React: Map components, TypeScript

AIT
Hi-tech team

Повторим;)

Что такое useState?

Что возвращает useState?

Что можно хранить внутри useState?

ЦЕЛЬ

Правила использования map для отображения компонентов.

ПЛАН ЗАНЯТИЯ

1. Метод map для компонентов
2. Установка и настройка проекта TS

Map method for components

В React метод map() обычно используется для обхода массива данных и создания нового массива элементов React.

Это особенно полезно при рендеринге списков компонентов.

Пример использования map() для создания компонентов:

Создаём компонент, в котором с помощью props будут передаваться данные для создания списка

```
const DataList = ({ data }) => {
  return (
    <ul>
      {data.map((item, index) => (
        <li key={index}>{item}</li>
      ))}
    </ul>
  );
};
```

Пример использования map() для создания компонентов:

Передаём в компонент DataList массив через родительский компонент

```
const App = () => {
  const dataArray = ['Item 1', 'Item 2', 'Item 3'];

  return (
    <div>
      <h1>List of Items</h1>
      <DataList data={dataArray} />
    </div>
  );
};
```

Key

Когда вы создаете множество компонентов с использованием map() или других методов, React ожидает, что каждый компонент будет иметь уникальный key. Это помогает React эффективно обновлять и перерисовывать компоненты при изменении данных

Как идентификаторы для ключей, лучше использовать уникальные и стабильные значения. Это могут быть уникальные идентификаторы элементов, предоставляемые, например, сервером, базой данных или библиотекой, например **UUID**

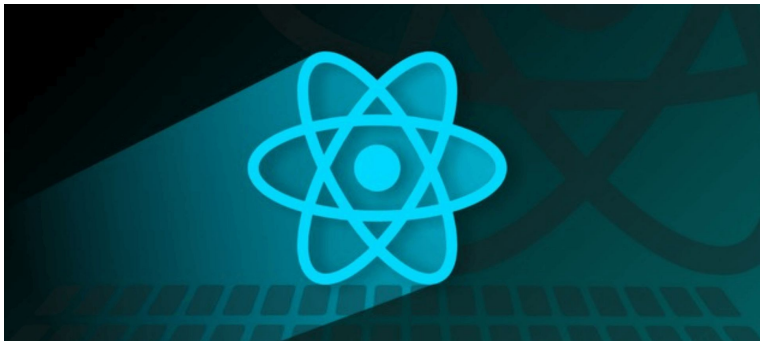
Установка - **npm install uuid**

Импорт `import {v4} from 'uuid';`

Использование `v4()`

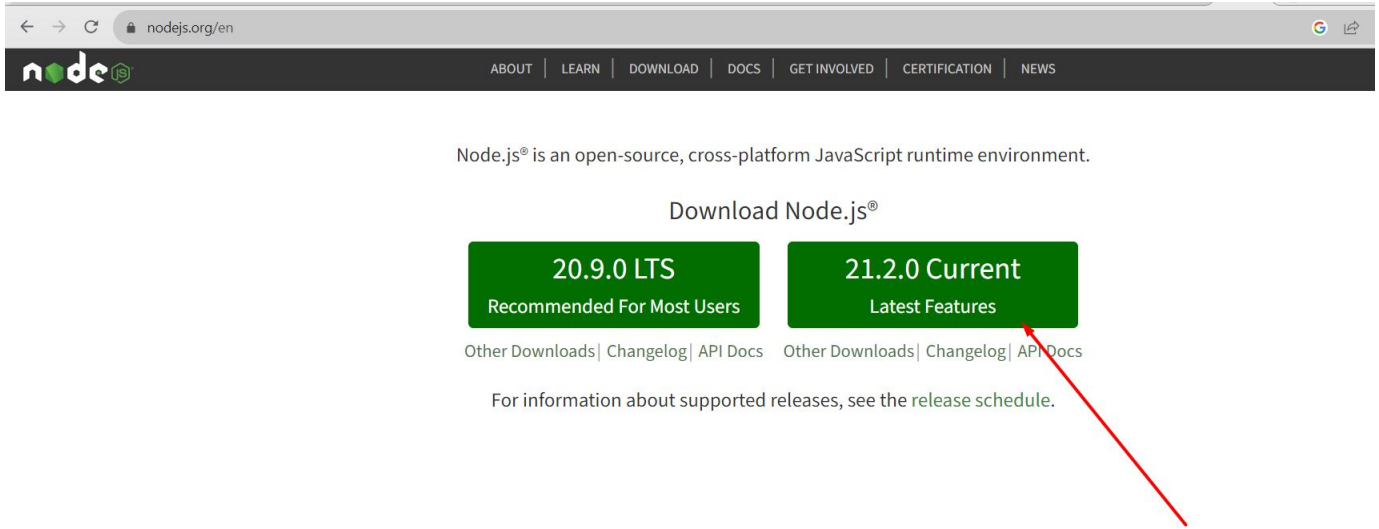
```
const ItemList = ({ items }) => {
  return (
    <ul>
      {items.map((item) => (
        <li key={item.id}>{item.name}</li>
      ))}
    </ul>
  );
};
```

Создание React проекта



Установка NodeJS

1. Перейдите на сайт https://nodejs.org/en
2. Скачайте и установите последнюю версию Node.js



Создание репозитория. Работа с GIT

```
# склонировать репозиторий с сервера на свой компьютер
git clone [url]


# скачать с сервера последние изменения
git pull


# Создать и ответвиться в новую ветку(делать такое от главной ветки, предварительно подтянув изменения с помощью git pull)
git checkout -b branch-name


# загрузить свои обновления на удаленный репозиторий (сделать коммит)
git add .
git commit -m "message"
git push

# пример push при работе с отдельной веткой
git push --set-upstream origin branch_name
```

Создание проекта

1. Откройте VSCode, перейдите в папку, в которой будет лежать ваш проект и в терминале введите следующую команду -

```
npx create-vite@latest . --force --template react-ts
```

1. Настройте зависимости - `npm install`

1. Запустите проект `npm run dev`

Примерная структура проекта

```
├── node_modules/       # Папка с установленными npm-зависимостями (создается после `npm install`)
├── public/             # Публичные статические файлы (например, `favicon.ico`)
│   └── vite.svg        # Логотип Vite (пример статического ресурса)
├── src/                # Исходный код приложения
│   ├── App.css         # Стили для компонента App
│   ├── App.tsx         # Главный React-компонент приложения
│   ├── index.css       # Глобальные стили для всего приложения
│   ├── main.jsx        # Точка входа в приложение (рендеринг React)
│   └── assets/         # Папка для статических ресурсов (например, изображений)
├── .gitignore          # Файлы и папки, игнорируемые Git
├── index.html          # Главный HTML-файл, который является базой для приложения
├── package.json        # Файл конфигурации npm с информацией о проекте и зависимостях
├── tsconfig.json       # Конфигурация TypeScript
├── tsconfig.node.json  # Дополнительная конфигурация TypeScript для Node.js
├── vite.config.ts      # Конфигурация Vite
└── README.md           # Файл с описанием проекта (по умолчанию может быть пустым)
```

Менеджер пакетов - npm

npm (Node Package Manager) — это инструмент для управления пакетами (библиотеками и модулями) в экосистеме Node.js. Он позволяет устанавливать, обновлять, удалять и управлять зависимостями в вашем проекте.

Примечание: для запуска команд npm не забудьте перейти в директорию вашего проекта.

Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH

AIT
Hi-tech team