

# Exam 09

--- 학과선택 --- ▾

## 학생 목록

학번	이름	아이디	학년	주민번호	생년월일	연락처	키	몸무게	소속학과번호	담당교수번호
검색결과가 없습니다										

## 교수 목록

교수번호	이름	아이디	직급	급여	입사일	보직수당	소속학과번호
검색결과가 없습니다							

# Exam 09

전자공학과 ▾

## 학생 목록

학번	이름	아이디	학년	주민번호	생년월일	연락처	키	몸무게	소속학과번호	담당교수번호
20101	이동훈	dals	1	831210-*****	198312-10	055)426-1752	172	64	201	
20102	박동진	ping2	1	851124-*****	198511-24	051)742-6384	182	70	201	
20103	김진경	lovely	2	830228-*****	198302-28	052)175-3941	166	51	201	9902
20104	조명훈	rader214	1	841214-*****	198412-14	02)785-6984	184	62	201	

## 교수 목록

교수번호	이름	아이디	직급	급여	입사일	보직수당	소속학과번호
9902	이재우	sweat413	조교수	320	199504-12		201

## Exam 09

기계공학과 ▼

### 학생 목록

학번	이름	아이디	학년	주민번호	생년월일	연락처	키	몸무게	소속학과번호	담당교수번호
검색결과가 없습니다										

### 교수 목록

교수번호	이름	아이디	직급	급여	입사일	보직수당	소속학과번호
9908	남은혁	bird13	부교수	400	199010-18	17	202

### src/Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import { BrowserRouter } from 'react-router-dom';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>
);
```

### src/App.js

```
import React from 'react';
import axios from 'axios';
import Spinner from './components/Spinner';
import Professor from './components/Professor';
import Student from './components/Student';
const App = () => {
  const [loading, setLoading] = React.useState(false);
  const [department, setDepartment] = React.useState([]);
  const [deptno, setDeptno] = React.useState(-1);
```

```

/**페이지가 처음 열렸을때 실행할 Hook*/
React.useEffect(() => {
  // Ajax 로딩 시작을 알림
  setLoading(true);
  (async () => {
    try {
      const response = await
axios.get('http://localhost:3001/department');
      setDepartment((department) => response.data);
    } catch (e) {
      console.error(e);
      alert('Ajax 연동 실패');
    } finally {
      // Ajax 로딩 종료를 알림
      setLoading(false);
    }
  })();
}, []);
/** 드롭다운의 선택 상태가 변경되었을 때 호출될 이벤트 핸들러 */
const onSelectChanged = React.useCallback((e) => {
  e.preventDefault();
  // 드롭다운의 선택 결과를 상태값으로 등록함
  const current = e.target;
  const id = parseInt(current[current.selectedIndex].value);
  setDeptno(id);
}, []);
return (
  <div>
    <Spinner visible={loading} />
    <h1>Exam 09</h1>
    <hr />
    <select onChange={onSelectChanged}>
      <option value="-1">--- 학과선택 ---</option>
      {department.map((v, i) => (
        <option key={v.id} value={v.id}>
          {v.dname}
        </option>
      ))}
    </select>
    <Student deptno={deptno} />
    <Professor deptno={deptno} />
  </div>
);
};
export default React.memo(App);

```

## src/components/Spinner.js

```

import React from 'react';
import PropTypes from 'prop-types';

```

```

import styled from 'styled-components';
/** 로딩바 컴포넌트 */
// --> https://mhnpd.github.io/react-loader-spinner/
import { Bars } from 'react-loader-spinner';
/** 로딩바 뒤에 표시될 반투명 막 */
const TransLayer = styled.div`
  position: fixed;
  left: 0;
  top: 0;
  z-index: 9999;
  background-color: #0003;
  width: 100%;
  height: 100%;
`;
const Spinner = ({ visible, color, width, height }) => {
  return (
    <>
    {visible && (
      <TransLayer>
        <Bars
          color={color}
          height={width}
          width={height}
          wrapperStyle={{
            position: 'absolute',
            zIndex: 10000,
            left: '50%',
            top: '50%',
            marginLeft: -width / 2 + 'px',
            marginTop: -height / 2 + 'px',
          }}
        />
      </TransLayer>
    )}
    </>
  );
};
/** 기본값 정의 */
Spinner.defaultProps = {
  visible: false,
  color: '#06f',
  width: 100,
  height: 100,
};
/** 데이터 타입 설정 */
Spinner.propTypes = {
  visible: PropTypes.bool.isRequired,
  color: PropTypes.string,
  width: PropTypes.number,
  height: PropTypes.number,
};
// React.memo()를 사용하여 함수형 컴포넌트의 리렌더링 성능을 최적화
export default React.memo(Spinner);

```

## src/components/Professor.js

```
import React from 'react';
import axios from 'axios';
import dayjs from 'dayjs';
import PropTypes from 'prop-types';
import Spinner from './Spinner';
const Professor = ({ deptno }) => {
  const [loading, setLoading] = React.useState(false);
  const [professor, setProfessor] = React.useState([]);
  /** 페이지가 처음 열렸을 때와 검색어가 변경되었을 때 실행할 hook */
  React.useEffect(() => {
    if (deptno > -1) {
      // Ajax 로딩 시작을 알림
      setLoading(true);
      (async () => {
        try {
          const response = await
            axios.get('http://localhost:3001/professor', {
              params: {
                deptno: deptno,
              },
            });
          setProfessor((professor) => response.data);
        } catch (e) {
          console.error(e);
          alert('Ajax 연동 실패');
        } finally {
          // Ajax 로딩 종료를 알림
          setLoading(false);
        }
      })();
    } else {
      setProfessor([]);
    }
  }, [deptno]);
  return (
    <div>
      <Spinner visible={loading} />
      <h2>교수목록</h2>
      <table border="1" cellPadding="5">
        <thead>
          <tr>
            <th>교수번호</th>
            <th>이름</th>
            <th>아이디</th>
            <th>직급</th>
            <th>급여</th>
            <th>입사일</th>
            <th>보직수당</th>
          </tr>
        </thead>
      </table>
    </div>
  );
}
```

```

        <th>소속학과번호</th>
      </tr>
    </thead>
    <tbody>
      {professor.length > 0 ? (
        professor.map((v, i) => {
          return (
            <tr key={v.id} align="center">
              {Object.keys(v).map((w, j) => {
                let value = v[w];
                if (w === 'hiredate') {
                  value = dayjs(value).format('YYYYMM-DD');
                }
                return <td key={`_${v.id}_${j}`}>{value}</td>;
              })}
            </tr>
          );
        })
      ) : (
        <tr>
          <td colSpan="8" align="center">
            검색결과가 없습니다
          </td>
        </tr>
      )}
    </tbody>
  </table>
</div>
);
};
/** 기본값 정의 */
Professor.defaultProps = {
  deptno: -1,
};
/** 데이터 타입 설정 */
Professor.propTypes = {
  deptno: PropTypes.number.isRequired,
};
export default React.memo(Professor);

```

## src/components/Student.js

```

import React from 'react';
import axios from 'axios';
import dayjs from 'dayjs';
import PropTypes from 'prop-types';
import Spinner from './Spinner';

const Student = ({ deptno }) => {
  const [loading, setLoading] = React.useState(false);

```

```

const [student, setStudent] = React.useState([]);
/** 페이지가 처음 열렸을 때와 검색어가 변경되었을 때 실행할 hook */
React.useEffect(() => {
  if (deptno > -1) {
    // Ajax 로딩 시작을 알림
    setLoading(true);
    (async () => {
      try {
        const response = await
axios.get('http://localhost:3001/student', {
          params: {
            deptno: deptno,
          },
        });
        setStudent((student) => response.data);
      } catch (e) {
        console.error(e);
        alert('Ajax 연동 실패');
      } finally {
        // Ajax 로딩 종료를 알림
        setLoading(false);
      }
    })();
  } else {
    setStudent([]);
  }
}, [deptno]);
return (
  <div>
    <Spinner visible={loading} />
    <h2>학생목록</h2>
    <table border="1" cellPadding="5">
      <thead>
        <tr>
          <th>학번</th>
          <th>이름</th>
          <th>아이디</th>
          <th>학년</th>
          <th>주민번호</th>
          <th>생년월일</th>
          <th>연락처</th>
          <th>키</th>
          <th>몸무게</th>
          <th>소속학과번호</th>
          <th>담당교수번호</th>
        </tr>
      </thead>
      <tbody>
        {student.length > 0 ? (
          student.map((v, i) => {
            return (
              <tr key={v.id} align="center">
                {Object.keys(v).map((w, j) => {
                  let value = v[w];

```

```

        if (w === 'idnum') {
            value = value.substring(0, 6) + '-*****';
        } else if (w === 'birthdate') {
            value = dayjs(value).format('YYYYMM-DD');
        }
        return <td key={` ${v.id}_${j}`}>{value}</td>;
    })}
</tr>
    );
})
) : (
    <tr>
        <td colSpan="11" align="center">
            검색결과가 없습니다
        </td>
    </tr>
    )}
</tbody>
</table>
</div>
);
};
/** 기본값 정의 */
Student.defaultProps = {
    deptno: -1,
};
/** 데이터 타입 설정 */
Student.propTypes = {
    deptno: PropTypes.number.isRequired,
};
export default React.memo(Student);

```