npm i [react-router-dom@5.3.4](mailto:react-router-dom@5.3.4)

Обернуть Routes в тег <BrowserRouter>.

Если не включить exact то будут выведены все подходящие компоненты чей path, удовлетворяет подстроки нынешнего url. (Например в website/about будет выведен Home и About. Чтобы этого не было, используем атрибут exact.

Чтобы вывести только один из компонентов, а не все которые удовлетворяют url, используем тег <Switch>.

      <BrowserRouter>

        <Switch>

          <Route exact path="/" component={Home} />

          <Route path="/about" component={About} />

          <Route path="/profile" component={Profile} />

        </Switch>

      </BrowserRouter>

Добавить тег <Link>, чтобы создать тег привязки и разрешить пользователю переключать страницы.

        <Link to="/">Home</Link>

We can specify basename for our website.

      <BrowserRouter basename="/tutorial">

Мы можем указать базовое имя для нашего сайта. В этом случае все наши компоненты будут начинаться с “tutorial/…”

Чтобы создать route, который всегда будет отображаться, если URL-адрес не соответствует ни одному из предоставленных маршрутов, не нужно указывать атрибут пути.

          <Route component={NotFound} /> // without path=“”

Когда у нас есть список компонентов, мы хотим отобразить их по ID.

          <Route path="/post/:id" component={Post} />

Каждый компонент, созданный с помощью Route, принимает свойство {match}. Мы можем взять :id из этого

const Post = ({ match }) => {

или мы можем использовать UseParams hook.

  const { id } = useParams();

чтобы взять переменные из URL-адреса, мы можем использовать хук useLocation.

<http://localhost:3000/post/first-post?name=prod-name>

  const { search } = useLocation(); // search="?name=prod-name"

Чтобы получить имя переменной из useLocation

const query = new URLSearchParams(useLocation().search);

  const name = query.get("name");

Для перенаправления на другую страницу мы можем использовать <Redirect>

          <Route path="/profile">

            {login ? <Profile /> : <Redirect to="/" />}

          </Route>

Если логин в состоянии True, профиль будет доступен <Profile />, в противном случае перенаправление на “/”

Или мы можем использовать хук useHistory.

          <Route path="/profile">

            <Profile login={login} />

          </Route>

const Profile = ({ login }) => {

  const history = useHistory();

  useEffect(() => {

    if (!login) history.push("/");

  }, []);

Если мы хотим добавить часть URL-адреса к текущему URL-адресу, мы можем использовать useRouteMatch()

    const { path, url } = useRouteMatch(); // profile - is current url

      <Switch>

        <Route path="/viewprofile" />

        <Route path="/editprofile" />

      </Switch>