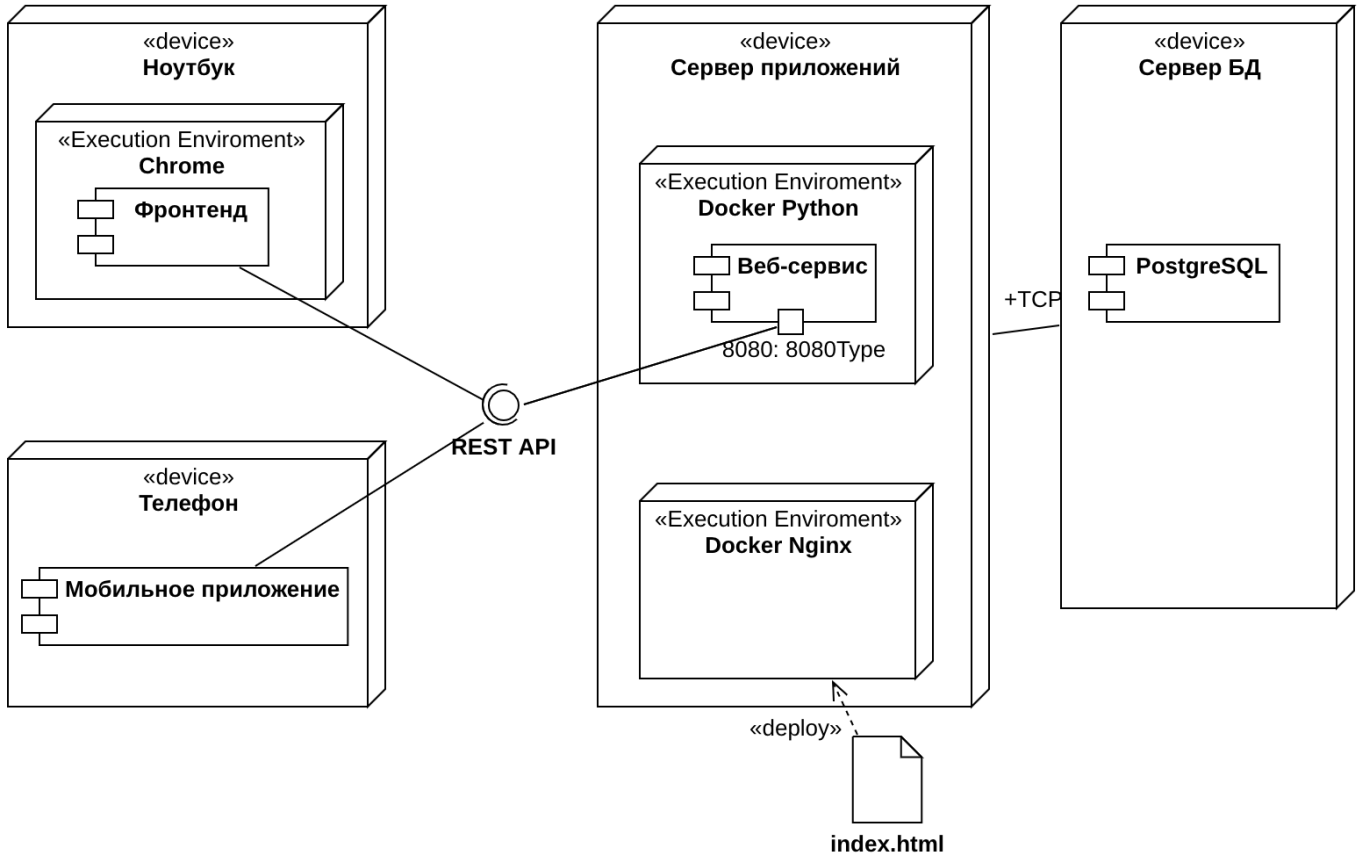
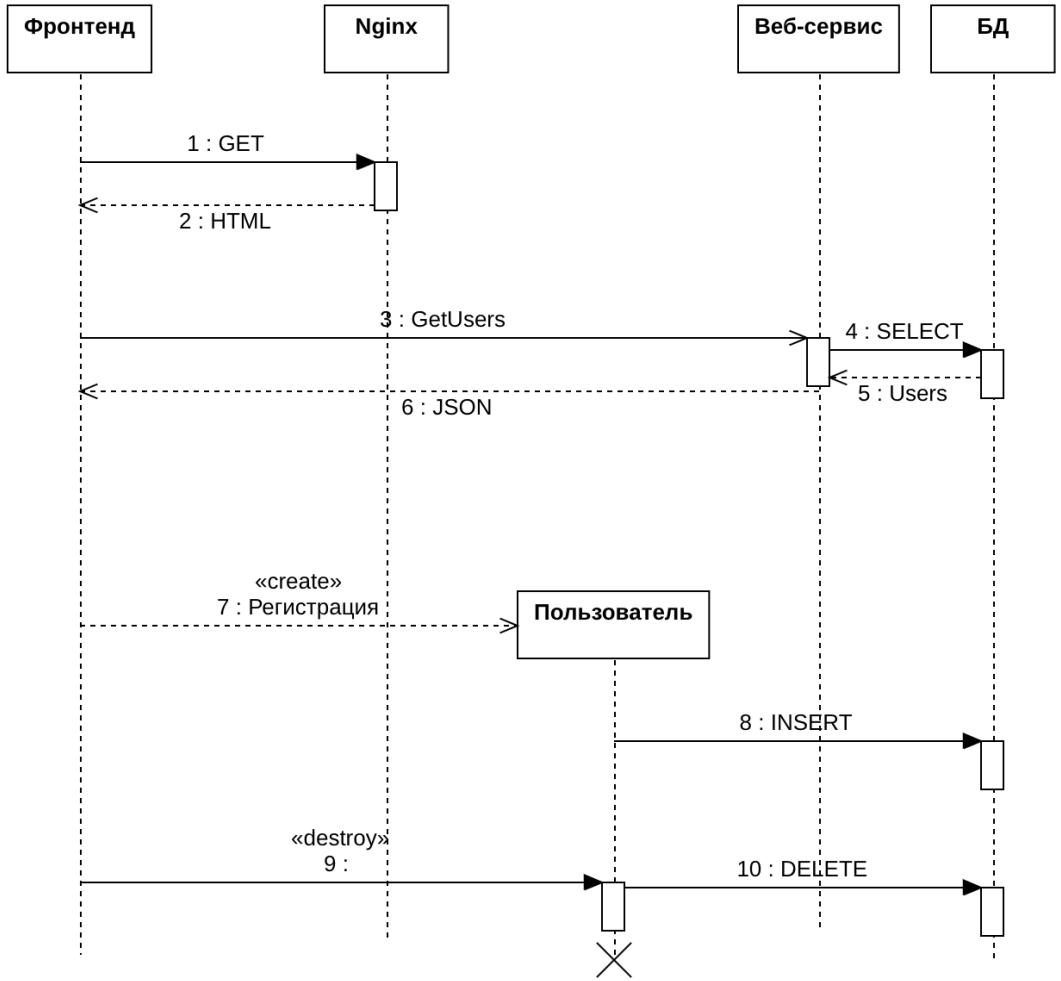
Лекция 7
Введение в фронтенд

Разработка интернет приложений

Канев Антон Игоревич

Трехзвенная архитектура. AJAX

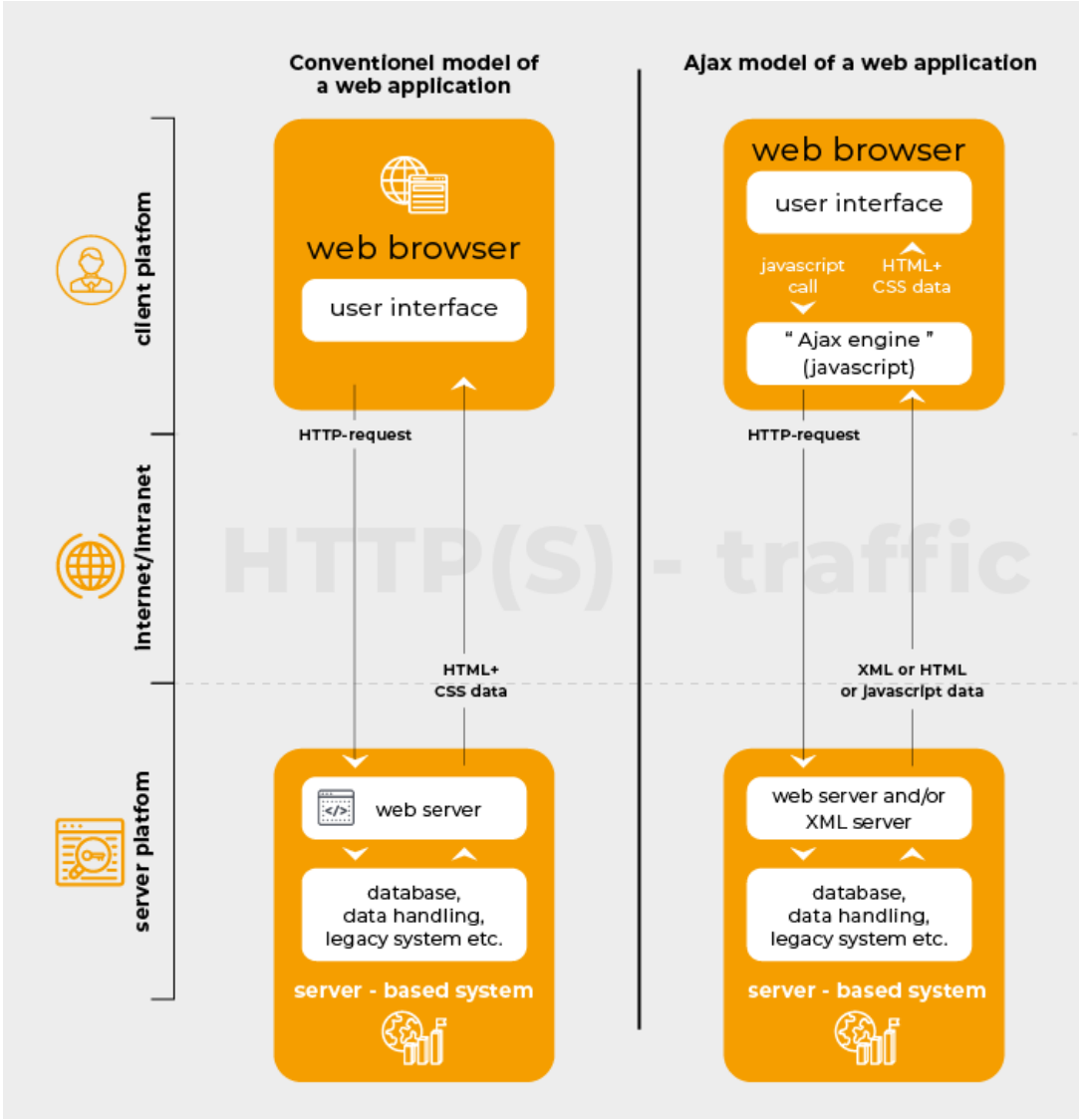




AJAX

- **AJAX**, Ajax (Asynchronous Javascript and XML — «асинхронный JavaScript и XML») — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером.
- В результате при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

- **JSON-RPC** (JavaScript Object Notation Remote Procedure Call — JSON-вызов удалённых процедур) — протокол удалённого вызова процедур, использующий JSON для кодирования сообщений.



XMLHttpRequest

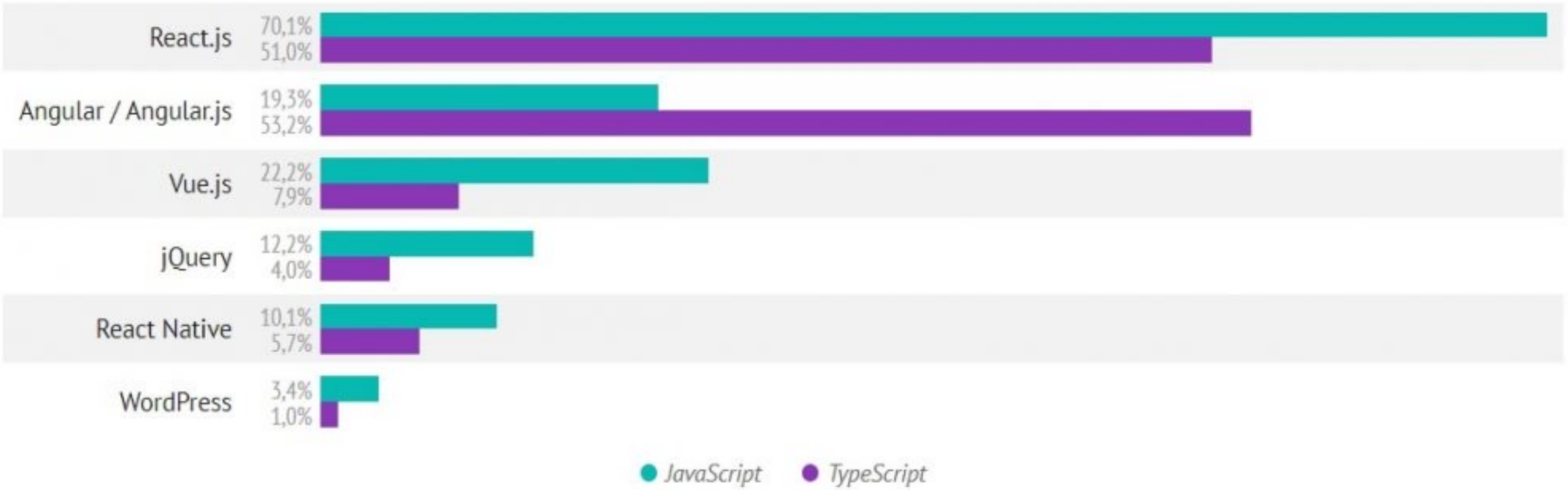
- **XMLHttpRequest** (XMLHTTP, XHR) — API, доступный в скриптовых языках браузеров, таких как JavaScript.
- Использует запросы HTTP или HTTPS напрямую к веб-серверу и загружает данные ответа сервера напрямую в вызывающий скрипт.
- Информация может передаваться в любом текстовом формате, например, в XML, HTML или JSON. Позволяет осуществлять HTTP-запросы к серверу без перезагрузки страницы.

```
var http_request = new XMLHttpRequest();
http_request.onreadystatechange = function () {
    if (http_request.readyState !== 4)
        return;

    if (http_request.status !== 200)
        throw new Error('request was defeated');

    do_something_with_object(JSON.parse(http_request.responseText));
    http_request = null;
};
http_request.open("GET", url, true);
http_request.send(null);
```

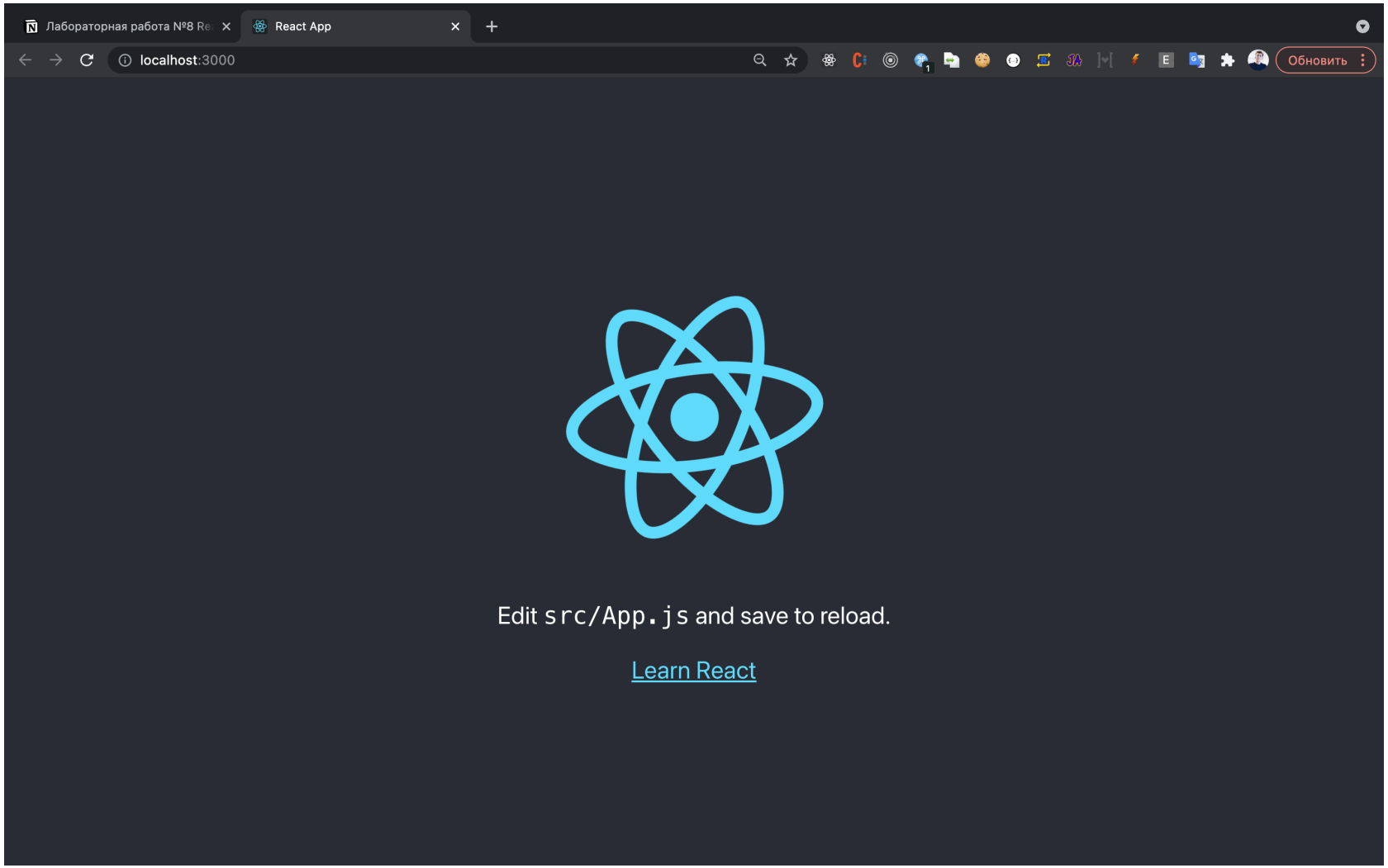
Web-фреймворки. Фронтенд



npm

- npx create-react-app my-app
- cd my-app
- npm start

React


Лабораторная работа №8 Re
React App
localhost:3000
Обновить
Edit src/App.js and save to reload.
Learn React

index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
document.getElementById('root')
);
```

App.js

```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

export default App;
```

Роутинг

```
"react-router": "^5.0.0",
"react-router-dom": "^5.0.0",
```

```
import { BrowserRouter, Route, Switch } from "react-router-dom";

function App() {

  return (
      <BrowserRouter basename="/">
          <Switch>
            <Route exact path="/">
              <h1>Это наша стартовая страница</h1>
            </Route>
            <Route path="/new">
              <h1>Это наша страница с чем-то новеньким</h1>
            </Route>
          </Switch>
      </BrowserRouter>
  );
}

export default App;
```

Link to

```
import { BrowserRouter, Route, Link, Switch } from "react-router-dom";

function App() {

  return (
      <BrowserRouter basename="/" >
        <div>
          <ul>
            <li>
              <Link to="/">Старт</Link>
            </li>
            <li>
              <Link to="/new">Хочу на страницу с чем-то новеньким</Link>
            </li>
          </ul>
          <hr />
          <Switch>
            <Route exact path="/">
              <h1>Это наша стартовая страница</h1>
            </Route>
            <Route path="/new">
              <h1>Это наша страница с чем-то новеньким</h1>
            </Route>
          </Switch>
        </div>
      </BrowserRouter>
  );
}

export default App;
```

Компоненты

- React-компоненты — это повторно используемые части кода, которые возвращают React-элементы для отображения на странице.

```
function Welcome(props) {
  return <h1>Привет, {props.name}</h1>;
}
```

```
class Welcome extends React.Component {
  render() {
    return <h1>Привет, {this.props.name}</h1>;
  }
}
```

Props

- props (пропсы) — это входные данные React-компонентов, передаваемые от родительского компонента дочернему компоненту.
- В любом компоненте доступны props.children. Это контент между открывающим и закрывающим тегом компонента.
- Для классовых компонентов используйте this.props.children

```
class Welcome extends React.Component {
  render() {
    return <h1>Привет, {this.props.name}</h1>;
  }
}
```

```
<BrowserRouter basename="/">
          <Switch>
            <Route exact path="/">
              <h1>Это наша стартовая страница</h1>
            </Route>
            <Route path="/new">
              <h1>Это наша страница с чем-то новеньким</h1>
            </Route>
          </Switch>
</BrowserRouter>
```

Состояние

- Компонент нуждается в state, когда данные в нём со временем изменяются.
- Например, компоненту Checkbox может понадобиться состояние isChecked.

- Разница между пропсами и состоянием заключается в основном в том, что состояние нужно для управления компонентом, а пропсы для получения информации.

Жизненный цикл приложения

- **1: Монтирование**

компонент запускает getDerivedStateFromProps(), потом запускается render(), возвращающий JSX. React «монтируется» в DOM

- **2: Обновление**

Данный этап запускается во время каждого изменения состояния либо свойств

- **3: Размонтирование**

React выполняет запуск componentWillUnmount() непосредственно перед удалением из DOM