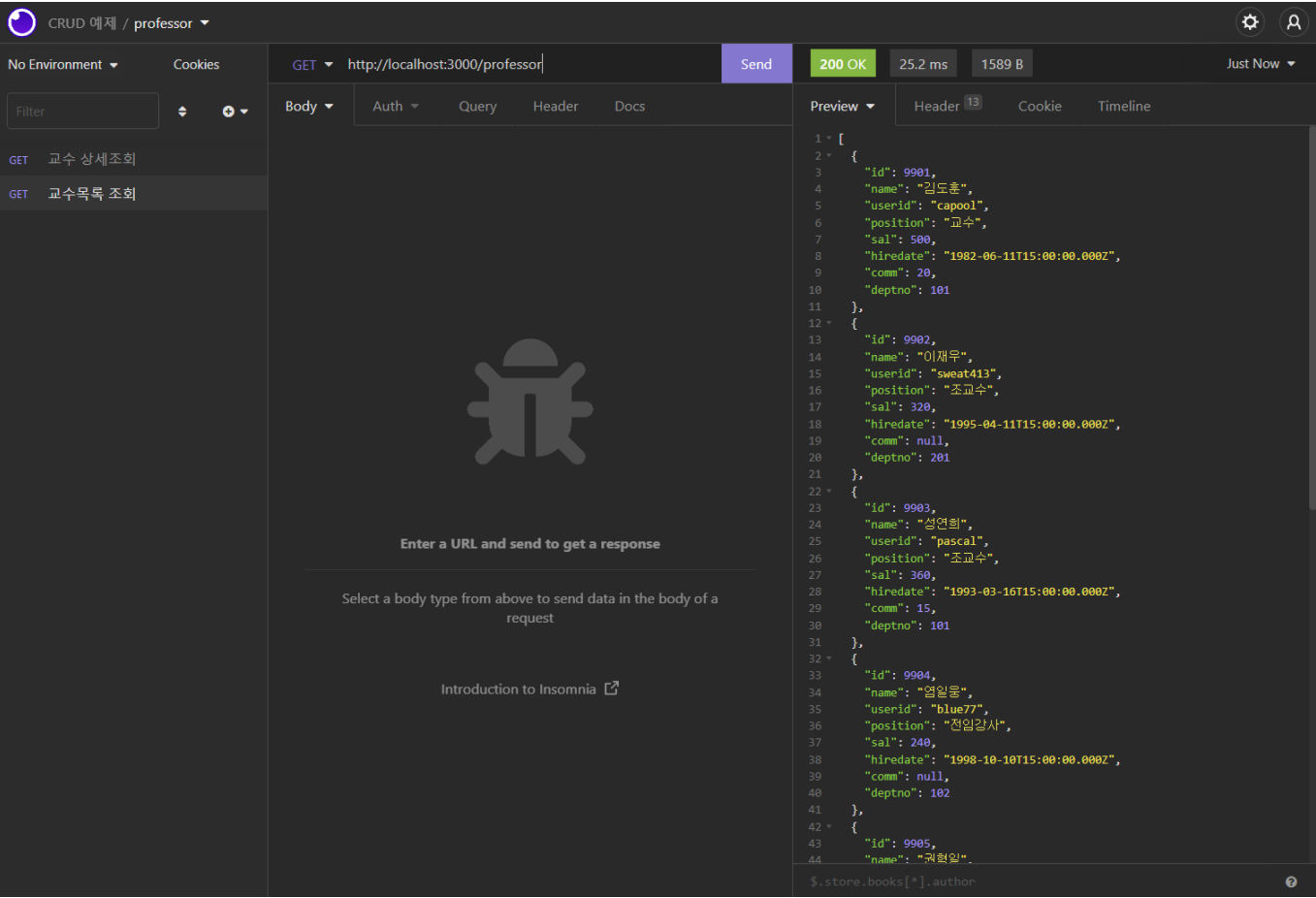


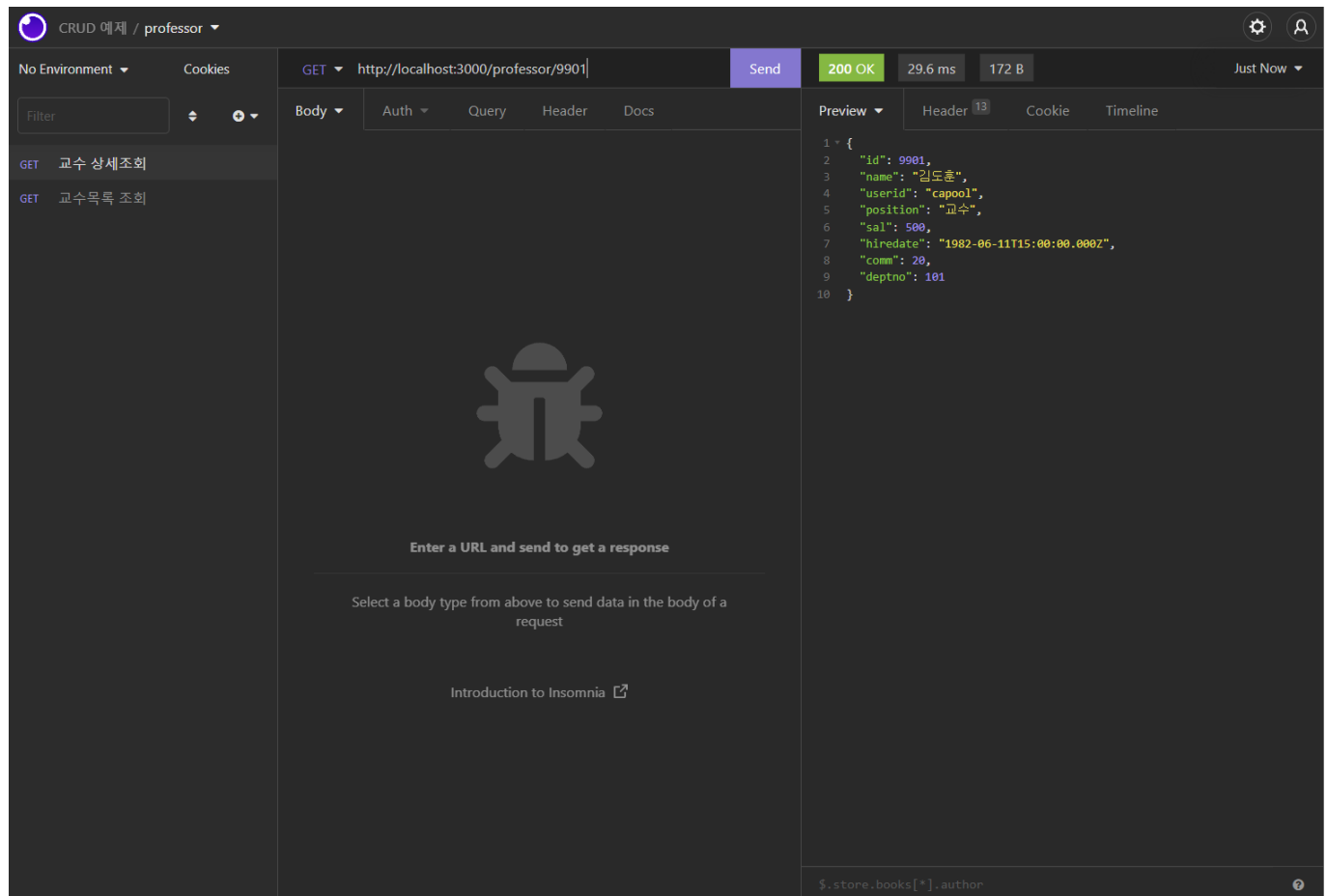
교수관리

insomnia 결과

교수목록조회(GET)



교수상세조회(GET/id)



교수 추가(POST)

CRUD 예제 / professor

No Environment Cookies

POST http://localhost:3000/professor Send 201 Created 20.4 ms 184 B Just Now

Filter

DEL 교수삭제
POST 교수추가
GET 교수 상세조회
GET 교수목록 조회

Form 7 Auth Query Header 1 Docs

name	신지섭	✓	🗑
userid	pon03046	✓	🗑
position	전임강사	✓	🗑
sal	210	✓	🗑
hiredate	2020-05-10T15:00:00.000Z	✓	🗑
comm	value	✓	🗑
deptno	101	✓	🗑
New name	New value		

Preview

```

1 {
2   "name": "신지섭",
3   "userid": "pon03046",
4   "position": "전임강사",
5   "sal": "210",
6   "hiredate": "2020-05-10T15:00:00.000Z",
7   "comm": "",
8   "deptno": "101",
9   "id": 9909
10 }

```

\$.store.books[*].author

CRUD 예제 / professor

No Environment Cookies

GET http://localhost:3000/professor Send 200 OK 27.1 ms 1795 B Just Now

Filter

DEL 교수삭제
POST 교수추가
GET 교수 상세조회
GET 교수목록 조회

Body Auth Query Header Docs

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

Preview

```

51 },
52 {
53   "id": 9906,
54   "name": "이만식",
55   "userid": "pocari",
56   "position": "부교수",
57   "sal": 420,
58   "hiredate": "1988-07-10T14:00:00.000Z",
59   "comm": null,
60   "deptno": 101
61 },
62 {
63   "id": 9907,
64   "name": "전문지",
65   "userid": "totoro",
66   "position": "전임강사",
67   "sal": 210,
68   "hiredate": "2001-05-10T15:00:00.000Z",
69   "comm": null,
70   "deptno": 101
71 },
72 {
73   "id": 9908,
74   "name": "남은현",
75   "userid": "bird13",
76   "position": "부교수",
77   "sal": 400,
78   "hiredate": "1990-10-17T15:00:00.000Z",
79   "comm": 17,
80   "deptno": 202
81 },
82 {
83   "name": "신지섭",
84   "userid": "pon03046",
85   "position": "전임강사",
86   "sal": "210",
87   "hiredate": "2020-05-10T15:00:00.000Z",
88   "comm": "",
89   "deptno": "101",
90   "id": 9909
91 }
92 ]

```

\$.store.books[*].author

교수 수정(PUT)

CRUD 예제 / professor

No Environment Cookies

PUT http://localhost:3000/professor/9909 Send 200 OK 22.7 ms 186 B Just Now

Filter

DEL 교수삭제
PUT 교수 수정
POST 교수추가
GET 교수 상세조회
GET 교수목록 조회

Form 7 Auth Query Header 1 Docs

name	신지섭	✓	🗑
userid	pon0304616	✓	🗑
position	전임강사	✓	🗑
sal	215	✓	🗑
hierdate	2020-05-11T15:00:00.000Z	✓	🗑
comm	value	✓	🗑
deptno	102	✓	🗑
New name	New value		

Preview

```

1 - {
2   "name": "신지섭",
3   "userid": "pon0304616",
4   "position": "전임강사",
5   "sal": "215",
6   "hierdate": "2020-05-11T15:00:00.000Z",
7   "comm": "",
8   "deptno": "102",
9   "id": 9909
10 }

```

\$.store.books[*].author

CRUD 예제 / professor

No Environment Cookies

GET http://localhost:3000/professor Send 200 OK 30.9 ms 1797 B Just Now

Filter

DEL 교수삭제
PUT 교수 수정
POST 교수추가
GET 교수 상세조회
GET 교수목록 조회

Body Auth Query Header Docs

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

Preview

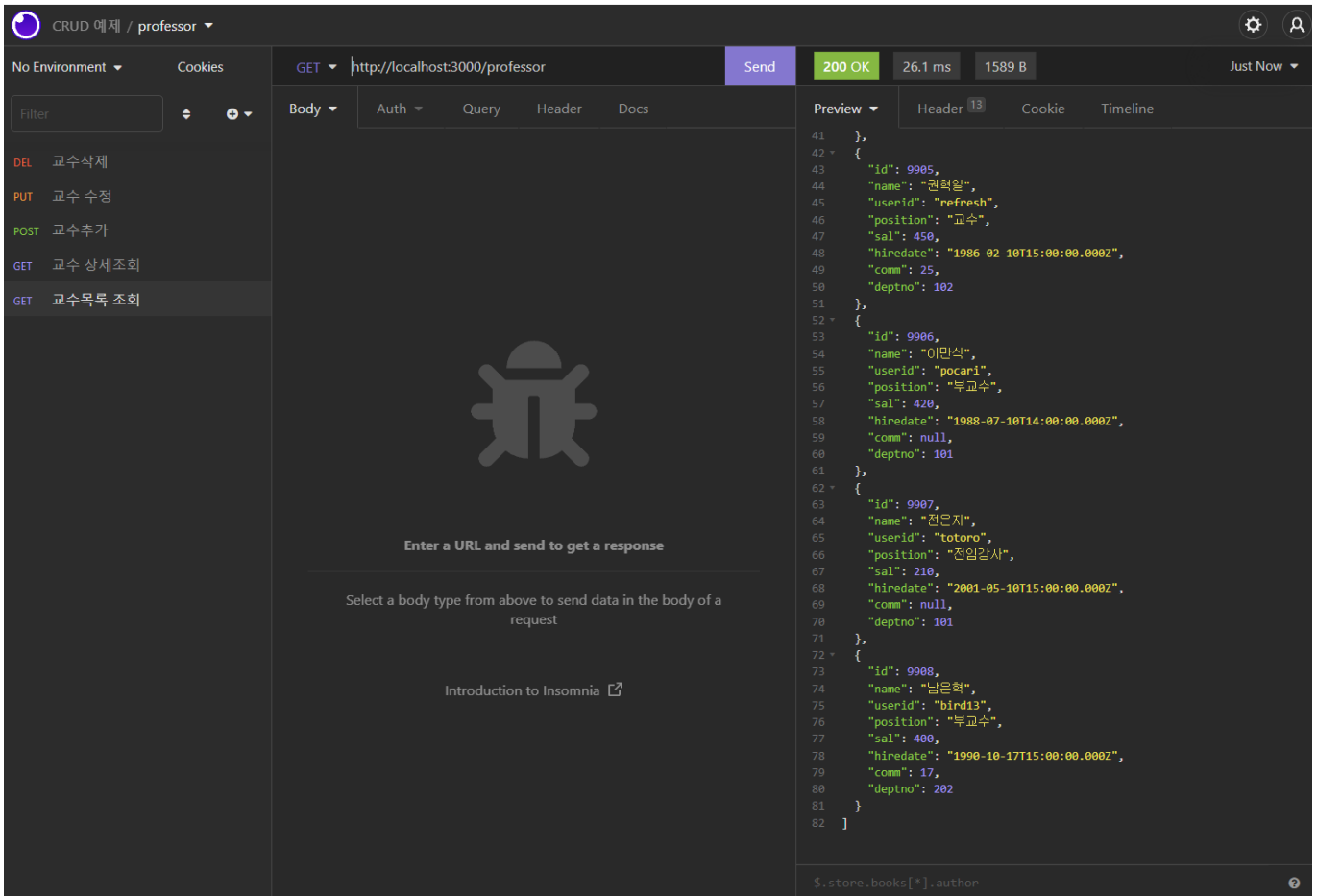
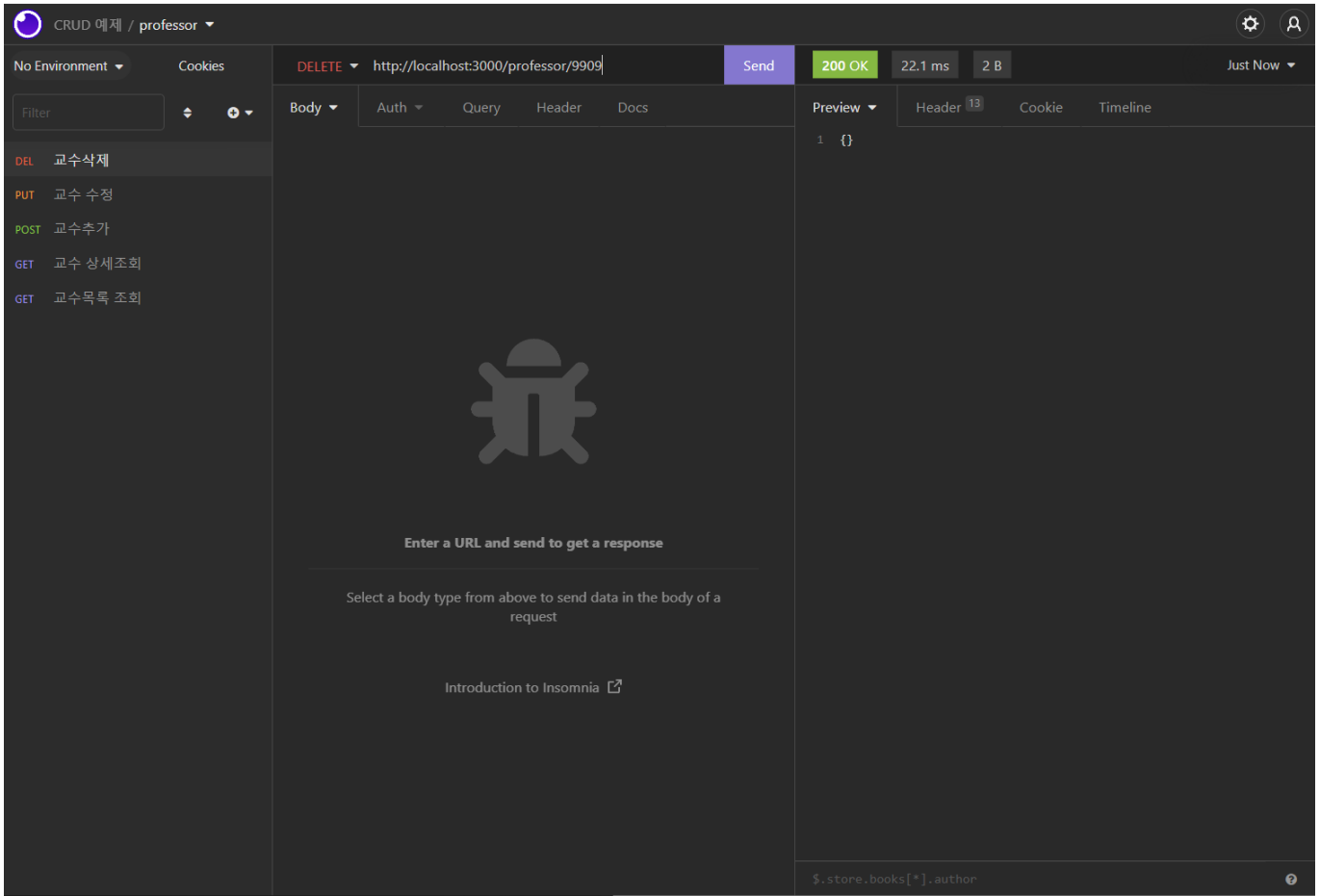
```

51 },
52 {
53   "id": 9906,
54   "name": "이만식",
55   "userid": "pocari",
56   "position": "부교수",
57   "sal": 420,
58   "hiredate": "1988-07-10T14:00:00.000Z",
59   "comm": null,
60   "deptno": 101
61 },
62 {
63   "id": 9907,
64   "name": "전은지",
65   "userid": "totoro",
66   "position": "전임강사",
67   "sal": 210,
68   "hiredate": "2001-05-10T15:00:00.000Z",
69   "comm": null,
70   "deptno": 101
71 },
72 {
73   "id": 9908,
74   "name": "남은희",
75   "userid": "bird13",
76   "position": "부교수",
77   "sal": 400,
78   "hiredate": "1990-10-17T15:00:00.000Z",
79   "comm": 17,
80   "deptno": 202
81 },
82 {
83   "name": "신지섭",
84   "userid": "pon0304616",
85   "position": "전임강사",
86   "sal": "215",
87   "hierdate": "2020-05-11T15:00:00.000Z",
88   "comm": "",
89   "deptno": "102",
90   "id": 9909
91 }
92 ]

```

\$.store.books[*].author

교수 삭제(DELETE)



professor 연습문제

index

```
//페이지가 실행되면서 자동으로 동작해야 하므로 즉시실행함수형태 구현.
(async () => {
  //ajax 결과가 저장될 json
  let json = null
  //ajax 요청
  try {
    json = await axios.get('http://localhost:3000/professor')
  } catch(e) {
    console.error(e);
    alert('요청을 처리하는데 실패했습니다.');
```

return;

```
  }

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

    data.forEach((v, i) => {
      //내용담을 tr 생성
      const tr = document.createElement('tr');

      //id 내용 추가
      const td1 = document.createElement('td');
      td1.innerHTML = v.id
      //교수이름 내용 추가
      const td2 = document.createElement('td');
      const a1 = document.createElement('a');
      a1.setAttribute('href', `view.html?id=${v.id}`)
      a1.innerHTML = v.name
      td2.appendChild(a1);
      //아이디 내용 추가
      const td3 = document.createElement('td');
      td3.innerHTML = v.userid;
      //직급 내용 추가
      const td4 = document.createElement('td');
      td4.innerHTML = v.position;
      //급여 내용 추가
      const td5 = document.createElement('td');
      td5.innerHTML = v.sal
      //입사일 내용 추가
      const td6 = document.createElement('td');
      td6.innerHTML = v.hiredate.substring(0,10);
      //보직수당 내용 추가
      const td7 = document.createElement('td');
      td7.innerHTML = v.comm;
      //부서번호(학과번호) 내용 추가
      const td8 = document.createElement('td');
      td8.innerHTML = v.deptno;
      //수정 및 삭제 추가
      const td9 = document.createElement('td');
      const a2 = document.createElement('a');
      //수정
```

```

a2.setAttribute('href', `edit.html?id=${v.id}`);
a2.innerHTML = '수정'
td9.appendChild(a2)
//삭제
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);

//tr에 모든내용 추가
tr.appendChild(td1);
tr.appendChild(td2);
tr.appendChild(td3);
tr.appendChild(td4);
tr.appendChild(td5);
tr.appendChild(td6);
tr.appendChild(td7);
tr.appendChild(td8);
tr.appendChild(td9);

//최종결과물 listBody에 추가
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 ${name}(을)를 삭제하시겠습니까?`)){
    return;
  }
  let json = null
  try {
    json = await
axios.delete(`http://localhost:3000/professor/${id}`);
  } catch(e) {
    console.error(e);
    alert('요청을 처리하는데 실패했습니다.')
    return;
  }
  //현재 클릭된 곳에서 가장 가까운 tr선택 삭제
  if(json != null) {
    current.closest('tr').remove();
  }
})
})
}
})();

```

교수관리

[새 교수 추가](#)

교수번호	이름	아이디	직급	급여	입사일	보직수당	부서번호(학과번호)	-
9901	김도훈	capool	교수	500	1982-06-11	20	101	수정 삭제
9902	이재우	sweat413	조교수	320	1995-04-11		201	수정 삭제
9903	성연희	pascal	조교수	360	1993-03-16	15	101	수정 삭제
9904	염일웅	blue77	전임강사	240	1998-10-10		102	수정 삭제
9905	권혁일	refresh	교수	450	1986-02-10	25	102	수정 삭제
9906	이만식	pocari	부교수	420	1988-07-10		101	수정 삭제
9907	전은지	totoro	전임강사	210	2001-05-10		101	수정 삭제
9908	남은혁	bird13	부교수	400	1990-10-17	17	202	수정 삭제

copyright©2022. ezenac.co.kr

교수관리

[새 교수 추가](#)

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

[확인](#) [취소](#)

교수번호	이름	아이디	직급	급여	입사일	보직수당	부서번호(학과번호)	-
9901	김도훈	capool	교수	500	1982-06-11	20	101	수정 삭제
9902	이재우	sweat413	조교수	320	1995-04-11		201	수정 삭제
9903	성연희	pascal	조교수	360	1993-03-16	15	101	수정 삭제
9904	염일웅	blue77	전임강사	240	1998-10-10		102	수정 삭제
9905	권혁일	refresh	교수	450	1986-02-10	25	102	수정 삭제
9906	이만식	pocari	부교수	420	1988-07-10		101	수정 삭제
9907	전은지	totoro	전임강사	210	2001-05-10		101	수정 삭제
9908	남은혁	bird13	부교수	400	1990-10-17	17	202	수정 삭제
9909	신지섭	pon0304616	전임강사	215	2020-05-11	5	102	수정 삭제

copyright©2022. ezenac.co.kr

교수관리

[새 교수 추가](#)

교수번호	이름	아이디	직급	급여	입사일	보직수당	부서번호(학과번호)	-
9901	김도훈	capool	교수	500	1982-06-11	20	101	수정 삭제
9902	이재우	sweat413	조교수	320	1995-04-11		201	수정 삭제
9903	성연희	pascal	조교수	360	1993-03-16	15	101	수정 삭제
9904	염일웅	blue77	전임강사	240	1998-10-10		102	수정 삭제
9905	권혁일	refresh	교수	450	1986-02-10	25	102	수정 삭제
9906	이만식	pocari	부교수	420	1988-07-10		101	수정 삭제
9907	전은지	totoro	전임강사	210	2001-05-10		101	수정 삭제
9908	남은현	bird13	부교수	400	1990-10-17	17	202	수정 삭제

copyright©2022. ezenac.co.kr

view

```
//페이지가 실행되면서 자동으로 동작해야 하므로 즉시실행함수형태로 구현.
(async () => {
  //GET 파라미터 받기
  //queryString값이 key-value형태로 객체로 분리됨;
  const params = new URLSearchParams(window.location.search);
  // 객체에서 key사용하여 값을 추출 --> 10101;
  const id = params.get('id');
  console.log(id)

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

  // ajax 요청
  let json = null

  try {
    json = await axios.get(`http://localhost:3000/professor/${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.substring(0,10);
        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;
    }
})();

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 = null;

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

    if (json != null) {
        window.location = 'index.html';
    }
})

```

교수관리

- 교수번호 | 9901
- 이름 | 김도훈
- 아이디 | capool
- 직급 | 교수
- 급여 | 500
- 입사일 | 1982-06-11
- 보직수당 | 20
- 부서번호(학과번호) | 101

목록보기 수정하기 삭제하기

교수관리

- 교수번호 | 9909
- 이름 | 신지섭
- 아이디 | pon0304616
- 직급 | 전임강사
- 급여 | 215
- 입사일 | 2020-05-11
- 보직수당 | 5
- 부서번호(학과번호) | 102

목록보기 수정하기 삭제하기

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

확인취소

교수관리

[새 교수 추가](#)

교수번호	이름	아이디	직급	급여	입사일	보직수당	부서번호(학과번호)	-
9901	김도훈	capool	교수	500	1982-06-11	20	101	수정 삭제
9902	이재우	sweat413	조교수	320	1995-04-11		201	수정 삭제
9903	성연희	pascal	조교수	360	1993-03-16	15	101	수정 삭제
9904	염일웅	blue77	전임강사	240	1998-10-10		102	수정 삭제
9905	권혁일	refresh	교수	450	1986-02-10	25	102	수정 삭제
9906	이만식	pocari	부교수	420	1988-07-10		101	수정 삭제
9907	전은지	totoro	전임강사	210	2001-05-10		101	수정 삭제
9908	남은현	bird13	부교수	400	1990-10-17	17	202	수정 삭제

copyright©2022. ezenac.co.kr

add

```
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;

    let json = null;

    try {
        json = await axios.post('http://localhost:3000/professor', {
            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;

window.location = 'view.html?id=' + data.id

})
})
```

교수관리

이름 * 신지섭

아이디* pon03046

직급* 전임강사

급여 210

입사일 2020-05-10T15:00:00.000Z

보직수당

부서번호(학과번호) 101

저장하기 다시작성

교수관리

- ◎ 교수번호 | 9909
- ◎ 이름 | 신지섭
- ◎ 아이디 | pon03046
- ◎ 직급 | 전임강사
- ◎ 급여 | 210
- ◎ 입사일 | 2020-05-10
- ◎ 보직수당
- ◎ 부서번호(학과번호) | 101

목록보기 수정하기 삭제하기

교수관리

[새 교수 추가](#)

교수번호	이름	아이디	직급	급여	입사일	보직수당	부서번호(학과번호)	-
9901	김도훈	capool	교수	500	1982-06-11	20	101	수정 삭제
9902	이재우	sweat413	조교수	320	1995-04-11		201	수정 삭제
9903	성연희	pascal	조교수	360	1993-03-16	15	101	수정 삭제
9904	염일웅	blue77	전임강사	240	1998-10-10		102	수정 삭제
9905	권현일	refresh	교수	450	1986-02-10	25	102	수정 삭제
9906	이만식	pocari	부교수	420	1988-07-10		101	수정 삭제
9907	전은지	totoro	전임강사	210	2001-05-10		101	수정 삭제
9908	남은현	bird13	부교수	400	1990-10-17	17	202	수정 삭제
9909	신지섭	pon03046	전임강사	210	2020-05-10		101	수정 삭제

copyright©2022. ezenac.co.kr

edit

```
//페이지가 실행되면서 자동으로 동작해야 하므로 즉시실행함수형태로 구현.
(async () => {
  const params = new URLSearchParams(window.location.search);
  const id = params.get('id');
  console.log(id)
  if (!id) {
    alert('학생번호가 없습니다. ');
    history.back();
    return;
  }

  //ajax요청
  let json = null
  try {
    json = await axios.get(`http://localhost:3000/professor/${id}`);
  } catch(e) {
    console.error(e);
    alert('요청을 처리하는데 실패했습니다. ');
    return;
  }
  if (json != null) {
    const {data} = json;
    //가져온 데이터가 있다면 <input> 태그의 value속성으로 가져온 값을 적용한다.
    //일련번호와 같이 시스템이 관리하는 값은 hidden요소로 포함시킨다.
    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;
    }
})();

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;

    let json = null;
    try {
        json = await axios.put(`http://localhost:3000/professor/${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}`;
    }
})
```

copyright©2022. ezenac.co.kr

copyright©2022. ezenac.co.kr

교수관리

- ◎ 교수번호
| 9909
- ◎ 이름
| 신지섭
- ◎ 아이디
| pon0304616
- ◎ 직급
| 전임강사
- ◎ 급여
| 215
- ◎ 입사일
| 2020-05-11
- ◎ 보직수당
| 5
- ◎ 부서번호(학과번호)
| 102

[목록보기](#) [수정하기](#) [삭제하기](#)

copyright©2022. ezenac.co.kr

교수관리

[새 교수 추가](#)

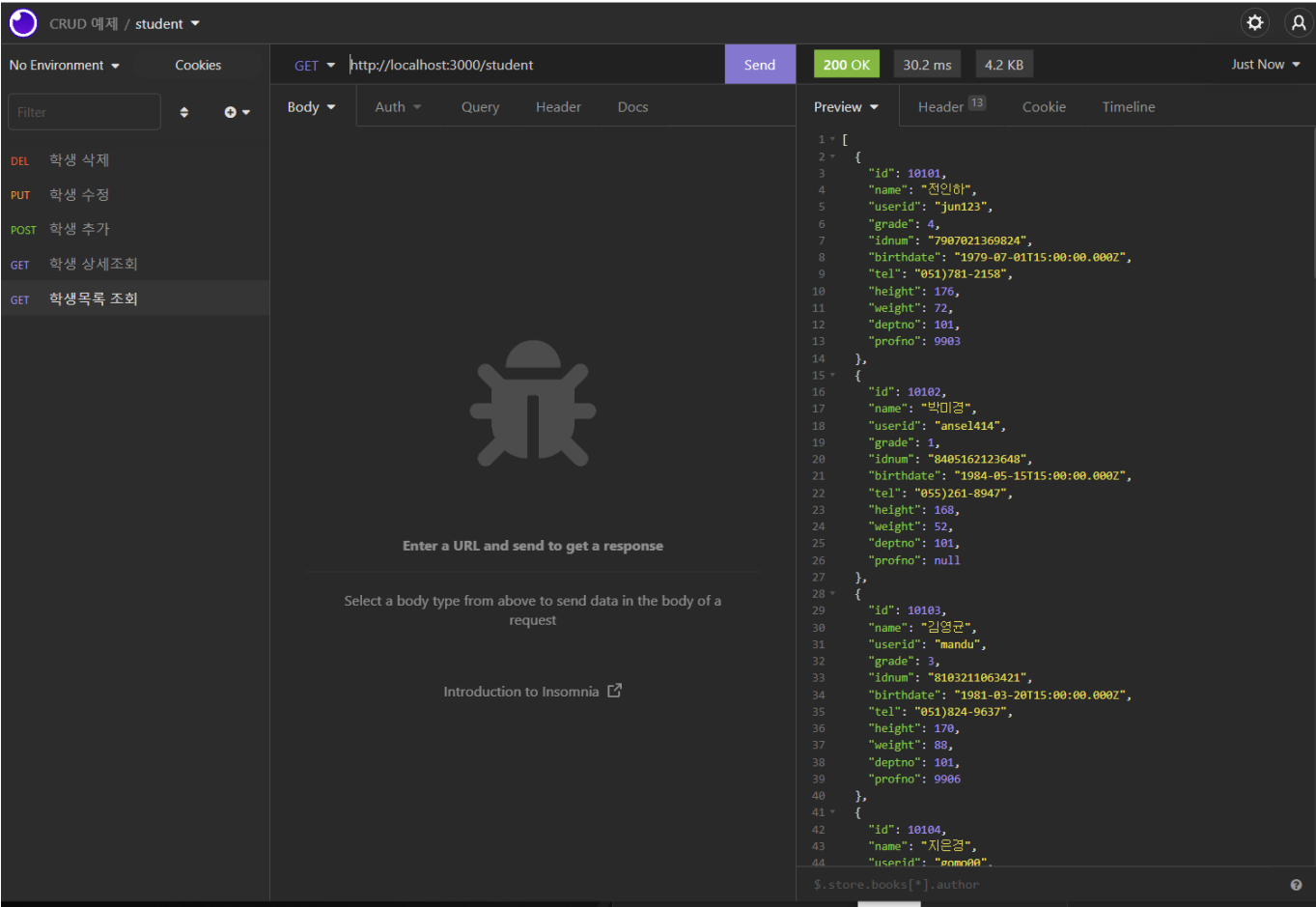
교수번호	이름	아이디	직급	급여	입사일	보직수당	부서번호(학과번호)	-
9901	김도훈	capool	교수	500	1982-06-11	20	101	수정 삭제
9902	이재우	sweat413	조교수	320	1995-04-11		201	수정 삭제
9903	성연희	pascal	조교수	360	1993-03-16	15	101	수정 삭제
9904	염일웅	blue77	전임강사	240	1998-10-10		102	수정 삭제
9905	권혁일	refresh	교수	450	1986-02-10	25	102	수정 삭제
9906	이만식	pocari	부교수	420	1988-07-10		101	수정 삭제
9907	전은지	totoro	전임강사	210	2001-05-10		101	수정 삭제
9908	남은형	bird13	부교수	400	1990-10-17	17	202	수정 삭제
9909	신지섭	pon0304616	전임강사	215	2020-05-11	5	102	수정 삭제

copyright©2022. ezenac.co.kr

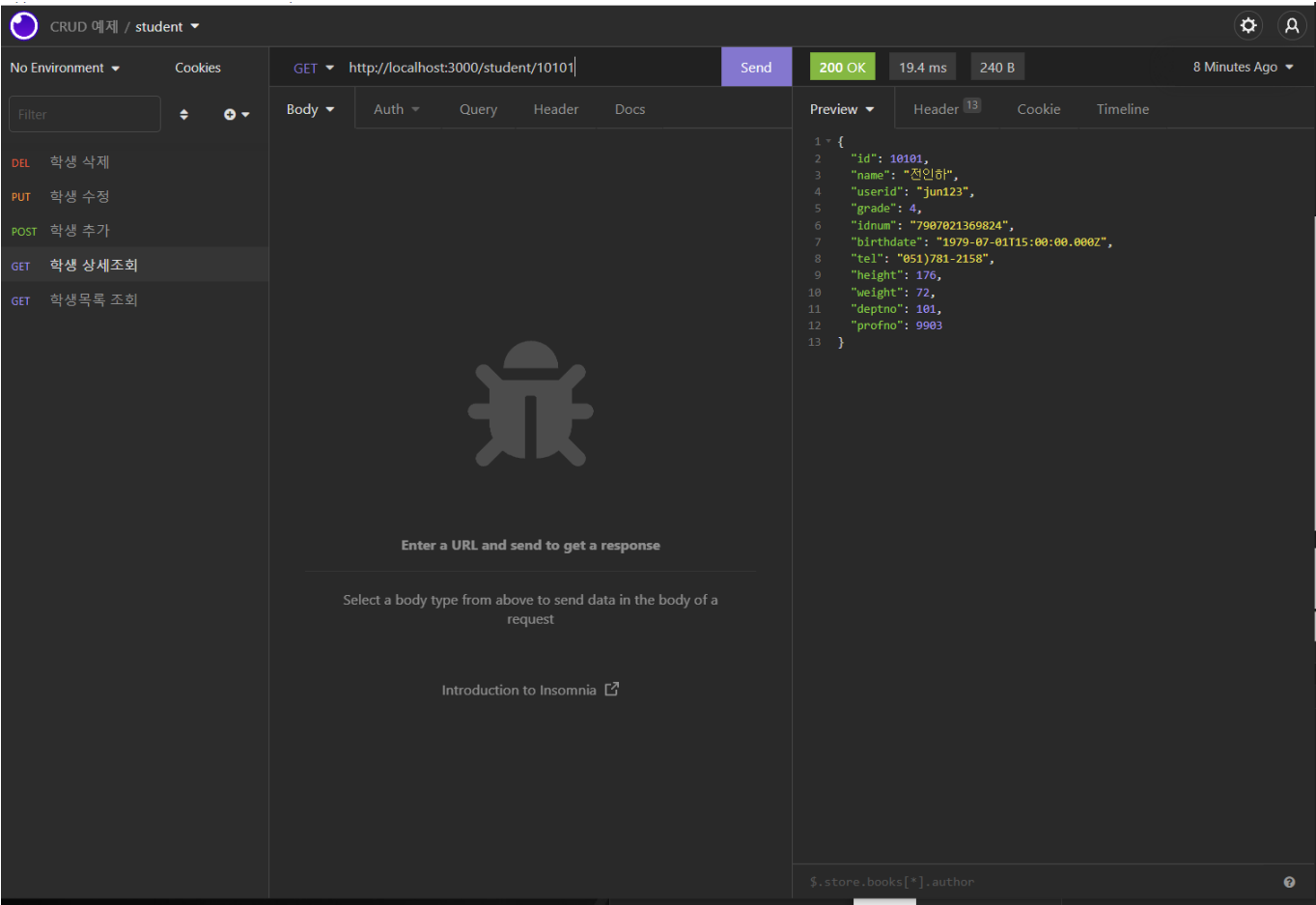
학생관리

insomnia 결과

학생목록조회(GET)



학생상세조회(GET/id)



학생추가(POST)

CRUD 예제 / student

No Environment Cookies

Filter

DEL 학생 삭제

PUT 학생 수정

POST 학생 추가

GET 학생 상세조회

GET 학생목록 조회

POST http://localhost:3000/student

Send

201 Created23.6 ms251 BJust Now

Form10AuthQueryHeader1Docs

name신지섭

useridpon03046

grade4

idnum9611011000000

birthdate1986-11-01T15:00:00.000Z

tel01012345678

height170

weight75

deptno101

profno9905

New nameNew value

Preview

Header15CookieTimeline

```
1 {
2   "name": "신지섭",
3   "userid": "pon03046",
4   "grade": "4",
5   "idnum": "9611011000000",
6   "birthdate": "1986-11-01T15:00:00.000Z",
7   "tel": "01012345678",
8   "height": "170",
9   "weight": "75",
10  "deptno": "101",
11  "profno": "9905",
12  "id": 20105
13 }
```

CRUD 예제 / student

No Environment Cookies

Filter

DEL 학생 삭제

PUT 학생 수정

POST 학생 추가

GET 학생 상세조회

GET 학생목록 조회

GET http://localhost:3000/student

Send

200 OK21.8 ms4.5 KBJust Now

BodyAuthQueryHeaderDocs

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

Preview

Header13CookieTimeline

```
182   "profno": null
183 },
184 {
185   "id": 20103,
186   "name": "김진영",
187   "userid": "lovely",
188   "grade": 2,
189   "idnum": "8302282169387",
190   "birthdate": "1983-02-27T15:00:00.000Z",
191   "tel": "052)175-3941",
192   "height": 166,
193   "weight": 51,
194   "deptno": 201,
195   "profno": 9902
196 },
197 {
198   "id": 20104,
199   "name": "조명훈",
200   "userid": "rader214",
201   "grade": 1,
202   "idnum": "8412141254963",
203   "birthdate": "1984-12-13T15:00:00.000Z",
204   "tel": "02)785-6984",
205   "height": 184,
206   "weight": 62,
207   "deptno": 201,
208   "profno": null
209 },
210 {
211   "name": "신지섭",
212   "userid": "pon03046",
213   "grade": "4",
214   "idnum": "9611011000000",
215   "birthdate": "1986-11-01T15:00:00.000Z",
216   "tel": "01012345678",
217   "height": "170",
218   "weight": "75",
219   "deptno": "101",
220   "profno": "9905",
221   "id": 20105
222 }
223 ]
```

학생수정(PUT)

19 / 37

CRUD 예제 / student

No Environment Cookies

PUT http://localhost:3000/student/20105 Send 200 OK 22.7 ms 53 B Just Now

Filter

DEL 학생 삭제
PUT 학생 수정
POST 학생 추가
GET 학생 상세조회
GET 학생목록 조회

Form Auth Query Header Docs

name 신지섭섭

name value

New name New value

Preview Header Cookie Timeline

```
1 {  
2   "name": "신지섭섭",  
3   "": "",  
4   "id": 20105  
5 }
```

\$.store.books[*].author

CRUD 예제 / student


No Environment Cookies

GET http://localhost:3000/student Send 200 OK 26.9 ms 4.3 KB Just Now

Filter

DEL 학생 삭제
PUT 학생 수정
POST 학생 추가
GET 학생 상세조회
GET 학생목록 조회

Body Auth Query Header Docs



Enter a URL and send to get a response

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

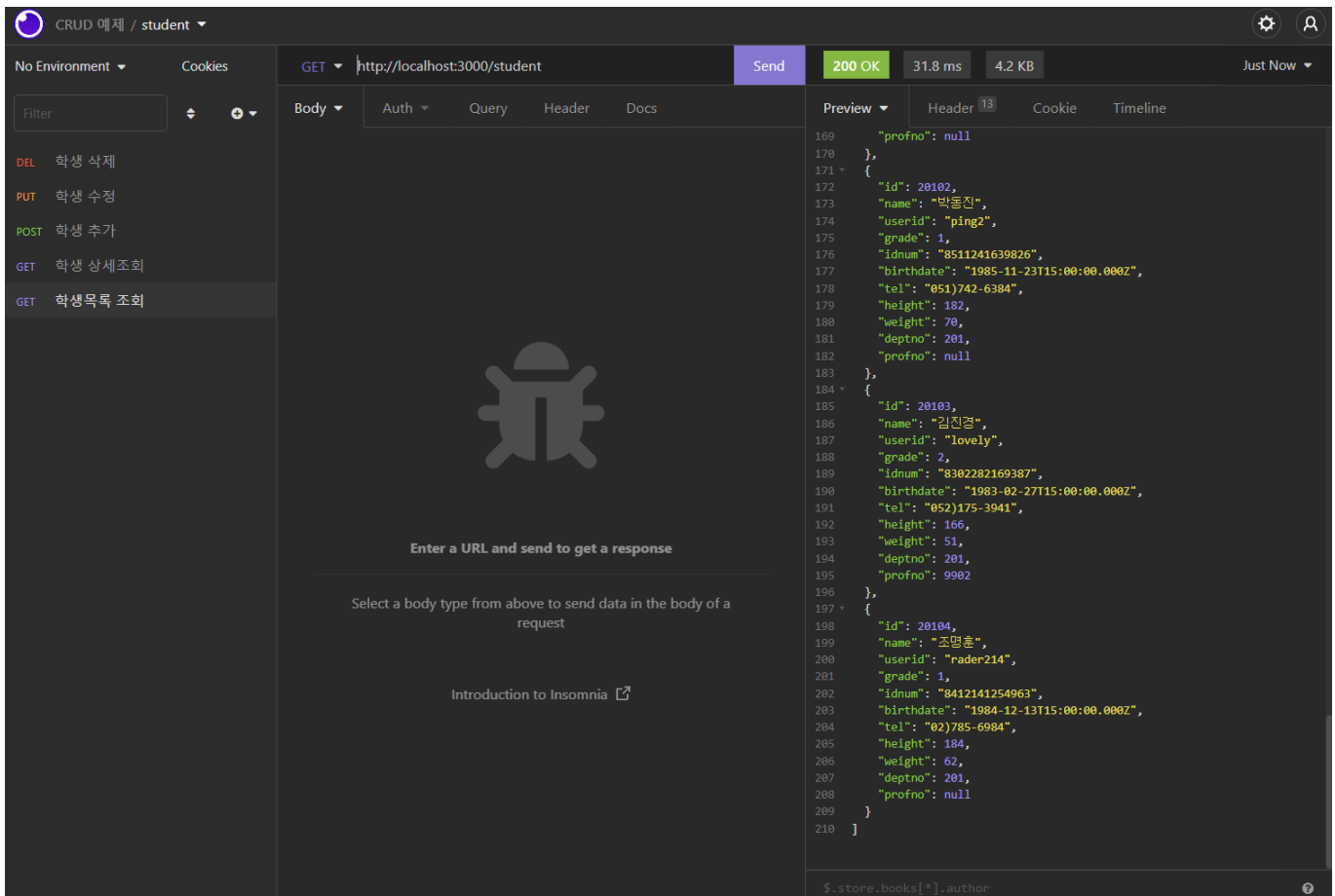
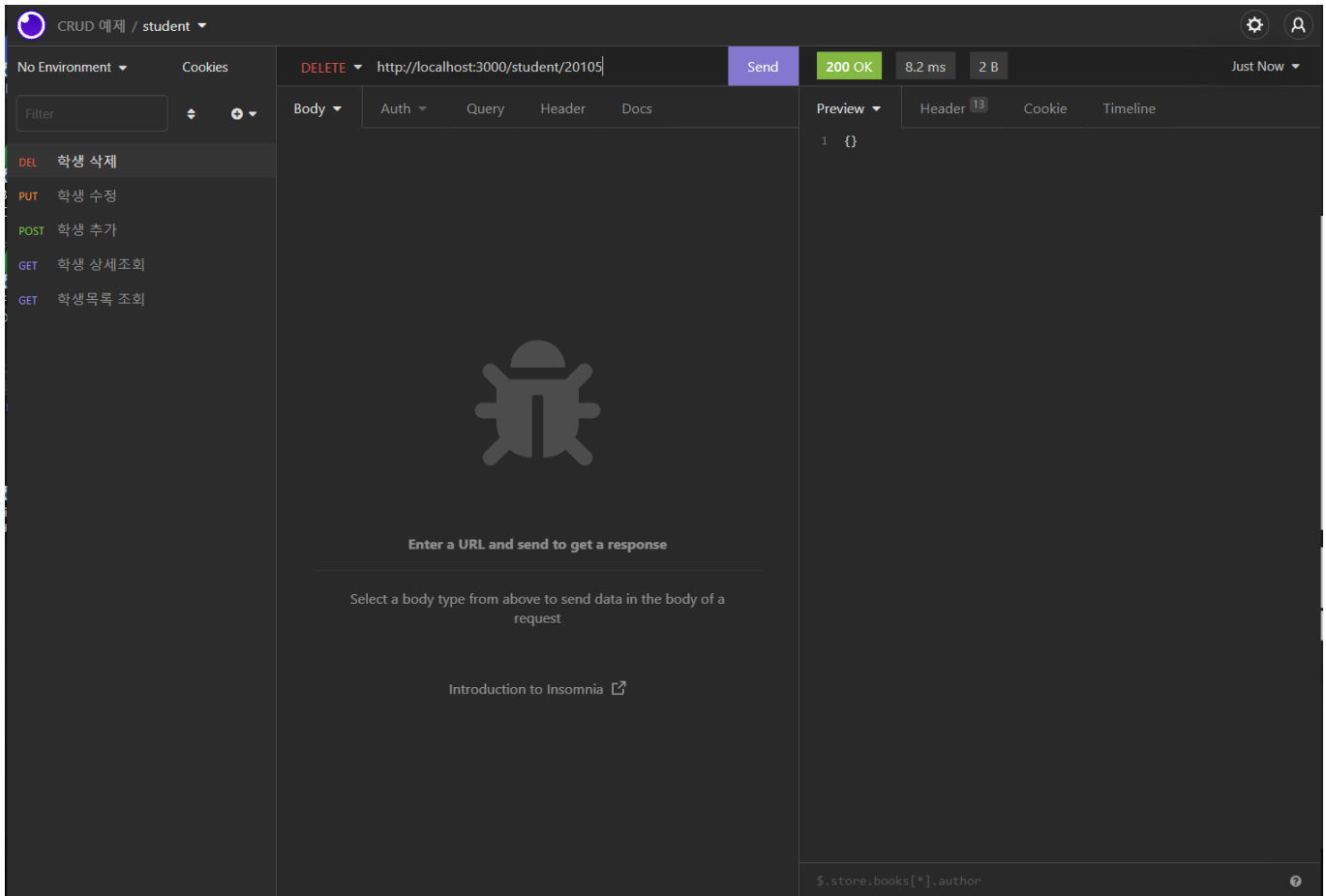
[Introduction to Insomnia](#)

Preview Header Cookie Timeline

```
174   "userid": "ping2",  
175   "grade": 1,  
176   "idnum": "8511241639826",  
177   "birthdate": "1985-11-23T15:00:00.000Z",  
178   "tel": "051)742-6384",  
179   "height": 182,  
180   "weight": 70,  
181   "deptno": 201,  
182   "profno": null  
183 },  
184 {  
185   "id": 20103,  
186   "name": "김진경",  
187   "userid": "lovely",  
188   "grade": 2,  
189   "idnum": "8302282169387",  
190   "birthdate": "1983-02-27T15:00:00.000Z",  
191   "tel": "052)175-3941",  
192   "height": 166,  
193   "weight": 51,  
194   "deptno": 201,  
195   "profno": 9902  
196 },  
197 {  
198   "id": 20104,  
199   "name": "조명훈",  
200   "userid": "rader214",  
201   "grade": 1,  
202   "idnum": "8412141254963",  
203   "birthdate": "1984-12-13T15:00:00.000Z",  
204   "tel": "02)785-6984",  
205   "height": 184,  
206   "weight": 62,  
207   "deptno": 201,  
208   "profno": null  
209 },  
210 {  
211   "name": "신지섭섭",  
212   "": "",  
213   "id": 20105  
214 }  
215 ]
```

\$.store.books[*].author

학생삭제(DELETE)



student 연습문제

index

```

//페이지가 실행되면서 자동으로 동작해야 하므로 즉시실행함수형태 구현.
(async () => {
  //ajax 결과가 저장될 json
  let json = null
  //ajax 요청
  try {
    json = await axios.get('http://localhost:3000/student')
  } catch(e) {
    console.error(e);
    alert('요청을 처리하는데 실패했습니다.');
```

return;

```

  }

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

    data.forEach((v, i) => {
      //내용답을 tr 생성
      const tr = document.createElement('tr');

      //id 내용 추가
      const td1 = document.createElement('td');
      td1.innerHTML = v.id
      //학생이름 내용 추가
      const td2 = document.createElement('td');
      const a1 = document.createElement('a');
      a1.setAttribute('href', `view.html?id=${v.id}`)
      a1.innerHTML = v.name
      td2.appendChild(a1);
      //아이디 내용 추가
      const td3 = document.createElement('td');
      td3.innerHTML = v.userid;
      //학년 내용 추가
      const td4 = document.createElement('td');
      td4.innerHTML = v.grade;
      //주민번호 내용 추가
      const td5 = document.createElement('td');
      td5.innerHTML = v.idnum.substring(0,6) + "-" + v.idnum.substring(6,7)
+ "*****";
      //생년월일 내용 추가
      const td6 = document.createElement('td');
      td6.innerHTML = v.birthdate.substring(0,10);
      //번호 내용 추가
      const td7 = document.createElement('td');
      td7.innerHTML = v.tel;
      //키 내용 추가
      const td8 = document.createElement('td');
      td8.innerHTML = v.height;
      //몸무게 내용 추가
      const td9 = document.createElement('td');
      td9.innerHTML = v.weight;
    });
  }
});

```

```

//학과번호
const td10 = document.createElement('td');
td10.innerHTML = v.deptno;
//담당교수의 일련번호
const td11 = document.createElement('td');
td11.innerHTML = v.profno;
//수정 및 삭제 추가
const td12 = document.createElement('td');
const a2 = document.createElement('a');
//수정
a2.setAttribute('href', `edit.html?id=${v.id}`);
a2.innerHTML = '수정'
td12.appendChild(a2)
//삭제
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);

//tr에 모든내용 추가
tr.appendChild(td1);
tr.appendChild(td2);
tr.appendChild(td3);
tr.appendChild(td4);
tr.appendChild(td5);
tr.appendChild(td6);
tr.appendChild(td7);
tr.appendChild(td8);
tr.appendChild(td9);
tr.appendChild(td10);
tr.appendChild(td11);
tr.appendChild(td12);

//최종결과물 listBody에 추가
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 ${name}(을)를 삭제하시겠습니까?`)){
        return;
    }
    let json = null
    try {
        json = await
    axios.delete(`http://localhost:3000/student/${id}`);
    } catch(e) {

```

```
        console.error(e);
        alert('요청을 처리하는데 실패했습니다.')
        return;
    }

    if(json != null) {
        current.closest('tr').remove();
    }
    })
    })
    }
    })();
```


학생관리

[새 학생 추가](#)

학생번호	이름	아이디	학년	주민번호	생년월일	전화번호	키	몸무게	학과번호	담당교수의 일련번호	-
10101	전인하	jun123	4	790702-1*****	1979-07-01	051)781-2158	176	72	101	9903	수정 삭제
10102	박미경	ansel414	1	840516-2*****	1984-05-15	055)261-8947	168	52	101		수정 삭제
10103	김영균	mandu	3	810321-1*****	1981-03-20	051)824-9637	170	88	101	9906	수정 삭제
10104	지은경	gomo00	2	800412-2*****	1980-04-11	055)418-9627	161	42	101	9907	수정 삭제
10105	임유진	youjin12	2	830121-2*****	1983-01-20	02)312-9838	171	54	101	9907	수정 삭제
10106	서재진	seolly	1	851129-1*****	1985-11-28	051)239-4861	186	72	101		수정 삭제
10107	이광훈	huriky	4	810913-1*****	1981-09-12	055)736-4981	175	92	101	9903	수정 삭제
10108	류민정	cleansky	2	810819-2*****	1981-08-18	055)248-3679	162	72	101	9907	수정 삭제
10201	김진영	simply	2	820606-2*****	1982-06-05	055)419-6328	164	48	102	9905	수정 삭제
10202	오유석	yousuk	4	770912-1*****	1977-09-11	051)724-9618	177	92	102	9905	수정 삭제
10203	하나리	hanal	1	850109-2*****	1985-01-08	055)296-3784	160	68	102		수정 삭제
10204	윤진욱	samba7	3	790402-1*****	1979-04-01	053)487-2698	171	70	102	9905	수정 삭제
20101	이동훈	dals	1	831210-1*****	1983-12-09	055)426-1752	172	64	201		수정 삭제
20102	박동진	ping2	1	851124-1*****	1985-11-23	051)742-6384	182	70	201		수정 삭제
20103	김진경	lovely	2	830228-2*****	1983-02-27	052)175-3941	166	51	201	9902	수정 삭제
20104	조명훈	rader214	1	841214-1*****	1984-12-13	02)785-6984	184	62	201		수정 삭제

copyright©2022. ezenac.co.kr

학생관리

127.0.0.1:5500 내용:

정말 신지섭(을)을 삭제하시겠습니까?

[확인](#)

[취소](#)

[새 학생 추가](#)

학생번호	이름	아이디	학년	주민번호	생년월일	전화번호	키	몸무게	학과번호	담당교수의 일련번호	-
10101	전인하	jun123	4	790702-1*****	1979-07-01	051)781-2158	176	72	101	9903	수정 삭제
10102	박미경	ansel414	1	840516-2*****	1984-05-15	055)261-8947	168	52	101		수정 삭제
10103	김영균	mandu	3	810321-1*****	1981-03-20	051)824-9637	170	88	101	9906	수정 삭제
10104	지은경	gomo00	2	800412-2*****	1980-04-11	055)418-9627	161	42	101	9907	수정 삭제
10105	임유진	youjin12	2	830121-2*****	1983-01-20	02)312-9838	171	54	101	9907	수정 삭제
10106	서재진	seolly	1	851129-1*****	1985-11-28	051)239-4861	186	72	101		수정 삭제
10107	이광훈	huriky	4	810913-1*****	1981-09-12	055)736-4981	175	92	101	9903	수정 삭제
10108	류민정	cleansky	2	810819-2*****	1981-08-18	055)248-3679	162	72	101	9907	수정 삭제
10201	김진영	simply	2	820606-2*****	1982-06-05	055)419-6328	164	48	102	9905	수정 삭제
10202	오유석	yousuk	4	770912-1*****	1977-09-11	051)724-9618	177	92	102	9905	수정 삭제
10203	하나리	hanal	1	850109-2*****	1985-01-08	055)296-3784	160	68	102		수정 삭제
10204	윤진욱	samba7	3	790402-1*****	1979-04-01	053)487-2698	171	70	102	9905	수정 삭제
20101	이동훈	dals	1	831210-1*****	1983-12-09	055)426-1752	172	64	201		수정 삭제
20102	박동진	ping2	1	851124-1*****	1985-11-23	051)742-6384	182	70	201		수정 삭제
20103	김진경	lovely	2	830228-2*****	1983-02-27	052)175-3941	166	51	201	9902	수정 삭제
20104	조명훈	rader214	1	841214-1*****	1984-12-13	02)785-6984	184	62	201		수정 삭제
20105	신지섭	pon0304616	2	961101-1*****	1996-11-01	01012345678	171	76	102		수정 삭제

copyright©2022. ezenac.co.kr

학생관리

[새 학생 추가](#)

학생번호	이름	아이디	학년	주민번호	생년월일	전화번호	키	몸무게	학과번호	담당교수의 일련번호	-
10101	전인하	jun123	4	790702-1*****	1979-07-01	051)781-2158	176	72	101	9903	수정 삭제
10102	박미경	ansel414	1	840516-2*****	1984-05-15	055)261-8947	168	52	101		수정 삭제
10103	김영균	mandu	3	810321-1*****	1981-03-20	051)824-9637	170	88	101	9906	수정 삭제
10104	지은경	gomo00	2	800412-2*****	1980-04-11	055)418-9627	161	42	101	9907	수정 삭제
10105	임유진	youjin12	2	830121-2*****	1983-01-20	02)312-9838	171	54	101	9907	수정 삭제
10106	서재진	seolly	1	851129-1*****	1985-11-28	051)239-4861	186	72	101		수정 삭제
10107	이광훈	huriky	4	810913-1*****	1981-09-12	055)736-4981	175	92	101	9903	수정 삭제
10108	류민정	cleansky	2	810819-2*****	1981-08-18	055)248-3679	162	72	101	9907	수정 삭제
10201	김진영	simply	2	820606-2*****	1982-06-05	055)419-6328	164	48	102	9905	수정 삭제
10202	오유석	yousuk	4	770912-1*****	1977-09-11	051)724-9618	177	92	102	9905	수정 삭제
10203	하나리	hanal	1	850109-2*****	1985-01-08	055)296-3784	160	68	102		수정 삭제
10204	윤진욱	samba7	3	790402-1*****	1979-04-01	053)487-2698	171	70	102	9905	수정 삭제
20101	이동훈	dals	1	831210-1*****	1983-12-09	055)426-1752	172	64	201		수정 삭제
20102	박동진	ping2	1	851124-1*****	1985-11-23	051)742-6384	182	70	201		수정 삭제
20103	김진경	lovely	2	830228-2*****	1983-02-27	052)175-3941	166	51	201	9902	수정 삭제
20104	조명훈	rader214	1	841214-1*****	1984-12-13	02)785-6984	184	62	201		수정 삭제

copyright©2022. ezenac.co.kr

view

```
//페이지가 실행되면서 자동으로 동작해야 하므로 즉시실행함수형태로 구현.
(async () => {
  //GET 파라미터 받기
  //queryString값이 key-value형태로 객체로 분리됨;
  const params = new URLSearchParams(window.location.search);
  // 객체에서 key사용하여 값을 추출 --> 10101;
  const id = params.get('id');
  console.log(id)

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

  // ajax 요청
  let json = null

  try {
    json = await axios.get(`http://localhost:3000/student/${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.substring(0,6) +
        "-" + data.idnum.substring(6,7) + "*****";
        document.querySelector("#.birthdate").innerHTML =
        data.birthdate.substring(0,10);
        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;
    }
})();

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 = null;

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

    if (json != null) {
        window.location = 'index.html';
    }

```

```
    }  
  })
```

←

→

↺

127.0.0.1:5500/04-VanillaJS/CRUD연습문제stu/view.html?id=10101

🔗

☆

🔑

🔧

🛡️

💡

🔖

⚙️

🗑️

👤

⋮

학생관리

◎ 학생번호

| 10101

◎ 이름

| 전인하

◎ 아이디

| jun123

◎ 학년

| 4

◎ 주민번호

| 790702-1*****

◎ 생년월일

| 1979-07-01

◎ 전화번호

| 051)781-2158

◎ 키

| 176

◎ 몸무게

| 72

◎ 학과번호

| 101

◎ 담당교수의 일련번호

| 9903

[목록보기](#)

[수정하기](#)

[삭제하기](#)

copyright©2022. ezenac.co.kr

←

→

↺

127.0.0.1:5500/04-VanillaJS/CRUD연습문제stu/view.html?id=20105

🔗

☆

🔑

🔧

🛡️

💡

🔖

⚙️

🗑️

👤

⋮

학생관리

◎ 학생번호

| 20105

◎ 이름

| 신지섭

◎ 아이디

| pon0304616

◎ 학년

| 2

◎ 주민번호

| 961101-1*****

◎ 생년월일

| 1996-11-01

◎ 전화번호

| 01012345678

◎ 키

| 171

◎ 몸무게

| 76

◎ 학과번호

| 102

127.0.0.1:5500 내용:

정말 신지섭(을)를 삭제하시겠습니까?

확인

취소

29 / 37

© 담당교수의 일련번호

[목록보기](#) [수정하기](#) [삭제하기](#)

copyright©2022. ezenac.co.kr

← → ↺ 127.0.0.1:5500/04-VanillaJS/CRUD연습문제stu/index.html

🔍 ☆ 📌 📄 📁 📂 📅 📆 📇 📈 📉 📊 📋 📌 📍 📎 📏 📐 📑 📒 📓 📔 📕 📖 📗 📙 📚 📛 📞 📟 📠 📡 📢 📣 📤 📥 📦 📧 📨 📩 📪 📫 📬 📭 📮 📯 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📽 📾 📿 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📽 📾 📿

학생관리

[새 학생 추가](#)

학생번호	이름	아이디	학년	주민번호	생년월일	전화번호	키	몸무게	학과번호	담당교수의 일련번호	-
10101	전인하	jun123	4	790702-1*****	1979-07-01	051)781-2158	176	72	101	9903	수정 삭제
10102	박미경	ansel414	1	840516-2*****	1984-05-15	055)261-8947	168	52	101		수정 삭제
10103	김영균	mandu	3	810321-1*****	1981-03-20	051)824-9637	170	88	101	9906	수정 삭제
10104	지은경	gomo00	2	800412-2*****	1980-04-11	055)418-9627	161	42	101	9907	수정 삭제
10105	임유진	youjin12	2	830121-2*****	1983-01-20	02)312-9838	171	54	101	9907	수정 삭제
10106	서재진	seolly	1	851129-1*****	1985-11-28	051)239-4861	186	72	101		수정 삭제
10107	이광훈	huriky	4	810913-1*****	1981-09-12	055)736-4981	175	92	101	9903	수정 삭제
10108	류민정	cleansky	2	810819-2*****	1981-08-18	055)248-3679	162	72	101	9907	수정 삭제
10201	김진영	simply	2	820606-2*****	1982-06-05	055)419-6328	164	48	102	9905	수정 삭제
10202	오유석	yousuk	4	770912-1*****	1977-09-11	051)724-9618	177	92	102	9905	수정 삭제
10203	하나리	hanal	1	850109-2*****	1985-01-08	055)296-3784	160	68	102		수정 삭제
10204	윤진욱	samba7	3	790402-1*****	1979-04-01	053)487-2698	171	70	102	9905	수정 삭제
20101	이동훈	dals	1	831210-1*****	1983-12-09	055)426-1752	172	64	201		수정 삭제
20102	박동진	ping2	1	851124-1*****	1985-11-23	051)742-6384	182	70	201		수정 삭제
20103	김진경	lovely	2	830228-2*****	1983-02-27	052)175-3941	166	51	201	9902	수정 삭제
20104	조명훈	rader214	1	841214-1*****	1984-12-13	02)785-6984	184	62	201		수정 삭제

copyright©2022. ezenac.co.kr

add

```

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;

    let json = null;

    try {
        json = await axios.post('http://localhost:3000/student', {
            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;

    window.location = 'view.html?id=' + data.id
}
})
```

127.0.0.1:5500/04-VanillaJS/CRUD연습문제stu/add.html

☆

학생관리

이름 *

신지섭

아이디*

pon03046

학년*

4

주민번호*

961101000000

생년월일

1984-05-15T15:00:00.000Z

전화번호

01000000000

키

170

몸무게

75

학과번호

101

담당교수의 일련번호

저장하기

다시작성

copyright©2022. ezenac.co.kr

127.0.0.1:5500/04-VanillaJS/CRUD연습문제stu/view.html?id=20105

☆

학생관리

◎ 학생번호

20105

◎ 이름

신지섭

◎ 아이디

pon03046

◎ 학년

4

◎ 주민번호

961101-0*****

◎ 생년월일

1984-05-15

◎ 전화번호

01000000000

◎ 키

170

◎ 몸무게

75

◎ 학과번호

101

◎ 담당교수의 일련번호

[목록보기](#) [수정하기](#) [삭제하기](#)

copyright©2022. ezenac.co.kr

학생관리

[새 학생 추가](#)

학생번호	이름	아이디	학년	주민번호	생년월일	전화번호	키	몸무게	학과번호	담당교수의 일련번호	-
10101	전인하	jun123	4	790702-1*****	1979-07-01	051)781-2158	176	72	101	9903	수정 삭제
10102	박미경	ansel414	1	840516-2*****	1984-05-15	055)261-8947	168	52	101		수정 삭제
10103	김영균	mandu	3	810321-1*****	1981-03-20	051)824-9637	170	88	101	9906	수정 삭제
10104	지은경	gomo00	2	800412-2*****	1980-04-11	055)418-9627	161	42	101	9907	수정 삭제
10105	임유진	youjin12	2	830121-2*****	1983-01-20	02)312-9838	171	54	101	9907	수정 삭제
10106	서재진	seolly	1	851129-1*****	1985-11-28	051)239-4861	186	72	101		수정 삭제
10107	이광훈	huriky	4	810913-1*****	1981-09-12	055)736-4981	175	92	101	9903	수정 삭제
10108	류민정	cleansky	2	810819-2*****	1981-08-18	055)248-3679	162	72	101	9907	수정 삭제
10201	김진영	simply	2	820606-2*****	1982-06-05	055)419-6328	164	48	102	9905	수정 삭제
10202	오유석	yousuk	4	770912-1*****	1977-09-11	051)724-9618	177	92	102	9905	수정 삭제
10203	하나리	hanal	1	850109-2*****	1985-01-08	055)296-3784	160	68	102		수정 삭제
10204	윤진욱	samba7	3	790402-1*****	1979-04-01	053)487-2698	171	70	102	9905	수정 삭제
20101	이동훈	dals	1	831210-1*****	1983-12-09	055)426-1752	172	64	201		수정 삭제
20102	박동진	ping2	1	851124-1*****	1985-11-23	051)742-6384	182	70	201		수정 삭제
20103	김진경	lovely	2	830228-2*****	1983-02-27	052)175-3941	166	51	201	9902	수정 삭제
20104	조명훈	rader214	1	841214-1*****	1984-12-13	02)785-6984	184	62	201		수정 삭제
20105	신지섭	pon03046	4	961101-0*****	1984-05-15	01000000000	170	75	101		수정 삭제

copyright©2022. ezenac.co.kr

edit

```
//페이지가 실행되면서 자동으로 동작해야 하므로 즉시실행함수형태로 구현.
(async () => {
  const params = new URLSearchParams(window.location.search);
  const id = params.get('id');
  console.log(id)
  if (!id) {
    alert('학생번호가 없습니다. ');
    history.back();
    return;
  }

  //ajax요청
  let json = null
  try {
    json = await axios.get(`http://localhost:3000/student/${id}`);
  } catch(e) {
    console.error(e);
    alert('요청을 처리하는데 실패했습니다. ');
    return;
  }
  if (json !== null) {
    const {data} = json;
    //가져온 데이터가 있다면 <input> 태그의 value속성으로 가져온 값을 적용한다.
```

```

//일련번호와 같이 시스템이 관리하는 값은 hidden요소로 포함시킨다.
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;
    }
})();

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;

    let json = null;
    try {
        json = await axios.put(`http://localhost:3000/student/${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}`;
    }
}

```

```
}  
})
```

학생관리

이름 *

신지섭

아이디*

pon03046

학년*

4

주민번호*

961101000000

생년월일

1984-05-15T15:00:00.000Z

전화번호

01000000000

키

170

몸무게

75

학과번호

101

담당교수의 일련번호

수정하기

다시작성

학생관리

이름 *

신지섭

아이디*

pon0304616

학년*

4

주민번호*

9611011000000

생년월일

1996-11-01T15:00:00.000Z

전화번호

01012345678

키

171

몸무게

76

학과번호

102

담당교수의 일련번호

수정하기

다시작성

127.0.0.1:5500/04-VanillaJS/CRUD연습문제stu/view.html?id=20105

☆

학생관리

◎ 학생번호

20105

◎ 이름

신지섭

◎ 아이디

pon0304616

◎ 학년

2

◎ 주민번호

961101-1*****

◎ 생년월일

1996-11-01

◎ 전화번호

01012345678

◎ 키

171

◎ 몸무게

76

◎ 학과번호

102

◎ 담당교수의 일련번호

[목록보기](#)

[수정하기](#)

[삭제하기](#)

copyright©2022. ezenac.co.kr

학생관리

새 학생 추가

학생번호	이름	아이디	학년	주민번호	생년월일	전화번호	키	몸무게	학과번호	담당교수의 일련번호	-
10101	전인하	jun123	4	790702-1*****	1979-07-01	051)781-2158	176	72	101	9903	수정 삭제
10102	박미경	ansel414	1	840516-2*****	1984-05-15	055)261-8947	168	52	101		수정 삭제
10103	김영균	mandu	3	810321-1*****	1981-03-20	051)824-9637	170	88	101	9906	수정 삭제
10104	지은경	gomo00	2	800412-2*****	1980-04-11	055)418-9627	161	42	101	9907	수정 삭제
10105	임유진	yujin12	2	830121-2*****	1983-01-20	02)312-9838	171	54	101	9907	수정 삭제
10106	서재진	seolly	1	851129-1*****	1985-11-28	051)239-4861	186	72	101		수정 삭제
10107	이광훈	huriky	4	810913-1*****	1981-09-12	055)736-4981	175	92	101	9903	수정 삭제
10108	류민정	cleansky	2	810819-2*****	1981-08-18	055)248-3679	162	72	101	9907	수정 삭제
10201	김진영	simply	2	820606-2*****	1982-06-05	055)419-6328	164	48	102	9905	수정 삭제
10202	오유석	yousuk	4	770912-1*****	1977-09-11	051)724-9618	177	92	102	9905	수정 삭제
10203	하나리	hanal	1	850109-2*****	1985-01-08	055)296-3784	160	68	102		수정 삭제
10204	윤진욱	samba7	3	790402-1*****	1979-04-01	053)487-2698	171	70	102	9905	수정 삭제
20101	이동훈	dals	1	831210-1*****	1983-12-09	055)426-1752	172	64	201		수정 삭제
20102	박동진	ping2	1	851124-1*****	1985-11-23	051)742-6384	182	70	201		수정 삭제
20103	김진경	lovely	2	830228-2*****	1983-02-27	052)175-3941	166	51	201	9902	수정 삭제
20104	조명훈	rader214	1	841214-1*****	1984-12-13	02)785-6984	184	62	201		수정 삭제
20105	신지섭	pon0304616	2	961101-1*****	1996-11-01	01012345678	171	76	102		수정 삭제