

Формы

№ урока: 5 **Курс:** React Essential

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

Обзор, цель и назначение урока

Узнать, как можно работать с формами в React. Познакомиться с такими понятиями как подъем состояния и валидация с PropTypes.

Изучив материал данного занятия, учащийся сможет

- Обрабатывать события.
- Работать с формами.
- Добавлять валидацию props с помощью PropTypes.

Содержание урока

1. Контролируемые и не контролируемые компоненты
2. Работа с Input
3. Подъем состояния
4. Валидация параметров с PropTypes
5. Мышление в стиле React

Резюме

- **Контролируемые и не контролируемые компоненты** - чаще всего вы нужно будет использовать контролируемые компоненты, за несколькими исключениями. Контролируемыми являются те компоненты, которые не сами содержат свое состояние, а его передают им. В React мутабельное состояние обычно содержится в свойстве компонентов state и обновляется только через вызов `setState()`.
- **Обработка событий** — события в React именуются в стиле camelCase, вместо нижнего регистра. В React нельзя предотвратить обработчик события по умолчанию, вернув false. Нужно явно вызвать `preventDefault`.
- **Добавление обработчика события** - React определяет синтетические события в соответствии со [спецификацией W3C](#). Обработчики событий получают экземпляр `SyntheticEvent`, это кроссбраузерная обёртка над нативным экземпляром события. У неё такой же интерфейс, как и у нативного события, включая методы `stopPropagation()` и `preventDefault()`. Эта обёртка помогает событиям работать одинаково во всех браузерах. Если вам всё-таки нужно получить нативное браузерное событие, обратитесь к атрибуту `nativeEvent`. Если вы хотите обратиться к полям события асинхронно, вам нужно вызвать `event.persist()` на событии. Тогда оно будет извлечено из пула, что позволит вашему коду удерживать ссылки на это событие.
- **Привязка контекста «this»** - при обращении к this в JSX-колбэках необходимо учитывать, что методы класса в JavaScript по умолчанию не [привязаны](#) к контексту. Если вы забудете привязать метод и передать его, к примеру, в onClick, значение this будет undefined в момент вызова функции.

- **Ter textarea** - использует атрибут value. Таким образом, форму с <textarea> можно написать почти тем же способом, что и форму с однострочным <input>.
- **Ter select** - пункт списка, который выбран, выбирается из-за установленного атрибута selected. React вместо этого атрибута использует value в корневом теге select. В управляемом компоненте так удобнее, потому что обновлять значение нужно только в одном месте (state). В атрибут value можно передать массив, что позволит выбрать несколько опций в теге select.
- **Загрузка файла** - в React <input type="file"> всегда является неуправляемым компонентом, потому что его значение может быть установлено только пользователем, а не программным путём. Для взаимодействия с файлами следует использовать File API.
- **Подъем состояния** - в React совместное использование состояния достигается перемещением его до ближайшего предка компонентов, которым оно требуется. Это называется «подъем состояния».
- **PropTypes** - React предоставляет встроенные возможности для проверки типов. Для запуска этой проверки на пропсах компонента вам нужно использовать специальное свойство propTypes.

Закрепление материала

- Что такое Synthetic Event?
- Зачем привязывать контекст к jsx-колбэкам?
- Что является контролируемым компонентом в React?

Дополнительное задание

Задание

Прочитайте документацию react, а именно - раздел с объяснением Synthetic Event и Философия React.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Создайте компонент формы с тремя контролируемыми компонентами, которые рендерят input. После отправки формы - страница не перезагружается, а содержимое полей выводится в виде json под формой.

Задание 3

Создайте компонент с двумя контролируемыми компонентами, которые рендерят input. Они должны принимать только числа. После того, как пользователь вводит число в input, его содержимое выводится в блоке под ним с конвертацией - number * 0,5 и number * 0,05 соответственно.

Рекомендуемые ресурсы

<https://ru.reactjs.org/docs/handling-events>

<https://ru.reactjs.org/docs/uncontrolled-components>

<https://ru.reactjs.org/docs/events>
<https://ru.reactjs.org/docs/thinking-in-react>