

연습문제11

Name

localhost

bundle.js

ws

react_devtools_backend.js

professor

professor

favicon.ico

manifest.json

logo192.png

×

Headers

Preview

Response

Initiator

Timing

▼

[{id: 9901, name: "김도훈", userid: "capool", position: "교수", sal: 5

▶0: {id: 9901, name: "김도훈", userid: "capool", position: "교수", sa

▶1: {id: 9902, name: "이재우", userid: "sweat413", position: "조교수"

▶2: {id: 9903, name: "성연희", userid: "pascal", position: "조교수", s

▶3: {id: 9904, name: "염일웅", userid: "blue77", position: "전임강사",

▶4: {id: 9905, name: "권혁일", userid: "refresh", position: "교수", s

▶5: {id: 9906, name: "이만식", userid: "pocari", position: "부교수", s

▶6: {id: 9907, name: "전은지", userid: "totoro", position: "전임강사",

▶7: {id: 9908, name: "남은혁", userid: "bird13", position: "부교수", s

22.05.19-리엑트_교수관리_CRUD

React App

localhost:3000/add

Exam 11

이름	asdf
아이디	
직급	<input type="radio"/> 교수 <input type="radio"/> 부교수 <input type="radio"/> 조교수 <input type="radio"/> 전임강사
급여(만원)	숫자만 입력
입사일	dd/mm/yyyy
보직수당(만원)	숫자만 입력
소속학과	----- 선택하세요 -----

localhost:3000 says

이름은 한글로 입력하세요.

OK

저장하기

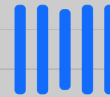
이름	asdf
아이디	asdf
직급	<input checked="" type="radio"/> 교수 <input type="radio"/> 부교수 <input type="radio"/> 조교수 <input type="radio"/> 전임강사
급여(만원)	12345
입사일	dd/mm/yyyy
보직수당(만원)	12
소속학과	<div> <div>✓ ----- 선택하세요 -----</div> <div> 컴퓨터공학과 멀티미디어학과 전자공학과 기계공학과 </div> </div>

Exam 11

localhost:3000 says

저장되었습니다.

OK



저장하기

Exam 11

교수 등록하기									
No.	이름	아이디	직급	급여	입사일	보직수당	소속학과번호	수정	삭제
9901	김도훈	capool	교수	500만원	82/06/12	20만원	101	수정하기	삭제하기
9902	이재우	sweat413	조교수	320만원	95/04/12		201	수정하기	삭제하기
9903	상연희	pascal	조교수	360만원	93/03/17	15만원	101	수정하기	삭제하기
9904	엄일용	blue77	전임강사	240만원	98/10/11		102	수정하기	삭제하기
9905	권혁일	refresh	교수	450만원	86/02/11	25만원	102	수정하기	삭제하기
9906	이만석	pocari	부교수	420만원	88/07/11		101	수정하기	삭제하기
9907	전은지	totoro	전임강사	210만원	01/05/11		101	수정하기	삭제하기
9908	남은혁	bird13	부교수	400만원	90/10/18	17만원	202	수정하기	삭제하기
9909	가나다	asdf	교수	12345만원	22/05/16	12만원	101	수정하기	삭제하기
9909	가나다	asdf	교수	12345만원	22/05/16	12만원	101	수정하기	삭제하기

Network tab showing a list of requests and their details.

Filter: ☐ Invert ☐ Hide data URLs

Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

☐ Has blocked cookies ☐ Blocked Requests ☐ 3rd-party requests

☐ Use large request rows ☐ Group by frame

☒ Show overview ☐ Capture screenshots

Timeline: 20000 ms, 40000 ms, 60000 ms, 80000 ms, 100000 ms, 120000 ms, 140000 ms, 160000 ms, 180000 ms

Request details for 'professor' (id: 9901, name: "김도훈", userid: "capool", position: "교수"):

- 0: {id: 9901, name: "김도훈", userid: "capool", position: "교수"}
- 1: {id: 9902, name: "이재우", userid: "sweat413", position: "조교수"}
- 2: {id: 9903, name: "상연희", userid: "pascal", position: "조교수"}
- 3: {id: 9904, name: "엄일용", userid: "blue77", position: "전임강사"}
- 4: {id: 9905, name: "권혁일", userid: "refresh", position: "교수"}
- 5: {id: 9906, name: "이만석", userid: "pocari", position: "부교수"}
- 6: {id: 9907, name: "전은지", userid: "totoro", position: "전임강사"}
- 7: {id: 9908, name: "남은혁", userid: "bird13", position: "부교수"}

Request details for 'professor' (id: 9905, name: "권혁일", userid: "refresh", position: "교수"):

- 4: {id: 9905, name: "권혁일", userid: "refresh", position: "교수"}
- 5: {id: 9906, name: "이만석", userid: "pocari", position: "부교수"}
- 6: {id: 9907, name: "전은지", userid: "totoro", position: "전임강사"}
- 7: {id: 9908, name: "남은혁", userid: "bird13", position: "부교수"}

localhost:3000

Exam 11

교수 등록하기

No.	이름	아이디	직급	급여	입사일	보직수당	소속학과번호	수정	삭제
9901	김도훈	capool	교수	500만원	82/06/12	20만원	101	수정하기	삭제하기
9902	이재우	sweat413	조교수	320만원	95/04/12		201	수정하기	삭제하기
9903	성연희	pascal	조교수	360만원	93/03/17	15만원	101	수정하기	삭제하기
9904	염일웅	blue77	전임강사	240만원	98/10/11		102	수정하기	삭제하기
9905	권혁일	refresh	교수	450만원	86/02/11	25만원	102	수정하기	삭제하기
9906	이만식	pocari	부교수	420만원	88/07/11		101	수정하기	삭제하기
9907	전은지	totoro	전임강사	210만원	01/05/11		101	수정하기	삭제하기
9908	남은혁	bird13	부교수	400만원	90/10/18	17만원	202	수정하기	삭제하기
9909	가나다	asdf	교수	12345만원	22/05/16	12만원	101	수정하기	삭제하기

9909 가나다 asdf 교수 12345만원 22/05/16 12만원 101 [수정하기](#) [삭제하기](#)

Network

Filter: All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

Preserve log Disable cache No throttling

Use large request rows Group by frame

Show overview Capture screenshots

Name: localhost, bundle.js, ws, professor, professor, favicon.ico, manifest.json, logo192.png

Preview: [{"id": 9901, "name": "김도훈", "userid": "capool", "position": "교수", "salary": 50000000, "hiredate": "2022-05-16", "department": 101}, {"id": 9902, "name": "이재우", "userid": "sweat413", "position": "조교수", "salary": 32000000, "hiredate": "2022-05-16", "department": 201}, {"id": 9903, "name": "성연희", "userid": "pascal", "position": "조교수", "salary": 36000000, "hiredate": "2022-05-16", "department": 101}, {"id": 9904, "name": "염일웅", "userid": "blue77", "position": "전임강사", "salary": 24000000, "hiredate": "2022-05-16", "department": 102}, {"id": 9905, "name": "권혁일", "userid": "refresh", "position": "교수", "salary": 45000000, "hiredate": "2022-05-16", "department": 102}, {"id": 9906, "name": "이만식", "userid": "pocari", "position": "부교수", "salary": 42000000, "hiredate": "2022-05-16", "department": 101}, {"id": 9907, "name": "전은지", "userid": "totoro", "position": "전임강사", "salary": 21000000, "hiredate": "2022-05-16", "department": 101}, {"id": 9908, "name": "남은혁", "userid": "bird13", "position": "부교수", "salary": 40000000, "hiredate": "2022-05-16", "department": 202}, {"id": 9909, "name": "가나다", "userid": "asdf", "position": "교수", "salary": 123450000, "hiredate": "2022-05-16", "department": 101}]]

localhost:3000/edit/9909

Exam 11

localhost:3000 says
수정되었습니다.

이름: 마바사
아이디: asdf
직급: ☒ 교수 ☐ 부교수 ☐ 조교수 ☐ 전임강사
급여(만원): 11111
입사일: 16 / 05 / 2022
보직수당(만원): 11
소속학과: 멀티미디어학과

저장하기

Network

Filter: All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

Preserve log Disable cache No throttling

Use large request rows Group by frame

Show overview Capture screenshots

Name: localhost, bundle.js, ws, professor, professor, favicon.ico, manifest.json, logo192.png, department, 9909

Preview: [{"id": 9901, "name": "김도훈", "userid": "capool", "position": "교수", "salary": 50000000, "hiredate": "2022-05-16", "department": 101}, {"id": 9902, "name": "이재우", "userid": "sweat413", "position": "조교수", "salary": 32000000, "hiredate": "2022-05-16", "department": 201}, {"id": 9903, "name": "성연희", "userid": "pascal", "position": "조교수", "salary": 36000000, "hiredate": "2022-05-16", "department": 101}, {"id": 9904, "name": "염일웅", "userid": "blue77", "position": "전임강사", "salary": 24000000, "hiredate": "2022-05-16", "department": 102}, {"id": 9905, "name": "권혁일", "userid": "refresh", "position": "교수", "salary": 45000000, "hiredate": "2022-05-16", "department": 102}, {"id": 9906, "name": "이만식", "userid": "pocari", "position": "부교수", "salary": 42000000, "hiredate": "2022-05-16", "department": 101}, {"id": 9907, "name": "전은지", "userid": "totoro", "position": "전임강사", "salary": 21000000, "hiredate": "2022-05-16", "department": 101}, {"id": 9908, "name": "남은혁", "userid": "bird13", "position": "부교수", "salary": 40000000, "hiredate": "2022-05-16", "department": 202}, {"id": 9909, "name": "가나다", "userid": "asdf", "position": "교수", "salary": 123450000, "hiredate": "2022-05-16", "department": 101}]]

localhost:3000

Exam 11

localhost:3000 says
정말 undefined 교수의 정보를 삭제하시겠습니까?

교수 등록하기

No.	이름	아이디	직급	급여	입사일	보직수당	소속학과번호	수정	삭제
9901	김도훈	capool	교수	500만원	82/06/12	20만원	101	수정하기	삭제하기
9902	이재우	sweat413	조교수	320만원	95/04/12		201	수정하기	삭제하기
9903	성연희	pascal	조교수	360만원	93/03/17	15만원	101	수정하기	삭제하기
9904	염일웅	blue77	전임강사	240만원	98/10/11		102	수정하기	삭제하기
9905	권혁일	refresh	교수	450만원	86/02/11	25만원	102	수정하기	삭제하기
9906	이만식	pocari	부교수	420만원	88/07/11		101	수정하기	삭제하기
9907	전은지	totoro	전임강사	210만원	01/05/11		101	수정하기	삭제하기
9908	남은혁	bird13	부교수	400만원	90/10/18	17만원	202	수정하기	삭제하기
9909	마바사	asdf	교수	11111만원	22/05/16	11만원	102	수정하기	삭제하기

9909 마바사 asdf 교수 11111만원 22/05/16 11만원 102 [수정하기](#) [삭제하기](#)

Network

Filter: All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

Preserve log Disable cache No throttling

Use large request rows Group by frame

Show overview Capture screenshots

Name: localhost, bundle.js, ws, professor, professor, favicon.ico, manifest.json, logo192.png

Preview: [{"id": 9901, "name": "김도훈", "userid": "capool", "position": "교수", "salary": 50000000, "hiredate": "2022-05-16", "department": 101}, {"id": 9902, "name": "이재우", "userid": "sweat413", "position": "조교수", "salary": 32000000, "hiredate": "2022-05-16", "department": 201}, {"id": 9903, "name": "성연희", "userid": "pascal", "position": "조교수", "salary": 36000000, "hiredate": "2022-05-16", "department": 101}, {"id": 9904, "name": "염일웅", "userid": "blue77", "position": "전임강사", "salary": 24000000, "hiredate": "2022-05-16", "department": 102}, {"id": 9905, "name": "권혁일", "userid": "refresh", "position": "교수", "salary": 45000000, "hiredate": "2022-05-16", "department": 102}, {"id": 9906, "name": "이만식", "userid": "pocari", "position": "부교수", "salary": 42000000, "hiredate": "2022-05-16", "department": 101}, {"id": 9907, "name": "전은지", "userid": "totoro", "position": "전임강사", "salary": 21000000, "hiredate": "2022-05-16", "department": 101}, {"id": 9908, "name": "남은혁", "userid": "bird13", "position": "부교수", "salary": 40000000, "hiredate": "2022-05-16", "department": 202}, {"id": 9909, "name": "마바사", "userid": "asdf", "position": "교수", "salary": 111110000, "hiredate": "2022-05-16", "department": 102}]]

Exam 11

교수 등록하기

No.	이름	아이디	직급	급여	입사일	보직수당	소속학과번호	수정하기	삭제하기
9901	김도훈	capool	교수	500만원	82/06/12	20만원	101	수정하기	삭제하기
9902	이재우	sweat413	조교수	320만원	95/04/12		201	수정하기	삭제하기
9903	성연희	pascal	조교수	360만원	93/03/17	15만원	101	수정하기	삭제하기
9904	염일웅	blue77	전임강사	240만원	98/10/11		102	수정하기	삭제하기
9905	권혁일	refresh	교수	450만원	86/02/11	25만원	102	수정하기	삭제하기
9906	이만식	pocari	부교수	420만원	88/07/11		101	수정하기	삭제하기
9907	전은지	totoro	전임강사	210만원	01/05/11		101	수정하기	삭제하기
9908	남은혁	bird13	부교수	400만원	90/10/18	17만원	202	수정하기	삭제하기
9909	마바사	asdf	교수	11111만원	22/05/16	11만원	102	수정하기	삭제하기

localhost:3000 says

정말 마바사 교수의 정보를 삭제하시겠습니까?

Cancel

OK

Network Performance Memory Application >> 1

ble cache No throttling

data URLs

c WS Wasm Manifest Other Has blocked cookies Blocked Requests

Use large request rows Show overview Group by frame Capture screenshots

200 ms 400 ms 600 ms 800 ms 1000 ms 1200 ms 1400 ms 1600 ms

Name Headers Preview Response Initiator Timing

localhost bundle.js ws professor professor favicon.ico manifest.json logo192.png

▼ [{id: 9901, name: "김도훈", userid: "capool", position: "교수", sal: 500,...}]

▶ 0: {id: 9901, name: "김도훈", userid: "capool", position: "교수", sal: 500,...}

▶ 1: {id: 9902, name: "이재우", userid: "sweat413", position: "조교수", sal: 320,...}

▶ 2: {id: 9903, name: "성연희", userid: "pascal", position: "조교수", sal: 360,...}

▶ 3: {id: 9904, name: "염일웅", userid: "blue77", position: "전임강사", sal: 240,...}

▶ 4: {id: 9905, name: "권혁일", userid: "refresh", position: "교수", sal: 450,...}

▶ 5: {id: 9906, name: "이만식", userid: "pocari", position: "부교수", sal: 420,...}

▶ 6: {id: 9907, name: "전은지", userid: "totoro", position: "전임강사", sal: 210,...}

▶ 7: {id: 9908, name: "남은혁", userid: "bird13", position: "부교수", sal: 400,...}

▶ 8: {name: "마바사", userid: "asdf", position: "교수", sal: 11111, hiredate: "2022-05-1

Exam 11

교수 등록하기

No.	이름	아이디	직급	급여	입사일	보직수당	소속학과번호	수정하기	삭제하기
9901	김도훈	capool	교수	500만원	82/06/12	20만원	101	수정하기	삭제하기
9902	이재우	sweat413	조교수	320만원	95/04/12		201	수정하기	삭제하기
9903	성연희	pascal	조교수	360만원	93/03/17	15만원	101	수정하기	삭제하기
9904	염일웅	blue77	전임강사	240만원	98/10/11		102	수정하기	삭제하기
9905	권혁일	refresh	교수	450만원	86/02/11	25만원	102	수정하기	삭제하기
9906	이만식	pocari	부교수	420만원	88/07/11		101	수정하기	삭제하기
9907	전은지	totoro	전임강사	210만원	01/05/11		101	수정하기	삭제하기
9908	남은혁	bird13	부교수	400만원	90/10/18	17만원	202	수정하기	삭제하기
9909	마바사	asdf	교수	11111만원	22/05/16	11만원	102	수정하기	삭제하기

localhost:3000 says

정말 마바사 교수의 정보를 삭제하시겠습니까?

Cancel

OK

Network

ble cache

data URLs

c WS Wa

Use large request rows Show overview

200 ms 400 ms 600 ms

Name Headers Preview

localhost bundle.js ws professor professor favicon.ico manifest.json logo192.png

▼ [{id: 9901, name: "김도훈", userid: "capool", position: "교수", sal: 500,...}]

▶ 0: {id: 9901, name: "김도훈", userid: "capool", position: "교수", sal: 500,...}

▶ 1: {id: 9902, name: "이재우", userid: "sweat413", position: "조교수", sal: 320,...}

▶ 2: {id: 9903, name: "성연희", userid: "pascal", position: "조교수", sal: 360,...}

▶ 3: {id: 9904, name: "염일웅", userid: "blue77", position: "전임강사", sal: 240,...}

▶ 4: {id: 9905, name: "권혁일", userid: "refresh", position: "교수", sal: 450,...}

▶ 5: {id: 9906, name: "이만식", userid: "pocari", position: "부교수", sal: 420,...}

▶ 6: {id: 9907, name: "전은지", userid: "totoro", position: "전임강사", sal: 210,...}

▶ 7: {id: 9908, name: "남은혁", userid: "bird13", position: "부교수", sal: 400,...}

▶ 8: {name: "마바사", userid: "asdf", position: "교수", sal: 11111, hiredate: "2022-05-1

9908	남은혁	bird13	부교수	400만원	90/10/18	17만원	202	수정하기	삭제하기
9907	전은지	totoro	전임강사	210만원	01/05/11		101	수정하기	삭제하기
9908	남은혁	bird13	부교수	400만원	90/10/18	17만원	202	수정하기	삭제하기

professor favicon.ico manifest.j... logo192.p... 9909

▶ 3: {id: 9904, name: "염일웅", userid: "blue77", position: "전임강사", sal: 240,...}

▶ 4: {id: 9905, name: "권혁일", userid: "refresh", position: "교수", sal: 450,...}

▶ 5: {id: 9906, name: "이만식", userid: "pocari", position: "부교수", sal: 420,...}

▶ 6: {id: 9907, name: "전은지", userid: "totoro", position: "전임강사", sal: 210,...}

▶ 7: {id: 9908, name: "남은혁", userid: "bird13", position: "부교수", sal: 400,...}

▶ 8: {name: "마바사", userid: "asdf", position: "교수", sal: 11111, hiredate: "2022-05-15T15:00:00.000Z",...}

src/pages/ProfessorList.js

```
import React from 'react';
import useAxios from 'axios-hooks';
import styled from 'styled-components';
import { NavLink } from 'react-router-dom';
import dayjs from 'dayjs';
import Spinner from '../components/Spinner';
import Table from '../components/Table';
const LinkContainer = styled.div`
  position: sticky;
  top: 0;
  background-color: #fff;
  border-top: 1px solid #eee;
  border-bottom: 1px solid #eee;
  padding: 10px 0;
`;
```

```

const TopLink = styled(NavLink)`
  margin-right: 15px;
  display: inline-block;
  font-size: 16px;
  padding: 7px 10px 5px 10px;
  border: 1px solid #ccc;
  background-color: #fff;
  color: #000;
  text-decoration: none;
  &:hover {
    background-color: #06f2;
  }
`;

const ProfessorList = () => {
  const [professor, setProfessor] = React.useState([]);
  const [{ data, loading1, error }] = useAxios(
    'http://localhost:3001/professor',
    {
      useCache: false,
    }
  );

  React.useEffect(() => {

    setProfessor(data);
  }, [data]);
  const [{ loading2 }, sendDelete] = useAxios(
    {
      method: 'DELETE',
    },
    {
      useCache: false,
      manual: true,
    }
  );

  const onDeleteClick = (e) => {
    e.preventDefault();
    const current = e.target;
    const id = parseInt(current.dataset.id);
    const name = current.dataset.name;
    if (window.confirm(`정말 ${name} 교수의 정보를 삭제하시겠습니까?`)) {
      (async () => {
        let json = null;
        try {
          const response = await sendDelete({
            method: 'DELETE',
            url: `http://localhost:3001/professor/${id}`,
          });
          json = response.data;
        } catch (e) {
          console.error(e);
          window.alert(`[${e.response.status}]
${e.response.statusText}\n${e.message}`);
        }
      })
    }
  }
}

```

```

        if (json != null) {
            setProfessor((professor) => professor.filter((v, i) => v.id !==
id));
        }
    })();
}
};
return (
    <div>
        <Spinner visible={loading1 || loading2} />
        <LinkContainer>
            <TopLink to="add">교수 등록하기</TopLink>
        </LinkContainer>
        {error ? (
            <div>
                <h1>Oops~!!! {error.code} Error.</h1>
                <hr />
                <p>{error.message}</p>
            </div>
        ) : (
            <Table>
                <thead>
                    <tr>
                        <th>No.</th>
                        <th>이름</th>
                        <th>아이디</th>
                        <th>직급</th>
                        <th>급여</th>
                        <th>입사일</th>
                        <th>보직수당</th>
                        <th>소속학과번호</th>
                        <th>수정</th>
                        <th>삭제</th>
                    </tr>
                </thead>
                <tbody>
                    {professor &&
                    professor.map(
                        (
                            { id, name, userid, position, sal, hiredate, comm,
deptno },
                            i
                        ) => {
                            return (
                                <tr key={id}>
                                    <td>{id}</td>
                                    <td>{name}</td>
                                    <td>{userid}</td>
                                    <td>{position}</td>
                                    <td>{sal}<td>{dayjs(hiredate).format('YY/MM/DD')}</td>
                                    <td>{comm && `${comm}만원`}</td>
                                    <td>{deptno}</td>
                                    <td>

```

```

        <NavLink to={`edit/${id}`}>수정하기</NavLink>
      </td>
      <td>
        <a
          href="#"
          data-id={id}
          data-name={name}
          onClick={onDeleteClick}
        >
          삭제하기
        </a>
      </td>
    </tr>
  );
}
})
</tbody>
</Table>
)}
</div>
);
};
export default React.memo(ProfessorList);

```

src/pages/ProfessorAdd.js

```

import React from 'react';
import useAxios from 'axios-hooks';
import styled from 'styled-components';
import { useNavigate } from 'react-router-dom';
import dayjs from 'dayjs';
import Spinner from '../components/Spinner';
import Table from '../components/Table';
import regexHelper from '../libs/RegexHelper';
const TableEx = styled(Table)`
  margin-top: 50px;
  margin-bottom: 15px;
  .inputWrapper {
    padding: 0;
    position: relative;
    text-align: left;
    .field {
      box-sizing: border-box;
      display: block;
      position: absolute;
      left: 0;
      top: 0;
      width: 100%;
      height: 100%;
      border: 0;

```

```

padding: 0 10px;
outline: none;
font-size: 14px;
}
label {
margin-left: 7px;
margin-right: 10px;
input {
margin-right: 10px;
}
}
}
};
const ProfessorAdd = () => {
const navigate = useNavigate();
const [{ data, loading: loading1, error }] = useAxios(
'http://localhost:3001/department'
);
const [{ loading: loading2 }, refetch] = useAxios(
{
url: 'http://localhost:3001/professor',
method: 'POST',
},
{ manual: true }
);
const onSubmit = React.useCallback(
(e) => {
e.preventDefault();
const current = e.target;
try {
regexHelper.value(current.name, '이름을 입력하세요. ');
regexHelper.kor(current.name, '이름은 한글로 입력하세요. ');
regexHelper.minLength(
current.name,
2,
'이름은 최소 2글자 이상 입력해야 합니다.'
);
regexHelper.maxLength(
current.name,
10,
'이름은 최대 10글자 까지 입력가능합니다.'
);
regexHelper.value(current.userid, '아이디를 입력하세요. ');
regexHelper.engNum(
current.userid,
'아이디는 영문과 숫자 조합으로만 입력 가능합니다.'
);
regexHelper.minLength(
current.userid,
2,
'아이디는 최소 2글자 이상 입력해야 합니다.'
);
regexHelper.maxLength(
current.userid,

```



```

        20,
        '아이디는 최대 20글자 까지입력 가능합니다.'
    );
    regexHelper.check(current.position, '직급을 선택하세요. ');
    regexHelper.value(current.sal, '급여를 입력하세요. ');
    regexHelper.num(current.sal, '급여는 숫자만 입력 가능합니다. ');
    regexHelper.value(current.hiredate, '입사일을 입력하세요. ');
    regexHelper.value(current.comm, '보직수당을 입력하세요. ');
    regexHelper.num(current.comm, '보직수당은 숫자만 입력 가능합니다. ');
    regexHelper.value(current.deptno, '소속학과를 선택하세요. ');
} catch (e) {
    window.alert(e.message);
    e.field.focus();
    return;
}
let json = null;
// 입력, 수정, 삭제 처리는 async~await 문법을 사용해야 한다.
(async () => {
    try {
        const response = await refetch({
            data: {
                name: current.name.value,
                userid: current.userid.value,
                position: current.position.value,
                sal: parseInt(current.sal.value),
                hiredate: dayjs(current.hiredate.value).toISOString(),
                comm: parseInt(current.comm.value),
                deptno: parseInt(current.deptno.value),
            },
        });
        json = response.data;
    } catch (e) {
        console.error(e);
        window.alert(`[${e.response.status}]
${e.response.statusText}\n${e.message}`);
    }
    if (json !== null) {
        window.alert('저장되었습니다. ');
        navigate('/');
    }
    })();
},
[refetch, navigate]
);
return (
    <>
    <Spinner visible={loading1 || loading2} />
    {error ? (
        <div>
            <h1>Oops~!!! {error.code} Error.</h1>
            <hr />
            <p>{error.message}</p>
        </div>
    ) : (

```

```

<form onSubmit={onSubmit}>
  <TableEx>
    <colgroup>
      <col width="120" />
      <col />
    </colgroup>
    <tbody>
      <tr>
        <th>이름</th>
        <td className="inputWrapper">
          <input className="field" type="text" name="name" />
        </td>
      </tr>
      <tr>
        <th>아이디</th>
        <td className="inputWrapper">
          <input className="field" type="text" name="userid" />
        </td>
      </tr>
      <tr>
        <th>직급</th>
        <td className="inputWrapper">
          <label>
            <input type="radio" name="position" value="교수" />
            교수
          </label>
          <label>
            <input type="radio" name="position" value="부교수" />
            부교수
          </label>
          <label>
            <input type="radio" name="position" value="조교수" />
            조교수
          </label>
          <label>
            <input type="radio" name="position" value="전임강사" />
            전임강사
          </label>
        </td>
      </tr>
      <tr>
        <th>급여(만원)</th>
        <td className="inputWrapper">
          <input
            className="field"
            type="number"
            name="sal"
            placeholder="숫자만 입력"
          />
        </td>
      </tr>
      <tr>
        <th>입사일</th>
        <td className="inputWrapper">

```

```

        <input className="field" type="date" name="hiredate" />
      </td>
    </tr>
    <tr>
      <th>보직수당(만원)</th>
      <td className="inputWrapper">
        <input
          className="field"
          type="number"
          name="comm"
          placeholder="숫자만 입력"
        />
      </td>
    </tr>
    <tr>
      <th>소속학과</th>
      <td className="inputWrapper">
        <select name="deptno" className="field">
          <option value="">----- 선택하세요 -----</option>
          {data &&
            data.map((v, i) => {
              return (
                <option key={i} value={v.id}>
                  {v.dname}
                </option>
              );
            })}
          </select>
        </td>
      </tr>
    </tbody>
  </TableEx>
  <div style={{ textAlign: 'center' }}>
    <button type="submit">저장하기</button>
  </div>
</form>
  )}
</>
);
};
export default React.memo(ProfessorAdd);

```

src/pages/ProfessorEdit.js

```

import React from 'react';
import useAxios from 'axios-hooks';
import styled from 'styled-components';
import { useNavigate, useParams } from 'react-router-dom';
import dayjs from 'dayjs';
import Spinner from '../components/Spinner';

```

```

import Table from '../components/Table';
import regexHelper from '../libs/RegexHelper';
// Table 컴포넌트의 CSS를 확장한 컴포넌트
const TableEx = styled(Table)`
  margin-top: 50px;
  margin-bottom: 15px;
  .inputWrapper {
    padding: 0;
    position: relative;
    text-align: left;
    .field {
      box-sizing: border-box;
      display: block;
      position: absolute;
      left: 0;
      top: 0;
      width: 100%;
      height: 100%;
      border: 0;
      padding: 0 10px;
      outline: none;
      font-size: 14px;
    }
    label {
      margin-left: 7px;
      margin-right: 10px;
      input {
        margin-right: 10px;
      }
    }
  }
`;

const ProfessorEdit = () => {
  /** Path 파라미터로 전송된 교수 일련번호 얻기 */
  const { id } = useParams();
  /** 저장 완료 후 목록으로 되돌아가기 위한 페이지 강제 이동 함수 생성 */
  const navigate = useNavigate();
  /** 드롭다운에 표시할 학과 목록 조회 */
  const [{ data: department, loading: loading1, error: err1 }] = useAxios(
    'http://localhost:3001/department'
  );
  /** 저장되어 있는 교수 데이터 조회 */
  const [{ data: professor, loading: loading2, error: err2 }, refetch] =
    useAxios(`http://localhost:3001/professor/${id}`);
  /** <form>의 submit 버튼이 눌렸을 때 호출될 이벤트 핸들러 */
  const onSubmit = React.useCallback(
    (e) => {
      e.preventDefault();
      // 이벤트가 발생한 폼 객체
      const current = e.target;
      // 입력값에 대한 유효성 검사
      try {
        regexHelper.value(current.name, '이름을 입력하세요. ');
        regexHelper.kor(current.name, '이름은 한글로 입력하세요. ');
      }
    }
  );

```

```

    regexHelper.minLength(
        current.name,
        2,
        '이름은 최소 2글자 이상 입력해야 합니다.'
    );
    regexHelper.maxLength(
        current.name,
        10,
        '이름은 최대 10글자 까지 입력가능합니다.'
    );
    regexHelper.value(current.userid, '아이디를 입력하세요. ');
    regexHelper.engNum(
        current.userid,
        '아이디는 영문과 숫자 조합으로만 입력 가능합니다.'
    );
    regexHelper.minLength(
        current.userid,
        2,
        '아이디는 최소 2글자 이상 입력해야 합니다.'
    );
    regexHelper.maxLength(
        current.userid,
        20,
        '아이디는 최대 20글자 까지 입력 가능합니다.'
    );
    regexHelper.check(current.position, '직급을 선택하세요. ');
    regexHelper.value(current.sal, '급여를 입력하세요. ');
    regexHelper.num(current.sal, '급여는 숫자만 입력 가능합니다. ');
    regexHelper.value(current.hiredate, '입사일을 입력하세요. ');
    regexHelper.value(current.comm, '보직수당을 입력하세요. ');
    regexHelper.num(current.comm, '보직수당은 숫자만 입력 가능합니다. ');
    regexHelper.value(current.deptno, '소속학과를 선택하세요. ');
} catch (e) {
    window.alert(e.message);
    e.field.focus();
    return;
}
let json = null;
// 입력, 수정, 삭제 처리는 async~await 문법을 사용해야 한다.
(async () => {
    try {
        const response = await refetch({
            method: 'PUT',
            data: {
                name: current.name.value,
                userid: current.userid.value,
                position: current.position.value,
                sal: parseInt(current.sal.value),
                hiredate: dayjs(current.hiredate.value).toISOString(),
                comm: parseInt(current.comm.value),
                deptno: parseInt(current.deptno.value),
            },
        });
        json = response.data;
    }

```

```

    } catch (e) {
      console.error(e);
      window.alert(`[${e.response.status}]
${e.response.statusText}\n${e.message}`);
    }

    // 정상적으로 저장되어 응답을 받았다면?
    if (json !== null) {
      window.alert('수정되었습니다.');
```

// 페이지 강제 이동 (window.location.href = URL 기능과 동일함)

```

      navigate('/');
    }
  })();
},
[navigate, refetch]
);
return (
  <>
    <Spinner visible={loading1 || loading2} />
    {err1 || err2 ? (
      <div>
        <h1>Oops~!!! {(err1 || err2).code} Error.</h1>
        <hr />
        <p>{(err1 || err2).message}</p>
      </div>
    ) : (
      professor && (
        <form onSubmit={onSubmit}>
          <TableEx>
            <colgroup>
              <col width="120" />
              <col />
            </colgroup>
            <tbody>
              <tr>
                <th>이름</th>
                <td className="inputWrapper">
                  <input
                    className="field"
                    type="text"
                    name="name"
                    defaultValue={professor.name}
                  />
                </td>
              </tr>
              <tr>
                <th>아이디</th>
                <td className="inputWrapper">
                  <input
                    className="field"
                    type="text"
                    name="userid"
                    defaultValue={professor.userid}
                  />
                </td>
              </tr>
            </tbody>
          </TableEx>
        </form>
      )
    )
  )
);

```

```

        </td>
      </tr>
      <tr>
        <th>직급</th>
        <td className="inputWrapper">
          <label>
            <input
              type="radio"
              name="position"
              value="교수"
              defaultChecked={professor.position === '교수'}
            />
            교수
          </label>
          <label>
            <input
              type="radio"
              name="position"
              value="부교수"
              defaultChecked={professor.position === '부교수'}
            />
            부교수
          </label>
          <label>
            <input
              type="radio"
              name="position"
              value="조교수"
              defaultChecked={professor.position === '조교수'}
            />
            조교수
          </label>
          <label>
            <input
              type="radio"
              name="position"
              value="전임강사"
              defaultChecked={professor.position === '전임강사'}
            />
            전임강사
          </label>
        </td>
      </tr>
      <tr>
        <th>급여(만원)</th>
        <td className="inputWrapper">
          <input
            className="field"
            type="number"
            name="sal"
            placeholder="숫자만 입력"
            defaultValue={professor.sal}
          />
        </td>
      </tr>
    </tbody>
  </table>

```

```

    </tr>
    <tr>
      <th>입사일</th>
      <td className="inputWrapper">
        <input
          className="field"
          type="date"
          name="hiredate"
          defaultValue={dayjs(professor.hiredate).format(
            'YYYY-MM-DD'
          )}
        />
      </td>
    </tr>
    <tr>
      <th>보직수당 (만원)</th>
      <td className="inputWrapper">
        <input
          className="field"
          type="number"
          name="comm"
          placeholder="숫자만 입력"
          defaultValue={professor.comm}
        />
      </td>
    </tr>
    <tr>
      <th>소속학과</th>
      <td className="inputWrapper">
        <select
          name="deptno"
          className="field"
          defaultValue={professor.deptno}
        >
          <option value="">----- 선택하세요 -----</option>
          {department &&
            department.map((v, i) => {
              return (
                <option key={i} value={v.id}>
                  {v.dname}
                </option>
              );
            })}
        </select>
      </td>
    </tr>
  </tbody>
</TableEx>
<div style={{ textAlign: 'center' }}>
  <button type="submit">저장하기</button>
</div>
</form>
)
})

```



```
    </>  
  );  
};  
export default React.memo(ProfessorEdit);
```