Состояние и жизненный цикл

№ урока: 4 **Курс:** React Essential

Средства обучения: Браузер Chrome, редактор кода VS Code или любой другой.

Обзор, цель и назначение урока

Познакомиться с состоянием и жизненным циклом в React компонентах. Узнать разницу между функциональными и классовыми компонентами, а также научиться их использовать. Кратко ознакомиться с Redux.

Изучив материал данного занятия, учащийся сможет

- Использовать state в компонентах.
- Использовать методы жизненного цикла компонента.
- Писать функциональные и классовые компоненты.

Содержание урока

1. Функциональный компонент
2. Классовый компонент
3. Преобразование функционального компонента в классовый
4. Добавление методов жизненного цикла
5. Знакомство с Redux

Резюме

- Компоненты в React бывают двух видов - функциональные и классовые.
- Компонент — получает данные в одном объекте («пропсы») в качестве параметра и возвращает React-элемент.
- Композиция компонентов - компоненты могут ссылаться на другие компоненты в возвращённом ими дереве. Это позволяет нам использовать одну и ту же абстракцию — компоненты — на любом уровне нашего приложения. Неважно, пишем ли мы кнопку, форму или целый экран.
- Когда React встречает компонент, он собирает все JSX-атрибуты и дочерние элементы в один объект и передаёт их нашему компоненту. Такие объекты называются «пропсы» (props).
- **Пропсы можно только читать** - компонент никогда не должен что-то записывать в свои пропсы — вне зависимости от того, функциональный он или классовый.

Методы жизненного цикла

- Жизненный цикл компонента в целом делится на четыре части:

1) инициализация (монтирование)
2) монтаж
3) обновление
4) размонтирование

CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Первоначальный рендеринг компонента в DOM называется «монтирование» (mounting).
- Каждый раз, когда DOM-узел, созданный компонентом, удаляется, происходит «размонтирование» (unmounting).
- В приложениях с множеством компонентов очень важно освобождать используемые системные ресурсы, когда компоненты удаляются. Для этого мы отписываемся от прослушивания событий в методе жизненного цикла - componentWillUnmount.

Как правильно использовать состояние:

- Не изменяйте состояние напрямую - вместо этого используйте setState().

Обновления состояния могут быть асинхронными:

- React может сгруппировать несколько вызовов setState() в одно обновление для улучшения производительности.
- Есть **два варианта использования setState()** :
    - Первый - передать объект нового состояния.
    - Второй - вариант вызова setState() который принимает функцию, а не объект. Эта функция получит предыдущее состояние в качестве первого аргумента и значения пропсов непосредственно во время обновления в качестве второго аргумента.

Обновления состояния объединяются

- Когда мы вызываем setState(), React объединит аргумент (новое состояние) с текущим состоянием.
- **Состояния объединяются поверхностно,** поэтому вызов this.setState({comments}) оставляет this.state.posts нетронутым, но полностью заменяет this.state.comments.
- **Redux** — это шаблон для управления состоянием приложения.
- **Redux** — библиотека управления состоянием для приложений, написанных на JavaScript. Она помогает писать приложения, которые ведут себя стабильно/предсказуемо, работают на разных окружениях (клиент/сервер/нативный код) и легко тестируемы.

Закрепление материала

- Что можно делать с пропсами компонента?
- Сколько есть вариантов использования setState, в чем разница?
- Какие вы знаете методы жизненного цикла?

Дополнительное задание

Задание
Прочитайте документацию react, а именно - раздел с объяснением жизненных циклов и состояния.

Самостоятельная деятельность учащегося

Задание 1
Выучите основные понятия, рассмотренные на уроке.

CyberBionic systematics
CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Задание 2
Создайте компонент, который будет появляться при нажатии на кнопку и прятаться при еще одном нажатии. После чего добавьте методы жизненного цикла, в которых будет запускаться код, показывающий «alert()» с информацией о том закрылся компонент или открылся.

Задание 3
Ознакомьтесь с Redux, посмотрите примеры использования.

Рекомендуемые ресурсы

https://ru.reactjs.org/docs/state-and-lifecycle
https://ru.reactjs.org/docs/react-component
https://redux.js.org/
https://rajdee.gitbooks.io/redux-in-russian/content/

CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com