

React VI

Routing to other components

Agenda

- React Router

Learning Objectives

- You know the basics how to create big applications
- You know how to switch between components (pages)
- You know how to use links

React Router

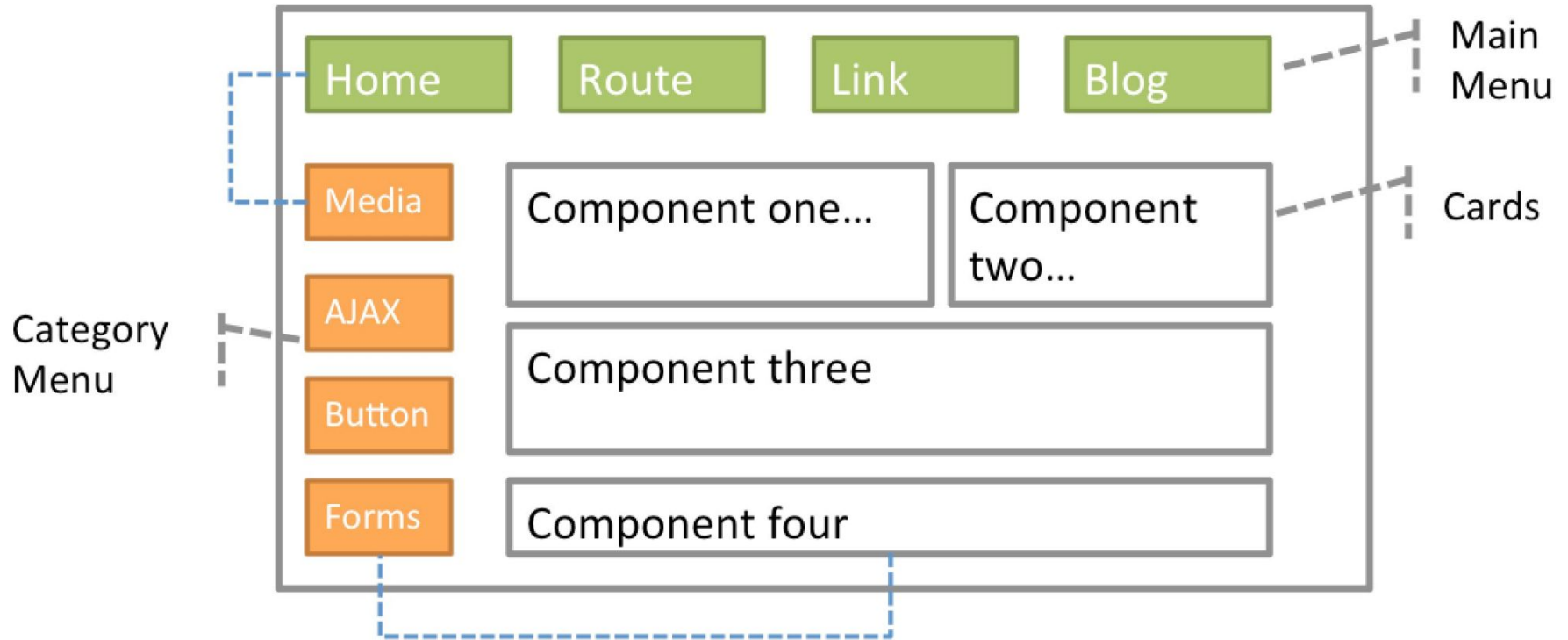
- One can build own application using only one root element
- An application with a root component and some sub components will work perfectly
- But, managing the UI view-changes can be tricky in this case, if the the application has more than one different views (dashboard, products, category).
- The react router comes to rescue for this situation.
- You can bind many components as Root-Components and switch between them over a navigation system, just like in normal html.
- React Router provides different mechanisms to support browser standards.

A react component <<TODO-List>>

A nested cmp <<TODO-Item>>

<<TODO-Item>>

The application has only one big view, all-in-all. It's perfectly manageable by react.



What about too many components in an app?



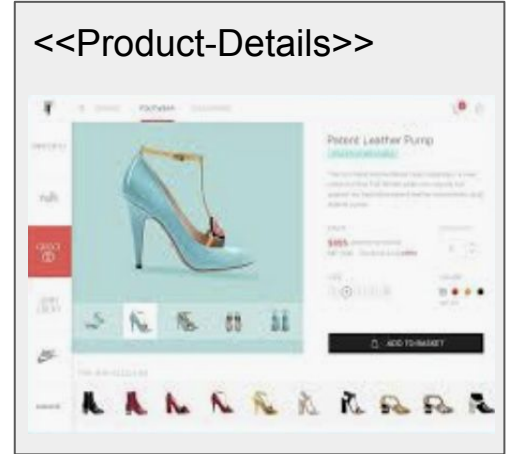
<<Dashboard>>



<<Products>>

	Name	Dimensions	Weight	Shipping Class	Updells
<input type="checkbox"/>	Chair large	30 x 30 x 20 cm	8 kg	Over 30kg	Ad. stool - Red
<input type="checkbox"/>	Estimote	1 x 2.2 x 6.2 cm	5.7 kg	Small Item	-
<input type="checkbox"/>	Washbasin single drawer	60 x 38 x 15 cm	8 kg	Bulky	Series Plastic Side Chair
<input type="checkbox"/>	Chair with case	30 x 63 x 88 cm	15 kg	Bulky	Series Magn.
<input type="checkbox"/>	Classic wicker chair	23 x 33 x 67 cm	6 kg	Discount	Magazine chair
<input type="checkbox"/>	Magazine chair	23 x 33 x 63 cm	30 kg	Over 30kg	-

<<Product-Details>>



How to switch between this react components (big!) and back again?



React Router

<<Dashboard>>

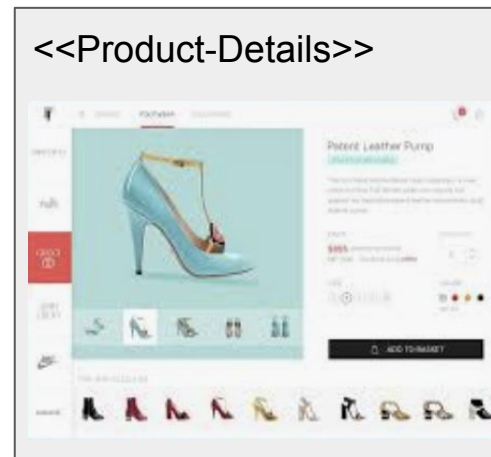


<<Products>>

The products list UI shows a table with columns: Name, Dimensions, Weight, Shipping Class, and Upsets. The table contains five rows of product data.

	Name	Dimensions	Weight	Shipping Class	Upsets
<input type="checkbox"/>	Chair large	30 x 30 x 20 cm	8 kg	Over 2kg	Ad. stool - Red
<input type="checkbox"/>	Estimote	1 x 2.2 x 6.2 cm	5.7 kg	Small Item	-
<input type="checkbox"/>	Wooden single drawer	60 x 38 x 15 cm	8 kg	Bulky	Series Plastic Side Chair
<input type="checkbox"/>	Chair with case	30 x 63 x 88 cm	15 kg	Bulky	Series Magic
<input type="checkbox"/>	Classic wooden chair	23 x 33 x 87 cm	8 kg	Discount	Magazine chair
<input type="checkbox"/>	Magazine chair	23 x 33 x 87 cm	30 kg	Over 30kg	-

<<Product-Details>>



React Router helps to switch these components and it keeps tracks of these ui changes.

Everything in a nutshell

```
<BrowserRouter>
  <div>
    <h1>React Router Simple Starter</h1>
    <ul className="header">
      <li><Link exact to="/">Home</Link> </li>
      <li><Link to="/blog">Blog</Link></li>
      <li><Link to="/contact">Contact</Link></li>
    </ul>
    <div className="content">
      <Route path="/" component={Home} />
      <Route path="/blog" component={Blog} />
      <Route path="/contact">
        <Contact/>
      </Route>
    </div>
  </div>
</BrowserRouter>
```

Does not work for version 6!

What happens if a link is clicked

- The application has already wrapped react router components
- It knows how to manage the clicks
- If one link is clicked, it basically gets the control over (hash changed or state changed)
- The given url in the link defines which route to select
- React router uses this info to select the related route
- If a route is selected, then the component in the route is displayed in the app

Basic Example

**[https://stackblitz.com/github/remix-run/react-router/tree/main/
examples/basic?file=src/App.tsx](https://stackblitz.com/github/remix-run/react-router/tree/main/examples/basic?file=src/App.tsx)**

How to install

```
npm install react-router-dom
```

It includes react-router core module!

src/main.jsx

```
1  import { render } from "react-dom";
+  import { BrowserRouter } from "react-router-dom";
3  import App from "./App";
4
5  const rootElement = document.getElementById("root");
6  render(
+    <BrowserRouter>
+    <App />
+    </BrowserRouter>,
10   rootElement
11 );
```

Setup

Main Objects

- **BrowserRouter**
 - Wraps the browser functionality
- **Route**
 - Define which components when to render
- **Link (NavLink)**
 - Gives the functionality to use native links in react
- **Routes**
 - A container holds routes together and matches exactly one route (if any possible)

Router Types in react router

- Browser Router

- Better for dynamic pages from a server
- It uses the history API under the hood
- `// <BrowserRouter>`
- `http://example.com/about`

- Hash Router

- Better for static pages
- It uses the hash functionality in the URL
- `// <HashRouter>`
- `http://example.com/#/about`

Path Match

- If the searched url fragment matches with the path in router, then component in the router is displayed
- Let's say, you have entered the address "http://localhost:3000/hi"

- `<Route path="/hi" element={Home} />`

- `<Route path="/coders" element={About} />`

- If you call, "<http://localhost:3000/hi/coders>"

- `<Route path="/hi" element={Home} />`

- `<Route path="/coders" element={About} />`

- `<Route path="/hi/coders" element={Details} />`

Using Routes

- Matching paths are all displayed above.
- If you want to show only one (the first one from top), the you use the container "Routes"
- <https://webshop.ch/products/28>

```
<Routes>
  <Route path="/" element={Home} >
    <Route path="/category" element={Category} ></Route>
    <Route path="/products" element={Products} >
      <Route path="/:id">
        <p>This text will render for any route other than those defined above</p>
      </Route>
    </Route>
  </Route>
</Routes>
```

src/main.jsx

```
1  <Routes>
2    <Route path="/" element={<App />}>
3      <Route path="expenses" element={<Expenses />} />
+    <Route path="invoices" element={<Invoices />}>
+      <Route path=":invoiceId" element={<Invoice />} />
+    </Route>
7    <Route
8      path="*"
9      element={
10        <main style={{ padding: "1rem" }}>
11          <p>There's nothing here!</p>
12        </main>
13      }
14    />
15  </Route>
16 </Routes>
```

No Match Route

Try it!

```
import React from "react";
import { Link, Route, Switch } from "react-router-dom";
import Category from "../Category";
import Products from "../Products";

const Home = () => (
  <div>
    <h2>Home</h2>
  </div>
);

export default function App() {
  return (
    <div>
      <nav className="navbar navbar-light">
        <ul className="nav navbar-nav">
          <li>
            <Link to="/">Home</Link>
          </li>
          <li>
            <Link to="/category">Category</Link>
          </li>
          <li>
            <Link to="/products">Products</Link>
          </li>
        </ul>
      </nav>

      <Switch>
        <Route exact path="/"><Home /></Route>
        <Route path="/category"><Category /></Route>
        <Route path="/products"><Products /></Route>
      </Switch>
    </div>
  );
}
```

Questions?