

박세영 연습문제

문제1

```
<!--promise 방식으로 axios를 활용한 다중행 조회-->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    #loading{
      width: 100px;
      height: 100px;
      background-image: url(./loading.gif);
      background-repeat: no-repeat;
      background-position: center center;
      display: block;
      position: absolute;
      left: 50%;
      top: 50%;
      margin-left: -50px;
      margin-top: -50px;
      z-index: 99999;
      display: none;
    }
    #loading.active{
      display: block;
    }
  </style>
</head>
<body>
  <div id="loading"></div>
  <h1>promise 방식으로 axios를 활용한 다중행 조회</h1>
  <button id="btn" type="button">데이터 가져오기</button>
  <hr/>

  <table border="1">
    <thead>
      <tr>
        <th>학과 번호</th>
        <th>학과 명</th>
        <th>학과 위치</th>
      </tr>
    </thead>
    <tbody id="list-body"></tbody>
  </table>
```

```
<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
<script>
    document.querySelector('#btn').addEventListener('click', async (e)=> {
        //로딩바 표시하기
        const loading= document.querySelector('#loading');
        loading.classList.add('active');

        //HTTP의 4대 접속방식 GET,POST,PUT,DELETE가 각각 함수로 존재
        axios
        .get(`http://localhost:3000/department/`)
        // .then((response)=>{
        .then(({data}) =>{
            data.map((v, i)=> {
                const tr= document.createElement('tr');
                const td1=document.createElement('td');
                td1.innerHTML= v.id;

                const td2=document.createElement('td');
                td2.innerHTML= v.dname;

                const td3= document.createElement('td');
                td3.innerHTML= v.loc;

                tr.appendChild(td1);
                tr.appendChild(td2);
                tr.appendChild(td3);

                document.querySelector('#list-body').appendChild(tr);
            });
        })
        .catch((error)=> {
            console.error(error);
            console.error(error.response.status);
            console.error(error.response.statusText);
            console.error(error.response.data);
            alert(error.response.statusText);
        })
        .finally(()=> {
            //성공,실패여부에 상관없이 마지막에 실행되는 함수
            loading.classList.remove('active');
        });

    });
</script>

</body>
</html>
```

출력 결과


```
#loading.active{
  display: block;
}
</style>
</head>
<body>
  <div id="loading"></div>
  <h1>Dept List</h1>
  <button id="btn" type="button">데이터 가져오기</button>
  <hr/>

  <table border="1">
    <thead>
      <tr>
        <th>학과 번호</th>
        <th>학과 명</th>
        <th>학과 위치</th>
      </tr>
    </thead>
    <tbody id="list-body"></tbody>
  </table>

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

  <script>
    document.querySelector('#btn').addEventListener('click', async (e)=> {
      //로딩바 표시하기
      const loading= document.querySelector('#loading');
      loading.classList.add('active');

      //Ajax 응답결과를 저장할 json
      let json= null;
      try{
        json= await axios.get('http://localhost:3000/department');
        console.log(json);
      }catch(e){
        console.error(error);
        console.error(error.response.status);
        console.error(error.response.statusText);
        console.error(error.response.data);
        alert(error.response.statusText);
      }finally{
        //로딩바 닫기
        loading.classList.remove('active');
      }
      if(json!= null){
        json.data.map((v, i)=> {
          const tr= document.createElement('tr');
          const td1=document.createElement('td');
          td1.innerHTML= v.id;

          const td2=document.createElement('td');
          td2.innerHTML= v.dname;
```

```
const td3= document.createElement('td');
td3.innerHTML= v.loc;

tr.appendChild(td1);
tr.appendChild(td2);
tr.appendChild(td3);

document.querySelector('#list-body').appendChild(tr);
});
}
});
</script>

</body>
</html>
```

출력 결과

Dept List

데이터 가져오기

학과 번호	학과 명	학과 위치
101	컴퓨터공학과	1호관
102	멀티미디어학과	2호관
201	전자공학과	3호관
202	기계공학과	4호관
101	컴퓨터공학과	1호관
102	멀티미디어학과	2호관
201	전자공학과	3호관
202	기계공학과	4호관