

교수_연습문제 Restful

GET http://localhost:3000/department/102 Send 200 OK 20.8 ms 176 B

Body Auth Query Header Docs

Preview Header 13 Cookie Timeline

```

1 {
2   "name": "김교수",
3   "userid": "교수형",
4   "position": "일반교수",
5   "sal": "100원",
6   "hiredate": "2020-04-07",
7   "comm": "10원",
8   "deptno": "1",
9   "id": 102
10 }

```

Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

[Introduction to Insomnia](#)

POST http://localhost:3000/department Send 201 Created 32 ms 179 B

Form 7 Auth Query Header 1 Docs

name	박교수	<input checked="" type="checkbox"/>	<input type="checkbox"/>
userid	교수동생	<input checked="" type="checkbox"/>	<input type="checkbox"/>
position	일반교수	<input checked="" type="checkbox"/>	<input type="checkbox"/>
sal	100원	<input checked="" type="checkbox"/>	<input type="checkbox"/>
hiredate	2020-04-06	<input checked="" type="checkbox"/>	<input type="checkbox"/>
comm	10원	<input checked="" type="checkbox"/>	<input type="checkbox"/>
deptno	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
New name	New value		

Preview Header 15 Cookie Timeline

```

1 {
2   "name": "박교수",
3   "userid": "교수동생",
4   "position": "일반교수",
5   "sal": "100원",
6   "hiredate": "2020-04-06",
7   "comm": "10원",
8   "deptno": "1",
9   "id": 103
10 }

```

PUT ▼ http://localhost:3000/department/102 Send 200 OK 21.8 ms 176 B

Form 7 ▼ Auth ▼ Query Header 1 Docs

name	김교수	▼	✓	🗑
userid	교수형	▼	✓	🗑
position	일반교수	▼	✓	🗑
sal	100원	▼	✓	🗑
hiredate	2020-04-07	▼	✓	🗑
comm	10원	▼	✓	🗑
deptno	1	▼	✓	🗑
New name	New value			

Preview ▼ Header 13 Cookie Timeline

```

1 {
2   "name": "김교수",
3   "userid": "교수형",
4   "position": "일반교수",
5   "sal": "100원",
6   "hiredate": "2020-04-07",
7   "comm": "10원",
8   "deptno": "1",
9   "id": 102
10 }

```

DELETE ▼ http://localhost:3000/department/106 Send 200 OK 30.3 ms 2 B


Body ▼ Auth ▼ Query Header Docs

Preview ▼ Header 13 Cookie

```


1 {}

```



Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

[Introduction to Insomnia](#) 

내 코드

index.html

```

<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>교수 정보관리 :: MySchool</title>
    <link rel="stylesheet" href="assets/css//style.css" />
  </head>
  <body>
    <div class="container">
      <!-- header -->
      <div data-include="inc/header.html"></div>
      <!-- body -->
      <section>
        <p>
          <a href="add.html">새 교수 추가</a>
        </p>
        <table class="table">
          <colgroup>
            <col width="10%" />
            <col width="10%" />
            <col width="10%" />
            <col width="10%" />
            <col width="10%" />
            <col width="10%" />
            <col width="10%" />
            <col width="10%" />
            <col width="10%" />
          </colgroup>
          <thead>
            <tr>
              <th class="text-center">교수번호</th>
              <th class="text-center">이름</th>
              <th class="text-center">아이디</th>
              <th class="text-center">직급</th>
              <th class="text-center">급여</th>
              <th class="text-center">입사일</th>
              <th class="text-center">보직수당</th>
              <th class="text-center">부서번호</th>
              <th class="text-center">-</th>
            </tr>
          </thead>
          <tbody id="listBody"></tbody>
        </table>
      </section>
      <!-- footer -->
      <div data-include="inc/footer.html"></div>
    </div>

    <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
    <script src="assets/js/include.js"></script>
    <script>
      (async () => {

```

```
let json = null;

try {
  json = await axios.get("http://localhost:3000/department");
} catch (e) {
  console.error(e);
  alert("요청 처리 실패");
  return;
}

if (json !== null) {
  const listBody = document.querySelector("#listBody");
  const { data } = json;

  data.forEach((v, i) => {
    const tr = document.createElement("tr");

    const td1 = document.createElement("td");
    td1.innerHTML = v.id;
    tr.appendChild(td1);

    const td2 = document.createElement("td");
    tr.appendChild(td2);

    const a1 = document.createElement("a");
    a1.setAttribute("href", `view.html?id=${v.id}`);
    a1.innerHTML = v.name;
    td2.appendChild(a1);

    const td3 = document.createElement("td");
    td3.innerHTML = v.userid;
    tr.appendChild(td3);

    const td4 = document.createElement("td");
    td4.innerHTML = v.position;
    tr.appendChild(td4);

    const td5 = document.createElement("td");
    td5.innerHTML = v.sal;
    tr.appendChild(td5);

    const td6 = document.createElement("td");
    td6.innerHTML = v.hiredate;
    tr.appendChild(td6);

    const td7 = document.createElement("td");
    td7.innerHTML = v.comm;
    tr.appendChild(td7);

    const td8 = document.createElement("td");
    td8.innerHTML = v.deptno;
    tr.appendChild(td8);

    const td9 = document.createElement("td");
```

```

tr.appendChild(td9);

const a2 = document.createElement("a");
a2.setAttribute("href", `edit.html?id=${v.id}`);
a2.innerHTML = "수정";
td9.appendChild(a2);

const a3 = document.createElement("a");
a3.setAttribute("href", "#");
a3.dataset.id = v.id;
a3.dataset.name = v.name;
a3.innerHTML = "삭제";
a3.classList.add("btn-delete");
td9.appendChild(a3);

listBody.appendChild(tr);

// 삭제 버튼에 대한 클릭 이벤트 구현
a3.addEventListener("click", async (e) => {
  e.preventDefault();
  const current = e.currentTarget;
  const id = current.dataset.id;
  const name = current.dataset.name;

  if (!confirm(`정말 ${name}(을)를 삭제하시겠습니까?`)) {
    return;
  }

  let json = null;

  try {
    json = axios.delete(`http://localhost:3000/department/${id}`);
  } catch (e) {
    console.error(e);
    alert("요청 처리 실패");
    return;
  }

  if (json != null) {
    current.closest("tr").remove();
  }
});
});
})();
</script>
</body>
</html>

```

view.html

```

<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>교수 정보관리 :: MySchool</title>
  <link rel="stylesheet" href="assets/css//style.css" />
</head>
<body>
  <div class="container">
    <!-- header -->
    <div data-include="inc/header.html"></div>
    <!-- body -->
    <section>
      <dl>
        <dt>교수 번호</dt>
        <dd class="id"></dd>
        <dt>교수 이름</dt>
        <dd class="name"></dd>
        <dt>교수 아이디</dt>
        <dd class="userid"></dd>
        <dt>교수 직급</dt>
        <dd class="position"></dd>
        <dt>교수 급여</dt>
        <dd class="sal"></dd>
        <dt>교수 입사일</dt>
        <dd class="hiredate"></dd>
        <dt>교수 보직수당</dt>
        <dd class="comm"></dd>
        <dt>교수 부서번호</dt>
        <dd class="deptno"></dd>
      </dl>
      <a href="index.html">목록보기</a>
      <a href="#" class="btn-edit">수정하기</a>
      <a href="#" class="btn-delete">삭제하기</a>
    </section>
    <!-- footer -->
    <div data-include="inc/footer.html"></div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
  <script src="assets/js/include.js"></script>
  <script>
    (async () => {
      /** GET 파라미터 받기 */
      // window.location.search : "?id=101"
      console.log(`search값 => ${window.location.search}`);

      // querystring값이 key-value 형태로 객체로 분리됨
      const params = new URLSearchParams(window.location.search);

      // 객체에서 key사용하여 값을 추출 --> 101
      const id = params.get('id');
    })();
  </script>

```

```

        console.log('id');

        // 파라미터가 정상적이지 않으므로 메시지 출력, 전페이지 이동 처리 후 수행
중단(return)
        if (!id) {
            alert('교수 번호가 없습니다.');
```

history.back();

```

            return;
        }

        /** Ajax요청 */
        let json = null;

        try {
            json = await axios.get(`http://localhost:3000/department/${id}`);
        } catch (e) {
            console.error(e);
            alert('요청을 처리하는데 실패했습니다.');
```

return;

```

        }

        if (json != null) {
            const {data} = json;

            document.querySelector('.id').innerHTML = data.id;
            document.querySelector('.name').innerHTML = data.name;
            document.querySelector('.userid').innerHTML = data.userid;
            document.querySelector('.position').innerHTML = data.position;
            document.querySelector('.sal').innerHTML = data.sal;
            document.querySelector('.hiredate').innerHTML = data.hiredate;
            document.querySelector('.comm').innerHTML = data.comm;
            document.querySelector('.deptno').innerHTML = data.deptno;

            // 가져온 데이터의 일련번호를 사용하여 어떤 대상을 수정할지에 대한 링크
URL 구성
            document.querySelector('.btn-edit').setAttribute('href',
            'edit.html?id=' + data.id);

            // 가져온 데이터의 일련번호를 사용하여 어떤 대상을 삭제할지에 대한
dataset 구성
            const btnDelete = document.querySelector('.btn-delete');
            btnDelete.dataset.id = data.id;
            btnDelete.dataset.name = data.name;
        }
    })();

    /** 삭제 버튼에 대한 클릭 이벤트 */
    document.querySelector('.btn-delete').addEventListener('click', (e) => {
        e.preventDefault();

        const current = e.currentTarget;
        const id = current.dataset.id;
        const name = current.dataset.name;

```

```

        if (!confirm(`정말 ${name}(을)를 삭제하시겠습니까?`)) {
            return;
        }

        let json;

        try {
            json = axios.delete(`http://localhost:3000/department/${id}`);
        } catch (e) {
            console.error(e);
            alert('요청을 처리하는데 실패했습니다. ');
            return;
        }

        if (json != null) {
            window.location = 'index.html';
        }
    });
</script>
</body>
</html>

```

add.html

```

<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>교수 정보관리 :: MySchool</title>
    <link rel="stylesheet" href="assets/css//style.css" />
  </head>
  <body>
    <div class="container">
      <!-- header -->
      <div data-include="inc/header.html"></div>
      <!-- body -->
      <section>
        <form name="form" id="department-form">
          <div class="form-group">
            <label for="name"> 교수 이름</label>
            <div>
              <input type="text" name="name" id="name" />
            </div>
          </div>
          <div class="form-group">
            <label for="userid"> 교수 아이디</label>
            <div>
              <input type="text" name="userid" id="userid" />
            </div>
          </div>
        </form>
      </section>
    </div>
  </body>
</html>

```



```

<div class="form-group">
  <label for="position"> 교수 직급</label>
  <div>
    <input type="text" name="position" id="position" />
  </div>
</div>
<div class="form-group">
  <label for="sal"> 교수 급여</label>
  <div>
    <input type="text" name="sal" id="sal" />
  </div>
</div>
<div class="form-group">
  <label for="hiredate"> 교수 입사일</label>
  <div>
    <input type="date" name="hiredate" id="hiredate" />
  </div>
</div>
<div class="form-group">
  <label for="comm"> 교수 보직수당</label>
  <div>
    <input type="text" name="comm" id="comm" />
  </div>
</div>
<div class="form-group">
  <label for="deptno"> 교수 부서번호</label>
  <div>
    <input type="number" name="deptno" id="deptno" />
  </div>
</div>
<hr />
<div class="text-center">
  <button type="submit">저장하기</button>
  <button type="reset">다시작성</button>
</div>
</form>
</section>
<!-- footer -->
<div data-include="inc/footer.html"></div>
</div>

<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
<script src="assets/js/include.js"></script>
<script>
  document
    .querySelector("#department-form")
    .addEventListener("submit", async (e) => {
      e.preventDefault();

      // 입력값 받아오기
      const name = document.querySelector("#name").value;
      const userid = document.querySelector("#userid").value;
      const position = document.querySelector("#position").value;
      const sal = document.querySelector("#sal").value;

```

```

const hiredate = document.querySelector("#hiredate").value;
const comm = document.querySelector("#comm").value;
const deptno = document.querySelector("#deptno").value;

// 유효성 검사는 생략한다.

let json = null;

try {
  // 대부분의 backend가 일련번호는 자체적으로 생성하기 때문에 create에 해당
  하는 데이터 추가 처리에서는
  // 백엔드에게 일련번호를 제외한 변수만 전달해야 한다.
  json = await axios.post("http://localhost:3000/department", {
    name: name,
    userid: userid,
    position: position,
    sal: sal,
    hiredate: hiredate,
    comm: comm,
    deptno: deptno,
  });
} catch (e) {
  console.error(e);
  alert('요청을 처리하는데 실패했습니다.');
```

return;

```

}

if (json !== null) {
  const {data} = json;
  console.log(data);
  // 백엔드에서 데이터 저장에 대한 응답으로 보낸 JSON에는 생성된 신규 항목에
  대한 일련번호가 포함되어 있다.
  // 이 값을 활용하여 상세 페이지로 이동해야 한다.
  // 이 때, 어떤 대상을 열람할 것인지에 대한 GET 파라미터를 구성한다.
  window.location = 'view.html?id=' + data.id;
}
});
</script>
</body>
</html>

```

edit.html

```

<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>교수 정보관리 :: MySchool</title>
    <link rel="stylesheet" href="assets/css//style.css" />
  </head>

```

```

<body>
  <div class="container">
    <!-- header -->
    <div data-include="inc/header.html"></div>
    <!-- body -->
    <section>
      <form name="form" id="department-form">
        <!-- 수정할 대상에 대한 일련번호 값을 포함해야 하므로 hidden 필드로 처리한다.
-->
        <input type="hidden" name="id" id="id" />
        <div class="form-group">
          <label for="name">
            교수 이름 <span class="text-danger">*</span></label>
          <div>
            <input type="text" name="name" id="name" />
          </div>
        </div>
        <div class="form-group">
          <label for="userid"> 교수 아이디</label>
          <div>
            <input type="text" name="userid" id="userid" />
          </div>
        </div>
        <div class="form-group">
          <label for="position"> 교수 직급</label>
          <div>
            <input type="text" name="position" id="position" />
          </div>
        </div>
        <div class="form-group">
          <label for="sal"> 교수 급여</label>
          <div>
            <input type="text" name="sal" id="sal" />
          </div>
        </div>
        <div class="form-group">
          <label for="hiredate"> 교수 입사일</label>
          <div>
            <input type="date" name="hiredate" id="hiredate" />
          </div>
        </div>
        <div class="form-group">
          <label for="comm"> 교수 보직수당</label>
          <div>
            <input type="text" name="comm" id="comm" />
          </div>
        </div>
        <div class="form-group">
          <label for="deptno"> 교수 부서번호</label>
          <div>
            <input type="number" name="deptno" id="deptno" />
          </div>
        </div>
      </form>
    </section>
  </div>
</body>

```

```

        <hr />
        <div class="text-center">
            <button type="submit">수정하기</button>
            <button type="reset">다시작성</button>
        </div>
    </form>
</section>
<!-- footer -->
<div data-include="inc/footer.html"></div>
</div>

<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
<script src="assets/js/include.js"></script>
<script>
    (async () => {
        /** GET 파라미터 받기 */
        const params = new URLSearchParams(window.location.search);
        const id = params.get("id");
        console.log(id);

        // 파라미터가 정상적이지 않으므로 메시지 출력, 전페이지로 이동 처리 후 수행 중단
        (return)
        if (!id) {
            alert("교수 번호가 없습니다.");
            history.back();
            return;
        }

        /** Ajax요청 */
        // 기존에 저장되어 있는 내용을 <form>안에 셋팅하기 위해 Ajax로 데이터를 가져온
다.
        let json;

        try {
            json = await axios.get(`http://localhost:3000/department/${id}`);
        } catch (e) {
            console.error(e);
            alert("요청을 처리하는데 실패했습니다.");
            return;
        }

        // 가져온 데이터가 있다면 <input> 태그의 value 속성으로 가져온 값을 적용한다.
        // 일련번호와 같이 시스템이 관리하는 값을 hidden 요소로 포함시킨다.
        if (json != null) {
            const { data } = json;
            document.querySelector("#id").value = data.id;
            document.querySelector("#name").value = data.name;
            document.querySelector("#userid").value = data.userid;
            document.querySelector("#position").value = data.position;
            document.querySelector("#sal").value = data.sal;
            document.querySelector("#hiredate").value = data.hiredate;
            document.querySelector("#comm").value = data.comm;
            document.querySelector("#deptno").value = data.deptno;
        }
    })();

```

```
})();

/** 수정 버튼의 submit 이벤트 */
document
  .querySelector("#department-form")
  .addEventListener("submit", async (e) => {
    e.preventDefault();

    // 입력값 받아오기
    const id = document.querySelector("#id").value;
    const name = document.querySelector("#name").value;
    const userid = document.querySelector("#userid").value;
    const position = document.querySelector("#position").value;
    const sal = document.querySelector("#sal").value;
    const hiredate = document.querySelector("#hiredate").value;
    const comm = document.querySelector("#comm").value;
    const deptno = document.querySelector("#deptno").value;

    // 입력값에 대한 유효성 검사 진행.... (이 예제에서는 생략)

    let json = null;

    try {
      json = await axios.put(`http://localhost:3000/department/${id}`, {
        name: name,
        userid: userid,
        position: position,
        sal: sal,
        hiredate: hiredate,
        comm: comm,
        deptno: deptno,
      });
    } catch (e) {
      console.error(e);
      alert("요청을 처리하는데 실패했습니다.");
      return;
    }

    if (json !== null) {
      window.location.href = `view.html?id=${id}`;
    }
  });
</script>
</body>
</html>
```

구현 스크린샷

교수 정보관리

새 교수 추가

교수번호	이름	아이디	직급	급여	입사일	보직수당	부서번호	-
102	김교수	교수형	일반교수	100원	2020-04-07	10원	1	수정 삭제
103	박교수	교수동생	일반교수	100원	2020-04-06	10원	1	수정 삭제
104	이교수	교수형동생	일반교수	120원	2022-03-28	15원	1	수정 삭제
105	최교수	교수결	일반교수	110원	2022-03-30	12원	1	수정 삭제

copyright©2022. ezenac.co.kr

교수 정보관리

교수 이름 *

이교수

교수 아이디

교수보이

교수 직급

일반교수

교수 급여

150원

교수 입사일

2022. 03. 28.

교수 보직수당

15원

교수 부서번호

2

수정하기

다시작성

copyright©2022. ezenac.co.kr

교수 정보관리

- ⊙ 교수 번호

104
- ⊙ 교수 이름

이교수
- ⊙ 교수 아이디

교수보이
- ⊙ 교수 직급

일반교수
- ⊙ 교수 급여

150원
- ⊙ 교수 입사일

2022-03-28
- ⊙ 교수 보직수당

15원
- ⊙ 교수 부서번호

2

[목록보기](#) [수정하기](#) [삭제하기](#)

copyright©2022. ezenac.co.kr

교수 정보관리

127.0.0.1:5500 내용:
정말 이교수(을)를 삭제하시겠습니까?

확인

취소

[새 교수 추가](#)

교수번호	이름	아이디	직급	급여	입사일	보직수당	부서번호	-
102	김교수	교수형	일반교수	100원	2020-04-07	10원	1	수정 삭제
103	박교수	교수동생	일반교수	100원	2020-04-06	10원	1	수정 삭제
104	이교수	교수보이	일반교수	150원	2022-03-28	15원	2	수정 삭제
105	최교수	교수걸	일반교수	110원	2022-03-30	12원	1	수정 삭제

copyright©2022. ezenac.co.kr


교수 정보관리

교수 이름

교수 아이디

교수 직급

교수 급여

교수 입사일 

교수 보직수당

교수 부서번호

[저장하기](#) [다시작성](#)

copyright©2022. ezenac.co.kr

교수 정보관리

[새 교수 추가](#)

교수번호	이름	아이디	직급	급여	입사일	보직수당	부서번호	-
102	김교수	교수형	일반교수	100원	2020-04-07	10원	1	수정 삭제
103	박교수	교수동생	일반교수	100원	2020-04-06	10원	1	수정 삭제
105	최교수	교수결	일반교수	110원	2022-03-30	12원	1	수정 삭제
106	정교수	교수인형	일반교수	200원	2022-01-01	20원	1	수정 삭제

copyright©2022. ezenac.co.kr

학생_연습문제 Restful

GET http://localhost:3000/department/ Send 200 OK 20.9 ms 525 B

Body Auth Query Header Docs

Preview Header 13 Cookie Timeline

1 [

2 {

3 "name": "김학생",

4 "userid": "kim",

5 "grade": "1학년",

6 "idnum": "060606-4123456",

7 "birthdate": "060606",

8 "tel": "010-1234-5678",

9 "height": "165",

10 "weight": "55",

11 "deptno": "201",

12 "profno": "1534",

13 "id": 1

14 },

15 {

16 "name": "박학생",

17 "userid": "park",

18 "grade": "2학년",

19 "idnum": "050505-3234567",

20 "birthdate": "050505",

21 "tel": "010-1234-5678",

22 "height": "175",

23 "weight": "70",

24 "deptno": "202",

25 "profno": "1535",

26 "id": 2

27 }

28]

Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

Introduction to Insomnia

POST http://localhost:3000/department Send 201 Created 29.6 ms 233 B

Form 10 Auth Query Header 1 Docs

Preview Header 15 Cookie Timeline

name	김학생		
userid	kim		
grade	1학년		
idnum	060606-1234567		
birthdate	060606		
tel	010-1234-5678		
height	170		
weight	65		
deptno	201		
profno	1534		

1 {

2 "name": "김학생",

3 "userid": "kim",

4 "grade": "1학년",

5 "idnum": "060606-1234567",

6 "birthdate": "060606",

7 "tel": "010-1234-5678",

8 "height": "170",

9 "weight": "65",

10 "deptno": "201",

11 "profno": "1534",

12 "id": 1

13 }

PUT http://localhost:3000/department/1 Send 200 OK 34.2 ms 233 B

Form 10 Auth Query Header 1 Docs

name	김학생	✓	✕
userid	kim	✓	✕
grade	1학년	✓	✕
idnum	060606-4123456	✓	✕
birthdate	060606	✓	✕
tel	010-1234-5678	✓	✕
height	165	✓	✕
weight	55	✓	✕
deptno	201	✓	✕
profno	1534	✓	✕

Preview Header 13 Cookie Timeline

```

1 {
2   "name": "김학생",
3   "userid": "kim",
4   "grade": "1학년",
5   "idnum": "060606-4123456",
6   "birthdate": "060606",
7   "tel": "010-1234-5678",
8   "height": "165",
9   "weight": "55",
10  "deptno": "201",
11  "profno": "1534",
12  "id": 1
13 }

```

DELETE http://localhost:3000/department/3 Send 200 OK 34.9 ms 2 B


Body Auth Query Header Docs

Preview Header 13 Cookie Timeline

```

1 {}

```



내 코드

index.html

```

<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>학생 관리 :: MySchool</title>
    <link rel="stylesheet" href="assets/css//style.css" />
  </head>
  <body>
    <div class="container">

```

```

<!-- header -->
<div data-include="inc/header.html"></div>
<!-- body -->
<section>
  <p>
    <a href="add.html">새 학생 추가</a>
  </p>
  <table class="table">
    <colgroup>
      <col width="8%" />
      <col width="8%" />
      <col width="8%" />
      <col width="8%" />
      <col width="8%" />
      <col width="8%" />
      <col width="8%" />
      <col width="8%" />
      <col width="8%" />
      <col width="8%" />
      <col width="8%" />
    </colgroup>
    <thead>
      <tr>
        <th class="text-center">학생번호</th>
        <th class="text-center">이름</th>
        <th class="text-center">아이디</th>
        <th class="text-center">학년</th>
        <th class="text-center">주민번호</th>
        <th class="text-center">생년월일</th>
        <th class="text-center">전화번호</th>
        <th class="text-center">키</th>
        <th class="text-center">몸무게</th>
        <th class="text-center">학과번호</th>
        <th class="text-center">담당교수 일련번호</th>
        <th class="text-center">-</th>
      </tr>
    </thead>
    <tbody id="listBody"></tbody>
  </table>
</section>
<!-- footer -->
<div data-include="inc/footer.html"></div>
</div>

<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
<script src="assets/js/include.js"></script>
<script>
  (async () => {
    let json = null;

    try {
      json = await axios.get("http://localhost:3000/department");
    } catch (e) {
      console.error(e);
    }
  })();

```

```
    alert("요청 처리 실패");
    return;
}

if (json != null) {
    const listBody = document.querySelector("#listBody");
    const { data } = json;

    data.forEach((v, i) => {
        const tr = document.createElement("tr");

        const td1 = document.createElement("td");
        td1.innerHTML = v.id;
        tr.appendChild(td1);

        const td2 = document.createElement("td");
        tr.appendChild(td2);

        const a1 = document.createElement("a");
        a1.setAttribute("href", `view.html?id=${v.id}`);
        a1.innerHTML = v.name;
        td2.appendChild(a1);

        const td3 = document.createElement("td");
        td3.innerHTML = v.userid;
        tr.appendChild(td3);

        const td4 = document.createElement("td");
        td4.innerHTML = v.grade;
        tr.appendChild(td4);

        const td5 = document.createElement("td");
        td5.innerHTML = v.idnum;
        tr.appendChild(td5);

        const td6 = document.createElement("td");
        td6.innerHTML = v.birthdate;
        tr.appendChild(td6);

        const td7 = document.createElement("td");
        td7.innerHTML = v.tel;
        tr.appendChild(td7);

        const td8 = document.createElement("td");
        td8.innerHTML = v.height;
        tr.appendChild(td8);

        const td9 = document.createElement("td");
        td9.innerHTML = v.weight;
        tr.appendChild(td9);

        const td10 = document.createElement("td");
        td10.innerHTML = v.deptno;
        tr.appendChild(td10);
    });
}
```

```
const td11 = document.createElement("td");
td11.innerHTML = v.profno;
tr.appendChild(td11);

const td12 = document.createElement("td");
tr.appendChild(td12);

const a2 = document.createElement("a");
a2.setAttribute("href", `edit.html?id=${v.id}`);
a2.innerHTML = "수정";
td12.appendChild(a2);

const a3 = document.createElement("a");
a3.setAttribute("href", "#");
a3.dataset.id = v.id;
a3.dataset.name = v.name;
a3.innerHTML = "삭제";
a3.classList.add("btn-delete");
td12.appendChild(a3);

listBody.appendChild(tr);

// 삭제 버튼에 대한 클릭 이벤트 구현
a3.addEventListener("click", async (e) => {
  e.preventDefault();
  const current = e.currentTarget;
  const id = current.dataset.id;
  const name = current.dataset.name;

  if (!confirm(`정말 ${name}(을)를 삭제하시겠습니까?`)) {
    return;
  }

  let json = null;

  try {
    json = axios.delete(`http://localhost:3000/department/${id}`);
  } catch (e) {
    console.error(e);
    alert("요청 처리 실패");
    return;
  }

  if (json != null) {
    current.closest("tr").remove();
  }
});
});
}
})();
</script>
</body>
</html>
```

view.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>교수 정보관리 :: MySchool</title>
  <link rel="stylesheet" href="assets/css//style.css" />
</head>
<body>
  <div class="container">
    <!-- header -->
    <div data-include="inc/header.html"></div>
    <!-- body -->
    <section>
      <dl>
        <dt>학생번호</dt>
        <dd class="id"></dd>
        <dt>이름</dt>
        <dd class="name"></dd>
        <dt>아이디</dt>
        <dd class="userid"></dd>
        <dt>학년</dt>
        <dd class="grade"></dd>
        <dt>주민번호</dt>
        <dd class="idnum"></dd>
        <dt>생년월일</dt>
        <dd class="birthdate"></dd>
        <dt>전화번호</dt>
        <dd class="tel"></dd>
        <dt>키</dt>
        <dd class="height"></dd>
        <dt>몸무게</dt>
        <dd class="weight"></dd>
        <dt>학과번호</dt>
        <dd class="deptno"></dd>
        <dt>담당교수 일련번호</dt>
        <dd class="profno"></dd>
      </dl>
      <a href="index.html">목록보기</a>
      <a href="#" class="btn-edit">수정하기</a>
      <a href="#" class="btn-delete">삭제하기</a>
    </section>
    <!-- footer -->
    <div data-include="inc/footer.html"></div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
  <script src="assets/js/include.js"></script>
```

```

<script>
  (async () => {
    /** GET 파라미터 받기 */
    // window.location.search : "?id=101"
    console.log(`search값 => ${window.location.search}`);

    // querystring값이 key-value 형태로 객체로 분리됨
    const params = new URLSearchParams(window.location.search);

    // 객체에서 key사용하여 값을 추출 --> 101
    const id = params.get('id');
    console.log('id');

    // 파라미터가 정상적이지 않으므로 메세지 출력, 전페이지 이동 처리 후 수행
    중단(return)
    if (!id) {
      alert('학생번호가 없습니다. ');
      history.back();
      return;
    }

    /** Ajax요청 */
    let json = null;

    try {
      json = await axios.get(`http://localhost:3000/department/${id}`);
    } catch (e) {
      console.error(e);
      alert('요청을 처리하는데 실패했습니다. ');
      return;
    }

    if (json != null) {
      const {data} = json;

      document.querySelector('.id').innerHTML = data.id;
      document.querySelector('.name').innerHTML = data.name;
      document.querySelector('.userid').innerHTML = data.userid;
      document.querySelector('.grade').innerHTML = data.grade;
      document.querySelector('.idnum').innerHTML = data.idnum;
      document.querySelector('.birthdate').innerHTML = data.birthdate;
      document.querySelector('.tel').innerHTML = data.tel;
      document.querySelector('.height').innerHTML = data.height;
      document.querySelector('.weight').innerHTML = data.weight;
      document.querySelector('.deptno').innerHTML = data.deptno;
      document.querySelector('.profno').innerHTML = data.profno;

      // 가져온 데이터의 일련번호를 사용하여 어떤 대상을 수정할지에 대한 링크
      URL 구성
      document.querySelector('.btn-edit').setAttribute('href',
        'edit.html?id=' + data.id);

      // 가져온 데이터의 일련번호를 사용하여 어떤 대상을 삭제할지에 대한
      dataset 구성

```

```

        const btnDelete = document.querySelector('.btn-delete');
        btnDelete.dataset.id = data.id;
        btnDelete.dataset.name = data.name;
    }
})();

/** 삭제 버튼에 대한 클릭 이벤트 */
document.querySelector('.btn-delete').addEventListener('click', (e) => {
    e.preventDefault();

    const current = e.currentTarget;
    const id = current.dataset.id;
    const name = current.dataset.name;

    if (!confirm(`정말 ${name}(을)를 삭제하시겠습니까?`)) {
        return;
    }

    let json;

    try {
        json = axios.delete(`http://localhost:3000/department/${id}`);
    } catch (e) {
        console.error(e);
        alert('요청을 처리하는데 실패했습니다.');
        return;
    }

    if (json != null) {
        window.location = 'index.html';
    }
});
</script>
</body>
</html>

```

add.html

```

<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>학새 관리 :: MySchool</title>
    <link rel="stylesheet" href="assets/css//style.css" />
  </head>
  <body>
    <div class="container">
      <!-- header -->
      <div data-include="inc/header.html"></div>
      <!-- body -->
    </div>
  </body>
</html>

```



```
<section>
  <form name="form" id="department-form">
    <div class="form-group">
      <label for="name"> 이름</label>
      <div>
        <input type="text" name="name" id="name" />
      </div>
    </div>
    <div class="form-group">
      <label for="userid"> 아이디</label>
      <div>
        <input type="text" name="userid" id="userid" />
      </div>
    </div>
    <div class="form-group">
      <label for="grade"> 학년</label>
      <div>
        <input type="text" name="grade" id="grade" />
      </div>
    </div>
    <div class="form-group">
      <label for="idnum"> 주민번호 </label>
      <div>
        <input type="text" name="idnum" id="idnum" />
      </div>
    </div>
    <div class="form-group">
      <label for="birthdate">생년월일</label>
      <div>
        <input type="text" name="birthdate" id="birthdate" />
      </div>
    </div>
    <div class="form-group">
      <label for="tel"> 전화번호</label>
      <div>
        <input type="text" name="tel" id="tel" />
      </div>
    </div>
    <div class="form-group">
      <label for="height"> 키</label>
      <div>
        <input type="number" name="height" id="height" />
      </div>
    </div>
    <div class="form-group">
      <label for="weight"> 몸무게</label>
      <div>
        <input type="number" name="weight" id="weight" />
      </div>
    </div>
    <div class="form-group">
      <label for="deptno"> 학과번호</label>
      <div>
        <input type="number" name="deptno" id="deptno" />
      </div>
    </div>
  </form>
</section>
```

```

        </div>
    </div>
    <div class="form-group">
        <label for="profno"> 교수일련번호</label>
        <div>
            <input type="number" name="profno" id="profno" />
        </div>
    </div>
    <hr />
    <div class="text-center">
        <button type="submit">저장하기</button>
        <button type="reset">다시작성</button>
    </div>
</form>
</section>
<!-- footer -->
<div data-include="inc/footer.html"></div>
</div>

<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
<script src="assets/js/include.js"></script>
<script>
    document
        .querySelector("#department-form")
        .addEventListener("submit", async (e) => {
            e.preventDefault();

            // 입력값 받아오기
            const name = document.querySelector("#name").value;
            const userid = document.querySelector("#userid").value;
            const grade = document.querySelector("#grade").value;
            const idnum = document.querySelector("#idnum").value;
            const birthdate = document.querySelector("#birthdate").value;
            const tel = document.querySelector("#tel").value;
            const height = document.querySelector("#height").value;
            const weight = document.querySelector("#weight").value;
            const deptno = document.querySelector("#deptno").value;
            const profno = document.querySelector("#profno").value;

            // 유효성 검사는 생략한다.

            let json = null;

            try {
                // 대부분의 backend가 일련번호는 자체적으로 생성하기 때문에 create에 해당
                // 하는 데이터 추가 처리에서는
                // 백엔드에게 일련번호를 제외한 변수만 전달해야 한다.
                json = await axios.post("http://localhost:3000/department", {
                    name: name,
                    userid: userid,
                    grade: grade,
                    idnum: idnum,
                    birthdate: birthdate,
                    tel: tel,

```

```

        height: height,
        weight: weight,
        deptno: deptno,
        profno: profno,
    });
} catch (e) {
    console.error(e);
    alert('요청을 처리하는데 실패했습니다.');
```

return;

```

}

if (json !== null) {
    const {data} = json;
    console.log(data);
    // 백엔드에서 데이터 저장에 대한 응답으로 보낸 JSON에는 생성된 신규 항목에
    대한 일련번호가 포함되어 있다.
    // 이 값을 활용하여 상세 페이지로 이동해야 한다.
    // 이 때, 어떤 대상을 열람할 것인지에 대한 GET 파라미터를 구성한다.
    window.location = 'view.html?id=' + data.id;
}
});
</script>
</body>
</html>

```

edit.html

```

<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>학과관리 :: MySchool</title>
    <link rel="stylesheet" href="assets/css//style.css" />
  </head>
  <body>
    <div class="container">
      <!-- header -->
      <div data-include="inc/header.html"></div>
      <!-- body -->
      <section>
        <form name="form" id="department-form">
          <!-- 수정할 대상에 대한 일련번호 값을 포함해야 하므로 hidden 필드로 처리한다. -->
          <input type="hidden" name="id" id="id" />
          <div class="form-group">
            <label for="name"> 이름 <span class="text-danger">*</span></label>
            <div>
              <input type="text" name="name" id="name" />
            </div>
          </div>

```

```
<div class="form-group">
  <label for="userid"> 아이디</label>
  <div>
    <input type="text" name="userid" id="userid" />
  </div>
</div>
<div class="form-group">
  <label for="grade"> 학년</label>
  <div>
    <input type="text" name="grade" id="grade" />
  </div>
</div>
<div class="form-group">
  <label for="idnum"> 주민번호</label>
  <div>
    <input type="text" name="idnum" id="idnum" />
  </div>
</div>
<div class="form-group">
  <label for="birthdate"> 생년월일</label>
  <div>
    <input type="text" name="birthdate" id="birthdate" />
  </div>
</div>
<div class="form-group">
  <label for="tel"> 전화번호</label>
  <div>
    <input type="text" name="tel" id="tel" />
  </div>
</div>
<div class="form-group">
  <label for="height"> 키</label>
  <div>
    <input type="number" name="height" id="height" />
  </div>
</div>
<div class="form-group">
  <label for="weight"> 몸무게</label>
  <div>
    <input type="number" name="weight" id="weight" />
  </div>
</div>
<div class="form-group">
  <label for="deptno"> 학과번호</label>
  <div>
    <input type="number" name="deptno" id="deptno" />
  </div>
</div>
<div class="form-group">
  <label for="profno"> 담당교수의 일련번호</label>
  <div>
    <input type="number" name="profno" id="profno" />
  </div>
</div>
```

```

        <hr />
        <div class="text-center">
            <button type="submit">수정하기</button>
            <button type="reset">다시작성</button>
        </div>
    </form>
</section>
<!-- footer -->
<div data-include="inc/footer.html"></div>
</div>

<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
<script src="assets/js/include.js"></script>
<script>
    (async () => {
        /** GET 파라미터 받기 */
        const params = new URLSearchParams(window.location.search);
        const id = params.get("id");
        console.log(id);

        // 파라미터가 정상적이지 않으므로 메시지 출력, 전페이지로 이동 처리 후 수행 중단
        (return)
        if (!id) {
            alert("학생번호가 없습니다.");
            history.back();
            return;
        }

        /** Ajax요청 */
        // 기존에 저장되어 있는 내용을 <form>안에 셋팅하기 위해 Ajax로 데이터를 가져온
다.
        let json;

        try {
            json = await axios.get(`http://localhost:3000/department/${id}`);
        } catch (e) {
            console.error(e);
            alert("요청을 처리하는데 실패했습니다.");
            return;
        }

        // 가져온 데이터가 있다면 <input> 태그의 value 속성으로 가져온 값을 적용한다.
        // 일련번호와 같이 시스템이 관리하는 값을 hidden 요소로 포함시킨다.
        if (json != null) {
            const { data } = json;
            document.querySelector("#id").value = data.id;
            document.querySelector("#name").value = data.name;
            document.querySelector("#userid").value = data.userid;
            document.querySelector("#grade").value = data.grade;
            document.querySelector("#idnum").value = data.idnum;
            document.querySelector("#birthdate").value = data.birthdate;
            document.querySelector("#tel").value = data.tel;
            document.querySelector("#height").value = data.height;
            document.querySelector("#weight").value = data.weight;

```

```

        document.querySelector("#deptno").value = data.deptno;
        document.querySelector("#profno").value = data.profno;
    }
})();

/** 수정 버튼의 submit 이벤트 */
document
    .querySelector("#department-form")
    .addEventListener("submit", async (e) => {
        e.preventDefault();

        // 입력값 받아오기
        const id = document.querySelector("#id").value;
        const name = document.querySelector("#name").value;
        const userid = document.querySelector("#userid").value;
        const grade = document.querySelector("#grade").value;
        const idnum = document.querySelector("#idnum").value;
        const birthdate = document.querySelector("#birthdate").value;
        const tel = document.querySelector("#tel").value;
        const height = document.querySelector("#height").value;
        const weight = document.querySelector("#weight").value;
        const deptno = document.querySelector("#deptno").value;
        const profno = document.querySelector("#profno").value;

        // 입력값에 대한 유효성 검사 진행.... (이 예제에서는 생략)

        let json = null;

        try {
            json = await axios.put(`http://localhost:3000/department/${id}`, {
                name: name,
                userid: userid,
                grade: grade,
                idnum: idnum,
                birthdate: birthdate,
                tel: tel,
                height: height,
                weight: weight,
                deptno: deptno,
                profno: profno,
            });
        } catch (e) {
            console.error(e);
            alert("요청을 처리하는데 실패했습니다.");
            return;
        }

        if (json != null) {
            window.location.href = `view.html?id=${id}`;
        }
    });
</script>
</body>
</html>

```

구현 스크린샷

학생 관리

새 학생 추가

학생번호	이름	아이디	학년	주민번호	생년월일	전화번호	키	몸무게	학과번호	담당교수 일련번호	
1	김학생	kim	1학년	060606-4123456	060606	010-1234-5678	165	55	101	1534	수정 삭제
2	박학생	park	2학년	050505-3234567	050505	010-1234-5678	175	70	202	1535	수정 삭제

copyright©2022. ezenac.co.kr

학생 관리

이름

최학생

아이디

choi

학년

2학년

주민번호

050505'

생년월일

040404

전화번호

010-1234-5678

키

175

몸무게

70

학과번호

203

교수일련번호

1536

저장하기

다시작성

copyright©2022. ezenac.co.kr

학생 관리

- ⊙ 학생번호
 - | 3
- ⊙ 이름
 - | 최학생
- ⊙ 아이디
 - | choi
- ⊙ 학년
 - | 2학년
- ⊙ 주민번호
 - | 050505-3123456
- ⊙ 생년월일
 - | 050505
- ⊙ 전화번호
 - | 010-1234-5678
- ⊙ 키
 - | 175
- ⊙ 몸무게
 - | 70
- ⊙ 학과번호
 - | 203
- ⊙ 담당교수 일련번호
 - | 1536

[목록보기](#) [수정하기](#) [삭제하기](#)

copyright©2022. ezenac.co.kr

학생 관리

이름 *	<input type="text" value="김학생"/>
아이디	<input type="text" value="kim"/>
학년	<input type="text" value="3학년"/>
주민번호	<input type="text" value="040404-4123456"/>
생년월일	<input type="text" value="040404"/>
전화번호	<input type="text" value="010-1234-5678"/>
키	<input type="text" value="165"/>
몸무게	<input type="text" value="55"/>
학과번호	<input type="text" value="301"/>
담당교수의 일련번호	<input type="text" value="1534"/>

copyright©2022. ezenac.co.kr

학생 관리

새 학생 추가

학생번호	이름	아이디	학년	주민번호	생년월일	전화번호	키	몸무게	학과번호	담당교수 일련번호	-
1	김학생	kim	3학년	040404-4123456	040404	010-1234-5678	165	55	301	1534	수정 삭제
2	박학생	park	2학년	050505-3234567	050505	010-1234-5678	175	70	202	1535	수정 삭제
3	최학생	choi	2학년	050505-3123456	050505	010-1234-5678	175	70	203	1536	수정 삭제

copyright©2022. ezenac.co.kr

127.0.0.1:5500 내용:

정말 최학생(을)를 삭제하시겠습니까?

확인

취소

새 학생 추가

학생번호	이름	아이디	학년	주민번호	생년월일	전화번호	키	몸무게	학과번호	담당교수 일련번호	-
1	김학생	kim	3학년	040404-4123456	040404	010-1234-5678	165	55	301	1534	수정 삭제
2	박학생	park	2학년	050505-3234567	050505	010-1234-5678	175	70	202	1535	수정 삭제
3	최학생	choi	2학년	050505-3123456	050505	010-1234-5678	175	70	203	1536	수정 삭제

copyright©2022. ezenac.co.kr

학생 관리

새 학생 추가

학생번호	이름	아이디	학년	주민번호	생년월일	전화번호	키	몸무게	학과번호	담당교수 일련번호	-
1	김학생	kim	3학년	040404-4123456	040404	010-1234-5678	165	55	301	1534	수정 삭제
2	박학생	park	2학년	050505-3234567	050505	010-1234-5678	175	70	202	1535	수정 삭제

copyright©2022. ezenac.co.kr