권채림 simple ajax 연습문제

> 2022-11-04

문제1

Professor

```
import React, { useCallback } from 'react';
import axios from 'axios';
import Spinner from '../components/Spinner';
import {useLocation, useNavigate} from 'react-router-dom';



const Professor = () => {

 const [professor, setProfessor] = React.useState([]);
 const [loading, setLoading] = React.useState(false);
 const [updateId, setUpdateId] = React.useState(-1);

 const {search} = useLocation();
 const query = new URLSearchParams(search);
 const {keyword} = Object.fromEntries(query);
 const navigate = useNavigate();

 React.useEffect(()=>{
   (async () => {
     setLoading(true);

     let json = null;

     try {
       const response = await axios.get('/professor', {
         params: keyword ? {name: keyword} : null
       })
       json = response.data;
       console.log("@@@@@@useEffect")
     } catch(e) {
       console.error(e);
       alert('Ajax 연동실패')
     } finally {
       setLoading(false)
     }

     setProfessor(json);
   })();
 }, [keyword])

 const onSearchSubmit = useCallback((e) => {
```

```
    e.preventDefault();
    console.log('submit');
    navigate(`/professor?keyword=${e.currentTarget.keyword.value}`)

  },[navigate])

  const onDataAddSubmit = useCallback((e)=> {
    e.preventDefault();

    const form = e.currentTarget;
    const name = form.name.value;
    const userid = form.userid.value;
    const position = form.position.value;
    const sal = form.sal.value;
    const hiredate = form.hiredate.value;
    const comm = form.comm.value;
    const deptno = form.deptno.value;

    (async ()=> {
      setLoading(true);

      let json = null;

      try {
        const response = await axios.post("/professor", {
          name: name,
          userid : userid,
          position : position,
          sal: sal,
          hiredate: hiredate,
          comm: comm,
          deptno: deptno
        })
        json = response.data;

        console.group('데이터 저장 결과')
                console.log(json)
                console.groupEnd();
      } catch(e) {
        console.error(e);
        alert(`데이터 저장에 실패했습니다. \n${e.message}`)
      } finally {
        setLoading(loading => false);
      }

      setProfessor(professor => professor.concat(json));
      form.reset();
    })();
  }, [])

  //데이터 수정
  const onDataEditClick = useCallback((e)=> {
    e.preventDefault();
    const current =  e.currentTarget;
```

```
    const id = parseInt(current.dataset.id);
    setUpdateId(id);
    console.log("@@@@edit click");
  }, [])

  const onDataEditSubmit = useCallback((e)=> {
    e.preventDefault();

    const current = e.target;
    const id = current.id.value;
    const name = current.name.value;
    const userid = current.userid.value;
    const position = current.position.value;
    const sal = current.sal.value;
    const hiredate = current.hiredate.value;
    const comm = current.comm.value;
    const deptno = current.deptno.value;

    console.log('@@@current id', id);

    (async ()=>{
        let json = null;
        setLoading(true);
        try {
          const response = await axios.put(`/professor/${id}`,{
            name: name,
            userid : userid,
            position : position,
            sal: sal,
            hiredate: hiredate,
            comm: comm,
            deptno: deptno
          })
          json = response.data;

          console.group('데이터 수정 결과')
          console.log(json)
          console.groupEnd();
        } catch(e) {
            console.error(e);
            alert(`데이터 수정에 실패했습니다. \n${e.message}`)
        } finally {
            setLoading(false)
        }

        setProfessor((professor) => {
          const editId = professor.findIndex((v, i) => v.id === json.id);
          console.log(`제거할 대상의 배열 인덱스: ${editId}`);

          professor.splice(editId, 1, json);
          return professor;
        })
    })();
```

```
    setUpdateId(-1);

  }, [])

  const onDataDeleteClick = useCallback((e)=> {
    e.preventDefault();

    const current = e.currentTarget;
    const id = parseInt(current.dataset.id);
    console.log(`삭제 대상의 id값:${id}`);

    (async ()=>{
      setLoading(true);

      try {
        await axios.delete(`/professor/${id}`)

    } catch(e) {
        console.error(e);
        alert(`데이터 삭제에 실패했습니다. \n${e.message}`)
        return;
    } finally {
        setLoading(loading => false)
    }

    setProfessor((professor) => {
      const dropId = professor.findIndex((v,i) => {
        return v.id === id;
      })
      console.log(`제거할 대상의 배열 인덱스 ${dropId}`);
      professor.splice(dropId, 1);

      return professor;
    })
    })();

  }, [])

  return (
    <div>
      <Spinner loading={loading}/>
      {/* 데이터 추가 */}
      <form onSubmit={onDataAddSubmit}>
        <div>
          <label htmlFor='name'>name :</label>
          <input type='text' name='name' id='name'/>
        </div>
        <div>
          <label htmlFor='userid'>userid :</label>
          <input type='text' name='userid' id='userid'/>
        </div>
        <div>
          <label htmlFor='position'>position :</label>
          <input type='text' name='position' id='position'/>
```

```
        </div>
        <div>
          <label htmlFor='sal'>sal :</label>
          <input type='text' name='sal' id='sal'/>
        </div>
        <div>
          <label htmlFor='hiredate'>hiredate :</label>
          <input type='text' name='hiredate' id='hiredate'/>
        </div>
        <div>
          <label htmlFor='comm'>comm :</label>
          <input type='text' name='comm' id='comm'/>
        </div>
        <div>
          <label htmlFor='deptno'>deptno :</label>
          <input type='text' name='deptno' id='deptno'/>
        </div>
        <button type='submit'>저장하기</button>
      </form>
      <hr/>
      {/* 데이터 검색 */}
      <form onSubmit={onSearchSubmit}>
          <input type='text' name='keyword'/>
          <button type='submit'>검색</button>
      </form>
      <hr/>

      <form onSubmit={onDataEditSubmit}>
        <table border='1'>
          <thead>
            <tr>
              <th>id</th>
              <th>name</th>
              <th>userid</th>
              <th>position</th>
              <th>sal</th>
              <th>hiredate</th>
              <th>comm</th>
              <th>deptno</th>
              <th>수정</th>
              <th>삭제</th>
            </tr>
          </thead>
          <tbody>
            {!professor.length ? (
              <tr>
                <td colSpan='10' align='center'>검색결과가 없습니다.</td>
              </tr>
            ) : (
              professor.map((item, index) => {
                if(item.id === updateId) {
                  return (
                    <tr key ={item.id}>
                      <input type='hidden' name='id' defaultValue={item.id}/>
```

```
                    <td>{item.id}</td>
                    <td><input type='text' name='name' defaultValue={item.name}/>
</td>
                    <td><input type='text' name='userid' defaultValue=
{item.userid}/></td>
                    <td><input type='text' name='position' defaultValue=
{item.position}/></td>
                    <td><input type='text' name='sal' defaultValue={item.sal}/>
</td>
                    <td><input type='text' name='hiredate' defaultValue=
{item.hiredate}/></td>
                    <td><input type='text' name='comm' defaultValue={item.comm}/>
</td>
                    <td><input type='text' name='deptno' defaultValue=
{item.deptno}/></td>
                    <td colSpan='2'>
                      <button type="submit">수정사항 저장</button>
                    </td>
                  </tr>
                )
              } else {
                return (
                  <tr key={item.id}>
                    <td>{item.id}</td>
                    <td>{item.name}</td>
                    <td>{item.userid}</td>
                    <td>{item.position}</td>
                    <td>{item.sal}</td>
                    <td>{item.hiredate}</td>
                    <td>{item.comm}</td>
                    <td>{item.deptno}</td>
                    <td><button type="button" data-id={item.id} onClick=
{onDataEditClick}>수정하기</button></td>
                    <td><button type="button" data-id={item.id} onClick=
{onDataDeleteClick}>삭제하기</button></td>
                  </tr>
                )
              }
            })
          )}
        </tbody>
      </table>
    </form>
  </div>
 )
}

export default Professor;
```

09-Simple-Ajax

뉴스목록 | 학과관리 | 교수관리

name :
userid :
position :
sal :
hiredate :
comm :
deptno :
저장하기

검색

id	name	userid	position	sal	hiredate	comm	deptno	수정	삭제
9901	김도훈22	capool	교수	500	1982-06-11T15:00:00.000Z	20	101	수정하기	삭제하기
9902	이재우	sweat413	조교수	320	1995-04-11T15:00:00.000Z		201	수정하기	삭제하기
9903	성연희	pascal	조교수	360	1993-03-16T15:00:00.000Z	15	101	수정하기	삭제하기
9904	염일웅	blue77	전임강사	240	1998-10-10T15:00:00.000Z		102	수정하기	삭제하기
9905	권혁일	refresh	교수	450	1986-02-10T15:00:00.000Z	25	102	수정하기	삭제하기
9906	이만식	pocari	부교수	420	1988-07-10T14:00:00.000Z		101	수정하기	삭제하기
9907	전은지	totoro	전임강사	210	2001-05-10T15:00:00.000Z		101	수정하기	삭제하기
9908	남은혁	bird13	부교수	400	1990-10-17	17	202	수정하기	삭제하기
9909	권채림333	t33	5	3	3	3	3	수정하기	삭제하기

09-Simple-Ajax

뉴스목록 | 학과관리 | 교수관리

name :
userid :
position :
sal :
hiredate :
comm :
deptno :
저장하기

검색

id	name	userid	position	sal	hiredate	comm	deptno	수정	삭제
9901	김도훈22	capool	교수	500	1982-06-11T15:00:00.000Z	20	101	수정하기	삭제하기
9902	이재우	sweat413	조교수	320	1995-04-11T15:00:00.000Z		201	수정하기	삭제하기
9903	성연희	pascal	조교수	360	1993-03-16T15:00:00.000Z	15	101	수정하기	삭제하기
9904	염일웅	blue77	전임강사	240	1998-10-10T15:00:00.000Z		102	수정하기	삭제하기
9905	권혁일	refresh	교수	450	1986-02-10T15:00:00.000Z	25	102	수정하기	삭제하기
9906	이만식	pocari	부교수	420	1988-07-10T14:00:00.000Z		101	수정하기	삭제하기
9907	전은지	totoro	전임강사	210	2001-05-10T15:00:00.000Z		101	수정하기	삭제하기
9908	남은혁	bird13	부교수	400	1990-10-17	17	202	수정하기	삭제하기
9909	권채림333	t33	5	3	3	3	3	수정사항 저장	

09-Simple-Ajax

뉴스목록 | 학과관리 | 교수관리

---

name :
userid :
position :
sal :
hiredate :
comm :
deptno :
저장하기

권채림 검색

---

id	name	userid	position	sal	hiredate	comm	deptno	수정	삭제
9909	권채림	t	5	3	3	3	3	수정하기	삭제하기