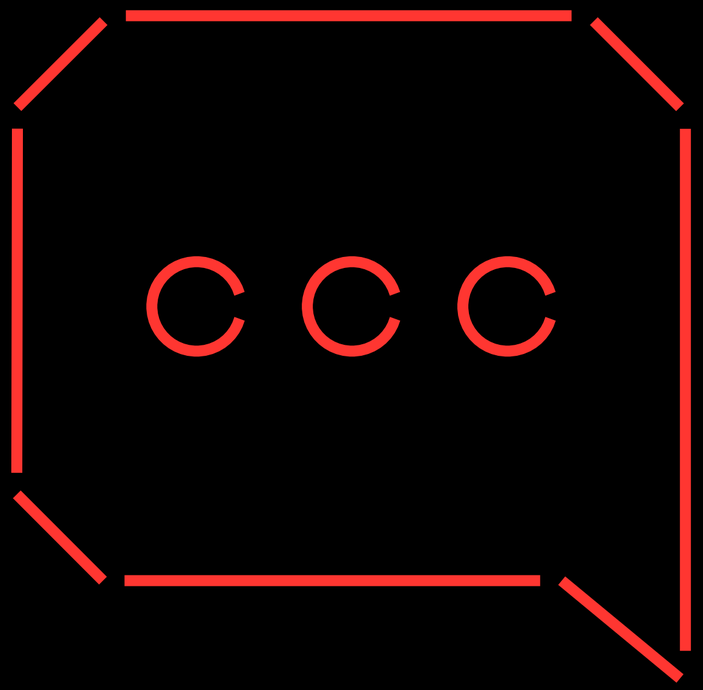
React.
Добавление страниц и роутинг

Михаил Зятьков,
Наставник, Яндекс.Практикум

Яндекс Практикум

Регламент



- Имя - настоящее как в Slack
- Включайте камеры
- Активно задавайте вопросы
- Не забывайте выключать микрофон

План

- Обсудим проект
- Поговорим о формах в React
- Проведем рефакторинг
- React router и многостраничные сайты на React
- Ответим на вопросы

Формы в React

Управляемые компоненты - текущие значения формы хранятся в стейте и при отображении формы значения берутся из стейта

Неуправляемые компоненты - для обращения к DOM элементам формы используются рефы

https://ru.reactjs.org/docs/forms.html

Управляемые компоненты

```
function Form(props) {
  const [value, setValue] = useState('');

  function handleChange(evt) {
    setValue(evt.target.value);
  }

  function handleSubmit(evt) {
    evt.preventDefault();
    alert('Отправленное имя: ' + this.state.value);
  }

  return (
    <form onSubmit={handleSubmit}>
      <label>
        Имя:
        <input type="text" value={value} onChange={handleChange} />
      </label>
      <input type="submit" value="Отправить" />
    </form>
  );
}
```

в стейте будет храниться значение поля ввода

обработчик события change поля ввода записывает значение поля ввода в стейт

полю ввода задается значение из стейта

полю ввода задается обработчик события change

https://ru.reactjs.org/docs/forms.html#controlled-components

Неуправляемые компоненты

```
function Form(props) {
  const inputRef = React.useRef();

  function handleSubmit(evt) {
    evt.preventDefault();
    alert('Отправленное имя: ' + inputRef.current.value);
  }

  return (
    <form onSubmit={handleSubmit}>
      <label>
        Имя:
        <input type="text" ref={inputRef}/>
      </label>
      <input type="submit" value="Отправить" />
    </form>
  );
}
```

создаем ref

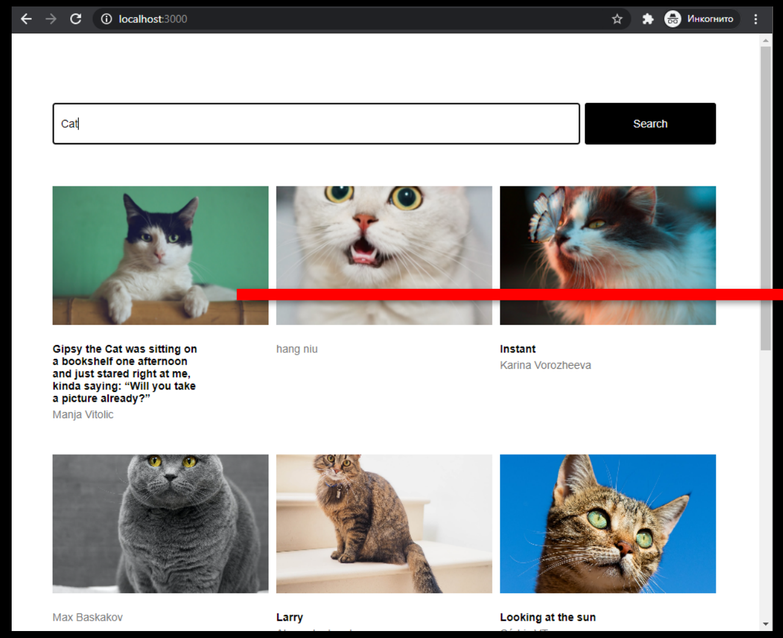
используя свойство current получаем DOM элемент на который назначен ref

передаем ref DOM элементу

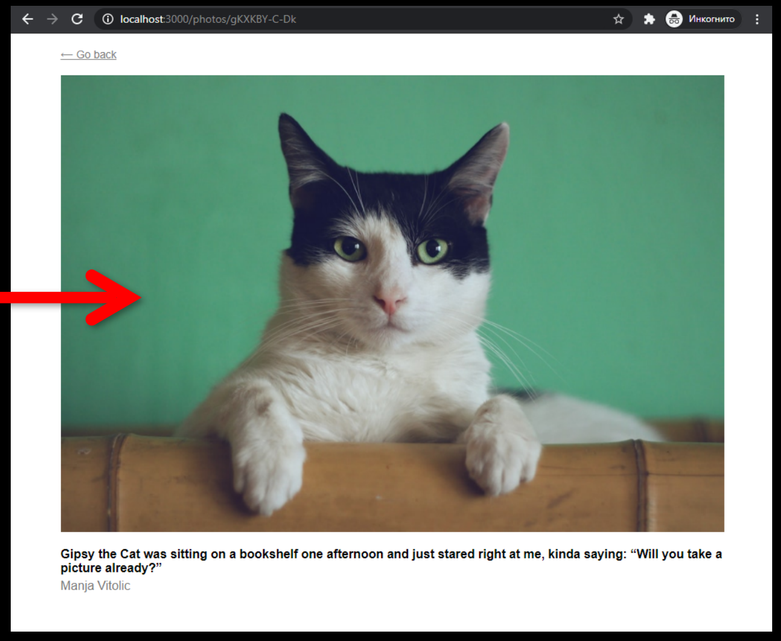
https://ru.reactjs.org/docs/refs-and-the-dom.html

Страницы проекта

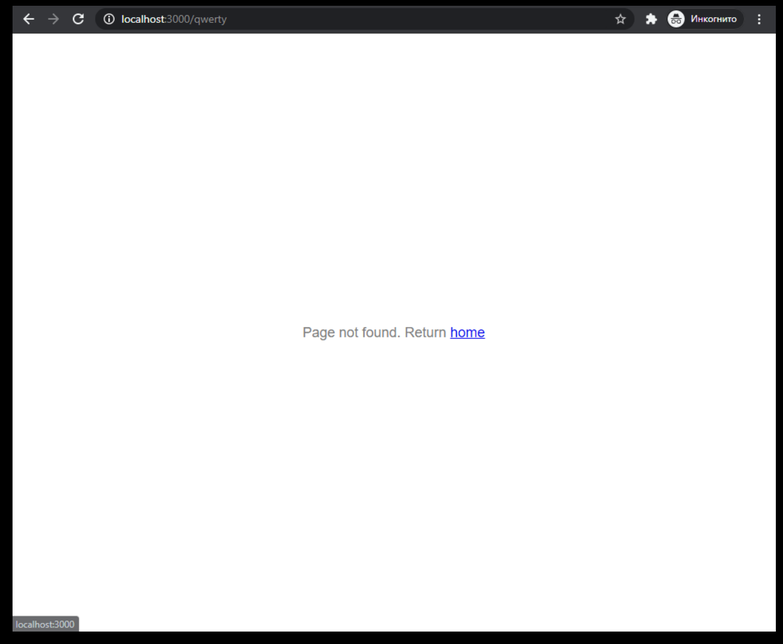
Главная страница "/"



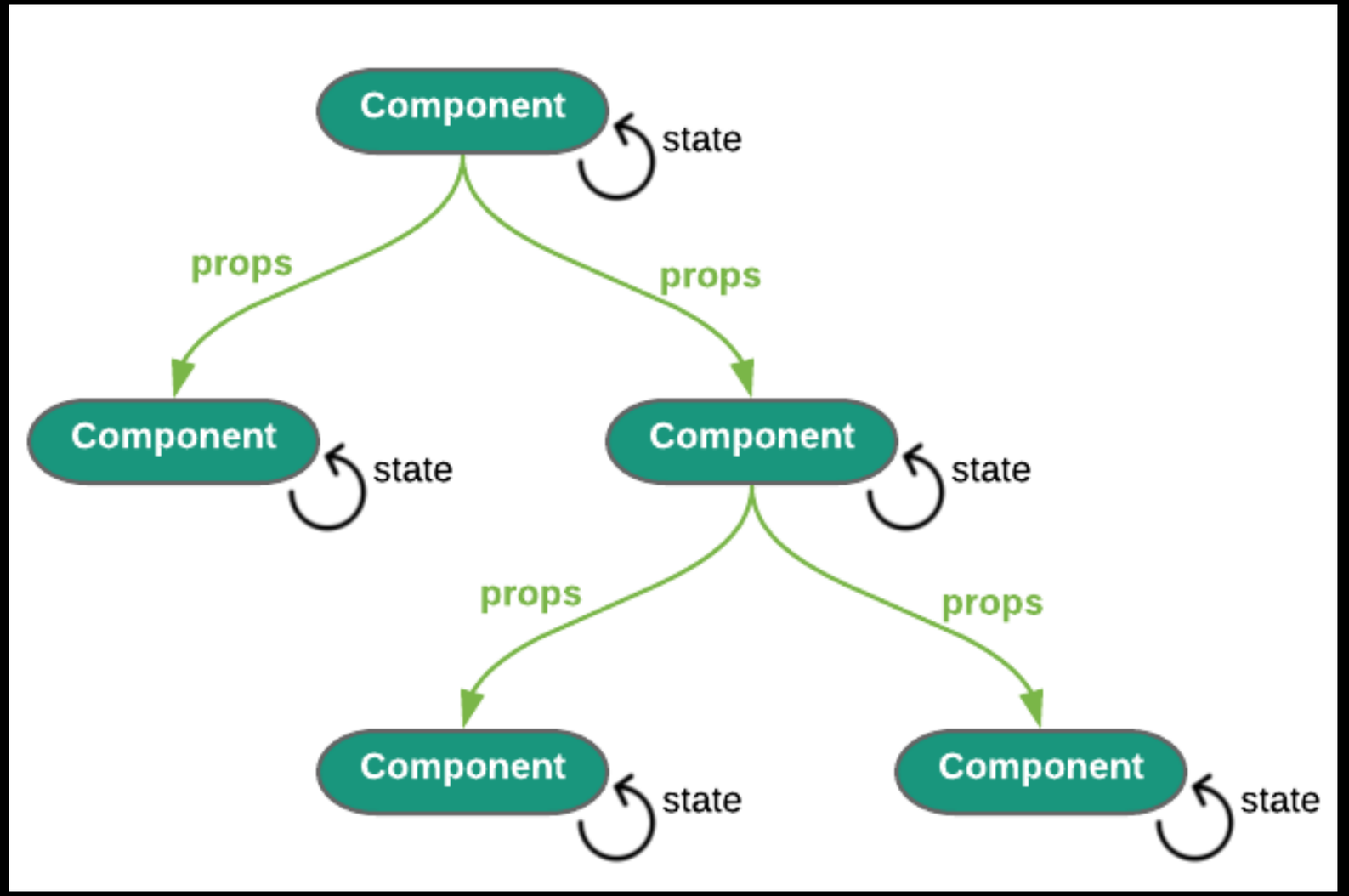
Страница фотографии "/photos/<id фото>"



Страница не найдена



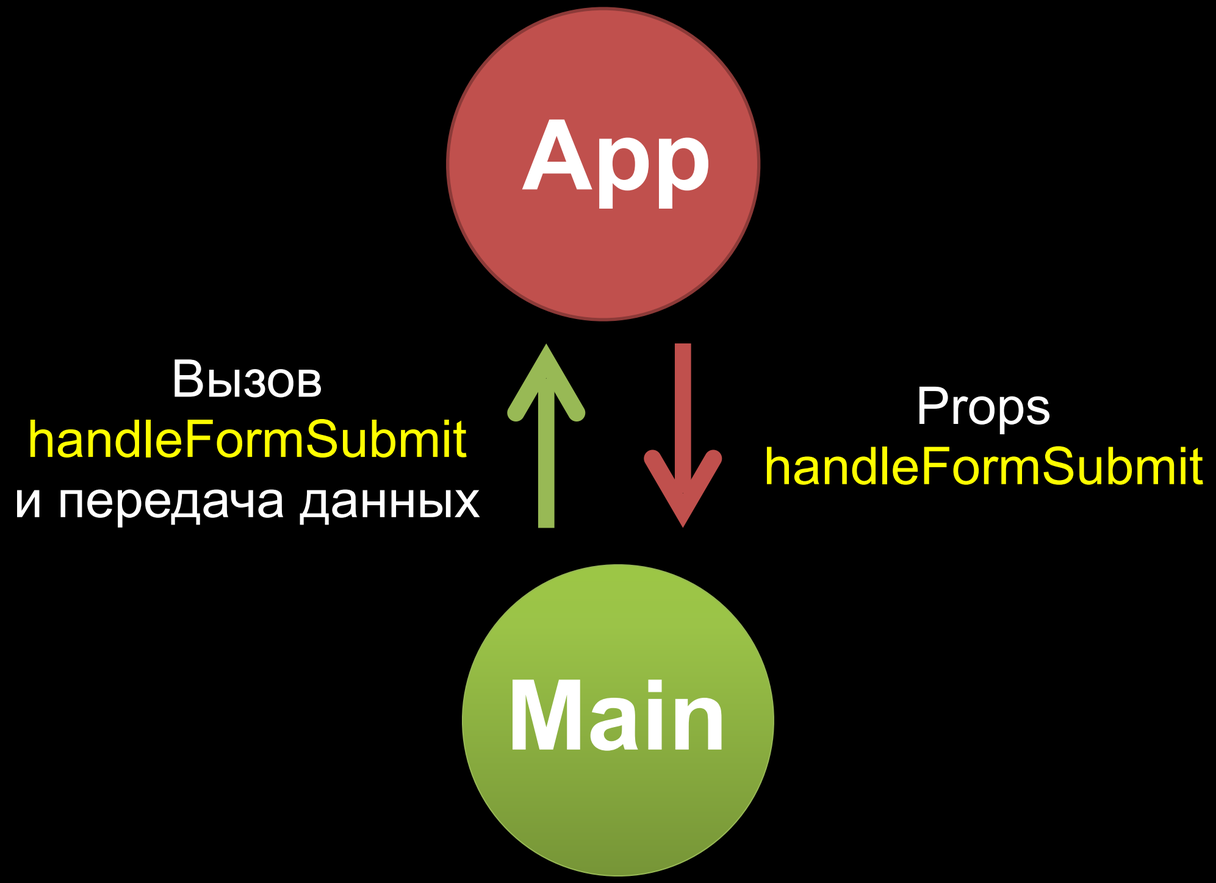
Поднятие состояния



https://ru.react.js.org/docs/lifting-state-up.html

Поднятие состояния

- Если мы передаем функцию, которая будет менять состояние, которое находится выше, то мы этот стейт (внутреннее, локальное состояние) поднимаем наружу.
- "Props down, Events up" – свойства, которые у нас есть, мы отправляем вниз, как значения (для отрисовки), а если нужно изменить состояние, то наверх

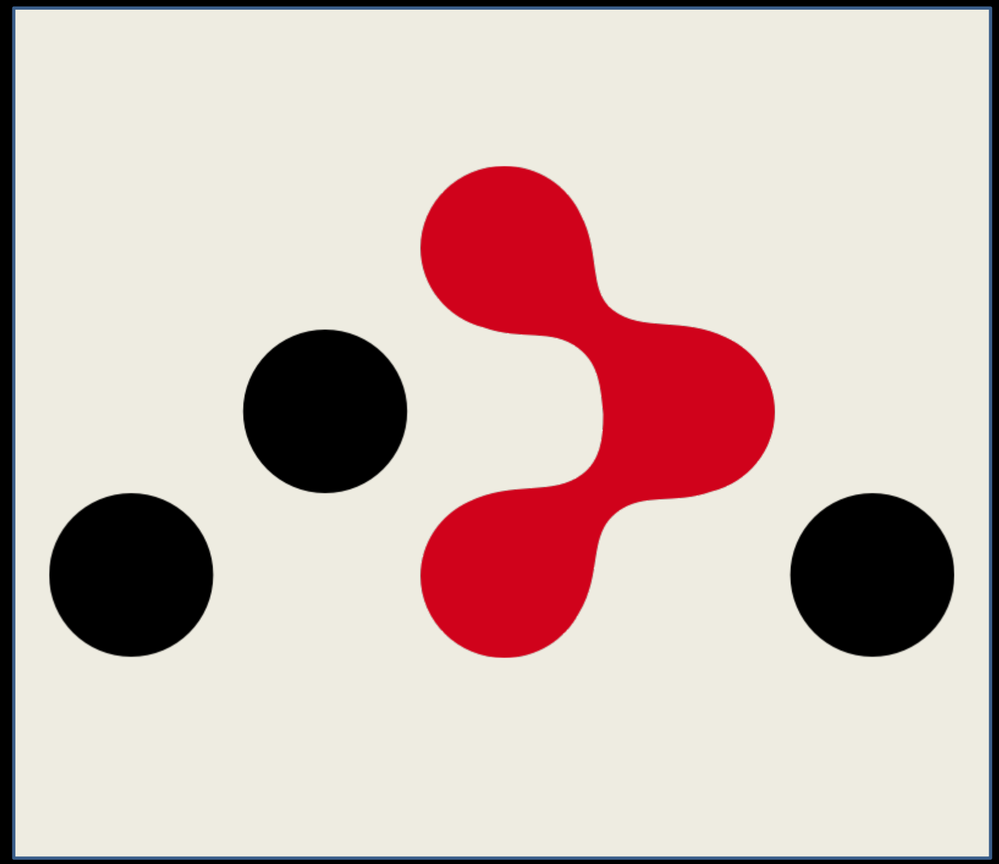


https://ru.react.js.org/docs/lifting-state-up.html

Роутинг

Особенности:

- переход между страницами внутри SPA приложения выполняется без перезагрузки страницы, меняется только адрес
- за отображение приложения по разным адресам отвечает js логика



Устанавливаем React Router

npm install react-router-dom

https://reactrouter.com/web/

React Router. Компоненты

<Router> - базовый компонент, отвечающий за ротинг внутри вложенных компонентов

Более частные реализации роутера:

- **<BrowserRouter>** - навигация через обычные url
- **<HashRouter>** - ссылка всегда на главную, но к ней добавляется путь после символа #
- **<MemoryRouter>** - адрес храниться в памяти, браузерная строка не используется
- **<StaticRouter>** - роутер никогда не изменяет адрес, используется для рендеринга на стороне сервера

<Route> - отображает компоненты, только если адрес заданный в пропс path совпадает с текущим адресом

<Switch> - в компонент оборачиваются несколько <Route>, отображается только первый <Route>, у которого совпал путь

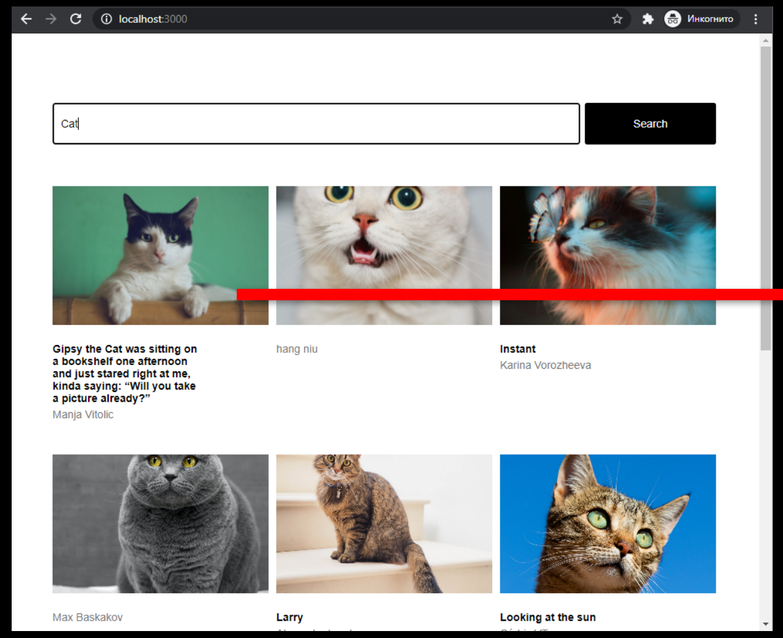
<Redirect> - при рендеренге этого компонента происходит переход на адрес указанный в пропсе to

<Link> - компонент ссылка для перехода внутри приложения, адрес задается в пропсе to.

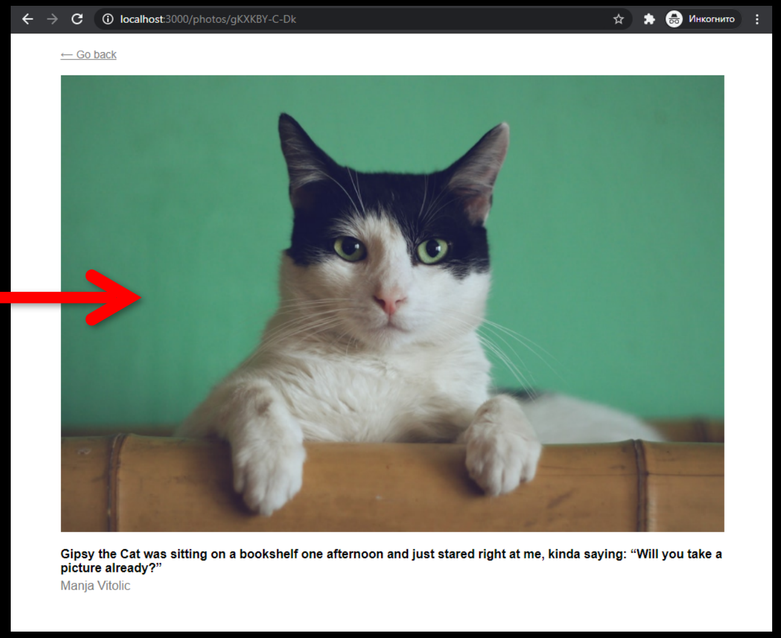
<NavLink> - работает как и <Link>, но так же позволяет задавать стили, если текущий адрес совпадает с указанным в пропсе to

Роуминг

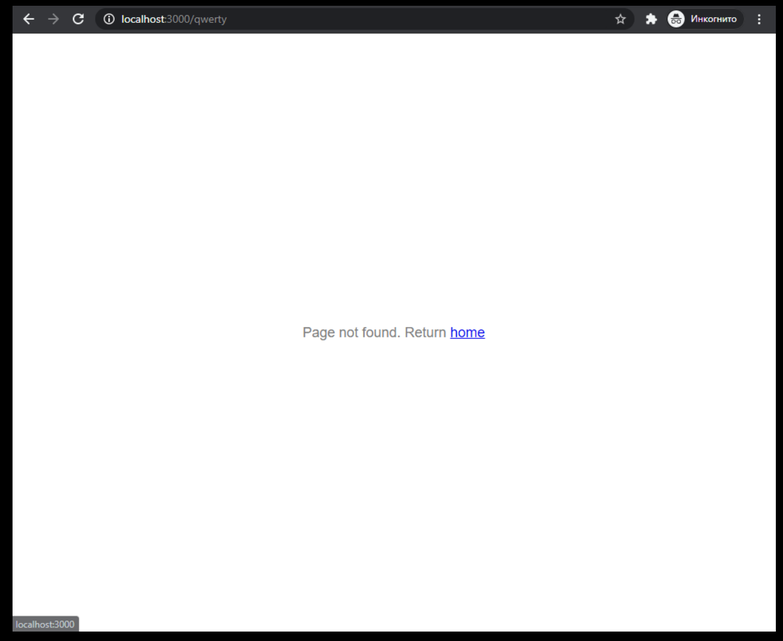
<Route path="/" exact>



<Route path="photos/:id">



<Route path="*">



Заключение

1. При работе с формой как с управляемым компонентом данные полей ввода храняться в стейте
2. Для доступа к DOM элементам используются ref
3. Поднятие стейта: состояние опускается вниз, но события поднимают его наружу
4. За каждой страницей закрепляется свой маршрут
5. Страница в реакте – это точно такой же компонент, как, например, кнопка
6. Самые необходимые компоненты для работы: BrowserRouter, Route, Switch, Link и хуки
7. Для ссылок внутри приложения используем <Link>, для внешних ссылок используем <a>

Вопросы