|  |
| --- |
| Министерство образования Республики Беларусь  УО «Полоцкий государственный университет им. Ефросинии Полоцкой» |

|  |  |
| --- | --- |
|  | Факультет информационных технологий  Кафедра технологий программирования |

|  |
| --- |
| ОТЧЕТ |
| **к лабораторной работе №8**  **Дисциплина**: «Программирование для Интернет» |

|  |  |
| --- | --- |
| Выполнил: | Студент группы 19-ИТ-2  Гилимович А.Н. |

|  |  |
| --- | --- |
| Проверила: | Преподователь  Виноградова А.Д. |

|  |
| --- |
| Полоцк, 2022 г. |

**Сетевые запросы.**

**Цель работы:** изучить принцип работы метода fetch(), научиться применять полученные знания на практике.

**Задание 1**

Необходимо создать приложение, в котором будет кнопка отправляющая дату и время нажатия на кнопку JSON серверу(POST запрос). После отправки фейковый сервер должен сохранять принятые данные.

**Задание 2**

Доработайте приложение из первого задания так, чтобы в нем отображался список дат и времени, который уже сохранен. Для этого вам необходимо выполнить GET запрос к JSON серверу и отобразить полученные данные на странице.

**Задание 3**

Доработайте ранее созданное приложение так, чтобы рядом с каждым элементом списка дат и времени была расположена кнопка обновления, которая по нажатию будет отправлять PUT запрос, обновляющий список так, чтобы дата, рядом с которой пользователь нажал на кнопку обновления, обновлялась на текущую. После обновления на сервере необходимо удалить данные и на экране у пользователя.

**Задание 4**

Доработать ранее созданное приложение так, чтобы рядом с каждым элементом списка появилась кнопка удаления(DELETE запрос). При нажатии на эту кнопку, элемент рядом с которым она расположена должен удаляться на фейковом сервере и у клиента.

**Ход работы**

Изучив теоритический материал, преступил к выполнению заданий.

Так как все задания связаны, то по выолнению лабораторной работы получилась одна программа.

**Листинг 1** – кнопка добавления задачи

import { Component } from "react";

class But extends Component {

constructor(props){

super(props);

this.state = {

url: "http://localhost:3000/dates",

dates: []

}

this.request();

}

request = () => {fetch(this.state.url)

.then(response => {

return response.json();

})

.then((data) => {

this.setState({dates: [...data]})

// this.state.dates = [...data];

console.log(this.state.dates);

})

}

render(){

return(

<div className="container">

<button

onClick={() => {

let data = {

date: (new Date())

}

let url = "http://localhost:3000/dates";

fetch(url, {

method: 'POST',

headers: {

'Content-Type': 'application/json;charset=utf-8'

},

body: JSON.stringify(data)

});

this.request();

}}

>Отправить текущую дату</button>

<div>

{

this.state.dates.map(date => {

return(

<>

<p

key={date.id}

>{date.date}</p>

<button

onClick={() => {

let url = "http://localhost:3000/dates/"+date.id;

let data = {

date: new Date()

}

fetch(url, {

method: 'PUT',

headers: {

'Content-Type': 'application/json;charset=utf-8'

},

body: JSON.stringify(data)

});

this.request();

}}

>Обновить</button>

<button

onClick={() => {

fetch(this.state.url+"/"+date.id, {

method: 'DELETE',

});

this.request();

}}

>

Удалить

</button>

</>

)

})

}

</div>

</div>

);

}

}

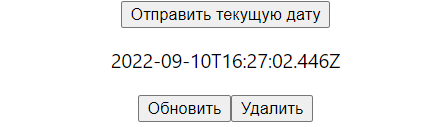
export default But;

Если в БД нечего не хранится то изначально на экране будет только одна кнопка для отпраки даты на сервер.



**Рисунок 1 –** Начальное состояние программы

После нажатия на кнопку на сервер отправляется текущая дата, состояние экрана показана на рисунке 2.



**Рисунок 2** – отправка даты.

JSON-сервер выглядит следующим образом:

**Листинг 1** -состояние сервера после отправки даты

{

"dates": [

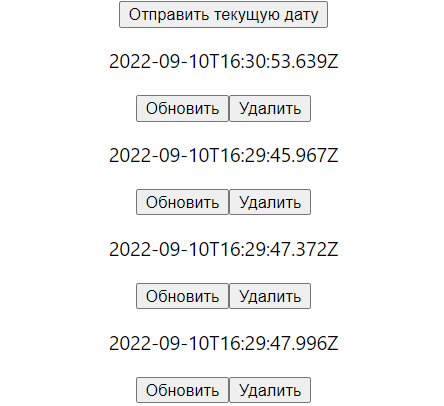
{

"date": "2022-09-10T16:27:02.446Z",

"id": 1

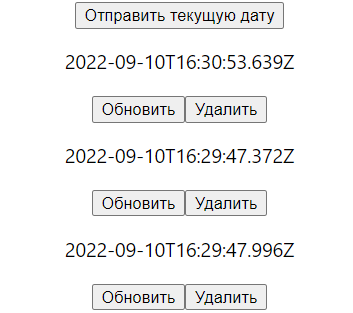
}]}

При нажатии обновить, на сервере заменяется дата на текущую и отображается обновление на экране.



**Рисунок 3** – обнавилась первая дата

При нажатии на кнопку удаления, происходит удаление этой даты на сервере и обнавление состояния экрана.



**Рисунок 4** – удалена вторая дата

**Вывод:** изучили принцип работы метода fetch(), научились применять полученные знания на практике.