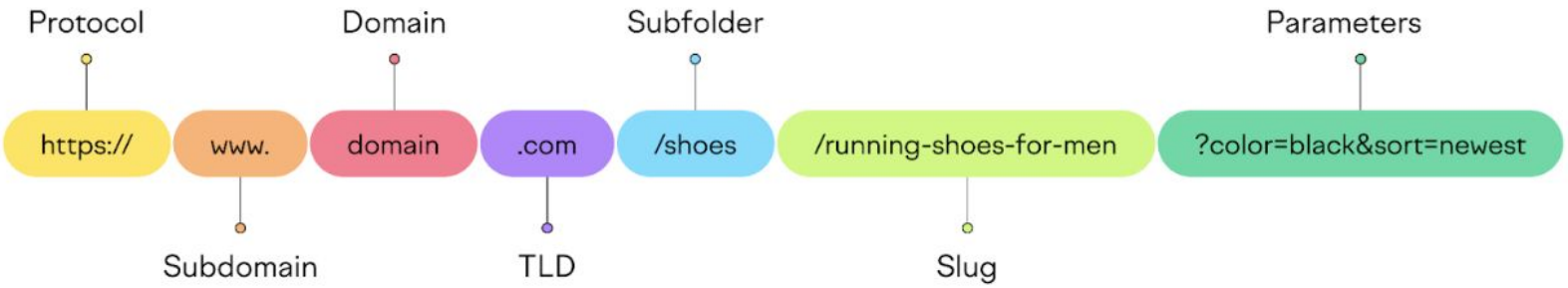
```
https://github.com/EvolveU-FSD/c11-FunWithUrls.git
```

INCEPTIONU

Big Picture Idea - What is React Router?

- What is a url and what are it's parts? Why are they important?
- Show how to use different parts of the URL in react projects
  - Subfolders
  - Error
  - Dynamic routes
  - Load another component
  - Ready Query Parameters
  - Change query parameters

Parts of a URL Structure



INCEPTIONU

Working with Subfolders

Subfolders are just a path to a different component. React-router-dom

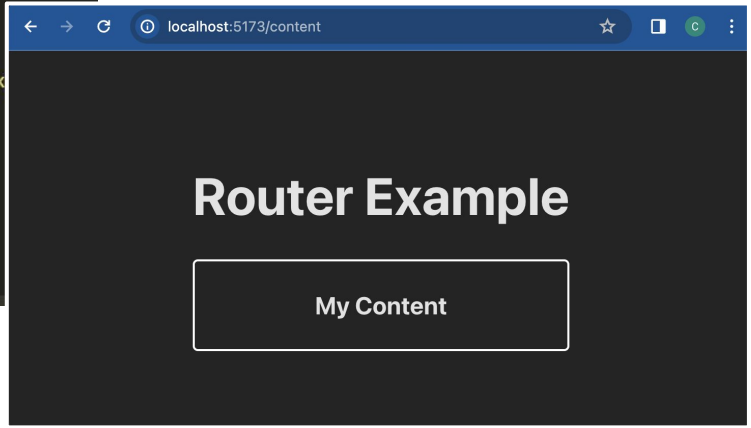
```jsx
import { createBrowserRouter, RouterProvider } from 'react-router-dom'

const router = createBrowserRouter ([
  { path: "/", element: <h2>Main Page</h2>},
  { path: "/content", element: <h2>My Content</h2>},
])

function App() {

  return (
    <>
      <h1>Router Example</h1>
      <div style={{border: "2px solid white", borderRadius: '5px', padding: '5px
        <RouterProvider router={router} />
      </div>
    </>
  )
}

export default App
```



npm create vite@latest 1-ReactRouter

INCEPTIONU

Adding some error handling

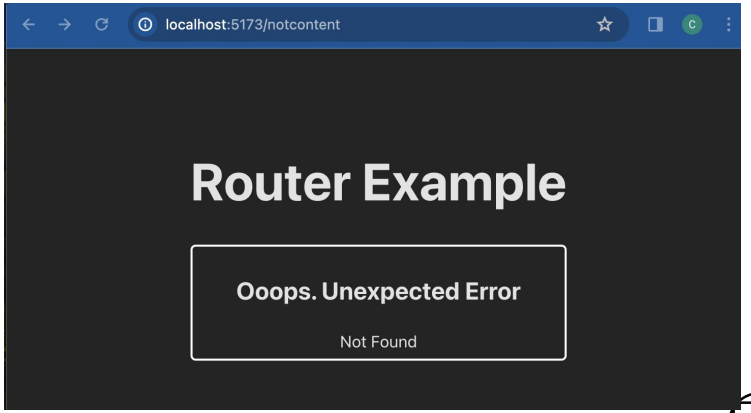
Now that we know we can control routes, let's leave a way to handle route errors.

```
import { createBrowserRouter, RouterProvider, useRouteError } from 'react-router-dom'

const router = createBrowserRouter ([
  { path: "/", element: <h2>Main Page</h2>, errorElement: <ErrorPage />},
  { path: "/content", element: <h2>My Content</h2>},
])
```

```
function ErrorPage(){
  const error = useRouteError();
  console.log(error)

  return <div>
    <h2>Ooops. Unexpected Error</h2>
    <div>{error.statusText || error.message}</div>
  </div>
}
```



INCEPTIONU

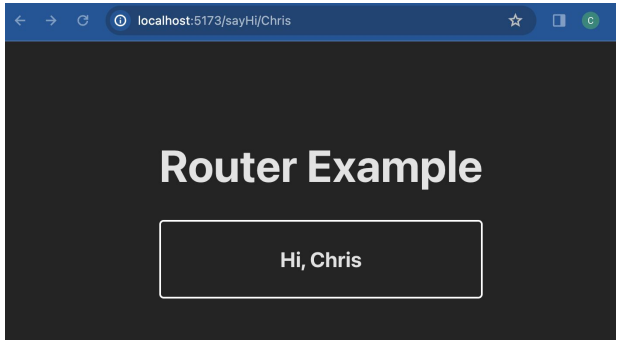
Dealing with slugs (dynamic routes)

```
import { createBrowserRouter, RouterProvider, useRouteError, useParams } from 'react-router-dom'

const router = createBrowserRouter ([
  { path: "/", element: <h2>Index</h2>, errorElement: <ErrorPage />},
  { path: "/foo", element: <h2>foo</h2>},
  { path: "/sayHi/:name", element: <SayHi />}
])
```

```
function SayHi(){
  const {name} = useParams();

  return <h2>Hi, {name}</h2>
}
```



INCEPTIONU

Now to pull query params

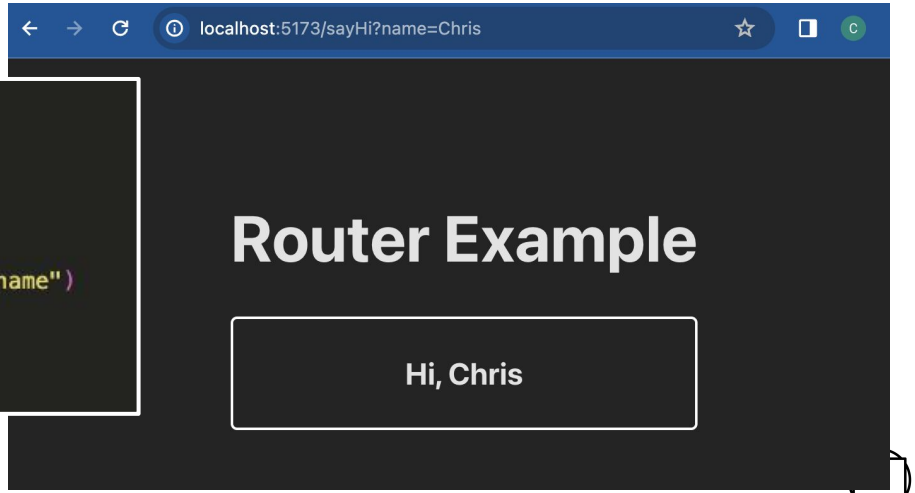
```
import { createBrowserRouter, RouterProvider, useRouteError, useParams, useLocation } from 'react-router-dom'

const router = createBrowserRouter ([
  { path: "/", element: <h2>Index</h2>, errorElement: <ErrorPage />},
  { path: "/foo", element: <h2>foo</h2>},
  { path: "/sayHi/:name", element: <SayHi />},
  { path: "/sayHi", element: <SayHi />}
])
```

```
function SayHi(){
  var {name} = useParams();
  var location = useLocation()

  if (name == null)
    name = new URLSearchParams(location.search).get("name")

  return <h2>Hi, {name}</h2>
}
```



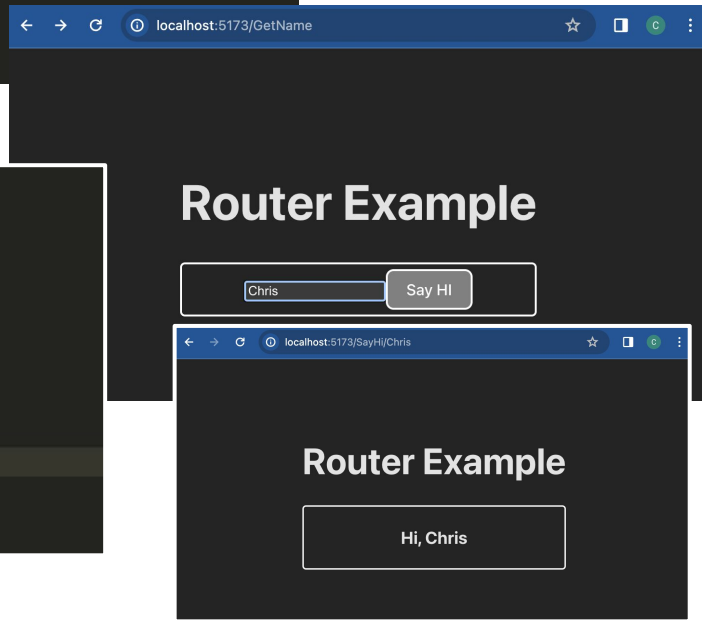
INCEPTIONU

Firestore

```
import { createBrowserRouter, RouterProvider, useRouteError, useParams, useLocation } from 'react-router-dom'

const router = createBrowserRouter ([
  { path: "/", element: <h2>Index</h2>, errorElement: <ErrorPage />},
  { path: "/foo", element: <h2>foo</h2>},
  { path: "/sayHi/:name", element: <SayHi />},
  { path: "/sayHi", element: <SayHi />},
  { path: "/GetName", element: <Redirect />}
])
```

```
function Redirect(){
  var [name, setName] = useState('')
  function redirectToSayHi(){
    window.location.href = `\\SayHi\\${name}`
  }
  return <div>
      <input type="text" value={name} onChange={(e)=>setName(e.target.value)} />
      <button onClick={redirectToSayHi}
        style={{backgroundColor: 'gray', border: "2px solid white"}}>
        Say HI</button>
  </div>
}
```



INCEPTIONU

Change Your URL in place

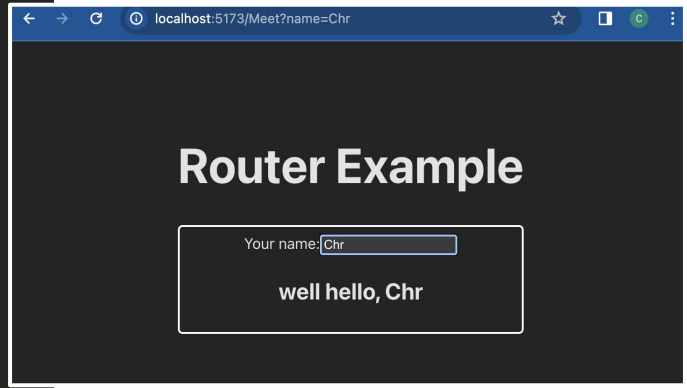
```
function ReactiveUrl(){
  var location = useLocation()
 var inboundName = new URLSearchParams(location.search).get("name")

  var [name, setName] = useState(inboundName);

  useEffect(()=>{
    const searchParams = new URLSearchParams(window.location.search);
    searchParams.set('name', name);
    window.history.replaceState(null, null, `?${searchParams.toString()}`)
  }, [name])

  return <div>
    Your name:
    <input type="text" value={name} onChange={e=>setName(e.target.value)} />

    <h2>well hello, {name}</h2>
  </div>
}
```



INCEPTIONU

Still have more questions?

INCEPTIONU