

## Создание веб-приложения с помощью библиотеки React.js

*Круглик Роман Игоревич*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

### **Аннотация**

В статье рассматривается создание веб-приложения, которое показывает параметры погоды, запрашиваемого города. Для выполнения задачи использовалась библиотека React.js.

**Ключевые слова:** React, программирование, JavaScript.

## Create a web application using the React.js library

*Kruglik Roman Igorevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### **Abstract**

In article discusses the creation of a web application that shows the parameters of the weather, the requested city. To perform the task, the React.js library was used.

**Keywords:** React, programming, JavaScript.

React - это библиотека для создания пользовательских интерфейсов. Одной из ее отличительных особенностей является возможность использовать JSX - язык программирования с близким к HTML синтаксисом, который компилируется в JavaScript. Разработчики могут добиваться высокой производительности приложений с помощью Virtual DOM. С React можно создавать изоморфные приложения.

Virtual DOM может повысить производительность высоконагруженных приложений, что может снизить вероятность возникновения возможных неудобств и улучшает пользовательский опыт. Использование изоморфного подхода помогает производить рендеринг страниц быстрее, тем самым позволяя пользователям чувствовать себя более комфортно во время работы с приложением. Поисковые системы индексируют такие страницы лучше, поскольку один и тот же код может быть использован как в клиентской, так и в серверной части приложения, нет необходимости в дублировании одного и того же функционала.

На сегодняшний день исследования в области применения паттернов программирования актуальны. В статье Н.О. Корнукова [1] проводится анализ JavaScript-библиотек, использующих MVC архитектуру, а также описываются их преимущества. В работе М.М. Хрусталева, Д.А. Шуклин [2] рассматривают целесообразность использования JS-фреймворков для

разработки веб-сайтов. В статье Д.Р. Богданова, Д.А. Шуклина [3] представлен сравнительный анализ фреймворков по следующим критериям: функциональность, сообщество разработчиков, защита от взлома и т.д. О.Ю. Лазарева, Д.Д. Ложкина рассматривают популярные графические библиотеки и возможности их использования для добавления интерактивности электронным изданиям.

На основе данной библиотеки создано веб-приложение, которое при вводе города выводит температуру в реальном времени.

Основным файлом в приложении является `app.js` (см. рис. 1).

```
import React from "react";
import Info from "../components/info";
import Form from "../components/form";
import Weather from "../components/weather";
const API_KEY = "68dc6836762268cb21c932eeb69f1ae1";
class App extends React.Component {
  state = {
    temp: undefined,
    city: undefined,
  };
  gettingWeather = async (e) => {
    {
      e.preventDefault();
      const city = e.target.elements.city.value;
      const api_url = await fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${API_KEY}&units=metric`);
      const data = await api_url.json();
      console.log(data);
      this.setState({
        temp: data.main.temp,
        city: data.name
      });
    }
  };
  render() {
    {
      return (
        <div>
          <Info />
          <Form weatherMethod={this.gettingWeather} />
          <Weather
            temp={this.state.temp}
            city={this.state.city}
          />
        </div>
      );
    }
  }
}
export default App;
```

Рисунок 1. Class App

На начальном этапе подключается библиотека и компоненты. Далее создаётся компонент `App`. Переменная `state` содержит в себе информации о городе и его температуре. `gettingWeather` является функцией, которая работает через API и собирает данные о введённом городе, записывая в переменную `state`. Далее выводятся компоненты (см. рис. 2).

```
import React from 'react';
class Info extends React.Component{
  render()
  {
    return(
      <div>
        <h2>Погодное приложение</h2>
        <p>Узнайте погоду в вашем городе!</p>
      </div>
    );
  }
}
export default Info;
```

Рисунок 2. Компонент Info

В компонент Info выводится вся текстовая информация в приложении. (см. рис. 3).

```
import React from 'react';
class Form extends React.Component{
  render()
  {
    return(
      <form onSubmit={this.props.weatherMethod}>
        <input type="text" name="city" placeholder="Город"/>
        <button>Получить погоду</button>
      </form>
    );
  }
}
export default Form;
```

Рисунок 3. Компонент Form

Далее следует компонент Form в котором создаётся форма и ставится обработчик нажатия на кнопку. Введённый город передаётся в главный файл (см. рис. 4).

```
import React from 'react';
class weather extends React.Component{
  render()
  {
    return(
      <div>
        Город - {this.props.city}
        <br/>
        Температура - {this.props.temp}
      </div>
    );
  }
}
export default weather;
```

Рисунок 4. Компонент Weather

Следующим компонентом является Weather в котором выводятся полученные данные через API. Предположим, что нужно узнать температуру в Москве (см. рис. 5).

---

## Погодное приложение

Узнайте погоду в вашем городе!

Moscow	Получить погоду
Город - Moscow	
Температура - -11.2	

Рисунок 5. Результат работы приложения

Приложение работает без обработки страницы. React упрощает процесс создания веб-приложений, а также делает его более легким для доработки. Технология JSX облегчает создание приложения, так как можно использовать HTML код, который будет автоматически конвертироваться в JS.

### Библиографический список

1. Корнуков Н.О. Сравнение javascript библиотек, основанных на mvc архитектуре // Молодежный научно-технический вестник. 2014. № 10. С. 20.
2. Хрусталева М.М., Шуклин Д.А. Методы оптимизации разработки сайтов с

- использованием js-фреймворков // Альманах научных работ молодых ученых Университета ИТМО в 5 т. 2016. С. 183-185.
3. Богданов Д.Р., Шуклин Д.А. Сравнение основных клиентских javascript фреймворков // Альманах научных работ молодых ученых Университета ИТМО Материалы XLVI научной и учебно-методической конференции. 2017. С. 30-32.
  4. Лазарева О.Ю., Ложкина Д.Д. Обзор графических javascript-библиотек для создания интерактивных электронных изданий // Вестник МГУП имени Ивана Федорова. 2016. № 1. С. 76-78.