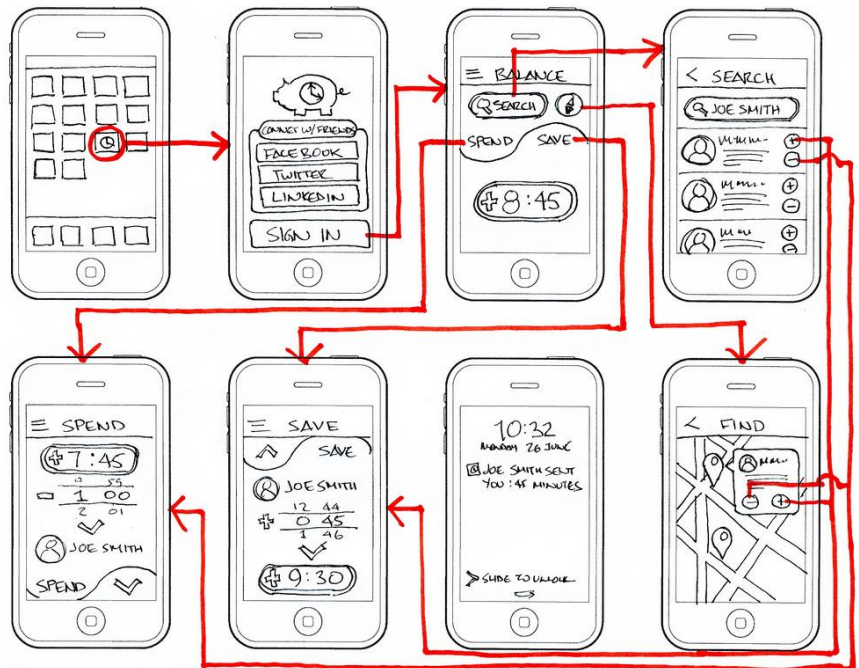


# React Router

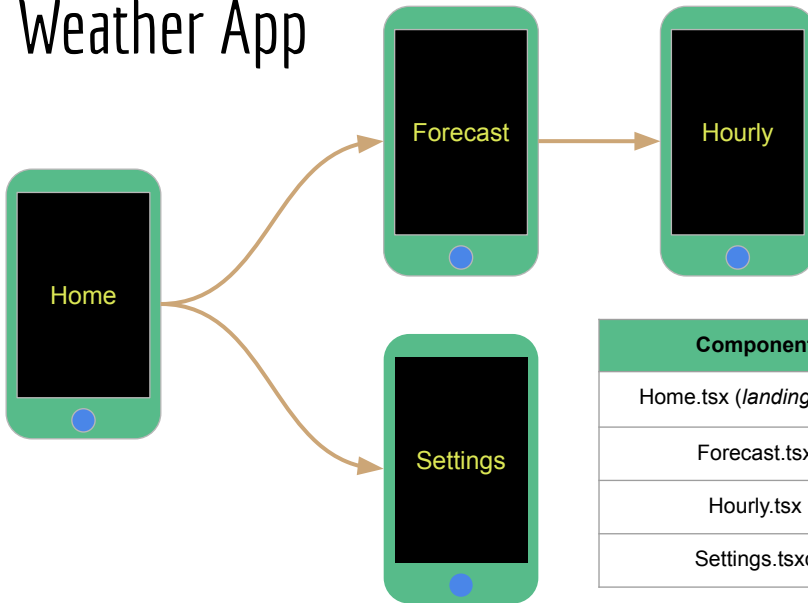
Navigate Among Multiple Views

## Routing

WebApp Navigation



# Weather App



Component	Path
Home.tsx ( <i>landing page</i> )	/
Forecast.tsx	/forecast
Hourly.tsx	/hourly
Settings.tsxc	/settings

3

# Installation & Setup

```
# Version 6.x  
yarn add react-router-dom@6
```

Component	Path
Home.tsx ( <i>landing page</i> )	/
Forecast.tsx	<b>/forecast</b>
Settings.tsx	<b>/settings</b>

[Stackblitz Playground](#)

```
main.tsx  
import {BrowserRouter} from "react-router-dom";  
import App from "./App"  
  
ReactDOM.render(  
  <BrowserRouter>  
    <App />  
  </BrowserRouter>,  
  document.getElementById("root"));
```

```
App.tsx  
import Home from "./views/Home";  
import Forecast from "./views/Forecast";  
import Settings from "./views/Settings";  
import {Routes, Route} from "react-router-dom";  
  
export default function App() {  
  return <>  
    <Routes>  
      <Route index element={ <Home/> }>  
      <Route path="/forecast" , element={ <Forecast /> }>  
      <Route path="/settings", element={ <Settings /> }>  
    </Routes>  
  </>;  
}
```

4

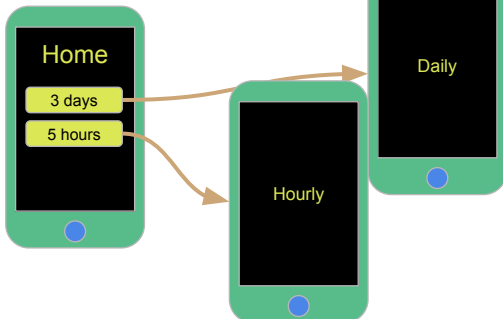
# React Hook #4: useNavigate()

## Navigate Programmatically

5

## Navigation Details

```
{ // routing table in main.ts
  path: '/daily',
  element: Daily
},
{
  path: "/hourly",
  element: Hourly
}
```



```
Home.vue
import {useNavigate} from "react-router-dom";

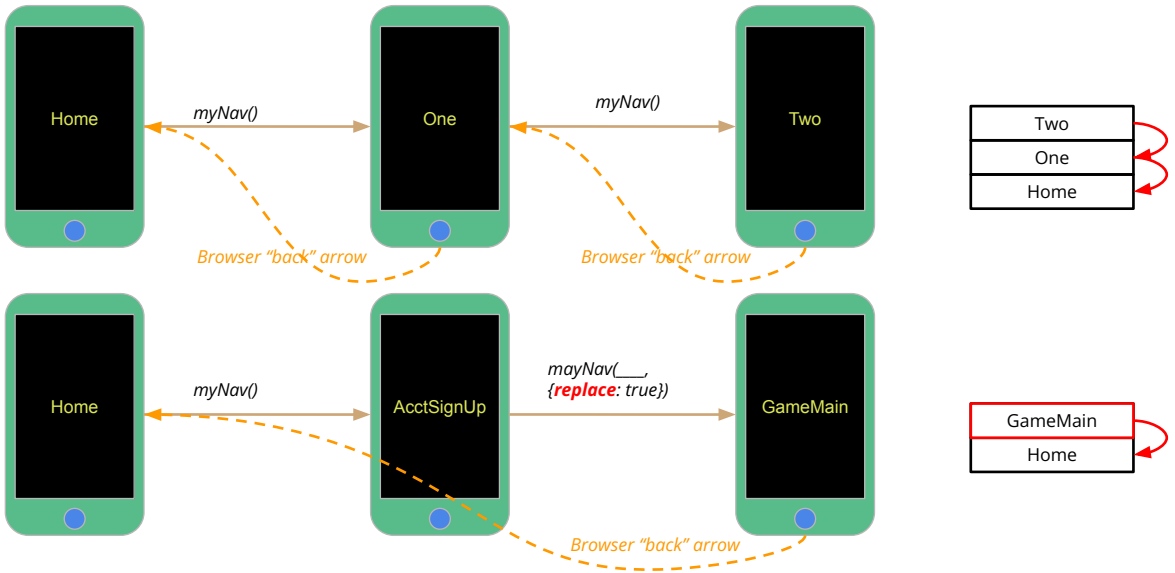
export default function Home() {
  const myNav = useNavigate();
  dailyWeather():void {
    myNav("/daily");
  }

  hourlyWeather(): void {
    myNav("/hourly");
  }

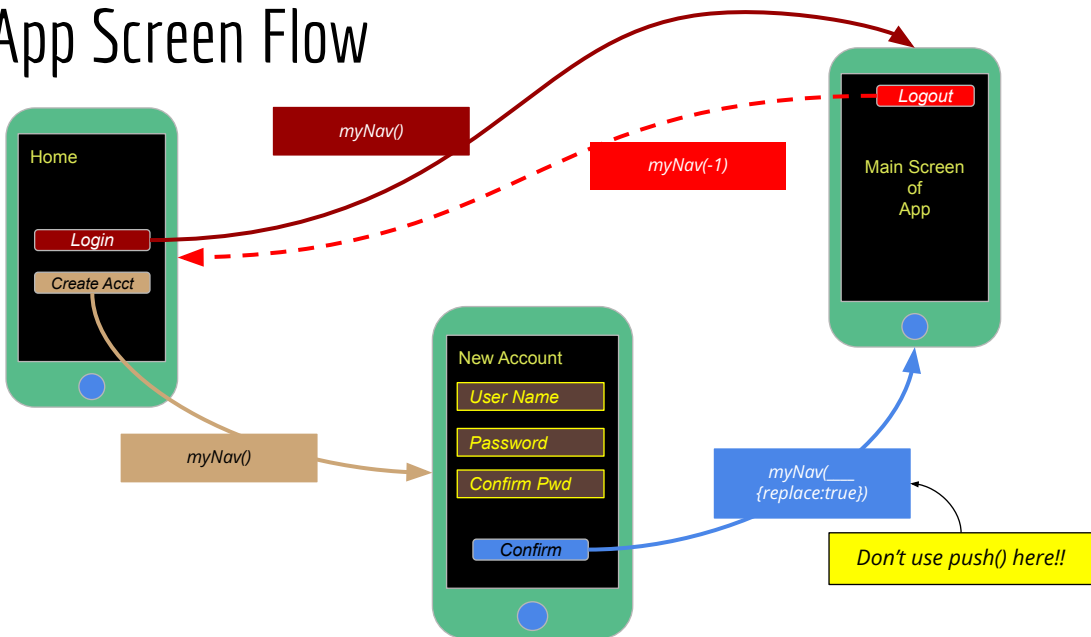
  return <>
    <button onClick={dailyWeather}>3 days</button>
    <button onClick={hourlyWeather}>5 hours</button>
  </>
}
```

6

# Browser History Stack: push() vs. replace()



# App Screen Flow



# Problem #2: Passing Data To Components

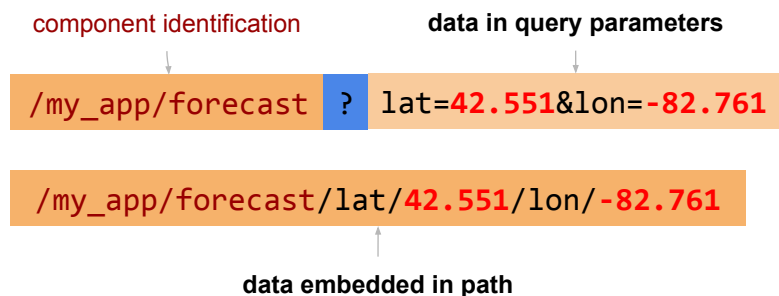
*Limited to "string-like" data*

9

## HTTP paths and query parameters

Passing data to a target component

- Via query parameters (useSearchParams)
- Embedded into path (useParams)



10

# React Hooks #5 and #6

## useParams() & useSearchParams()

12

## Passing Data to Components: useParams()

```
import Forecast from './components/Forecast';  
  
export default function App() {  
  return <Routes>  
    <Route path="/forecast/at/:zip/for/:duration"  
      element={ <Forecast /> }  
    />  
  </Routes>  
}
```

App.tsx

FromComponent.tsx (sender)

```
import {useNavigate} from "react-router-dom";  
  
export default function FromComponent() {  
  const nav = useNavigate();  
  
  doForecast() { // a button click handler?  
    nav("/forecast/at/49401/for/7d")  
  }  
}
```

Forecast.tsx (recipient)

```
// Option #1  
export default function ToComponent() {  
  const args = useParams();  
  
  // use: args.zip or args.duration  
}
```

Forecast.tsx (recipient)

```
// Option #2: object destructure  
export default function ToComponent() {  
  const {zip, duration} = useParams();  
  
  // Use: zip and duration  
}
```

13

# Passing Data to Components: useSearchParams()

```
import Forecast from './components/Forecast';
```

App.tsx

```
export default function App() {  
  return <Routes>  
    <Route path="/forecast/byzip"  
      element={ <Forecast /> }  
    />  
  </Routes>  
}
```

FromComponent.tsx (sender)

```
import {useNavigate} from "react-router-dom";  
  
export default function FromComponent() {  
  const nav = useNavigate();  
  
  doForecast() { // a button click handler?  
    nav("/forecast/byzip?zipCode=49401&duration=4");  
  }  
}
```

Forecast.tsx (recipient)

```
import {useSearchParams} from "react-router-dom"  
  
export default function ToComponent() {  
  const [q] = useSearchParams();  
  const zip = q.get("zipCode") ?? "00000"  
  const days = q.get("duration") ?? 0  
  
  return <>  
    <p>Incoming zip code is {zip}</p>  
  </a>  
}
```

14

Live Demo

useParams() & useSearchParams()

15