**Обзор библиотеки для создания реактивных приложений ReactJs**

***Волобуев Ю.С.***

*Севастопольский государственный университет, Севастополь*

*В статье приводится обзор библиотеки для построения пользовательских интерфейсов.*

*Ключевые слова: web, javascript, react, библиотека, приложение, программирование.*

React - это библиотека JavaScript с [открытым исходным](https://ibracilinks.com/blog/quest-ce-quun-logiciel-open-source-et-pourquoi-est-ce-important) кодом, которая используется для создания пользовательских интерфейсов специально для одностраничных приложений. Она используется для управления слоем отображения веб-приложений и мобильных приложений. React также позволяет нам создавать повторно используемые компоненты пользовательского интерфейса. React был создан [Джорданом Уолком](https://twitter.com/jordwalke), инженером-программистом, работающим в [Facebook](https://www.facebook.com/). React был впервые развернут в ленте новостей Facebook в 2011 году и на Instagram.com в 2012 году.

React позволяет разработчикам создавать крупные веб-приложения, которые могут изменять данные без перезагрузки страницы. Основные цели React – скорость, масштабируемость, простота. Он работает только с пользовательскими интерфейсами приложений. Это соответствует представлению модели MVC. Его можно использовать с комбинацией других библиотек или фреймворков JavaScript.

**JSX**

В React вместо обычного JavaScript для шаблонов используется JSX. JSX - это простой JavaScript, который использует синтаксис HTML-тегов для отображения подкомпонентов. Синтаксис HTML обрабатывается в вызовах JavaScript в React Framework. Также можно писать на чистом JavaScript.

**React Native**

В React есть собственные библиотеки, анонсированные Facebook в 2015 году, которые позволяют использовать React для мобильных приложений.

React-native - это фреймворк для разработки мобильных приложений с использованием только JavaScript. Он использует ту же концепцию, что и React, что позволяет использовать богатую библиотеку мобильных UI компонентов. Он использует те же основные модули пользовательского интерфейса, что и приложения для iOS и Android.

**Односторонний канал передачи данных**

В React есть набор неизменяемых значений, которые передаются как свойства в тегах HTML. Компонент не может напрямую изменять какое-либо свойство, но может передавать функцию обратного вызова, с помощью которой можно вносить изменения. Этот процесс называется “нисходящий поток свойств; восходящий поток действий”.

**Объектная модель виртуального документа**

React создает кеш структуры данных в памяти, вычисляет внесенные изменения, а затем обновляет браузер. В результате сравнения виртуального DOM React обновляет только те фрагменты, которые изменились.

**Почему React?**

Существует множество платформ с открытым исходным кодом для облегчения разработки веб-приложений, таких как Angular, Vue.js. Поскольку мир фронтенда меняется ежедневно, неэффективно тратить время на изучение нового фреймворка. Поэтому, если вы в поисках лучшего фреймворка, приведем несколько преимуществ React:

**1. Простота**

ReactJS легко освоить со старта. Компонентный подход, четко определенный жизненный цикл и прямое использование JavaScript делают React очень простым в изучении, создании профессионального веб-сайта (и мобильных приложений) и его поддержке. React использует специальный синтаксис JSX, который позволяет смешивать HTML и JavaScript.

**2. React Native**

React можно использовать для создания мобильных приложений (React Native). Таким образом, мы можем создавать как IOS, так и Android, и веб-приложения.

**3. Канал передачи данных**

React использует одностороннюю привязку данных, а архитектура приложения под названием Flux управляет потоком данных к компонентам через единую точку управления - диспетчер. Отдельные компоненты крупных ReactJS приложений легче отлаживать.

**4. Тестируемость**

Приложения ReactJS очень просты в тестировании. Представления React можно рассматривать как функции состояния, поэтому мы можем манипулировать с состоянием, которое мы передаем в представление ReactJS, и взглянуть на выход и действия, события, запускаемые функции и т. д.\

**СПИСОК ЛИТЕРАТУРЫ**

1. ReactJS [Электронный ресурс]. URL: https://reactjs.org (дата обращения: 12.04.2021).