

## 박찬우 리액트 연습문제08

2022-05-16

- src/App.jsx

```
import React, { memo, useCallback, useReducer, useRef } from 'react';
import Student from './components/Student';
import Professor from './components/Professor';

// TODO: reducer 함수 정의
const reducer = (state, action) => {
  switch (action.select) {
    case '101':
      return 101;
    case '102':
      return 102;
    case '201':
      return 201;
    case '202':
      return 202;
    default:
      return -1;
  };
};

function App() {
  // -----hooks 정의 시작
  // TODO: useReducer를 활용해 초기값 지정, 함수를 통해 값을 변경
  const [mySelect, mySetSelect] = useReducer(reducer, '');

  // TODO: 값을 가져올 태그에 useRef 지정
  const select = useRef();
  // -----hooks 정의 끝

  // TODO: select 선택값 가져오기
  const resultChange = useCallback(e => {
    mySetSelect({select: select.current.value })
  }, []);

  return (
    <div>
      <h1>EXAM 09</h1>

      <hr />

      <select ref={select} onChange={resultChange}>
        <option value='-1'>학과선택</option>
        <option value='101'>컴퓨터공학과</option>
      </select>
    </div>
  );
}
```

```

        <option value='102'>멀티미디어학과</option>
        <option value='201'>전자공학과</option>
        <option value='202'>기계공학과</option>
    </select>

    <Student deptno={mySelect} />
    <Professor deptno={mySelect} />
</div>
);
}

export default memo(App);

```

- src/component/Student.jsx

```

import React, { memo, useState, useEffect } from 'react';
import axios from 'axios';
import PropTypes from 'prop-types';
import Spinner from './Spinner';

const Student = ({deptno}) => {
    // -----hooks 정의 시작
    // // TODO: 현재 ajax가 데이터를 로딩중인지를 의미하는 상태값
    const [ loading, setLoading ] = useState(false);

    // TODO: 화면에 표시할 student 상태값 정의
    const [ student, setStudent ] = useState([]);
    // -----hooks 정의 끝

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

        const getStudent = async () => {
            try {
                const response = await axios.get(`http://localhost:3001/student?deptno=${deptno}`);
                setStudent(student => response.data);
            } catch(e) {
                console.error(e);
            } finally {
                setLoading(false);
            }
        };
        getStudent();
    }, [deptno]);

    const head = ['학번', '이름', '아이디', '학년', '주민번호', '생년월일', '연락처', '키', '몸무게', '소속학과번호', '담당교수번호'];

    return (
        <div>
            <Spinner visible={loading} />

```

```

    <h2>학생목록</h2>

    <table border='1'>
      <thead>
        <tr>
          {head.map((v,i) => <th key={i}>{v}</th>)}
        </tr>
      </thead>
      <tbody align='center'>
        {student.map((v,i) => {
          return (
            <tr key={v.id}>
              <td>{v.id}</td>
              <td>{v.name}</td>
              <td>{v.userid}</td>
              <td>{v.grade}</td>
              <td>{v.idnum.substring(0,6)}-*****</td>
              <td>{new Date(v.birthdate).toLocaleDateString()}</td>
              <td>{v.tel}</td>
              <td>{v.height}</td>
              <td>{v.weight}</td>
              <td>{v.deptno}</td>
              <td>{v.profno}</td>
            </tr>
          );
        })}
      </tbody>
    </table>
  </div>
);
};

// 속성들에 대한 타입 정의
Student.propTypes = {
  deptno: propTypes.number.isRequired,
}

// 속성들에 대한 기본값 정의
Student.defaultProps = {
  deptno: -1,
}

export default memo(Student);

```

- src/component/Professor.jsx

```

import React, { memo, useState, useEffect } from 'react';
import axios from 'axios';
import propTypes from 'prop-types';
import Spinner from './Spinner';

```

```

const Professor = ({deptno}) => {
  // -----hooks 정의 시작
  // TODO: 현재 ajax가 데이터를 로딩중인지를 의미하는 상태값
  const [ loading, setLoading ] = useState(false);

  // TODO: 화면에 표시할 student 상태값 정의
  const [ professor, setProfessor ] = useState([]);
  // -----hooks 정의 끝

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

    const getStudent = async () => {
      try {
        const response = await axios.get(`http://localhost:3001/professor?deptno=${deptno}`);
        setProfessor(professor => response.data);
      } catch(e) {
        console.error(e);
      } finally {
        setLoading(false);
      }
    };
    getStudent();
  }, [deptno]);

  const head = [ '교수번호', '이름', '아이디', '직급', '급여', '입사일', '보직수당', '소속학과번호' ];

  return (
    <div>
      <Spinner visible={loading} />

      <h2>교수목록</h2>

      <table border='1'>
        <thead>
          <tr>
            {head.map((v,i) => <th key={i}>{v}</th>)}
          </tr>
        </thead>
        <tbody align='center'>
          {professor.map((v,i) => {
            return (
              <tr key={v.id}>
                <td>{v.id}</td>
                <td>{v.name}</td>
                <td>{v.userid}</td>
                <td>{v.position}</td>
                <td>{v.sal}</td>
                <td>{new Date(v.hiredate).toLocaleDateString()}</td>
                <td>{v.comm}</td>
                <td>{v.deptno}</td>
              </tr>
            )
          })}
        </tbody>
      </table>
    </div>
  );
}

```

```
        </tr>
      );
    })}
  </tbody>
</table>
</div>
);
};

// 속성들에 대한 타입 정의
Professor.propTypes = {
  deptno: propTypes.number.isRequired,
}

// 속성들에 대한 기본값 정의
Professor.defaultProps = {
  deptno: -1,
}

export default memo(Professor);
```

---

## 실행결과

## EXAM 09

학과선택 ▼

### 학생목록

학번	이름	아이디	학년	주민번호	생년월일	연락처	키	몸무게	소속학과번호	담당교수번호
----	----	-----	----	------	------	-----	---	-----	--------	--------

### 교수목록

교수번호	이름	아이디	직급	급여	입사일	보직수당	소속학과번호
------	----	-----	----	----	-----	------	--------

## EXAM 09

컴퓨터공학과 ▼

### 학생목록

학번	이름	아이디	학년	주민번호	생년월일	연락처	키	몸무게	소속학과번호	담당교수번호
10101	전인하	jun123	4	790702-*****	1979. 7. 2.	051)781-2158	176	72	101	9903
10102	박미경	ansel414	1	840516-*****	1984. 5. 16.	055)261-8947	168	52	101	
10103	김영균	mandu	3	810321-*****	1981. 3. 21.	051)824-9637	170	88	101	9906
10104	지은경	gomo00	2	800412-*****	1980. 4. 12.	055)418-9627	161	42	101	9907
10105	임유진	youjin12	2	830121-*****	1983. 1. 21.	02)312-9838	171	54	101	9907
10106	서재진	seolly	1	851129-*****	1985. 11. 29.	051)239-4861	186	72	101	
10107	이광훈	huriky	4	810913-*****	1981. 9. 13.	055)736-4981	175	92	101	9903
10108	류민정	cleansky	2	810819-*****	1981. 8. 19.	055)248-3679	162	72	101	9907

### 교수목록

교수번호	이름	아이디	직급	급여	입사일	보직수당	소속학과번호
9901	김도훈	capool	교수	500	1982. 6. 12.	20	101
9903	성연희	pascal	조교수	360	1993. 3. 17.	15	101
9906	이만식	pocari	부교수	420	1988. 7. 11.		101
9907	전은지	totoro	전임강사	210	2001. 5. 11.		101

## EXAM 09

멀티미디어학과 ▼

### 학생목록

학번	이름	아이디	학년	주민번호	생년월일	연락처	키	몸무게	소속학과번호	담당교수번호
10201	김진영	simply	2	820606-*****	1982. 6. 6.	055)419-6328	164	48	102	9905
10202	오유석	yousuk	4	770912-*****	1977. 9. 12.	051)724-9618	177	92	102	9905
10203	하나리	hana1	1	850109-*****	1985. 1. 9.	055)296-3784	160	68	102	
10204	윤진욱	samba7	3	790402-*****	1979. 4. 2.	053)487-2698	171	70	102	9905

### 교수목록

교수번호	이름	아이디	직급	급여	입사일	보직수당	소속학과번호
9904	염일웅	blue77	전임강사	240	1998. 10. 11.		102
9905	권혁일	refresh	교수	450	1986. 2. 11.	25	102

# EXAM 09

전자공학과 ▾

## 학생목록

학번	이름	아이디	학년	주민번호	생년월일	연락처	키	몸무게	소속학과번호	담당교수번호
20101	이동훈	dals	1	831210-*****	1983. 12. 10.	055)426-1752	172	64	201	
20102	박동진	ping2	1	851124-*****	1985. 11. 24.	051)742-6384	182	70	201	
20103	김진경	lovely	2	830228-*****	1983. 2. 28.	052)175-3941	166	51	201	9902
20104	조명훈	rader214	1	841214-*****	1984. 12. 14.	02)785-6984	184	62	201	

## 교수목록

교수번호	이름	아이디	직급	급여	입사일	보직수당	소속학과번호
9902	이재우	sweat413	조교수	320	1995. 4. 12.		201

# EXAM 09

기계공학과 ▾

## 학생목록

학번	이름	아이디	학년	주민번호	생년월일	연락처	키	몸무게	소속학과번호	담당교수번호
----	----	-----	----	------	------	-----	---	-----	--------	--------

## 교수목록

교수번호	이름	아이디	직급	급여	입사일	보직수당	소속학과번호
9908	남은혁	bird13	부교수	400	1990. 10. 18.	17	202