

ex1) promise 방식으로 axios를 활용한 다중행 조회

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <style>
      #loading {
        width: 100px;
        height: 100px;
        background-image: url(img/loading.gif);
        background-size: cover;
        background-repeat: no-repeat;
        background-position: center center;
        display: block;
        position: absolute;
        left: 50%;
        top: 50%;
        margin-left: -50px;
        margin-top: -50px;
        z-index: 999999;
        display: none;
      }

      #loading.active {
        display: block;
      }
    </style>
  </head>
  <body>
    <div id="loading"></div>
    <h1>ex1</h1>

    <button id="btn" type="button">데이터 가져오기</button>

    <hr />

    <table border="1">
      <thead>
        <tr>
          <th>학과번호</th>
          <th>학과명</th>
          <th>학과위치</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", (e) => {

    // 로딩바 표시하기
    const loading = document.querySelector("#loading");
    loading.classList.add("active");

    axios
      .get("http://localhost:3000/department/")
      .then(({data}) => {
        const listBody = document.querySelector('#list-body');

        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);

          listBody.appendChild(tr);
        });

      })
      .catch((error) => {
        console.error(error);
        alert(error.response.statusText);
      })
      .finally(() => {
        // 성공, 실패 여부에 상관 없이 마지막에 실행되는 함수
        loading.classList.remove("active");
      });
  });
</script>
</body>
</html>
```

결과

ex1

데이터 가져오기

학과번호	학과명	학과위치
101	컴퓨터공학과	1호관
102	멀티미디어학과	2호관
201	전자공학과	3호관
202	기계공학과	4호관
101	컴퓨터공학과	1호관
102	멀티미디어학과	2호관
201	전자공학과	3호관
202	기계공학과	4호관

ex2) async~await 방식으로 axios를 활용한 다중행 조회

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <style>
      #loading {
        width: 100px;
        height: 100px;
        background-image: url(img/loading.gif);
        background-size: cover;
        background-repeat: no-repeat;
        background-position: center center;
        display: block;
        position: absolute;
        left: 50%;
        top: 50%;
        margin-left: -50px;
        margin-top: -50px;
        z-index: 999999;
        display: none;
      }

      #loading.active {
        display: block;
      }
    </style>
```

```
</head>
<body>
  <div id="loading"></div>
  <h1>ex2</h1>

  <button id="btn" type="button">데이터 가져오기</button>

  <hr />

  <table border="1">
    <thead>
      <tr>
        <th>학과번호</th>
        <th>학과명</th>
        <th>학과위치</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) => {
      e.preventDefault();

      const loading = document.querySelector("#loading");
      loading.classList.add("active");

      let json = null;

      try {
        json = await axios.get("http://localhost:3000/department");
      } catch (e) {
        console.error(e);
        console.error(e.response.status);
        alert(e.response.statusText);
      } finally {
        loading.classList.remove("active");
      }

      if (json != null) {
        const listBody = document.querySelector("#list-body");

        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);

        listBody.appendChild(tr);
    });
}
});
</script>
</body>
</html>
```

결과

ex2

데이터 가져오기

학과번호	학과명	학과위치
101	컴퓨터공학과	1호관
102	멀티미디어학과	2호관
201	전자공학과	3호관
202	기계공학과	4호관
101	컴퓨터공학과	1호관
102	멀티미디어학과	2호관
201	전자공학과	3호관
202	기계공학과	4호관