



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

Отчет по лабораторной работе №7

Студент Смыслов Максим Алексеевич
фамилия, имя, отчество

Группа ИУ5-55Б

Студент 18.12.2021 Смыслов М.А.
подпись, дата фамилия, и.о.

Преподаватель 18.12.2021 Гапанюк Ю.Е.
подпись, дата фамилия, и.о.

Описание задания:

1. На основе [методических указаний](#) разработайте пользовательский интерфейс для работы с REST API. Используйте REST API, разработанный Вами в предыдущей лабораторной работе.

Текст программы:

About.js

```
import { observer } from 'mobx-react-lite'
import React from 'react'
import { Card, Container } from 'react-bootstrap'

const About = observer(() => {
  return (
    <Container>
      <Card className='align-items-center mt-3 mb-2'>

        <h1 className='mt-2'>О сервисе</h1>
        <div className='p-4 pt-0 pb-0' style={{fontSize:'20px', textAlign:'justify'}}>
          <p >
            <span style={{fontWeight:'bold'}}>Добрый день или вечер!</span> Вы попали на сайт сервиса с очередями.
            Для того, чтобы полностью пользоваться его функционалом пожалуйста зарегистрируйтесь или авторизуйтесь.
          </p>
          <h4>О функционале</h4>
          <p>
            Этот сервис поможет вам и вашим коллегам по несчастью с гордостью отстоять возникшую очередь.
            Здесь вы можете найти нужную вам очередь и встать в неё.
            Пожалуйста не забывайте выходить из неё когда закончите.
          </p>
          <h4>Примечание</h4>
          <p>
            Данный сайт пока находится в далёкой pre alpha версии,
            и, если вы нашли какие-то недочёты, обязательно свяжитесь со мной.
          </p>
          <h4>Контакты</h4>
          <p>
            Почта: <span style={{fontStyle:'italic'}}>test@mail.ru</span><br></br>
            Телефон: <span style={{fontStyle:'italic'}}>+7(999)555-55-55</span><br></br>
          </p>
        </div>
      </Card>
    </Container>
  )
})
```

Create.js

```
import { observer } from 'mobx-react-lite'
import React, {useState} from 'react'
import { Container,Form,Card, Button } from 'react-bootstrap'
import { useHistory } from 'react-router-dom'
import { createQ } from '../http/queueAPI'
import { QUEUE_ROUTE } from '../utils/consts'

const clearInput = (idElem) =>{
  try {
    document.getElementById(idElem).value = ""
  } catch (error) {
    console.log(error)
  }
}

const Create = observer(() => {

  const [name, setName] = useState('')
  const [description, setDescription] = useState('')
  const history = useHistory()

  const checkInputs = () => {

    if (name && description) return false
    else return true
  }

  const create = async () => {
    try {
      const res = await createQ(name,description)
      history.push(QUEUE_ROUTE + `/${res.id}`)
    } catch (error) {
      alert(error.response.data.message)
    }
  }

}
```

```

return (
  <Container>
    <Card className='align-items-center mt-4 p-4'>
      <Form style={{width:'75%'}}>
        <Form.Label>Название</Form.Label>
        <Form.Control
          value={name}
          onChange={str=>setName(str.target.value)}
          id='nameInput'
          style={{width:'98.5%'}}
          placeholder='Введите название'
          className='mb-3'
        />
        <Form.Label>Описание</Form.Label>
        <Form.Control
          value={description}
          onChange={str=>setDescription(str.target.value)}
          id='descriptionInput'
          style={{width:'98.5%'}}
          placeholder='Напишите описание для вашей очереди'
          as="textarea"
          className='mb-3'
        />
      <Container className='d-flex justify-content-end'>
        <Button
          variant='outline-secondary'
          className='m-2'
          onClick={() => {
            clearInput('descriptionInput')
            clearInput('nameInput')
          }}
        >
          ОЧИСТИТЬ
        </Button>
      <Button

```

Find.js

```

1      const [number, setNumber] = useState('')
2      const [foundVisible, setFoundVisible] = useState(false)
3
4      const findByName = async () => {
5          try {
6
7              searchByName(name).then(data=>{
8                  queues.setQueues(data)
9                  console.log(queues.queues)
0              })
1
2          } catch (error) {
3              alert(error.response.data.message)
4          }
5      }
6      const findByNumber = async () => {
7          try {
8
9              searchById(number).then(data=>{
0                  queues.setQueues(data)
1                  console.log(queues.queues)
2              })
3
4          } catch (error) {
5              alert(error.response.data.message)
6          }
7      }
8
9
0      return (
1          <Container >
2              <Card className='align-items-center mt-4 p-4'>
3                  <h4 className='ml-0'>Поиск по номеру очереди</h4>
4                  <Form style={{width:'75%'}}>
5                      <Form.Control
6                          id='number'
7                          value={number}
8                          onChange={str=>setNumber(str.target.value)}
9                          placeholder='Введите номер'
0                          style={{width:'98.5%'}}
1                      />
2                  <Container className='d-flex justify-content-end'>
3                      <Button

```

MyQs.js

```

1
2
3
4 const MyQs = observer(() => {
5
6
7     const history = useHistory()
8     const {queues,user} = useContext(Context)
9     useEffect(() => {
10         getQs(user.user.id).then((data)=>{
11             try {
12                 queues.setQueues(data[0].queues)
13                 console.log(queues.queues)
14             } catch (error) {
15
16             }
17
18         })
19     }, [])
20
21     return (
22         <Container>
23             <Card className='p-4 pt-2 mt-3'>
24                 <h1>Мои очереди</h1>
25                 {queues.queues.length > 0? queues.queues.map(data=>{
26                     return(
27                         <Card
28                             style={{cursor: 'pointer'}}
29                             className='m-1'
30                             key={data.id}
31                             onClick={()=>history.push(Queue_ROUTE+ '/' + data.id)}
32                         >
33                             <h4 className='m-2 mt-0'>{data.name}</h4>
34                             <h5 className='m-2 mt-0' style={{font: 'italic 20px Arial', opacity: '0.75'}}>{data.description}</h5>
35                         </Card>
36                     )
37                 }): <h3>У тебя нет очередей, попробуй
38                 <Button
39                     onClick={() => history.push(FIND_ROUTE)}
40                 >поискать</Button>
41                 </h3>}
42             </Card>
43
44         </Container>
45     )
46 })
47
48 export default MyQs

```

Profile.js

```
import React, {useContext, useEffect, useState} from 'react'
import { Card, Container } from 'react-bootstrap'
import { useParams } from 'react-router-dom/cjs/react-router-dom.min'
import { Context } from '..'
import { getUser } from '../http/userAPI'

const Profile = () => {
  const {id} = useParams()
  const {curUsers} = useContext(Context)
  const [username, setUsername] = useState()

  useEffect(() => {
    getUser(id).then((data)=>{
      curUsers.setUsers(data)
      setUsername(curUsers.users.username)
      console.log(curUsers.users.username)
    })
  }, [curUsers, id])
  return (
    <Container>
      <Card className='mt-3 p-2'>
        <h1>Профиль пользователя</h1>
        <h2>{username}</h2>

      </Card>
    </Container>
  )
}
```

export default Profile

Queue.js

```

const Queue = observer(() => {

  const [nameQueue, setNameQueue] = useState('')
  const [descQueue, setDescQueue] = useState('')
  const {user, curUsers} = useContext(Context)
  const {id} = useParams()
  const history = useHistory()
  const update = async () => {
    let response = await fetch(process.env.REACT_APP_API_URL + 'api/queue?id='+id)

    let parse = await response.json()
    try {
      console.log(parse)
      if(!parse.message){
        document.getElementById('nobody').innerHTML = ''
        if(!(parse[0].users.length === curUsers.users.length) || !(parse[0].users.every(function(element, index) {
          return curUsers.users.filter((data)=>{
            return data.id === element.id
          }).length > 0
        })))) {
          clearInterval(timerId);
          curUsers.setUsers(parse[0].users)
        }
      } else {
        document.getElementById('nobody').innerHTML = "В очереди никого нет, станьте первым!"
        clearInterval(timerId);
        curUsers.setUsers([])
      }
    } catch (error) {
      return
    }
  }

}

const getIn = async () => {
  try {
    const res = await addToQ(user.user.id, id)
    await update()
    return res
  } catch (error) {
    alert(error.response.data.message)
  }
}

```

```

<Row>
  <Col className='mt-4' sm={9}>
    <Card className='p-4'>
      <h2 >{nameQueue}</h2>
      <h3>Уникальный номер: {id}</h3>
      {descQueue? <p style={{font:'italic 20px Arial', opacity:'0.75'}}>{descQueue}</p> : null}

      {curUsers.users? curUsers.users.map((data,index)=>{
        return(
          <Card
            onClick={()=>history.push(PROFILE_ROUTE+'/'+data.id)}
            key={index}
            style={{height:'50px', background:'#212529', color:'white', textAlign:'center', fontSize:'20px', border:'2px solid white', cursor:'pointer'}}
            >{` ${index+1}. ${data.username}`}</Card>
          )
        }) : null}
      <h4 id='nobody'></h4>
    </Card>
  </Col>
  <Col className='mt-4' sm={3}>
    <Card className='p-2'>
      {{{() =>{
        if(!user.isAuthenticated) return <h4>Вы не можете встать в очередь, пожалуйста авторизуйтесь</h4>
        else if(curUsers.users.filter((data)=>{
          return data.id === user.user.id
        })).length > 0)
          return(
            <Button
              variant='outline-danger'
              onClick={getOut}
            >
              Выйти
            </Button>
          )
        else return(
          <Button
            variant='outline-success'
            onClick={getIn}
          >
            Встать в очередь
          </Button>
        )
      }}}
    </Card>
  </Col>

```

userAPI.js

```
import { $authHost, $host } from "../index"
```

```
export const getUsers = async (id) => {
  const {data} = await $host.get('api/queue',{
    params:{
      id
    }
  })
  return data
}

export const searchById = async (id) => {
  const {data} = await $host.get('api/queue/searchById',{
    params:{
      id
    }
  })
  return data
}

export const searchByName = async (name) => {
  const {data} = await $host.get('api/queue/searchByName',{
    params:{
      name
    }
  })
  return data
}

export const addToQ = async (userId, queueId) => {
  const {data} = await $authHost.post('api/queue/addToQ',{
    userId,
    queueId
  })
  return data
}

export const createQ = async (name, description) => {
```

queueAPI.js

```
import { $authHost, $host } from "../index"

export const getUsers = async (id) => {
  const {data} = await $host.get('api/queue',{
    params:{
      id
    }
  })
  return data
}

export const searchById = async (id) => {
  const {data} = await $host.get('api/queue/searchById',{
    params:{
      id
    }
  })
  return data
}

export const searchByName = async (name) => {
  const {data} = await $host.get('api/queue/searchByName',{
    params:{
      name
    }
  })
  return data
}

export const addToQ = async (userId, queueId) => {
  const {data} = await $authHost.post('api/queue/addToQ',{
    userId,
    queueId
  })
  return data
}

export const createQ = async (name, description) => {
```

Новая очередь

Уникальный номер: 6

"Описание"

В очереди никого нет, станьте первым!

Встать в очередь

Очередь

admin

testtest

Выйти