

지산학 캡스톤 디자인 9/22 회의

지산학 캡스톤 디자인 9/22 회의

- 팀원: 엄경문, 이서준, 임호영, 이현노
- 회의내용

프로젝트 개요

- 학생이 성적표를 업로드하면 **자동으로 학점 분석 및 졸업요건 충족 여부 계산**
- 부족한 학점/필수과목을 파악하고, AI 기반으로 **맞춤형 과목 추천** 기능 제공
- 마이페이지를 통해 개인 수강 현황과 졸업 로드맵 확인 가능

기능별 논의

1. 회원가입/로그인

- 카카오 로그인 기반 간편 회원가입 도입
- 기본 계정 관리(로그인/로그아웃/탈퇴) 지원

2. 졸업요건 계산

- 학사일정 및 학과별 졸업요건 DB 구축 필요
- 사용자 입력 방식: PDF 업로드 + 직접 입력 혼합 지원
- 계산 항목: 전공필수/선택, 교양필수/선택, 공통교과(영어/설계 포함)

3. AI 과목 추천

- 부족 요건 기반 과목 우선 추천
- 사용자 관심사(선호 교수, 시간대, 과목 분야) 반영한 맞춤형 추천 로직 추가

4. 마이페이지

- 개인정보 수정, 비밀번호 변경/탈퇴 가능
- 수강 내역 조회 및 수정
- 졸업 조건 현황판 제공 (남은 학점, 필수과목 여부 표시)

5. 관리자 기능

- 학과별 졸업요건 등록/수정 페이지 제공
- 학년별 졸업요건 업데이트 가능 (PDF 업로드 기반)

추가로 기능명세서와 api 명세서 작성에 대한 논의도 진행하였다.

지산학 캡스톤 디자인 10/6 회의

기능명세서

☰ 표

🔍 분류	☰ 주기능	☰ 상세기능	☰ 설명	⚡ 중요도
📄 0. 회원가입	0.1 간편 회원가입	0.1.1 카카오로그인 기반		● 낮음
	0.2 로그인/로그아웃			● 낮음
📄 1. 졸업요건 계산	1.1 학사일정 DB 저장	1.1.1 크롤링/하드코딩 DB 구축	학교 졸업요건, 교양/전공 이수 조건	● 높음
	1.2 사용자 수강과목 입력	1.2.1 PDF 업로드		● 높음
		1.2.2 텍스트 입력	직접입력	● 높음
	1.3 졸업요건 비교	1.3.1 전공/교양 학점 계산	DB와 입력한 과목 비교	● 높음
		1.3.2 남은 졸업 요건 산출	전공/교양/필수이수 조건	● 높음
		1.4 결과 출력		● 높음
		1.4.1 전공필수과목 수료 충족		● 높음
		1.4.2 영어과목 수료 충족		● 높음
		1.4.3 설계과목 수료 충족		● 높음
		1.4.3 공교과목 수료 충족		● 높음
📄 2. AI 과목 추천	2.1 추천 로직 구현	2.1.1 부족한 요건 기준으로 추천	남은 전공,교양 추천 과목 제시	● 높음
	2.2 사용자 맞춤형 추천	2.2.1 관심사 기반 희망 수업 신청	선호 교수/과목/시간대(?)	● 높음
📄 3.마이페이지	3.1 개인정보	3.1.1 비밀번호 변경/탈퇴		● 낮음
	3.2 내 수강내역 관리	3.2.1 입력한 과목 확인/수정		● 높음
	3.3 졸업 조건 저장	3.3.1 현황판		● 보통
📄 4. 관리자 기능	4.1 학과별 졸업 요건 DB 등록/수정	4.1.1 학기별로 과목 업데이트		● 보통
	4.2 학년별 졸업 요건 관리	4.2.1 학년별 졸업요건 페이지	PDF파일을 올려워서 졸업요건하기	● 보통



API 명세서

표

기능	HTTP 주소	HTTP method	진행도
회원가입	/v1/auth/signup	POST	진행중
로그인	/v1/auth/login	POST	진행중
로그아웃	/v1/auth/logout	POST	진행중
회원탈퇴	v1/auth/user/me	DELETE	진행중
학사일정 DB 저장	/v1/requirements/db	POST	진행전
사용자 수강과목 입력	/v1/courses/pdf/{userid}	POST	진행전
사용자 수강과목 변경	/v1/courses/pdf/{userid}	PATCH	진행전
사용자 수강과목 직접 입력	/v1/courses	POST	진행전
졸업요건 비교	/v1/requirements/{userid}/com	GET	진행전
결과 출력	/v1/requirements/{userid}/resu	GET	진행전
추천 로직 구현	/v1/recommendations	GET	진행전
사용자 맞춤형 추천	/v1/recommendations/custom	GET	진행전
개인정보확인	/v1/mypage/info/{userid}	GET	진행전
개인정보변경	/v1/mypage/info/{userid}	PATCH	진행전
내 수강내역 확인	/v1/mypage/courses/{userid}	GET	진행전
내 수강내역 변경	/v1/mypage/courses/{userid}	PATCH	진행전
저장된 졸업 조건 조회	/v1/mypage/requirements	GET	진행전
개인 졸업 조건 저장	/v1/mypage/requirements/{use	POST	진행전
과별 졸업 요건 등록	/v1/admin/requirements/{depa	POST	진행전
과별 졸업 조건 수정	/v1/admin/requirements/{depa	PATCH	진행전
학년별 졸업 요건 페이지 조회	/v1/admin/requirements/yearly	GET	진행전

지산학 캡스톤 디자인 10/20 회의(피드백)

- DB 저장하는 형태 (연도별 VS 통합) → 통합으로 하고 변경사항만 따로 테이블 구성
- 과목 추천 AI → 랭그래프로 졸업 요건을 트리 형태로 구성. 여기에 맞춰서 컨설팅. 랭그래프 + LLM
- 테스트용 페이지
-

지산학 캡스톤 디자인 10/27 (피드백_이길섭교수님)

- DB에 어떻게 넣을지
- 확장기능 기가맥힌거 하나 선정해서 밀고가기
- 수행 계획서
 - 개요
 - 이 프로젝트를 하게 된 배경
 - 학생들이 어려움을 겪는 점 다듬기
 - 기대효과에 비교과 활동 관리 추가
 - 추진 배경
 - 배경까지는 잘 정리 됐는데, 필요성 추가
 - 타 대학교 서비스 참고
 - 선행 사례 분석해서 차별점 밀고 가기
 - 목표 및 내용
 - 선행 사례를 분석해서 목표 확실히 잡기
 - 목표별로 개발내용 세부사항 설립
 -

지산학 캡스톤 디자인 11/17 (교무처 회의)

- 취업자-미취업자 역량의 상관관계
 - 분석방법 : 스피어만 상관관계 등
 - 분석내용 : 취업자-미취업자의 역량(마일리지&자가진단) 특성
- 취업자에게 유의미한 이수 교과목
 - 분석방법 : 카이제곱, 랜덤포레스트 등
 - 분석내용 : 미취업자 대비 취업자에게 유의미한 교과목

허깅페이스 랭그래프

```
{
  "student": {
    "id": "2021112042",
    "program": "정보통신공학전공",
    "catalog_year": 2021,
    "flags": { "toeic": 0 }
  },
  "completed": {
    "total_credits": 100,
    "table_rows": [
      { "course_no": "RGC0017", "course_name": "자아와명상1", "credits": 1,
        "area": "필수교양", "english_yn": "N" },
      { "course_no": "RGC0005", "course_name": "기술보고서작성및발표", "credits": 3,
        "area": "필수교양", "english_yn": "N" },
      { "course_no": "RGC1074", "course_name": "커리어 디자인", "credits": 1,
        "area": "필수교양", "english_yn": "N" },
      { "course_no": "RGC0018", "course_name": "자아와명상2", "credits": 1,
        "area": "필수교양", "english_yn": "N" },
      { "course_no": "RGC0003", "course_name": "불교와인간", "credits": 2, "area": "필수교양", "english_yn": "N" },
      { "course_no": "RGC1052", "course_name": "테크노앙트레프레너십과리더십", "credits": 2, "area": "필수교양", "english_yn": "N" },
      { "course_no": "RGC1046", "course_name": "인공지능과미래사회", "credits": 2, "area": "필수교양", "english_yn": "N" },
      { "course_no": "RGC1080", "course_name": "EAS1", "credits": 2, "area": "필수교양", "english_yn": "Y" },

      { "course_no": "EGC3044", "course_name": "북한의이해", "credits": 3, "area": "일반교양", "english_yn": "N" },
      { "course_no": "EGC4040", "course_name": "인공지능프로그래밍기초와실습", "credits": 3, "area": "일반교양", "english_yn": "N" },
```



```

    { "course_no": "EGC9003", "course_name": "예술과경영의만남", "credits": 2, "area": "일반교양", "english_yn": "N" },
    { "course_no": "EGC9006", "course_name": "인도불교이야기", "credits": 1, "area": "일반교양", "english_yn": "N" },
    { "course_no": "EGC7026", "course_name": "기술창조와특허", "credits": 3, "area": "일반교양", "english_yn": "N" },
    { "course_no": "EGC4039", "course_name": "공학윤리", "credits": 3, "area": "일반교양", "english_yn": "N" },

    { "course_no": "PRI4001", "course_name": "미적분학및연습1", "credits": 3, "area": "msc", "english_yn": "N" },
    { "course_no": "PRI4002", "course_name": "일반물리학및실험1", "credits": 4, "area": "msc", "english_yn": "N" },
    { "course_no": "PRI4035", "course_name": "프로그래밍기초와실습", "credits": 3, "area": "msc", "english_yn": "N" },
    { "course_no": "PRI4012", "course_name": "미적분학및연습2", "credits": 3, "area": "msc", "english_yn": "N" },
    { "course_no": "PRI4013", "course_name": "일반물리학및실험2", "credits": 4, "area": "msc", "english_yn": "N" },
    { "course_no": "PRI4023", "course_name": "확률및통계학", "credits": 3, "area": "msc", "english_yn": "N" },
    { "course_no": "PRI4039", "course_name": "인터넷프로그래밍", "credits": 3, "area": "msc", "english_yn": "N" },
    { "course_no": "PRI4051", "course_name": "산업수학", "credits": 3, "area": "msc", "english_yn": "N" },

    { "course_no": "INC2034", "course_name": "ICT와소프트웨어", "credits": 3, "area": "전공", "english_yn": "N" },
    { "course_no": "INC2031", "course_name": "정보통신프로그래밍", "credits": 3, "area": "전공", "english_yn": "N" },
    { "course_no": "INC2033", "course_name": "어드벤처디자인", "credits": 3, "area": "전공", "english_yn": "N" },
    { "course_no": "INC2025", "course_name": "정보통신수학및실습", "credits": 3, "area": "전공", "english_yn": "N" },
    { "course_no": "INC2027", "course_name": "자료구조와실습", "credits": 3, "area": "전공", "english_yn": "N" },
    { "course_no": "INC2024", "course_name": "컴퓨터알고리즘및실습", "credits": 3, "area": "전공", "english_yn": "Y" },

```

```

    { "course_no": "INC2028", "course_name": "컴퓨터구성", "credits": 3, "area": "전공", "english_yn": "N" },
    { "course_no": "INC2029", "course_name": "객체지향언어와실습", "credits": 3, "area": "전공", "english_yn": "Y" },
    { "course_no": "INC2032", "course_name": "신호와시스템", "credits": 3, "area": "전공", "english_yn": "N" },
    { "course_no": "DES2001", "course_name": "커리어멘토링", "credits": 3, "area": "전공", "english_yn": "N" },
    { "course_no": "INC4058", "course_name": "데이터통신", "credits": 3, "area": "전공", "english_yn": "N" },
    { "course_no": "INC4059", "course_name": "운영체제", "credits": 3, "area": "전공", "english_yn": "N" },
    { "course_no": "INC4061", "course_name": "디지털신호처리", "credits": 3, "area": "전공", "english_yn": "N" },
    { "course_no": "INC4110", "course_name": "통신이론", "credits": 3, "area": "전공", "english_yn": "N" },
    { "course_no": "INC4116", "course_name": "임베디드소프트웨어와스마트모빌리티", "credits": 3, "area": "전공", "english_yn": "Y" }
  ]
}

```

- 일물실/일화실 → 개론 X (이거는 로직비교가 무의미할듯)
- 영어성적 S4→BASIC 듣고 학점짜리
- S0 → 토익 900이상 안들어도됨
- 선이수 과목
- 설계과목(전공 설계): 어벤디, 컴퓨터 구성, 객체지향,컴퓨터알고리즘, 운영체제,임베디드, 모바일 소프트웨어, 마이크로 프로세서, 데이터베이스 체제,디지털 통신, 소프트웨어 공학, 캡스톤 디자인 1,2
- 전공필수, 필수교양,필수 MSC

```

{
  "pass": false,

```

```
"summary": {
  "earned_credits": 100,
  "required_credits": 130
},
"by_rule": [
  {
    "rule_id": "TOTAL_CR",
    "type": "min_credits",
    "ok": false,
    "required": 130,
    "current": 100,
    "missing": 30
  },
  {
    "rule_id": "MAJOR_CR",
    "type": "category_min_credits",
    "category": "major",
    "category_label": "전공",
    "ok": false,
    "required": 60,
    "current": 45,
    "missing": 15,
    "suggest": {
      "take_more_credits_in": "major",
      "take_more_credits_in_label": "전공",
      "missing_credits": 15
    }
  },
  {
    "rule_id": "MSC_CR",
    "type": "category_min_credits",
    "category": "msc",
    "category_label": "MSC/기초",
    "ok": false,
    "required": 30,
    "current": 26,
    "missing": 4,
    "suggest": {
```

```

    "take_more_credits_in": "msc",
    "take_more_credits_in_label": "MSC/기초",
    "missing_credits": 4
  }
},
{
  "rule_id": "ENG_TOTAL_WITH_MAJOR_MIN",
  "type": "english_total_with_major_min",
  "ok": true,
  "required": {
    "min_total": 4,
    "min_major": 2
  },
  "current": {
    "total": 4,
    "major": 3
  },
  "missing": {
    "total": 0,
    "major": 0
  },
  "exempted": false,
  "suggest": null
},
{
  "rule_id": "ENGR_2OF3",
  "type": "at_least_k_from_list",
  "ok": true,
  "required_min_count": 2,
  "current_count": 2,
  "missing": 0,
  "taken_courses": [
    "EGC4039",
    "EGC7026"
  ],
  "taken_course_names": [
    "공학윤리",
    "기술창조와특허"
  ]
}

```

```

],
"remaining_courses": [
  "PRI4041"
],
"remaining_course_names": [
  null
],
"exempted": false
},
{
  "rule_id": "ENTR_10F3_LE_2022",
  "type": "at_least_k_from_list",
  "ok": true,
  "required_min_count": 1,
  "current_count": 1,
  "missing": 0,
  "taken_courses": [
    "RGC1052"
  ],
  "taken_course_names": [
    "테크노앙트레프레너십과리더십"
  ],
  "remaining_courses": [
    "RGC1050",
    "RGC1051"
  ],
  "remaining_course_names": [
    null,
    null
  ],
  "exempted": false
},
{
  "rule_id": "MSC_MATH_ALL",
  "type": "must_take_all_from_list",
  "label": "MSC 수학(미적분1·2, 산업수학)",
  "ok": true,
  "required_courses": [

```

```

    "PRI4001",
    "PRI4012",
    "PRI4051"
  ],
  "required_course_names": [
    "미적분학및연습1",
    "미적분학및연습2",
    "산업수학"
  ],
  "missing_courses": [],
  "missing_course_names": [],
  "missing_count": 0
},
{
  "rule_id": "MSC_SCI_PAIR",
  "type": "k_complete_groups",
  "ok": true,
  "required_min_groups": 1,
  "current_completed_groups": 1,
  "completed_group_indices": [
    0
  ],
  "groups": [
    {
      "index": 0,
      "required_courses": [
        "PRI4002",
        "PRI4013"
      ],
      "required_course_names": [
        "일반물리학및실험1",
        "일반물리학및실험2"
      ],
      "missing_courses": [],
      "missing_course_names": []
    },
    {
      "index": 1,

```

```

    "required_courses": [
      "PRI4003",
      "PRI4014"
    ],
    "required_course_names": [
      null,
      null
    ],
    "missing_courses": [
      "PRI4003",
      "PRI4014"
    ],
    "missing_course_names": [
      null,
      null
    ]
  },
  {
    "index": 2,
    "required_courses": [
      "PRI4004",
      "PRI4015"
    ],
    "required_course_names": [
      null,
      null
    ],
    "missing_courses": [
      "PRI4004",
      "PRI4015"
    ],
    "missing_course_names": [
      null,
      null
    ]
  }
],
"exempted": false

```

```

},
{
  "rule_id": "MSC_PROG_BASIC",
  "type": "must_take_all_from_list",
  "label": "프로그래밍기초와실습",
  "ok": true,
  "required_courses": [
    "PRI4035"
  ],
  "required_course_names": [
    "프로그래밍기초와실습"
  ],
  "missing_courses": [],
  "missing_course_names": [],
  "missing_count": 0
},
{
  "rule_id": "MAJOR_MUST_ALL",
  "type": "must_take_all_from_list",
  "label": "전공필수(학과 지정)",
  "ok": true,
  "required_courses": [
    "INC2027",
    "INC2028",
    "INC2032",
    "INC2033",
    "INC4058",
    "INC4110"
  ],
  "required_course_names": [
    "자료구조와실습",
    "컴퓨터구성",
    "신호와시스템",
    "어드벤처디자인",
    "데이터통신",
    "통신이론"
  ],
  "missing_courses": [],

```



```

    "missing_course_names": [],
    "missing_count": 0
  },
  {
    "rule_id": "DESIGN_CR",
    "type": "design_min_credits",
    "ok": false,
    "required": 15,
    "current": 15,
    "missing": 0,
    "capstone_required": 1,
    "capstone_ok": false,
    "capstone_taken": [],
    "capstone_taken_names": [],
    "capstone_candidates": [
      "INC4084",
      "INC4085"
    ],
    "capstone_candidate_names": [
      null,
      null
    ],
    "capstone_missing": 1
  },
  {
    "rule_id": "CAPSTONE_PRE_REQ",
    "type": "capstone_requires_design_credits",
    "ok": true,
    "capstone_taken": [],
    "capstone_taken_names": [],
    "min_pre_credits": 12,
    "current_pre_design_credits": 15,
    "missing": 0
  },
  {
    "rule_id": "ENG_TEST_TOEIC_MIN700",
    "type": "min_exam_score",
    "ok": false,

```

```

    "key": "toeic",
    "required": 700,
    "current": 0,
    "missing": 700,
    "exempted": false
  }
],
"todo": {
  "must_take_courses": [],
  "required_groups": [],
  "categories": [
    {
      "category": "major",
      "category_label": "전공",
      "missing_credits": 15
    },
    {
      "category": "msc",
      "category_label": "MSC/기초",
      "missing_credits": 4
    }
  ],
  "english": [],
  "design": {
    "total": {
      "required_credits": 15,
      "current_credits": 15,
      "missing_credits": 0
    },
    "capstone_presence": {
      "required_at_least": 1,
      "taken": [],
      "taken_names": [],
      "candidates": [
        "INC4084",
        "INC4085"
      ],
      "candidate_names": [

```

```

        null,
        null
    ]
}
},
"tests": [
{
    "rule_id": "ENG_TEST_TOEIC_MIN700",
    "key": "toEIC",
    "required": 700,
    "current": 0,
    "missing": 700
}
]
},
"echo": {
    "taken_course_codes": [
        "RGC0017",
        "RGC0005",
        "RGC1074",
        "RGC0018",
        "RGC0003",
        "RGC1052",
        "RGC1046",
        "RGC1080",
        "EGC3044",
        "EGC4040",
        "EGC9003",
        "EGC9006",
        "EGC7026",
        "EGC4039",
        "PRI4001",
        "PRI4002",
        "PRI4035",
        "PRI4012",
        "PRI4013",
        "PRI4023",
        "PRI4039",
    ]
}
}

```

```
"PRI4051",
"INC2034",
"INC2031",
"INC2033",
"INC2025",
"INC2027",
"INC2024",
"INC2028",
"INC2029",
"INC2032",
"DES2001",
"INC4058",
"INC4059",
"INC4061",
"INC4110",
"INC4116"
],
"taken_course_names": [
"자아와명상1",
"기술보고서작성및발표",
"커리어 디자인",
"자아와명상2",
"불교와인간",
"테크노앙트레프레너십과리더십",
"인공지능과미래사회",
"EAS1",
"북한의이해",
"인공지능프로그래밍기초와실습",
"예술과경영의만남",
"인도불교이야기",
"기술창조와특허",
"공학윤리",
"미적분학및연습1",
"일반물리학및실험1",
"프로그래밍기초와실습",
"미적분학및연습2",
"일반물리학및실험2",
"확률및통계학",
```

```

"인터넷프로그래밍",
"산업수학",
"ICT와소프트웨어",
"정보통신프로그래밍",
"어드벤처디자인",
"정보통신수학및실습",
"자료구조와실습",
"컴퓨터알고리즘및실습",
"컴퓨터구성",
"객체지향언어와실습",
"신호와시스템",
"커리어멘토링",
"데이터통신",
"운영체제",
"디지털신호처리",
"통신이론",
"임베디드소프트웨어와스마트모빌리티"
],
"excluded_course_codes": [],
"excluded_course_names": []
}
}

```

현재 코드기준으로는 이렇게반환인데 일단 선이수 조건 아직 안함 db들어오면함

```

{
  "major_courses": [
    "확률및랜덤프로세스",
    "컴퓨터알고리즘및실습",
    "정보통신수학및실습",
    "자료구조와실습",
    "컴퓨터구성",
    "객체지향언어와실습",
    "정보통신프로그래밍",
    "신호와시스템",
    "어드벤처디자인",
    "ICT와소프트웨어",
    "데이터베이스체제",

```

"데이터통신",
"운영체제",
"디지털신호처리",
"컴퓨터네트워크",
"네트워크보안",
"캡스톤디자인1",
"캡스톤디자인2",
"임베디드시스템",
"암(ARM)마이크로프로세서 어셈블리어프로그래밍",
"소프트웨어공학및설계",
"머신러닝",
"디지털영상처리",
"데이터압축",
"통신시스템소프트웨어",
"모바일소프트웨어",
"암호알고리즘및응용",
"딥러닝",
"네트워크프로그래밍",
"통신이론",
"디지털통신",
"무선통신",
"지능형IoT플랫폼및특허분석",
"지능형IoT시스템설계및지식재산권",
"회로이론1",
"회로이론2",
"전자회로1",
"시스템소프트웨어와 실습",
"프로그래밍언어개념",
"컴퓨터구조",
"객체지향설계와 패턴",
"모바일컴퓨팅",
"S/W 품질관리 및 테스트",
"인간컴퓨터상호작용시스템",
"공개SW프로젝트",
"SW비즈니스와 창업",
"테크니컬프리젠테이션",
"다학제캡스톤디자인",
"기술창업캡스톤디자인1",

```

    "기술창업캡스톤디자인2",
    "AI서비스기획",
    "지산학캡스톤디자인1",
    "지산학캡스톤디자인2",
    "커리어멘토링"
  ]
}

```

```

{
  "liberal_courses": [
    "심리학의거장을만나다:인간의생각에대한이해",
    "양반·사대부·사무라이:주제로읽는한·중·일비교 문화사",
    "인공지능윤리",
    "오랑캐의세계사:동아시아에서동부유라시아로",
    "동부유라시아와한국,갈등과교류의역사",
    "동국의역사와인물",
    "물품의글로벌히스토리",
    "금서의문화사",
    "역사학과빅데이터",
    "한국문화와불교",
    "사회리더를위한의사소통의원리와실제",
    "해외콘텐츠로보는근대불교문화",
    "만해한용운:함께길을걷다",
    "만해와불교의민족운동",
    "만해와근대지성",
    "근현대고승열전",
    "성장문학과나의자화상",
    "글로벌한국사",
    "삶과죽음의철학",
    "한미관계,협력과갈등의이중주",
    "만해의생애와사상",
    "21세기생명생태철학",
    "오리엔탈리즘과불교",
    "대학과청년문화",
    "세계문학속의인권",
    "글로벌에콜로지컬히스토리",
    "Introduction to History of Modern Art",
    "일본한자음쉽게이해하기",
  ]
}

```

"동아시아의역사와문화",
"인간과로봇의공존시대",
"한국문화낫설게보기",
"현대페미니즘의이해",
"불교와스토리텔링",
"동유라시아 세계 물품의 문명·문화사",
"이야기와 노래",
"지도 위의 문학",
"바다로 읽는 동아시아 문명문화사",
"시와예술적상상력",
"인간에 관하여",
"교유와 치유의 한시감상",
"문자로읽는아시아의문화와사상",
"영어권문화상품탐방",
"시사한국어1",
"시사한국어2",
"인도불교이야기",
"중국문자의이해",
"그리스문학과신화이야기",
"서구문학에나타난문제적인간읽기",
"행복학의인문학적문제들",
"현대요가와자기이해",
"LanugageandMind",
"인터넷정책과법",
"정보기술의이해",
"종교와법",
"인권의이해",
"영문학속의AI",
"광고홍보의이해",
"오페라속범죄학",
"글로벌시대의대중문화론",
"대학생을위한실용금융",
"자본주의의역사와전망",
"연애학개론",
"기업운영의기초:수요와공급의균형",
"소비인사이트",
"나의 생각: 나와 타인, 사회에 대한 인지",
"자녀와 소통하는 훈련중심 예비부모교육",

"포스트 코로나 시대와 일본의 서브컬처",
"사회적 경제와 대항 서사",
"주제와 이슈로 돌아보는 세계",
"교과서 밖의 심리학",
"여행지리와 도시산책",
"한일관계의과거,현재그리고미래",
"에너지산업과시장읽기",
"디지털시대의노동인류학",
"기술창조와특허",
"유라시아지역의정보기술의활용",
"관광과문화",
"동남아시아와불교문화",
"생명가치와바이오산업",
"환경과탄소",
"생활속의약",
"인간과우주",
"건강한삶을위한식생활",
"숲에서찾는바이오나노재료의이해",
"노벨상으로보는생명이야기",
"알기쉬운반도체",
"성의과학",
"미래사회와 식물",
"4차 산업혁명과 5G 무선통신",
"사람을위한미생물에의한미생물의환경공정",
"공학윤리",
"인공지능프로그래밍기초와실습",
"자연과학의융합적이해",
"파이썬프로그래밍기초",
"컴퓨터알고리즘의이해",
"로봇프로그래밍의세계",
"집합과수의세계로의초대",
"과학기술과사회",
"태양계와우주행성",
"현대자연과학과미래사회",
"골프",
"댄스스포츠",
"배드민턴",
"농구",

"호신술",
"태권도",
"음악즐기기",
"부조리문학",
"영화로만나는패션",
"서양현대미술의이해",
"사진이미지와표현",
"연기와창의적사고",
"작곡실습,소리를통한가능성",
"실전축구초급",
"실전농구초급",
"실전야구초급",
"요가&필라테스",
"스포츠산업의변화와이노베이션",
"현대사회와 영상문화",
"패션그리고미술관",
"스포츠메타버스콘텐츠개발",
"치유로서의미술",
"영화속클래식음악",
"방송댄스",
"예술과경영의만남",
"세계음식의문화적이해",
"English Through Drama",
"일본,문학,그리고불교",
"4차산업혁명과드론디자인",
"4차산업혁명시대,삶의변화와불교휴머니티",
"로봇의행복,인간의행복",
"마케팅에스노그래피와빅데이터기반디자인씽킹",
"인문사회계열를위한Python과AI기초",
"융합의이해와화쟁형인재",
"색채와과학",
"레트로 동아시아의 건축공간과 신화",
"언택트(Untact) 사회와 문화다양성",
"미래기술문명사회에서호모루덴스들의놀이여행",
"말은 마음이 지은 집",
"콘텐츠 기획을 위한 스토리텔링",
"지구시민과 다큐멘터리",
"도시에서 생태적으로 살아가기",

"융합콘텐츠산업과 캐릭터",
"데이터프로그래밍기초와실습",
"테마로 읽는 몸과 세계",
"AI시대의 인간관계와 감정 탐구",
"매체를 통한 소통, 사고를 디자인하다",
"행복 인터뷰와 자기서사의 테크닉",
"삼국유사와 문화 콘텐츠",
"AI·사랑·혐오",
"디지털 붓다와의 대화",
"디지털 전환 시대, 삶의 변화와 불교 휴머니티",
"메타버스와 통일 문화 콘텐츠",
"소셜이노베이션을 위한 질적 데이터 분석 실습(캡스톤 디자인)",
"종교와 과학",
"21세기 Leadership(경제사회)",
"사회봉사1",
"사회봉사2",
"취업 실전 전략",
"대학생의 자기주도적 학습 전략",
"소셜벤처",
"비즈니스 실무론",
"창의적 기획과 문제 해결",
"코칭 리더십",
"자기주도 학습으로 가는 유학생 학습 전략",
"도석완씨와 함께하는 도서관 지식 여행",
"기업 실무 기초(인사)",
"공감의 리더십",
"글로벌 리더십",
"비전 설정과 커리어 플래닝",
"직무 이해와 기업 분석",
"스타트업과 진로 탐색",
"실전 창업과 상권 분석",
"스타트업과 크라우드 펀딩",
"소셜벤처 프로젝트",
"스타트업 시뮬레이션",
"평범함을 위대함으로 바꾸는 홍보의 기술",
"더 행복해지기-행복 심리학",
"Presentation Skills and Public Speaking",
"커리어 디자인",

"자아와명상1",
"자아와명상2",
"불교와인간",
"소셜앙트레프레너십과리더십",
"글로벌앙트레프레너십과리더십",
"테크노앙트레프레너십과리더십",
"소통,미디어,윤리",
"다문화사회의삶",
"사랑,우정,관용",
"민주주의와시장경제",
"아시아의사회와문화",
"아메리카의사회와문화",
"유럽의사회와문화",
"아프리카의사회와문화",
"미래인간과 위험사회",
"시장경제와 위험사회",
"과학기술과 위험사회",
"미래환경과 위험사회",
"디지털시대의글쓰기",
"기술보고서작성및발표",
"외국인을위한대학글쓰기",
"Basic EAS",
"EAS1",
"EAS2",
"지혜와자비명작세미나",
"존재와역사명작세미나",
"경제와사회명작세미나",
"자연과기술명작세미나",
"문화와예술명작세미나",
"인공지능과미래사회",
"컴퓨팅사고",
"프로그래밍입문",
"자율사물입문",
"미적분학및연습1",
"일반물리학및실험1",
"일반화학및실험1",
"일반생물학및실험1",
"일반통계학및실습1",

"대학생물학및실험1",
"대학통계학및실습1",
"미적분학및연습2",
"일반물리학및실험2",
"일반화학및실험2",
"일반생물학및실험2",
"일반통계학및실습2",
"대학생물학및실험2",
"대학통계학및실습2",
"확률및통계학",
"공학선형대수학",
"공학수학1",
"이산수학",
"생물학개론",
"물리학개론",
"화학개론",
"지구환경과학",
"프로그래밍기초와실습",
"인터넷프로그래밍",
"공학경제",
"기초수학(인문)",
"산업수학",
"영어작문",
"TOEIC영어",
"Academic Reading and Discussion",
"Academic Listening and Discussion",
"시사영어",
"Current Issues and Discussion",
"English through Films",
"Academic Writing",
"Business writing",
"English Conversation 3",
"English Conversation 4",
"중국어(초급)",
"중국어(중급)",
"일본어(초급)",
"일본어(중급)",
"독일어(초급)",

```

"독일어(중급)",
"불어(초급)",
"불어(중급)",
"러시아어(초급)",
"스페인어(초급)",
"스페인어(중급)",
"아랍어(초급)",
"교양한문",
"라틴어(초급)",
"OPIc"
]
}

```

```

{
  "recommendations": [
    {
      "course": "EGC3051",
      "reason": "금융 기초 개념과 실무적인 돈의 흐름을 이해할 수 있어 금융권 진로 준비에 직접적으로 도움이 됩니다."
    },
    {
      "course": "INC4096",
      "reason": "머신러닝은 금융 데이터 분석과 예측 모델링의 핵심 기술로, 금융AI 분야 역량을 강화할 수 있습니다."
    },
    {
      "course": "EGC3059",
      "reason": "수요와 공급 구조를 다루어 금융 및 시장 분석의 기본 논리를 이해하는데 큰 도움이 됩니다."
    },
    {
      "course": "EGC5033",
      "reason": "데이터 프로그래밍 기초를 학습하여 금융 데이터 전처리와 분석 역량을 높일 수 있습니다."
    },
    {
      "course": "EGC3028",
      "reason": "인터넷 정책과 법률을 다루어 금융 데이터 관리와 규제 이해에 필수적

```

인 법·정책적 관점을 확보할 수 있습니다."

```
}  
]  
}
```



CustomError

```
/**
 * CustomException 생성
 */
@Getter
public class CustomException extends RuntimeException {

    //errorCode
    private final ErrorCode errorCode;

    public CustomException(ErrorCode errorCode) {
        super(errorCode.getMessage());
        this.errorCode = errorCode;
    }

    public CustomException(ErrorCode errorCode, String message) {
        super(message);
        this.errorCode = errorCode;
    }
}
```

```
public interface ErrorCode {
    int getStatus();
    String getCode();
    String getMessage();
}
```

이제 개발할때 커스텀에러코드 쓰는법(예시)


```

@Getter
@AllArgsConstructor
public enum UserErrorCode implements ErrorCode {

    //유저관련
    INVALID_INPUT_ID(400, "U001", "아이디 입력이 올바르지 않습니다."),
    INVALID_INPUT_PW(400, "U002", "비밀번호 입력이 올바르지 않습니다."),
    USER_NOT_FOUND(404, "U003", "사용자를 찾을 수 없습니다."),
    PASSWORD_NOT_EQUAL(401, "U004", "패스워드가 일치하지 않습니다."),
    ALREADY_JOINED(409, "U005", "이미 유저가 있습니다."),

    //토큰관련
    INVALID_TOKEN(401, "T001", "유효하지 않은 Refresh Token입니다"),
    IS_NOT_REFRESH_TOKEN(401, "T002", "Refresh 토큰이 아닙니다."),
    IS_NOT_SAME_TOKEN(401, "T003", "Token이 일치하지 않습니다.");

    private final int status;
    private final String code;
    private final String message;
}

```

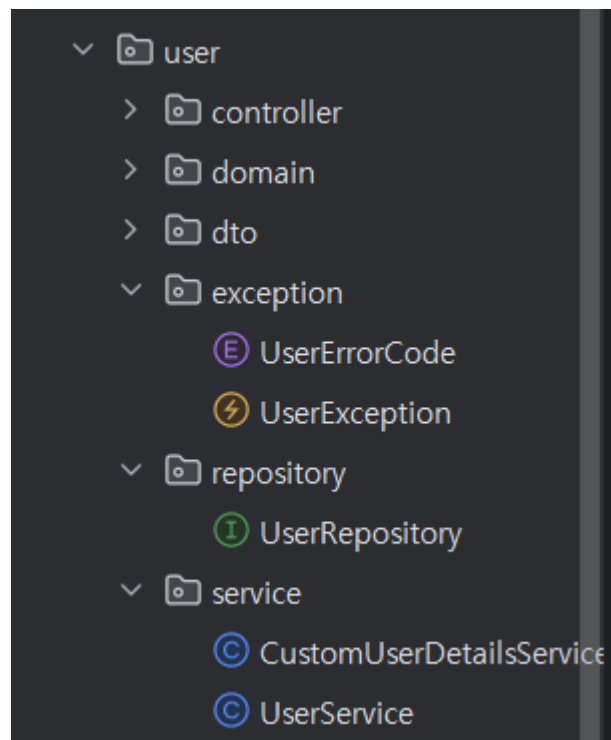
```

public class UserException extends CustomException { ;

    public UserException(ErrorCode errorCode) {
        super(errorCode);
    }

    public UserException(ErrorCode errorCode, String message) {
        super(errorCode, message);
    }
}

```





테스트

▼ 회원가입 및 로그인

✓ 1. 회원가입 (SignUp)

요청 (POST)

POST http://localhost:8081/v1/auth/signup
Content-Type: application/json

Body (예시)

```
{  
  "email": "test@dgu.ac.kr",  
  "password": "1234",  
  "name": "홍길동"  
}
```

예상 응답

"회원가입 성공, 이메일 인증 필요"

이메일 인증 완료! 이제 로그인할 수 있습니다.

- @dgu.ac.kr이 아니면 회원가입이 안 되게
- 이 시점에서는 DB에 `enabled=false` 상태로 저장됨.
- 이메일 인증 로직이 구현되어 있다면, 발송된 토큰으로 `/verify` 요청해서 활성화해야 함.

Request URL

http://localhost:8081/v1/auth/signup

Server response

Code

Details

200

Response body

```
{
  "success": true,
  "message": "회원가입 성공! 이메일을 확인하세요.",
  "data": null
}
```



Download

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-type: application/json
date: Sun, 02 Nov 2025 11:10:21 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers
x-content-type-options: nosniff
x-xss-protection: 0
```

Responses

Responses

Code

Description

Links

200

OK

No links

Media type

/

Controls Accept header.

Example Value | Schema

```
{
  "success": true,
  "message": "성공",
  "data": {}
}
```

hyl020415@gmail.com

오후 8:10 (1분 전)



나에게 ▼

안녕하세요, 동국대 학점관리 웹서비스입니다.

회원가입을 완료하기 위해 이메일 인증이 필요합니다.

아래 링크를 클릭하시면 인증이 완료됩니다.

<http://localhost:8081/v1/auth/verify?token=26cb6ef9-9f53-4d3c-9a428cdf1dc891>

⚠ 본 메일은 발신 전용입니다.

✓ 2. 이메일 인증

요청 (GET)

```
GET http://localhost:8081/v1/auth/verify?token={메일로 온 token}
Content-Type: application/json
```

예상 응답

```
"이메일 인증 완료! 이제 로그인할 수 있습니다."
```

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8081/v1/auth/verify?token=115ad308-81d2-4d33-b1db-c04ea47b6b75' \
  -H 'accept: */*' 
```



Request URL

http://localhost:8081/v1/auth/verify?token=115ad308-81d2-4d33-b1db-c04ea47b6b75

Server response

Code

Details

200

Response body

```
{
  "success": true,
  "message": "이메일 인증 완료!",
  "data": null
}
```



Download

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-type: application/json
date: Sun,02 Nov 2025 11:26:56 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
x-content-type-options: nosniff
x-xss-protection: 0
```

Responses

✓ 3. 로그인 (Login)

요청 (POST)

POST http://localhost:8081/v1/auth/login

Content-Type: application/json

Body

```
{  
  "email": "test@dgu.ac.kr",  
  "password": "1234"  
}
```

예상 응답

```
{  
  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."  
}
```

```
curl -X 'POST' \
  'http://localhost:8081/v1/auth/login' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "email": "hy10415@dgu.ac.kr",
    "password": "12341234!"
  }'
```

Request URL

http://localhost:8081/v1/auth/login

Server response

Code

Details

200

Response body

```
{
  "success": true,
  "message": "로그인 성공",
  "data": {
    "accessToken": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJoZWwwNDE1QGRndS5hYy5rciIsIm1hdCI6MTc2MjA4Mjg4MiwiZXhwIjoxNzYyMDg0NjgyfQ.TDS_AwA-KI3LnNq8V3PT5eIVw8iyFnTPrdLYo8HnKjs",
    "refreshToken": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJoZWwwNDE1QGRndS5hYy5rciIsIm1hdCI6MTc2MjA4Mjg4MiwiZXhwIjoxNzYyMDg0NjgyfQ.LbdaS8e23x-wN4QhC8EXwKkMouRXoc7tf07cIxc29yw"
  }
}
```



Download

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-type: application/json
date: Sun,02 Nov 2025 11:28:02 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
x-content-type-options: nosniff
x-xss-protection: 0
```

- token 2개 복사해주기

Available authorizations

x

bearerAuth (http, Bearer)

Authorized

Value: *****

Logout

Close

- 우측 상단 authorize 버튼 클릭해서, accessToken 등록
- 이제 로그아웃하고 다 사용가능

POST

/v1/auth/refresh

Parameters

Cancel

Reset

No parameters

Request body required

application/json

```
{
  "refreshToken":
  "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJoeWwwNDE1QGRndS5hYy5rciIsImhhdCI6MTc2MjA4Mjg4MiwiZXhwIjoxNzYzMjkyNDgyfQ.LbdaS8e23x-wN4QhC8EXwKkMouhXoc7tf07clxc29vw"
}
```

Execute

Responses

Request URL

```
http://localhost:8081/v1/auth/refresh
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "success": true, "message": "AccessToken 재발급 완료", "data": { "accessToken": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJoZWwwNDE1QGRndS5hYy5rciIsIm1hdCI6MTc2MjA4MzEyMSwiZXhwIjoxNzYyMDg0OTIxfQ.fzCyWfqQpCwh7Esy1emtj9oJfu5p4QPs8mvLcPSGjes", "refreshToken": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJoZWwwNDE1QGRndS5hYy5rciIsIm1hdCI6MTc2MjA4Mjg4MiwiZXhwIjoxNzYyMjkyNDgyfQ.LbdaS8e23x-wN4QhC8EXwKkMouRXoc7tf07cIxc29yw" } }</pre> <p>Response headers</p> <pre>cache-control: no-cache,no-store,max-age=0,must-revalidate connection: keep-alive content-type: application/json date: Sun,02 Nov 2025 11:32:01 GMT expires: 0 keep-alive: timeout=60 pragma: no-cache transfer-encoding: chunked vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers x-content-type-options: nosniff x-xss-protection: 0</pre>

Responses

- refreshToken으로 accessToken재발급
- 이때 refreshToken값은 기존과 동일

✓ 4. 로그아웃 (Logout)

요청 (POST)

POST http://localhost:8081/v1/auth/logout
 Authorization: Bearer {accessToken}

예상 응답

"로그아웃 성공"

[illegible]

Request URL


```
http://localhost:8081/v1/auth/logout
```

Server response	
Code	Details

200 **Response body**

Response body

```
{
  "success": true,
  "message": "로그아웃 완료",
  "data": null
}
```

 Download

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-type: application/json
date: Sun,02 Nov 2025 11:34:11 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
```

5. 회원탈퇴 (Withdraw)

요청 (DELETE)

```
DELETE http://localhost:8081/v1/auth/user/me
Authorization: Bearer {accessToken}
```

예상 응답

"회원탈퇴 완료"

[illegible]

▼ 교과목 DB 자동 업데이트

0. DB 초기화

```
SET FOREIGN_KEY_CHECKS = 0;
TRUNCATE TABLE course;
TRUNCATE TABLE course_version;
TRUNCATE TABLE course_update_log;
SET FOREIGN_KEY_CHECKS = 1;
```

✓ 1. 테스트용 데이터 삽입

```
INSERT INTO course (course_id, course_name, credit, course_type, is_force, is_eng, major_id)
VALUES
```

```
VALUES
```

```
('INC4058', '데이터통신', 3, '전문', true, false, 'ICE'),
```

```
('INC4093', '소프트웨어공학및설계', 3, '전문', false, true, 'ICE');
```

```
INSERT INTO course_version (previous_course_id, course_id, change_type, change_year, notes, major_id)
VALUES
```

```
VALUES
```

```
('INC4058', 'INC4060', '학수번호변경', 2025, '데이터통신 → 네트워크기초', 'ICE'),
```

```
('INC4093', 'INC4093', '이름변경', 2025, '소프트웨어공학및설계 → 소프트웨어공학', 'ICE');
```

The screenshot shows a SQL IDE interface with a Navigator on the left and a SQL editor on the right. The Navigator displays the 'hangover' database schema with various tables. The SQL editor contains two SQL statements: an INSERT INTO statement for the 'course' table and an INSERT INTO statement for the 'course_version' table. The 'course' table has columns: course_id, course_name, credit, course_type, is_force, is_eng, major_id. The 'course_version' table has columns: previous_course_id, course_id, change_type, change_year, notes, major_id. The SQL statements are as follows:

```
1 • INSERT INTO course (course_id, course_name, credit, course_type, is_force, is_eng, major_id)
2 VALUES
3 ('INC4058', '데이터통신', 3, '전문', true, false, 'ICE'),
4 ('INC4093', '소프트웨어공학및설계', 3, '전문', false, true, 'ICE');
5
6 INSERT INTO course_version (previous_course_id, course_id, change_type, change_year, notes, major_id)
7 VALUES
8 ('INC4058', 'INC4060', '학수번호변경', 2025, '데이터통신 → 네트워크기초', 'ICE'),
9 ('INC4093', 'INC4093', '이름변경', 2025, '소프트웨어공학및설계 → 소프트웨어공학', 'ICE');
```

The screenshot shows the same SQL IDE interface, but now the SQL editor contains a SELECT query: `SELECT * FROM hangover.course;`. Below the editor, the 'Result Grid' is displayed, showing the data for the 'course' table. The grid has columns: course_id, major_id, course_name, credit, course_type, is_force, is_eng, prerequisite. The data is as follows:

course_id	major_id	course_name	credit	course_type	is_force	is_eng	prerequisite
INC4058	ICE	데이터통신	3	전문	1	0	NULL
INC4093	ICE	소프트웨어공학및설계	3	전문	0	1	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Navigator: SCHEMAS

Filter objects

▼ hangover

▼ Tables

- course
- course_update_log
- course_version
- enrollment
- graduation_requirement
- major
- recommendation
- refresh_token
- transcript
- user
- users
- verification_token

Views

Stored Procedures

Functions

happy

happy_db

preboard

SQL File 6" course course_version x

Limit to 1000 rows

1 • `SELECT * FROM hangover.course_version;`

Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Conte

	version_id	change_type	change_year	course_id	major_id	notes	pre
▶	1	학수번호변경	2025	INC4060	ICE	데이터통신 → 네트워크기초	NULL
	2	이름변경	2025	INC4093	ICE	소프트웨어공학및설계 → 소프트웨어공학	NULL
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

✓ 2. Swagger

- <http://localhost:8081/swagger-ui/index.html>

GET /api/courseupdate/trigger 수동으로 교과목 배치 실행

자동 배치 주기를 기다리지 않고 즉시 교과목 버전을 업데이트합니다.

Parameters Cancel

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8081/api/courseupdate/trigger' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:8081/api/courseupdate/trigger
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "success": true, "message": "성공", "data": "수동 배치 실행 완료!" }</pre> <p>Download</p>

✓ 3. 결과

```
: Initializing Servlet 'dispatcherServlet'
: Completed initialization in 0 ms
: Init duration for springdoc-openapi is: 372 ms
: 이름 변경 반영: INC4093 → 소프트웨어공학
: 이름 변경 반영: INC4100 → 영상처리개론
```


Navigator

SQL File 6* course course_version course

Limit to 1000 rows

1 • `SELECT * FROM hangover.course;`

Result Grid

	course_id	major_id	course_name	credit	course_type	is_force	is_eng	prerequisite
▶	INC4093	ICE	소프트웨어공학	3	전문	0	1	NULL
	INC4100	ICE	영상처리개론	3	전문	0	0	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



스웨거 쓰는법

```
@RestController
@RequestMapping("v1/auth")
@RequiredArgsConstructor
@Tag(name = "인증 API", description = "회원가입 / 로그인 / 토큰 재발급 관련 API") //이부분추가
public class AuthController {

    private final UserService userService;

    /**
     * 회원가입
     */
    @Operation(
        summary = "회원가입",
        description = "새로운 사용자를 등록합니다.",
        responses = {
            @ApiResponse(
                responseCode = "200",
                description = "회원가입 성공",
                content = @Content(schema = @Schema(implementation
= ApiResponseDto.class))
            ),
            @ApiResponse(
                responseCode = "409",
                description = "이미 가입된 유저",
                content = @Content(schema = @Schema(implementation
= ErrorResponse.class))
            )
        }
    )
}
```

```
@PostMapping("/register")
public ResponseEntity<ApiResponseDto<Void>> register(@RequestBody
UserRequestDto.Register dto) {
    userService.registerUser(dto.getUsername(), dto.getPassword());
    return ResponseEntity.ok(ApiResponseDto.success("회원가입 성공", nul
l));
}
```

Code	Description	Links
200	<p>회원가입 성공</p> <p>Media type</p> <p>*/*</p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "status": 200, "message": "성공", "data": {} }</pre>	No links
409	<p>이미 가입된 유저</p> <p>Media type</p> <p>*/*</p> <p>Example Value Schema</p> <pre>{ "status": 0, "code": "string", "message": "string" }</pre>	No links

이런식으로 공통된응답이 뜹니다.

성공시와 실패시에는 저희가 만들었던 Custom예외처리와 공통응답처리로 처리가됩니다.

server response

Code	Details
400 <i>Undocumented</i>	<div>Error: response status is 400</div> <div>Response body</div> <pre>{ "status": 400, "code": "U902", "message": "비밀번호 입력이 올바르지 않습니다." }</pre> <div>Download</div>

Response header

Code	Details
200	<p>Response body</p> <pre>{ "status": 200, "message": "로그인 성공", "data": { "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IWRXbWV0QXQwIiwiaWF0IjE0NjwzODUwMDE5fQ.MjB6QUA5ZC_QXh6f6p6s64dzViyQ5MzszHr13-4", "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IWRXbWV0QXQwIiwiaWF0IjE0NjwzODUwMDE5fQ.MjB6QUA5ZC_QXh6f6p6s64dzViyQ5MzszHr13-4" } }</pre> <p>Response header</p>



응답 controller에서쓰는법

```
return ResponseEntity.noContent().build(); 204(delete일때 noContent)  
return ResponseEntity.status(HttpStatus.OK).body(null); //body가 필요없을  
때
```

예시

```
public ResponseEntity<TokenResponse> login(@RequestBody UserReque  
stDto.Login dto) {  
    TokenResponse tokenResponse = authService.login(dto.getUsername  
(), dto.getPassword());  
    return ResponseEntity.status(HttpStatus.CREATED).body(tokenRespon  
se);  
}
```

ResponseEntity에서

```
return ResponseEntity.status(HttpStatus.CREATED).body(tokenRespons  
e);
```

처럼 꺼내오기

HttpStatus.CREATED 는 상태코드따라 하면되고 .body는 출력 dto를 감싸서 보내
면됨

기본은 "성공"이라고 메시지가 뜨기때문에

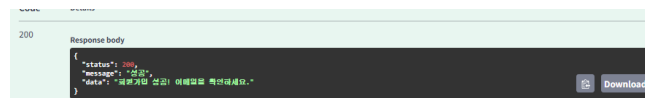
```
return ResponseEntity.status(HttpStatus.CREATED)  
    .body(ApiResponseDto.success(201, "회원가입 성공", createdUser));  
추가적으로 넣고싶은사람은 이렇게 넣으면돼
```

```
public static <T> ApiResponseDto<T> success(int status, T data) {  
    return new ApiResponseDto<>(status, "성공", data);  
}
```

```
public static <T> ApiResponseDto<T> success(int status, String message, T data) {
    return new ApiResponseDto<>(status, message, data);
}
```

이게 포맷형식

원래는 HttpHeaders에서 201,200처럼 번호받아와서 body에 똑같이 받아오는데 직접
 return ResponseEntity.status(HttpStatus.CREATED)
 .body(ApiResponseDto.success(201, "회원가입 성공", createdUser));
 이런식으로 넣는 경우는 HttpStatus랑 int로 넣는 성공코드가 일치하는지 확인후에 넣기



이렇게 돼 성공응답이 호영이 코드로 예시든