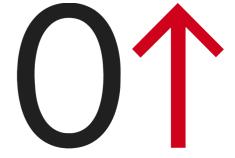
День 3

Работа с НТТР запросами



Жизненный цикл

```
constructor(props) {
   super(props);
   this.state = {};
}
```

```
componentWillMount() {
}
```

```
componentDidMount() {
}
```

```
componentWillUnmount() {
}
```

Введение в НТТР

HTTP — широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов (HyperText Transfer Protocol)

✓ Задача, которая традиционно решается с помощью протокола HTTP — обмен данными между пользовательским приложением и веб-сервером.

Клиент составляет и передает запрос на сервер, сервер обрабатывает и анализирует его, после этого создается ответ и отправляется пользователю.

Основные методы запроса

GET - используется для получения данных

POST - используется для отправки данных

Код состояния (status code) — часть первой строки ответа сервера при запросах по протоколу HTTP. Он представляет собой целое число из трёх цифр.

Описание кодов:

1xx - Informational (информационные)

2xx - Success (успешно)

3xx - Redirection (перенаправление)

4xx - Client Error (ошибка клиента)

5xx - Server Error (ошибка сервера)

Fetch

Fetch - функция, используемая в React Native для отправки запросов.



- **Отправляет** запрос на указанный URL-адрес.
- Возвращает Promise, который содержит в себе ответ запроса.

fetch('https://goodsite.com/data.json');

Fetch также может принимать второй аргумент, который позволяет настроить HTTP-запрос.

Вы можете указать <mark>дополнительные заголовки</mark> или сделать запрос POST

```
fetch('https://goodsite.com/endpoint/', {
    method: 'POST',
    headers: {
        Accept: 'application/json',
        'Content-Type': 'application/json',
        },
    body: JSON.stringify({
        firstParam: 'yourValue',
        secondParam: 'yourOtherValue',
    }),
});
```

Синтаксис функции .fetch()

```
let promise = fetch(url, options);
```

url - URL, на который нужно сделать запрос, options - объект с настройками запроса.

Свойства options:

method - метод запроса, headers - заголовки запроса (объект), body - тело запроса.

```
fetch(url).then(response => {
    response.json().then(data => {
        return data;
    });
})
```

Обработка ошибок при запросах

Пример запроса с использованием Promise

```
fetch(url)
  then(response => {
    if (response status !== 200) {
      console.log(`Problem. Status Code: ${response.status}`);
      return;
    response.json().then(data => {
      return data;
   });
  .catch(err => {
    console.log('Error', err);
  });
```

Homework

github.com/021NIS/Suretgram