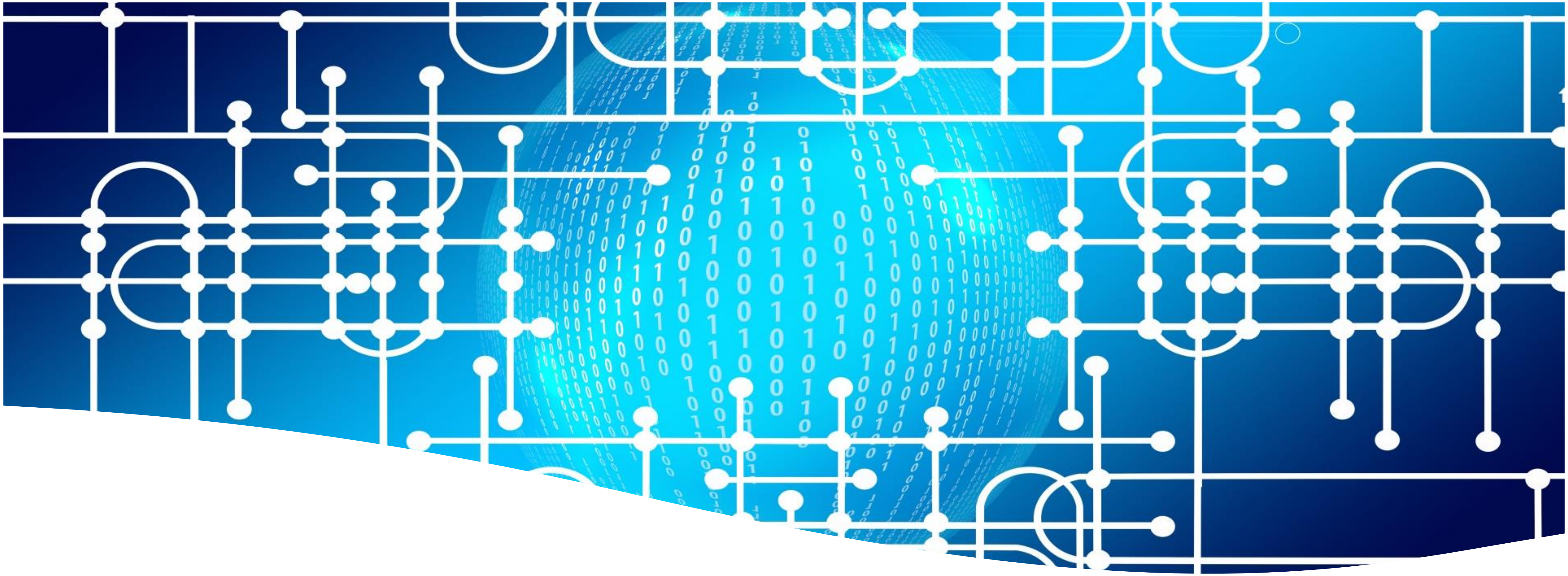


Full-stack Application Development

SPA Routing with React Router v6

Where to Find The Code and Materials?

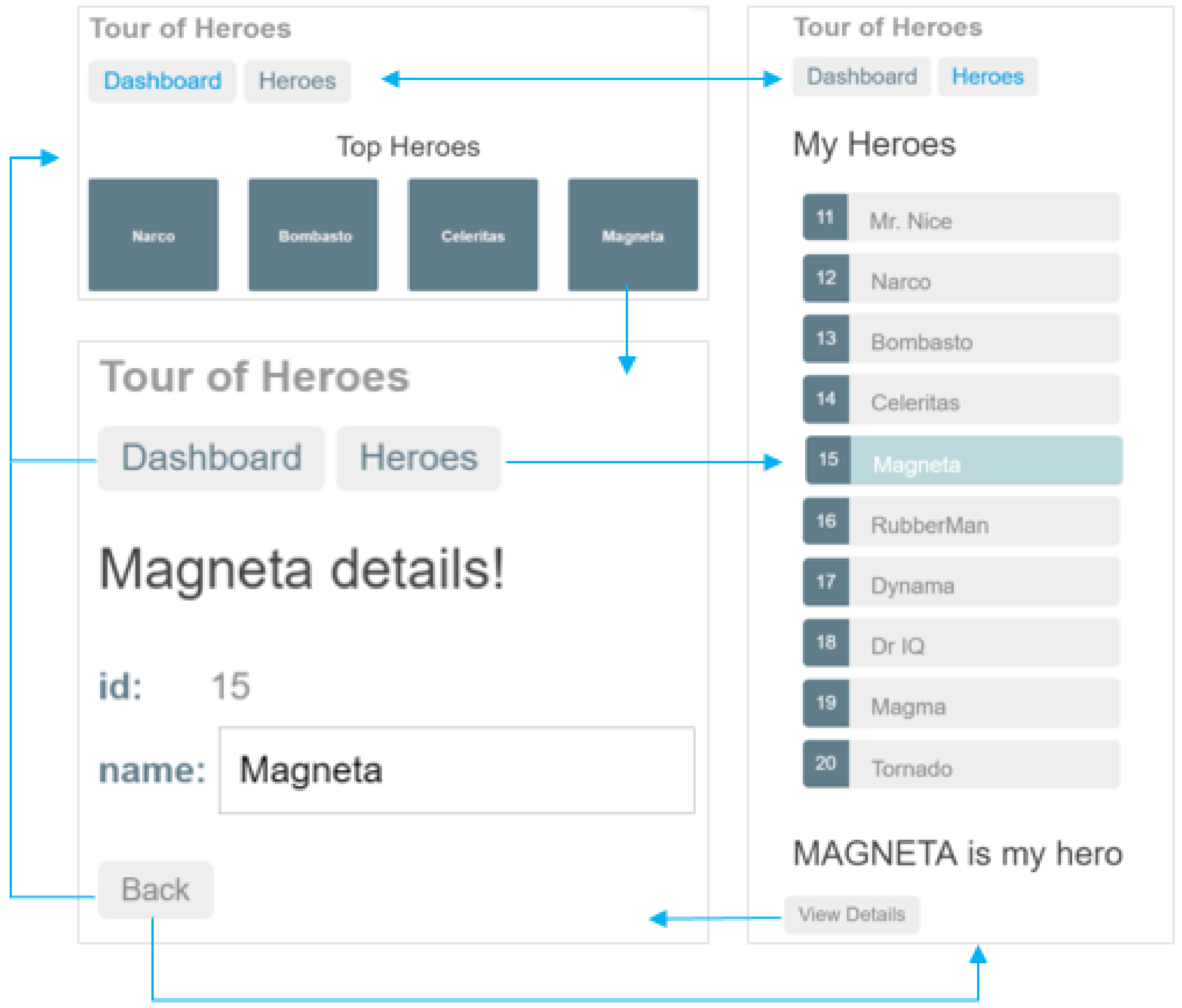
https://github.com/iproduct/fullstack-typescript-react

Agenda

Contemporary Web Applications

- Provide better **User Experience (UX)** by:
  - more interactive
  - loading and reacting faster in response (or even anticipation) of user's moves
  - able to work offline
  - supporting multiple devices and screen resolutions (responsive design)
  - are following design metaphors consistently (e.g. **Google Material Design - MD**)
  - looking more like desktop application than static web page

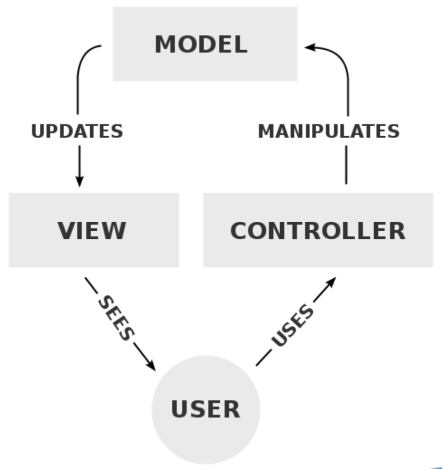
Single Page Applications (SPA)

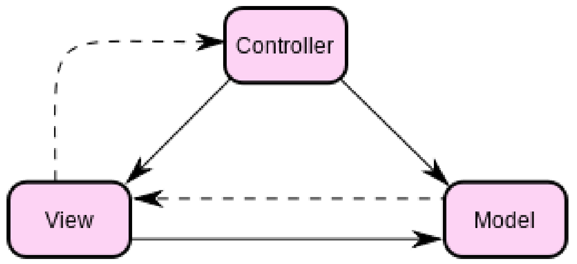


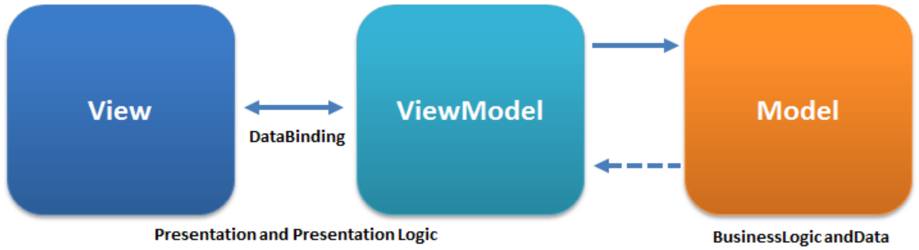
Source: Angular 2 Tutorial: Routing https://angular.io/docs/ts/latest/tutorial/toh-pt5.html
License: CC BY 4.0.

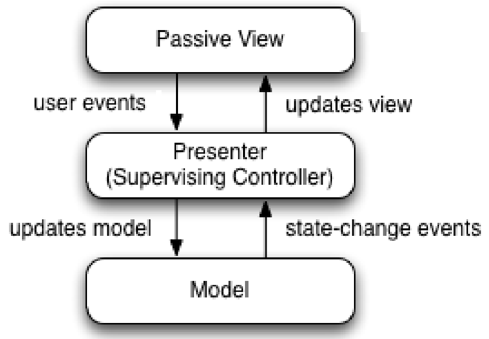
MVC Comes in Different Flavors

- MVC
- MVVM
- MVP









Sources: https://en.wikipedia.org/wiki/Model_View_ViewModel#/media/File:MVVMPattern.png,
https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93presenter#/media/File:Model_View_Presenter_GUI_Design_Pattern.png

Why SPA?

- Page does not flicker – seamless (or even animated) transitions
- Less data transferred – responses are cached
- Only raw data, not markup
- Features can be loaded on demand (lazy) or in background
- Most page processing happens on the client offloading the server: REST data services + snapshops for crawlers (SEO)
- Code reuse – REST endopints are general purpose
- Supporting multiple platforms (Web, iOS, Android) → React Native

Developing Sinagle Page Apps (SPA) in 3 steps

1)Setting up a build system – *npm, webpack, gulp* are common choices, *babel, typescript, JSX/TSX, CSS preprocessors (SASS, SCSS, LESS), jasmine, karma, protractor, live servers* ...

2)Designing front-end architecture components – *views & layouts* + *view models* (presentation data models) + *presentation logic* (event handling, messaging) + *routing paths* (essential for SPA)

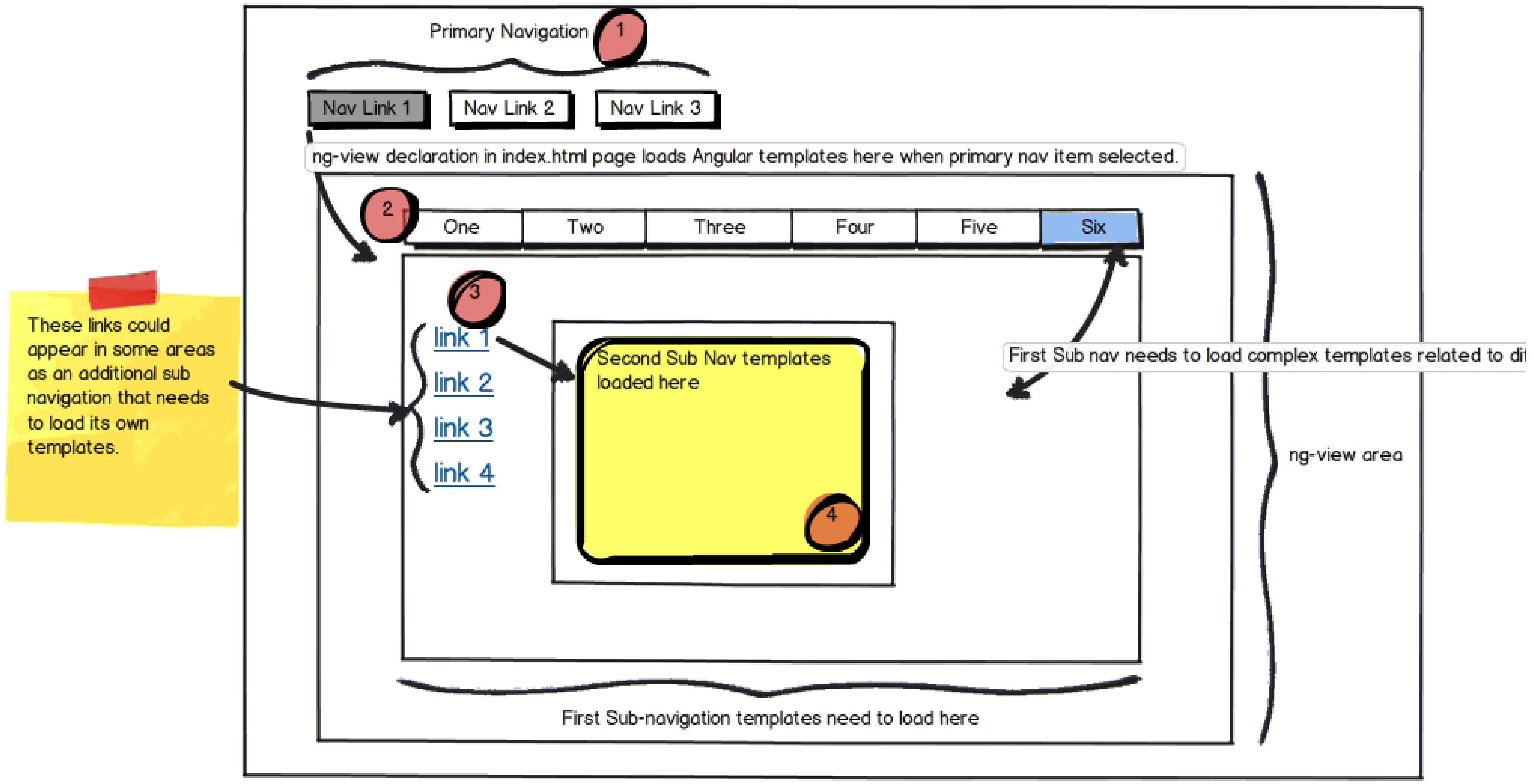
3)Better to use component model to boost productivity and maintainability.

4)End-to-end application design – front-end: wireframes → views,

5)data entities & data streams → service API and models design,

6)sitemap → router config

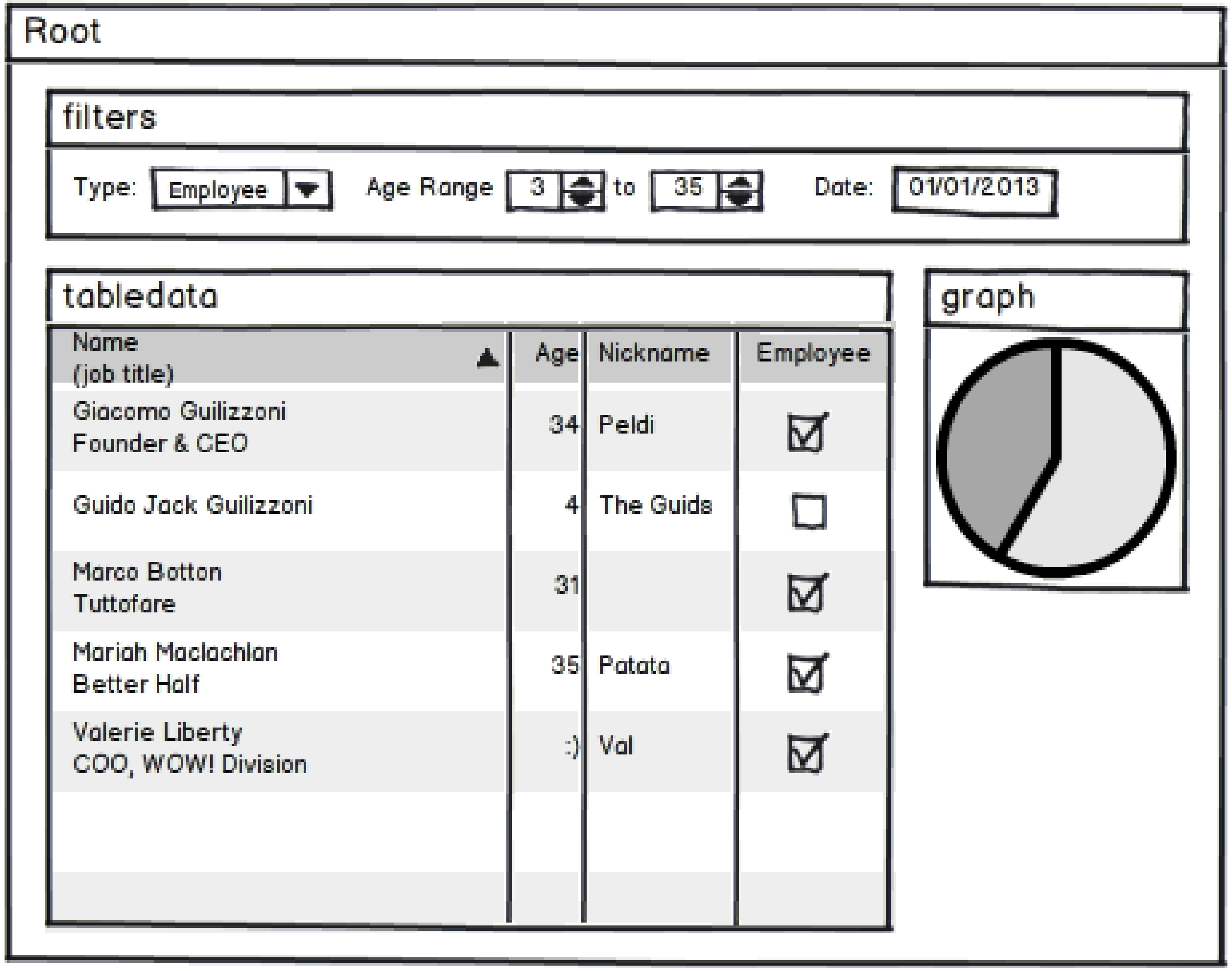
Hierarchical Routing



Primary Navigation
1
Nav Link 1
Nav Link 2
Nav Link 3
ng-view declaration in index.html page loads Angular templates here when primary nav item selected.
2
One
Two
Three
Four
Five
Six
3
link 1
link 2
link 3
link 4
These links could appear in some areas as an additional sub navigation that needs to load its own templates.
Second Sub Nav templates loaded here
4
First Sub nav needs to load complex templates related to di
ng-view area
First Sub-navigation templates need to load here

Source: http://stackoverflow.com/questions/12863663/complex-nesting-of-partials-and-templates
Author: PhillipKregg

SPA with Multiple Router Outlets



Name (job title) ▲	Age	Nickname	Employee
Giacomo Guilizzoni Founder & CEO	34	Peldi	☑
Guido Jack Guilizzoni	4	The Guids	☐
Marco Botton Tuttofare	31		☑
Mariah Maclachlan Better Half	35	Patata	☑
Valerie Liberty COO, WOW! Division	:)	Val	☑

Getting Started with React Router v6

- Create new project using *create-react-app*:

```
npx create-react-app demo-routing-app --template typescript
cd demo-routing-app
```

- Install *react-router-dom*:

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

- Implement routing in *src/App.js*

Basic Routing using React Router v6

```
import React from "react";
import { createRoot } from "react-dom/client";
import {
    createBrowserRouter,
    RouterProvider,
    Route,
    Link,
} from "react-router-dom";

const router = createBrowserRouter([
    {
        path: "/",
        element: (
            <div>
                <h1>Hello World</h1>
                <Link to="about">About Us</Link>
            </div>
        ),
    },
    {
        path: "about",
        element: <div>About</div>,
    },
]);

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

Basic Routing using React Router v6 - I

```
import * as React from "react";
import { Routes, Route, Outlet, Link } from "react-router-dom";
export default function App() {
  return (
    <div>
      {/* Routes nest inside one another. Nested route paths build upon
            parent route paths, and nested route elements render inside
            parent route elements. See the note about <Outlet> below. */}
      <Routes>
        <Route path="/" element={<Layout />}>
          <Route index element={<Home />} />
          <Route path="about" element={<About />} />
          <Route path="dashboard" element={<Dashboard />} />

          {/* Using path="*"" means "match anything", so this route
                acts like a catch-all for URLs that we don't have explicit
                routes for. */}
          <Route path="*" element={<NoMatch />} />
        </Route>
      </Routes>
    </div>
  );
}
```

Basic Routing using React Router v6 - II

```
function Layout() {
  return (
    <div>
      {/* A "layout route" is a good place to put markup you want to
          share across all the pages on your site, like navigation. */}
      <nav>
        <ul>
          <li>
            <Link to="/">Home</Link>
          </li>
          <li>
            <Link to="/about">About</Link>
          </li>
          <li>
            <Link to="/dashboard">Dashboard</Link>
          </li>
          <li>
            <Link to="/nothing-here">Nothing Here</Link>
          </li>
        </ul>
      </nav>
      <hr />
```

Basic Routing using React Router v6 - III

```
      {/* An <Outlet> renders whatever child route is currently active,
          so you can think about this <Outlet> as a placeholder for
          the child routes we defined above. */}
      <Outlet />
    </div>
  );
}

function Home() {
  return (
    <div>
      <h2>Home</h2>
    </div>
  );
}

function About() {
  return (
    <div>
      <h2>About</h2>
    </div>
  );
}
```

Basic Routing using React Router v6 - IV

```
function Dashboard() {
  return (
    <div>
      <h2>Dashboard</h2>
    </div>
  );
}

function NoMatch() {
  return (
    <div>
      <h2>Nothing to see here!</h2>
      <p>
        <Link to="/">Go to the home page</Link>
      </p>
    </div>
  );
}
```

Login Demo with Redirection

- There are 3 pages:
  - **public page** (demonstrating the public part of a web site)
  - **protected page** (demonstrating the private part of web site)
  - **login page**
- In order to see the protected page, you must login first. Upon login success, you will be redirected automatically to the required protected page.
- If you click the back button, would you expect to go back to the login page? No! You're already logged in. Going back, you should see the page you visited *before* logging in - the public page.

Thank’s for Your Attention!

Trayan Iliev

IPT – Intellectual Products & Technologies

http://iproduct.org/

http://robolearn.org/

https://github.com/iproduct

https://twitter.com/trayaniliev

https://www.facebook.com/IPT.EACAD