## **📘 Chapter 1. 시스템 개요**

### **1.1 시스템 명**

**스포츠지원포털 사용자 인증 및 회원관리 시스템**

### **1.2 시스템 목적**

본 시스템은 스포츠지원포털 사용자들의 인증 및 회원 관리를 전담하는 핵심 시스템으로, 강사와 교육생의 역할을 구분하여 각자의 권한에 따라 시스템을 이용할 수 있도록 하고, 외부 시스템(SSO)과의 연계를 통해 통합 인증도 지원합니다. 이 시스템은 보안성과 유연성을 갖춘 구조로 설계되어, 사용자 경험을 해치지 않으면서도 안정적인 인증 및 정보 제공을 수행합니다.

### **1.3 주요 기능**

#### **✅ 인증/회원 관리**

| **기능 명칭** | **설명** |
| --- | --- |
| 로그인 | 사용자 ID와 비밀번호를 이용한 로그인 처리 및 JWT 토큰 발급 |
| 토큰 갱신 | Refresh Token을 활용해 Access Token을 재발급 |
| SSO 콜백 처리 | 외부 SSO 포털 인증 완료 후 콜백 요청 처리 및 사용자 인증 토큰 생성 |
| 내 정보 조회 | 현재 로그인한 사용자(강사/교육생)의 프로필 정보 확인 |
| 내 정보 수정 | 이름, 이메일, 프로필 이미지 등의 정보 수정 기능 제공 |

#### **✅ 공지사항**

| **기능 명칭** | **설명** |
| --- | --- |
| 공지사항 목록 조회 | 전체 공지사항 목록 페이징 처리하여 조회 |
| 공지사항 상세 조회 | 개별 공지사항의 제목, 내용, 첨부파일 등 상세 조회 |

#### **✅ 교육 신청 및 이력**

| **기능 명칭** | **설명** |
| --- | --- |
| 교육 신청 내역 조회 | 교육생이 본인이 신청한 교육 과정 목록을 조회 |
| 교육 수료증 조회 | 교육이 종료된 과정에 대해 수료증을 확인 및 다운로드 |
| 교육 신청 취소 | 교육생이 신청한 교육 과정을 직접 취소 |

#### **✅ 강사 전용 기능**

| **기능 명칭** | **설명** |
| --- | --- |
| 강의 배정 정보 확인 | 본인이 강의하는 교육과정 정보 확인 |
| 강의자료 업로드 | 과정별 강의자료(파일) 등록 및 수정 |
| 출결 정보 조회 | 과정별 교육생 출결 정보 확인 |
| 출결 정보 수정 요청 | 출결 오류 수정 요청 기능 제공 |

#### **✅ 관리자/운영자 기능 (참고용)**

| **기능 명칭** | **설명** |
| --- | --- |
| 사용자 목록 관리 | 전체 사용자 계정(강사/교육생 포함) 조회 및 권한 관리 |
| 공지사항 등록/수정 | 공지사항을 작성하고 게시하는 관리자 전용 기능 |
| 수강 통계 조회 | 전체 수강 이력, 수료율 등 통계 조회 기능 |

### **1.4 시스템 구성 요약**

| **항목** | **내용** |
| --- | --- |
| 개발 언어 | Java 21 |
| 프레임워크 | Spring Boot 3.2 |
| 빌드 도구 | Gradle 8.7 |
| 데이터베이스 | PostgreSQL |
| 인증 방식 | JWT (Access Token + Refresh Token), 외부 SSO |
| API 통신 방식 | REST API + JSON |
| 보안 구성 | HTTPS, 토큰 기반 인증, 요청/응답 검증 |
| 배포 환경 | Kubernetes 또는 Docker 기반 MSA 구조 (상황에 따라 구성 확장 가능) |
| 클라이언트 유형 | Web/모바일 클라이언트 (SPA 기반 또는 Hybrid App) |

### **1.5 이해관계자 및 사용자 그룹**

| **사용자 유형** | **설명** |
| --- | --- |
| **강사** | 교육 콘텐츠를 제공하며, 강의자료 등록, 출결 확인 등 강사 전용 기능을 수행 |
| **교육생** | 교육 과정에 참여하고 신청/수료증/출결 등 정보를 확인하는 최종 사용자 |
| **관리자** | 전체 시스템 사용자 및 공지사항, 통계 등을 관리하는 운영자 |
| **외부 시스템** | SSO 인증 정보 제공 시스템 (예: 통합 스포츠 포털 인증 시스템) |

## **📘 Chapter 2. 시스템 아키텍처**

### **2.1 아키텍처 개요**

시스템은 **마이크로서비스 아키텍처(MSA)** 기반으로 구성되며, 각 서비스는 독립적으로 배포/운영이 가능합니다. 서비스 간 통신은 RESTful API 또는 필요 시 gRPC로 수행되며, 서비스 간 결합도를 최소화하여 확장성과 유지보수성을 확보합니다.

* 언어 및 프레임워크: Java 21, Spring Boot 3.2
* 데이터베이스: PostgreSQL
* 빌드 도구: Gradle 8.7
* 배포 환경: Docker, Kubernetes
* 인증 방식: JWT, SSO (Callback 처리)
* 메시징 시스템: 선택 사항 (Kafka 추천)
* CI/CD: Jenkins, GitHub Actions, ArgoCD 등 사용 가능

### **2.2 서비스 구성도 (5개 프로세스)**

| **서비스 명** | **설명** |
| --- | --- |
| ① Gateway Service | 클라이언트 요청을 라우팅하고 인증 필터 및 로깅을 담당 |
| ② Auth Service | 사용자 로그인, JWT 발급/검증, 토큰 갱신, SSO 콜백 처리 담당 |
| ③ User Service | 사용자 프로필 조회/수정, 권한 확인 등 회원 관련 로직 담당 |
| ④ Education Service | 교육 신청, 수료증 조회, 강의 이력 관리 등 교육 관련 도메인 처리 |
| ⑤ Notification Service | 공지사항 목록, 상세 조회, 메시지 알림 등의 기능 제공 |

※ 필요시 추가로 Admin Service, File Service, Monitoring Service 등을 확장 가능

### **2.3 서비스 간 통신 구조 (Logical View)**

복사편집

[Web / App Client]

│

┌────────▼────────┐

│ Gateway Service │

└──────┬──────┬──────┘

│ │

┌────────▼──┐ ┌──▼────────────┐

│ Auth Service │ │ User Service │

└─────────────┘ └───────────────┘

│ │

┌──────────▼─────┐ ┌──────▼────────────┐

│ Education Service │ │ Notification Service │

└────────────────┘ └────────────────────┘

* **Gateway → 각 서비스**: REST API, Spring Cloud Gateway 또는 NGINX 기반
* **서비스 간 통신**: REST (필요 시 gRPC 또는 Kafka 이벤트 전환 가능)

### **2.4 기술 스택 제안 목록**

| **영역** | **기술** | **설명** |
| --- | --- | --- |
| **API Gateway** | Spring Cloud Gateway, NGINX, Kong | 인증, 로깅, 라우팅 기능 |
| **인증/SSO** | Spring Security + JWT, OAuth2 Client, Keycloak (선택) | JWT 발급, SSO 연계 |
| **서비스 개발** | Spring Boot 3.2, WebFlux(Optional), Lombok, MapStruct | 비동기 처리, 객체 매핑 |
| **데이터베이스** | PostgreSQL, Flyway | 스키마 관리, 마이그레이션 |
| **파일 업로드** | AWS S3, Local/NFS | 강의자료, 이미지 저장소 연동 |
| **테스트/문서화** | JUnit 5, RestAssured, Spring REST Docs / Swagger3(OpenAPI) | 자동화 테스트 및 API 문서화 |
| **CI/CD** | Jenkins, GitHub Actions, ArgoCD | 자동 빌드, 배포 파이프라인 |
| **배포 인프라** | Docker, Kubernetes, Helm | 마이크로서비스 컨테이너 오케스트레이션 |
| **보안 강화** | HTTPS, Rate Limit (Bucket4j), CORS 설정, API Key 관리, OWASP 정책 적용 | API 보안 및 취약점 대응 |

※ **검색/캐시, 모니터링**은 아래 **2.6 확장 고려사항**으로 이동되었습니다.

### **2.5 데이터 흐름 상세 설명**

#### **1️⃣ 사용자 로그인 및 인증 처리**

1. 클라이언트에서 ID/PW 기반 로그인 요청
2. Gateway Service에서 요청 라우팅 및 로깅
3. Auth Service에서 사용자 정보를 조회 (User Service or 내부 DB)
4. 비밀번호 해시 검증 (BCrypt 등)
5. 검증 성공 시 Access Token + Refresh Token 발급
6. 토큰은 응답으로 클라이언트에 전달

#### **2️⃣ SSO 콜백 처리**

1. 사용자가 외부 SSO 인증 포털에서 인증을 완료
2. SSO 서버는 설정된 콜백 URL(/auth/sso/callback)로 사용자 인증 정보를 전달
3. Auth Service는 해당 정보를 검증하여 사용자 세션을 생성
4. 필요 시 신규 사용자 등록 (User Service 연동)
5. JWT 토큰 발급 후 클라이언트로 리디렉션 또는 응답 반환

#### **3️⃣ 사용자 정보 조회**

1. 클라이언트가 /users/me API 호출
2. JWT를 Authorization 헤더에 포함하여 Gateway를 통해 요청
3. Gateway에서 JWT 유효성 검증 후 User Service로 전달
4. User Service는 사용자 DB에서 프로필 정보를 조회
5. 사용자 정보 JSON으로 응답

#### **4️⃣ 교육 과정 정보 조회**

1. 교육생 또는 강사가 교육 목록, 수료증, 출결 정보를 요청
2. JWT 인증 후 Gateway → Education Service로 요청 전달
3. Education Service는 교육 수강/출결 정보를 DB에서 조회
4. 결과 반환 후 클라이언트에 전달

#### **5️⃣ 공지사항 확인**

1. 사용자가 공지사항 목록 API 요청
2. Gateway → Notification Service로 라우팅
3. Notification Service는 공지사항 DB를 조회하여 목록 반환
4. 상세 조회 시에도 동일한 구조

#### **6️⃣ 토큰 만료 및 갱신**

1. Access Token 만료 → 클라이언트는 Refresh Token 포함 요청
2. Gateway를 통해 Auth Service의 /auth/refresh 호출
3. Refresh Token 검증 후 새로운 Access Token 발급 및 응답

### **2.6 확장 고려사항**

| **영역** | **기술** | **설명** |
| --- | --- | --- |
| **검색/캐시** | Redis, Elasticsearch | 자주 조회되는 교육 정보/공지사항 캐싱 또는 사용자 로그 검색용 |
| **모니터링** | Prometheus + Grafana, ELK(ElasticSearch + Logstash + Kibana), Zipkin | 서비스 상태 모니터링, 호출 트래픽 분석, 장애 추적 |
| **이벤트 기반 처리** | Kafka, RabbitMQ | 교육 수강/출결 정보 비동기 처리, 알림 트리거 |
| **비동기 요청 처리** | WebFlux, CompletableFuture | 교육 서비스의 대용량 요청 대응 시 고려 |
| **인증 통합 관리** | Keycloak, Auth0, OAuth2 Server | 기업 통합 인증 도입 시 확장 가능 |
| **문서 자동화 및 게이트웨이 연동** | Spring Cloud Gateway + OpenAPI 연동 | API 라우팅에 따른 문서 자동 추출 기능 고려 |

## **📘 Chapter 3. 데이터베이스 설계**

### **3.1 주요 도메인 목록**

| **도메인 구분** | **설명** |
| --- | --- |
| 인증/회원 관리 | 로그인, 회원 정보, 권한 관리 등 |
| 교육 과정 및 프로그램 | 교육 콘텐츠, 일정, 강사 배정 등 |
| 수강 신청 및 정원 | 교육생 신청/취소, 정원 관리 |
| 수료 및 증명서 발급 | 수료 상태 및 수료증 관리 |
| 학습 활동 및 진도 | 강의 자료 열람, 진도율 추적 등 |
| 설문/시험 | 과정별 설문, 시험, 응답 및 채점 |
| 게시판 관리 | 공지사항, 게시판, 댓글 |
| 통계 (관리자) | 수강률, 활동률, 설문 통계 등 |
| 시스템 설정 (관리자) | 관리자 권한 설정, 기본 코드 관리 등 |
| 결제 및 환불 | 유료 과정 결제, 환불 내역 |
| 파일 관리 | 강의 자료 및 첨부 파일 업로드/관리 |

### **3.2 핵심 엔티티 및 관계 설명**

여기서는 핵심 테이블과 관계를 요약합니다. 전체 ERD는 이 구조를 바탕으로 다이어그램으로 생성 가능합니다.

#### **🧑‍🎓 사용자 (users)**

* id (PK)
* user\_id, name, role (교육생/강사/관리자)
* email, phone, profile\_image
* status (활성/탈퇴)
* created\_at, updated\_at

#### **🔐 토큰 (user\_tokens)**

* id (PK), user\_id (FK)
* refresh\_token, expired\_at

#### **🎓 교육과정 (courses)**

* id (PK), title, description, category
* start\_date, end\_date
* capacity, location, status

#### **👨‍🏫 강사 배정 (course\_instructors)**

* id (PK)
* course\_id (FK), instructor\_id (FK → users.id)

#### **📝 수강신청 (enrollments)**

* id (PK), course\_id, user\_id
* enroll\_date, status (신청/수료/취소), cancel\_date

#### **📈 진도 (learning\_progress)**

* id (PK), enrollment\_id (FK)
* content\_type, content\_id
* progress\_rate, last\_accessed\_at

#### **🗂 강의자료 (course\_materials)**

* id (PK), course\_id, title, file\_path, uploaded\_by

#### **🧪 시험/설문 (assessments)**

* id (PK), course\_id, type (시험/설문), title, description, open\_date, close\_date

#### **📋 문제 (assessment\_questions)**

* id (PK), assessment\_id, question\_type, question\_text, options, correct\_answer

#### **✍️ 응답 (assessment\_answers)**

* id (PK), assessment\_id, user\_id, question\_id, answer, score

#### **📜 수료증 (certificates)**

* id (PK), enrollment\_id, issued\_at, certificate\_url

#### **📢 공지사항 (notices)**

* id (PK), title, content, pinned, writer\_id, created\_at

#### **💬 댓글 (comments)**

* id (PK), notice\_id, user\_id, content, created\_at

#### **📊 통계 조회 로그 (stats\_logs)**

* id (PK), type (수강/진도/출결), target\_id, executed\_by, executed\_at

#### **⚙️ 시스템 코드 (system\_codes)**

* id (PK), code\_type, code\_key, code\_value, is\_active

#### **💳 결제 (payments)**

* id (PK), user\_id, course\_id, amount, paid\_at, status

#### **🔄 환불 (refunds)**

* id (PK), payment\_id, refund\_amount, refund\_date, status

#### **📂 파일 정보 (files)**

* id (PK), file\_name, file\_path, uploaded\_by, created\_at

### **📐 ERD in PlantUML**

@startuml

!define table(x) class x << (T,#FFAAAA) >>

!define primary\_key(x) <u>x</u>

!define foreign\_key(x) <i>x</i>

table(users) {

primary\_key(id): UUID

user\_id: String

name: String

role: String

email: String

phone: String

profile\_image: String

status: String

created\_at: Timestamp

updated\_at: Timestamp

}

table(user\_tokens) {

primary\_key(id): UUID

foreign\_key(user\_id): UUID

refresh\_token: String

expired\_at: Timestamp

}

table(courses) {

primary\_key(id): UUID

title: String

description: Text

category: String

start\_date: Date

end\_date: Date

capacity: Int

location: String

status: String

}

table(course\_instructors) {

primary\_key(id): UUID

foreign\_key(course\_id): UUID

foreign\_key(instructor\_id): UUID

}

table(enrollments) {

primary\_key(id): UUID

foreign\_key(course\_id): UUID

foreign\_key(user\_id): UUID

enroll\_date: Date

status: String

cancel\_date: Date

}

table(learning\_progress) {

primary\_key(id): UUID

foreign\_key(enrollment\_id): UUID

content\_type: String

content\_id: UUID

progress\_rate: Float

last\_accessed\_at: Timestamp

}

table(course\_materials) {

primary\_key(id): UUID

foreign\_key(course\_id): UUID

title: String

file\_path: String

uploaded\_by: UUID

}

table(assessments) {

primary\_key(id): UUID

foreign\_key(course\_id): UUID

type: String

title: String

description: Text

open\_date: Date

close\_date: Date

}

table(assessment\_questions) {

primary\_key(id): UUID

foreign\_key(assessment\_id): UUID

question\_type: String

question\_text: Text

options: Text

correct\_answer: String

}

table(assessment\_answers) {

primary\_key(id): UUID

foreign\_key(assessment\_id): UUID

foreign\_key(user\_id): UUID

foreign\_key(question\_id): UUID

answer: Text

score: Float

}

table(certificates) {

primary\_key(id): UUID

foreign\_key(enrollment\_id): UUID

issued\_at: Timestamp

certificate\_url: String

}

table(notices) {

primary\_key(id): UUID

title: String

content: Text

pinned: Boolean

foreign\_key(writer\_id): UUID

created\_at: Timestamp

}

table(comments) {

primary\_key(id): UUID

foreign\_key(notice\_id): UUID

foreign\_key(user\_id): UUID

content: Text

created\_at: Timestamp

}

table(stats\_logs) {

primary\_key(id): UUID

type: String

target\_id: UUID

executed\_by: UUID

executed\_at: Timestamp

}

table(system\_codes) {

primary\_key(id): UUID

code\_type: String

code\_key: String

code\_value: String

is\_active: Boolean

}

table(payments) {

primary\_key(id): UUID

foreign\_key(user\_id): UUID

foreign\_key(course\_id): UUID

amount: Int

paid\_at: Timestamp

status: String

}

table(refunds) {

primary\_key(id): UUID

foreign\_key(payment\_id): UUID

refund\_amount: Int

refund\_date: Timestamp

status: String

}

table(files) {

primary\_key(id): UUID

file\_name: String

file\_path: String

uploaded\_by: UUID

created\_at: Timestamp

}

' 관계 설정

users -- user\_tokens

users -- enrollments

users -- course\_instructors : instructor\_id

users -- assessment\_answers

users -- comments

users -- payments

courses -- course\_instructors

courses -- enrollments

courses -- course\_materials

courses -- assessments

courses -- payments

assessments -- assessment\_questions

assessments -- assessment\_answers

assessment\_questions -- assessment\_answers

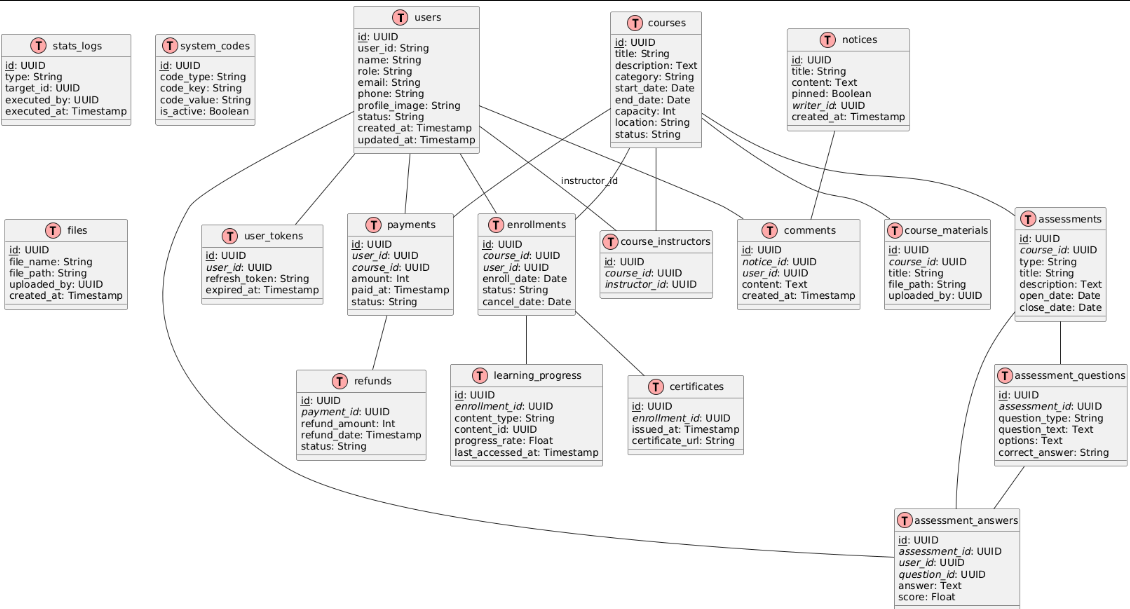
enrollments -- learning\_progress

enrollments -- certificates

notices -- comments

payments -- refunds

@enduml



## **📘 Chapter 4. API 명세서**

### **🧾 작성 양식**

| **항목** | **설명** |
| --- | --- |
| API ID | 인터페이스 설계서상의 식별자 |
| API 명칭 | 기능 이름 |
| URI | 요청 URL |
| Method | HTTP 메서드 (GET, POST 등) |
| 설명 | 기능의 목적 및 역할 |
| 요청값 | 입력 파라미터 구조 및 예시 |
| 응답값 | 정상 응답 구조 및 예시 |
| 인증 방식 | JWT, Refresh Token 등 |
| 호출 주체 | 클라이언트 or 내부 서비스 |
| 대상 서비스 | 실제 처리를 담당하는 MSA 서비스 명 |

## **[1] 인증/회원 관리**

### **🔐 API-AUTH-001 | 로그인 (ID/PW)**

* **URI**: /auth/login
* **Method**: POST
* **설명**: 사용자 ID/PW 기반 인증 후 JWT Access Token과 Refresh Token 발급
* **요청값 예시**:

json

복사편집

{

"userId": "johndoe",

"password": "secure123!"

}

* **응답값 예시**:

json

복사편집

{

"accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI...",

"refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI...",

"expiresIn": 3600

}

* **인증 방식**: 없음
* **호출 주체**: 클라이언트 (Web/Mobile)
* **대상 서비스**: Auth Service

### **🔄 API-AUTH-002 | 토큰 갱신**

* **URI**: /auth/refresh
* **Method**: POST
* **설명**: Refresh Token을 이용해 새로운 Access Token을 발급
* **요청값 예시**:

json

복사편집

{

"refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI..."

}

* **응답값 예시**:

json

복사편집

{

"accessToken": "new.access.token",

"expiresIn": 3600

}

* **인증 방식**: Refresh Token
* **호출 주체**: 클라이언트
* **대상 서비스**: Auth Service

### **🌐 API-AUTH-003 | SSO 콜백 처리**

* **URI**: /auth/sso/callback
* **Method**: GET 또는 POST
* **설명**: 외부 포털(Sports SSO 등)에서 인증 후 콜백 요청을 받아 JWT 발급
* **요청값 예시** (Query or Body):

json

복사편집

{

"ssoToken": "externalSsoToken",

"provider": "sportportal"

}

* **응답값 예시**:

json

복사편집

{

"accessToken": "generated.jwt.token",

"refreshToken": "generated.refresh.token"

}

또는 Redirect (302)

* **인증 방식**: 외부 SSO 검증
* **호출 주체**: 외부 인증 서버
* **대상 서비스**: Auth Service

### **👤 API-USER-001 | 내 정보 조회**

* **URI**: /users/me
* **Method**: GET
* **설명**: 현재 로그인된 사용자의 상세 정보 조회
* **요청값 예시**:  
  + Header: Authorization: Bearer <accessToken>
* **응답값 예시**:

json

복사편집

{

"userId": "johndoe",

"name": "John Doe",

"email": "john@example.com",

"role": "교육생"

}

* **인증 방식**: JWT (Bearer Token)
* **호출 주체**: 클라이언트
* **대상 서비스**: User Service

### **✏️ API-USER-002 | 내 정보 수정**

* **URI**: /users/me
* **Method**: PUT or PATCH
* **설명**: 로그인한 사용자의 이름, 이메일, 프로필 이미지 등 정보 수정
* **요청값 예시**:

json

복사편집

{

"name": "홍길동",

"email": "hong@example.com",

"profileImage": "https://.../image.png"

}

* **응답값 예시**:

json

복사편집

{

"userId": "hongildong",

"name": "홍길동",

"email": "hong@example.com"

}

* **인증 방식**: JWT
* **호출 주체**: 클라이언트
* **대상 서비스**: User Service

## **[2] 교육 과정 및 프로그램**

### **📚 API-COURSE-001 | 교육 과정 목록 조회**

* **URI**: /courses
* **Method**: GET
* **설명**: 검색 조건(카테고리, 제목, 진행 상태 등)을 기반으로 교육 과정 목록을 조회하며, 페이징 제공
* **요청값 예시 (Query)**:

복사편집

/courses?page=1&size=10&status=OPEN&category=HRD

* **응답값 예시**:

json

복사편집

{

"totalElements": 132,

"totalPages": 14,

"content": [

{

"id": "c-1001",

"title": "스프링부트 마이크로서비스",

"category": "Backend",

"startDate": "2025-05-01",

"endDate": "2025-06-30",

"status": "OPEN"

}

]

}

* **인증 방식**: JWT (교육생/강사/관리자)
* **호출 주체**: 클라이언트
* **대상 서비스**: Education Service

### **📘 API-COURSE-002 | 교육 과정 상세 조회**

* **URI**: /courses/{courseId}
* **Method**: GET
* **설명**: 특정 교육 과정의 상세 정보 조회 (강사, 일정, 수강 정원 등 포함)
* **요청값 예시**:

복사편집

/courses/c-1001

* **응답값 예시**:

json

복사편집

{

"id": "c-1001",

"title": "스프링부트 마이크로서비스",

"description": "Java 21 기반 Spring Boot 3.2를 활용한 실전 강의",

"category": "Backend",

"startDate": "2025-05-01",

"endDate": "2025-06-30",

"capacity": 30,

"enrolledCount": 24,

"instructors": [

{ "name": "홍길동", "email": "hong@edu.org" }

],

"status": "OPEN"

}

* **인증 방식**: JWT
* **호출 주체**: 클라이언트
* **대상 서비스**: Education Service

### **📤 API-COURSE-003 | 교육 과정 등록 (강사/관리자 전용)**

* **URI**: /courses
* **Method**: POST
* **설명**: 신규 교육 과정 등록
* **요청값 예시**:

json

복사편집

{

"title": "AI 기반 코딩 교육",

"description": "AI 보조 기반의 프로그래밍 학습 과정",

"category": "AI",

"startDate": "2025-07-01",

"endDate": "2025-08-31",

"capacity": 40,

"location": "온라인",

"instructorIds": ["u-102", "u-103"]

}

* **응답값 예시**:

json

복사편집

{

"id": "c-1101",

"message": "교육 과정이 등록되었습니다."

}

* **인증 방식**: JWT + ROLE\_ADMIN 또는 ROLE\_INSTRUCTOR
* **호출 주체**: 관리자 or 강사
* **대상 서비스**: Education Service

### **✏️ API-COURSE-004 | 교육 과정 수정**

* **URI**: /courses/{courseId}
* **Method**: PUT or PATCH
* **설명**: 교육 과정 정보(제목, 기간, 정원 등) 수정
* **요청값 예시**:

json

복사편집

{

"title": "AI 기반 코딩 교육 - 업데이트",

"capacity": 50

}

* **응답값 예시**:

json

복사편집

{

"id": "c-1101",

"message": "수정이 완료되었습니다."

}

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Education Service

### **❌ API-COURSE-005 | 교육 과정 삭제**

* **URI**: /courses/{courseId}
* **Method**: DELETE
* **설명**: 특정 교육 과정 삭제 처리 (신청자가 없을 경우에만 허용)
* **요청값 예시**:

복사편집

DELETE /courses/c-1101

* **응답값 예시**:

json

복사편집

{

"message": "교육 과정이 삭제되었습니다."

}

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Education Service

### **👥 API-COURSE-006 | 강사 목록 조회 (교육 과정별)**

* **URI**: /courses/{courseId}/instructors
* **Method**: GET
* **설명**: 교육 과정에 배정된 강사 목록 조회
* **응답값 예시**:

json

복사편집

[

{

"id": "u-102",

"name": "홍길동",

"email": "hong@edu.org"

}

]

* **인증 방식**: JWT
* **호출 주체**: 관리자, 강사
* **대상 서비스**: Education Service

**[3] 수강 신청 및 정원 관리**

### **📝 API-ENROLL-001 | 수강 신청**

* **URI**: /enrollments
* **Method**: POST
* **설명**: 교육생이 특정 교육 과정에 수강 신청
* **요청값 예시**:

json

복사편집

{

"courseId": "c-1001"

}

* **응답값 예시**:

json

복사편집

{

"enrollmentId": "e-9001",

"status": "ENROLLED",

"message": "수강 신청이 완료되었습니다."

}

* **인증 방식**: JWT (교육생)
* **호출 주체**: 클라이언트
* **대상 서비스**: Education Service

### **🗂 API-ENROLL-002 | 수강 신청 내역 조회**

* **URI**: /enrollments
* **Method**: GET
* **설명**: 로그인한 사용자의 수강 신청 내역을 조회 (페이징 지원)
* **요청값 예시 (Query)**:

복사편집

/enrollments?page=1&size=10&status=ENROLLED

* **응답값 예시**:

json

복사편집

{

"totalElements": 2,

"content": [

{

"enrollmentId": "e-9001",

"courseTitle": "Spring Boot 입문",

"status": "ENROLLED",

"enrollDate": "2025-04-10"

}

]

}

* **인증 방식**: JWT
* **호출 주체**: 클라이언트
* **대상 서비스**: Education Service

### **❌ API-ENROLL-003 | 수강 신청 취소**

* **URI**: /enrollments/{enrollmentId}
* **Method**: DELETE
* **설명**: 사용자가 신청한 교육 과정을 취소
* **요청값 예시**:

복사편집

DELETE /enrollments/e-9001

* **응답값 예시**:

json

복사편집

{

"message": "수강 신청이 취소되었습니다.",

"status": "CANCELED"

}

* **인증 방식**: JWT
* **호출 주체**: 클라이언트
* **대상 서비스**: Education Service

### **📊 API-ENROLL-004 | 수강 인원 현황 조회 (관리자/강사용)**

* **URI**: /courses/{courseId}/enrollments
* **Method**: GET
* **설명**: 특정 교육 과정에 신청한 사용자 목록 조회
* **요청값 예시**:

복사편집

/courses/c-1001/enrollments

* **응답값 예시**:

json

복사편집

[

{

"userId": "u-201",

"name": "김영희",

"status": "ENROLLED",

"enrollDate": "2025-04-12"

},

{

"userId": "u-202",

"name": "이철수",

"status": "CANCELED",

"enrollDate": "2025-04-10"

}

]

* **인증 방식**: JWT + ROLE\_INSTRUCTOR or ROLE\_ADMIN
* **호출 주체**: 강사, 관리자
* **대상 서비스**: Education Service

### **📌 API-ENROLL-005 | 수강 정원 초과 여부 확인**

* **URI**: /courses/{courseId}/availability
* **Method**: GET
* **설명**: 교육 과정의 현재 신청 인원이 정원 초과 상태인지 여부 확인
* **요청값 예시**:

복사편집

/courses/c-1001/availability

* **응답값 예시**:

json

복사편집

{

"courseId": "c-1001",

"capacity": 30,

"enrolled": 29,

"isAvailable": true

}

* **인증 방식**: JWT
* **호출 주체**: 클라이언트
* **대상 서비스**: Education Service

## **[4] 수료 및 증명서 발급**

### **🏁 API-CERT-001 | 수료 여부 조회**

* **URI**: /certificates/status
* **Method**: GET
* **설명**: 로그인한 사용자가 특정 교육 과정에 대해 수료했는지 여부를 조회
* **요청값 예시 (Query)**:

복사편집

/certificates/status?courseId=c-1001

* **응답값 예시**:

json

복사편집

{

"courseId": "c-1001",

"enrollmentId": "e-9001",

"isCompleted": true,

"completedAt": "2025-06-30"

}

* **인증 방식**: JWT (교육생)
* **호출 주체**: 클라이언트
* **대상 서비스**: Education Service

### **📜 API-CERT-002 | 수료증 조회**

* **URI**: /certificates/{certificateId}
* **Method**: GET
* **설명**: 수료한 교육 과정에 대한 수료증 정보 및 다운로드 링크 조회
* **요청값 예시**:

복사편집

/certificates/cert-102

* **응답값 예시**:

json

복사편집

{

"certificateId": "cert-102",

"courseTitle": "AI 코딩 교육",

"issuedAt": "2025-07-01",

"downloadUrl": "https://cdn.edu.org/certs/cert-102.pdf"

}

* **인증 방식**: JWT
* **호출 주체**: 클라이언트
* **대상 서비스**: Education Service

### **🧾 API-CERT-003 | 수료증 목록 조회 (내 수료 이력)**

* **URI**: /certificates
* **Method**: GET
* **설명**: 로그인한 사용자의 수료증 목록 조회
* **요청값 예시 (Query)**:

복사편집

/certificates?page=1&size=10

* **응답값 예시**:

json

복사편집

{

"totalElements": 2,

"content": [

{

"certificateId": "cert-101",

"courseTitle": "스프링부트 입문",

"issuedAt": "2025-03-01"

},

{

"certificateId": "cert-102",

"courseTitle": "AI 코딩 교육",

"issuedAt": "2025-07-01"

}

]

}

* **인증 방식**: JWT
* **호출 주체**: 클라이언트
* **대상 서비스**: Education Service

### **📥 API-CERT-004 | 수료 상태 반영 (관리자/자동화)**

* **URI**: /certificates
* **Method**: POST
* **설명**: 진도율 또는 시험 결과 기반으로 수료 조건을 만족한 사용자의 수료 상태 반영 및 수료증 발급
* **요청값 예시**:

json

복사편집

{

"enrollmentId": "e-9001"

}

* **응답값 예시**:

json

복사편집

{

"certificateId": "cert-103",

"message": "수료 처리 및 수료증이 발급되었습니다."

}

* **인증 방식**: JWT + ROLE\_ADMIN or 시스템 내부 호출
* **호출 주체**: 관리자 / 자동화 스케줄러
* **대상 서비스**: Education Service

## **[5] 학습 활동 및 진도 관리**

### **📂 API-LEARN-001 | 학습 콘텐츠 목록 조회**

* **URI**: /courses/{courseId}/contents
* **Method**: GET
* **설명**: 특정 교육 과정에 포함된 학습 콘텐츠 목록(강의자료, 영상 등) 조회
* **요청값 예시**:

복사편집

/courses/c-1001/contents

* **응답값 예시**:

json

복사편집

[

{

"contentId": "cnt-001",

"title": "오리엔테이션 영상",

"contentType": "VIDEO",

"duration": 780,

"order": 1

},

{

"contentId": "cnt-002",

"title": "스프링부트 구조",

"contentType": "PDF",

"order": 2

}

]

* **인증 방식**: JWT
* **호출 주체**: 클라이언트 (교육생, 강사)
* **대상 서비스**: Education Service

### **▶️ API-LEARN-002 | 학습 콘텐츠 열람 로그 저장**

* **URI**: /contents/{contentId}/progress
* **Method**: POST
* **설명**: 교육생이 학습 콘텐츠를 열람한 시점, 진도율, 시청 시간 등을 저장
* **요청값 예시**:

json

복사편집

{

"progressRate": 0.75,

"watchedSeconds": 585

}

* **응답값 예시**:

json

복사편집

{

"message": "학습 진도 기록이 저장되었습니다.",

"progressRate": 0.75

}

* **인증 방식**: JWT
* **호출 주체**: 클라이언트 (교육생)
* **대상 서비스**: Education Service

### **📊 API-LEARN-003 | 진도율 조회**

* **URI**: /courses/{courseId}/progress
* **Method**: GET
* **설명**: 로그인한 사용자가 특정 교육 과정에서 얼마나 학습했는지 진도율 조회
* **요청값 예시**:

복사편집

/courses/c-1001/progress

* **응답값 예시**:

json

복사편집

{

"courseId": "c-1001",

"userId": "u-2001",

"totalRate": 0.82,

"lastAccessed": "2025-04-20T21:34:12"

}

* **인증 방식**: JWT
* **호출 주체**: 교육생
* **대상 서비스**: Education Service

### **🧑‍🏫 API-LEARN-004 | 교육생 진도 관리 (강사용)**

* **URI**: /courses/{courseId}/students/progress
* **Method**: GET
* **설명**: 강사가 담당하는 교육 과정에서 교육생들의 진도율 현황을 일괄 조회
* **응답값 예시**:

json

복사편집

[

{

"userId": "u-2001",

"name": "김하늘",

"progressRate": 0.95

},

{

"userId": "u-2002",

"name": "박성호",

"progressRate": 0.42

}

]

* **인증 방식**: JWT + ROLE\_INSTRUCTOR
* **호출 주체**: 강사
* **대상 서비스**: Education Service

### **🧾 API-LEARN-005 | 진도 초기화 (관리자 전용)**

* **URI**: /progress/reset
* **Method**: POST
* **설명**: 교육 과정 종료 등 상황에서 진도 정보를 초기화 (재수강 시 등)
* **요청값 예시**:

json

복사편집

{

"courseId": "c-1001",

"userId": "u-2002"

}

* **응답값 예시**:

json

복사편집

{

"message": "진도 정보가 초기화되었습니다."

}

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Education Service

## **[6] 설문/시험**

### **🧾 API-EXAM-001 | 설문/시험 목록 조회**

* **URI**: /courses/{courseId}/assessments
* **Method**: GET
* **설명**: 특정 교육 과정에 등록된 설문 또는 시험 목록 조회
* **요청값 예시**:

복사편집

/courses/c-1001/assessments

* **응답값 예시**:

json

복사편집

[

{

"assessmentId": "a-1001",

"title": "수업 전 사전 설문",

"type": "SURVEY",

"status": "OPEN",

"openDate": "2025-04-15",

"closeDate": "2025-04-25"

},

{

"assessmentId": "a-1002",

"title": "중간 시험",

"type": "EXAM",

"status": "CLOSED",

"openDate": "2025-04-20",

"closeDate": "2025-04-21"

}

]

* **인증 방식**: JWT
* **호출 주체**: 교육생, 강사
* **대상 서비스**: Assessment Service

### **📄 API-EXAM-002 | 설문/시험 상세 조회**

* **URI**: /assessments/{assessmentId}
* **Method**: GET
* **설명**: 설문 또는 시험의 기본 정보 및 문제 목록을 포함한 상세 정보 조회
* **응답값 예시**:

json

복사편집

{

"assessmentId": "a-1002",

"title": "중간 시험",

"type": "EXAM",

"questions": [

{

"questionId": "q-1",

"questionText": "Spring Boot의 특징은?",

"questionType": "MULTIPLE\_CHOICE",

"options": ["경량성", "자동 설정", "복잡한 설정", "의존성 부족"]

}

]

}

* **인증 방식**: JWT
* **호출 주체**: 교육생
* **대상 서비스**: Assessment Service

### **📝 API-EXAM-003 | 응답 제출**

* **URI**: /assessments/{assessmentId}/submit
* **Method**: POST
* **설명**: 설문이나 시험에 대한 응답 제출 (객관식, 주관식 포함)
* **요청값 예시**:

json

복사편집

{

"answers": [

{

"questionId": "q-1",

"answer": "자동 설정"

},

{

"questionId": "q-2",

"answer": "Spring Security"

}

]

}

* **응답값 예시**:

json

복사편집

{

"assessmentId": "a-1002",

"submittedAt": "2025-04-21T09:10:00"

}

* **인증 방식**: JWT
* **호출 주체**: 교육생
* **대상 서비스**: Assessment Service

### **📊 API-EXAM-004 | 채점 결과 조회 (시험만 해당)**

* **URI**: /assessments/{assessmentId}/result
* **Method**: GET
* **설명**: 시험 응답 제출 이후 채점된 결과를 확인
* **응답값 예시**:

json

복사편집

{

"assessmentId": "a-1002",

"score": 85,

"totalQuestions": 10,

"correctAnswers": 8

}

* **인증 방식**: JWT
* **호출 주체**: 교육생
* **대상 서비스**: Assessment Service

### **🧑‍🏫 API-EXAM-005 | 설문/시험 결과 조회 (강사용)**

* **URI**: /assessments/{assessmentId}/responses
* **Method**: GET
* **설명**: 특정 설문/시험에 대한 교육생 응답 결과를 전체 조회 (강사/관리자 권한)
* **응답값 예시**:

json

복사편집

[

{

"userId": "u-202",

"name": "이민수",

"score": 90,

"submittedAt": "2025-04-21T09:10:00"

},

{

"userId": "u-203",

"name": "한소영",

"score": 75,

"submittedAt": "2025-04-21T09:11:10"

}

]

* **인증 방식**: JWT + ROLE\_INSTRUCTOR
* **호출 주체**: 강사, 관리자
* **대상 서비스**: Assessment Service

## **[7] 게시판 관리**

### **📢 API-BOARD-001 | 공지사항 목록 조회**

* **URI**: /notices
* **Method**: GET
* **설명**: 전체 공지사항 목록을 페이징으로 조회하며, 고정(Pinned) 여부, 작성일순 정렬
* **요청값 예시 (Query)**:

복사편집

/notices?page=1&size=10&sort=createdAt,DESC

* **응답값 예시**:

json

복사편집

{

"totalElements": 25,

"content": [

{

"noticeId": "n-101",

"title": "2025 상반기 교육 일정 안내",

"pinned": true,

"writer": "관리자",

"createdAt": "2025-04-01"

},

{

"noticeId": "n-102",

"title": "강의자료 업로드 방법",

"pinned": false,

"writer": "관리자",

"createdAt": "2025-04-10"

}

]

}

* **인증 방식**: JWT
* **호출 주체**: 클라이언트
* **대상 서비스**: Notification Service

### **📄 API-BOARD-002 | 공지사항 상세 조회**

* **URI**: /notices/{noticeId}
* **Method**: GET
* **설명**: 공지사항 제목, 본문, 첨부파일, 댓글 등 상세 내용 조회
* **응답값 예시**:

json

복사편집

{

"noticeId": "n-101",

"title": "2025 상반기 교육 일정 안내",

"content": "5월부터 개설되는 강의는 다음과 같습니다...",

"writer": "관리자",

"createdAt": "2025-04-01",

"attachments": ["교육일정.pdf"],

"comments": [

{

"commentId": "c-2001",

"userName": "홍길동",

"content": "감사합니다!",

"createdAt": "2025-04-02T12:00:00"

}

]

}

* **인증 방식**: JWT
* **호출 주체**: 클라이언트
* **대상 서비스**: Notification Service

### **🖊️ API-BOARD-003 | 공지사항 등록 (관리자 전용)**

* **URI**: /notices
* **Method**: POST
* **설명**: 관리자 권한으로 공지사항을 작성 및 등록
* **요청값 예시**:

json

복사편집

{

"title": "신규 강의 개설 안내",

"content": "6월부터 AI 코딩 실습 강의가 개설됩니다.",

"pinned": true,

"attachments": ["ai\_course\_info.pdf"]

}

* **응답값 예시**:

json

복사편집

{

"noticeId": "n-105",

"message": "공지사항이 등록되었습니다."

}

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Notification Service

### **💬 API-BOARD-004 | 공지사항 댓글 작성**

* **URI**: /notices/{noticeId}/comments
* **Method**: POST
* **설명**: 특정 공지사항에 대한 댓글 작성
* **요청값 예시**:

json

복사편집

{

"content": "좋은 강의 기대됩니다!"

}

* **응답값 예시**:

json

복사편집

{

"commentId": "c-2002",

"message": "댓글이 등록되었습니다."

}

* **인증 방식**: JWT
* **호출 주체**: 클라이언트
* **대상 서비스**: Notification Service

### **❌ API-BOARD-005 | 공지사항 삭제**

* **URI**: /notices/{noticeId}
* **Method**: DELETE
* **설명**: 관리자 권한으로 특정 공지사항 삭제
* **응답값 예시**:

json

복사편집

{

"message": "공지사항이 삭제되었습니다."

}

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Notification Service

## **[8] 통계 (관리자)**

### **📈 API-STATS-001 | 전체 수강 통계 조회**

* **URI**: /admin/stats/enrollments
* **Method**: GET
* **설명**: 전체 교육 과정에 대한 수강 신청자 수, 수료자 수, 수료율 등을 종합적으로 조회
* **요청값 예시 (Query)**:

복사편집

/admin/stats/enrollments?year=2025

* **응답값 예시**:

json

복사편집

{

"totalCourses": 38,

"totalApplicants": 1204,

"totalCompletions": 932,

"completionRate": 0.774

}

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Statistics Service

### **📊 API-STATS-002 | 과정별 진도율 통계**

* **URI**: /admin/stats/courses/{courseId}/progress
* **Method**: GET
* **설명**: 특정 교육 과정에 대한 전체 교육생의 평균 진도율과 분포도 조회
* **응답값 예시**:

json

복사편집

{

"courseId": "c-1001",

"averageProgress": 0.68,

"progressDistribution": {

"0~25%": 5,

"25~50%": 12,

"50~75%": 18,

"75~100%": 30

}

}

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Statistics Service

### **📋 API-STATS-003 | 설문 응답률 통계**

* **URI**: /admin/stats/assessments/{assessmentId}/responses
* **Method**: GET
* **설명**: 설문조사에 대한 응답 완료자 수 및 응답률 조회
* **응답값 예시**:

json

복사편집

{

"assessmentId": "a-1001",

"totalTarget": 60,

"responded": 48,

"responseRate": 0.8

}

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Assessment Service

### **🗓️ API-STATS-004 | 월별 수강 현황 차트 데이터**

* **URI**: /admin/stats/enrollments/monthly
* **Method**: GET
* **설명**: 월별로 신청자 수, 수료자 수를 통계 차트용으로 제공
* **응답값 예시**:

json

복사편집

{

"year": 2025,

"monthlyData": [

{ "month": "01", "enrolled": 50, "completed": 30 },

{ "month": "02", "enrolled": 80, "completed": 65 },

{ "month": "03", "enrolled": 120, "completed": 100 }

]

}

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Statistics Service

### **🔍 API-STATS-005 | 사용자 활동 로그 조회**

* **URI**: /admin/logs/activity
* **Method**: GET
* **설명**: 관리자용 사용자 활동 기록 조회 (조회 시점, API 호출 등)
* **요청값 예시 (Query)**:

복사편집

/admin/logs/activity?userId=u-2001&from=2025-04-01&to=2025-04-21

* **응답값 예시**:

json

복사편집

[

{

"timestamp": "2025-04-20T09:34:21",

"userId": "u-2001",

"action": "로그인",

"ip": "192.168.1.11"

},

{

"timestamp": "2025-04-20T09:40:12",

"userId": "u-2001",

"action": "강의자료 열람",

"ip": "192.168.1.11"

}

]

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Logging / Statistics Service

## **[9] 시스템 설정 (관리자)**

### **⚙️ API-CONFIG-001 | 시스템 코드 목록 조회**

* **URI**: /admin/system-codes
* **Method**: GET
* **설명**: 코드 유형별로 시스템 코드 목록을 조회
* **요청값 예시 (Query)**:

복사편집

/admin/system-codes?codeType=COURSE\_CATEGORY

* **응답값 예시**:

json

복사편집

[

{

"codeId": "cd-1001",

"codeType": "COURSE\_CATEGORY",

"codeKey": "AI",

"codeValue": "인공지능",

"isActive": true

},

{

"codeId": "cd-1002",

"codeType": "COURSE\_CATEGORY",

"codeKey": "HRD",

"codeValue": "인재개발",

"isActive": true

}

]

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Config Service

### **➕ API-CONFIG-002 | 시스템 코드 등록**

* **URI**: /admin/system-codes
* **Method**: POST
* **설명**: 신규 시스템 코드 등록
* **요청값 예시**:

json

복사편집

{

"codeType": "COURSE\_CATEGORY",

"codeKey": "ML",

"codeValue": "머신러닝",

"isActive": true

}

* **응답값 예시**:

json

복사편집

{

"codeId": "cd-1003",

"message": "코드가 등록되었습니다."

}

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Config Service

### **✏️ API-CONFIG-003 | 시스템 코드 수정**

* **URI**: /admin/system-codes/{codeId}
* **Method**: PUT or PATCH
* **설명**: 코드 값 또는 활성 여부 수정
* **요청값 예시**:

json

복사편집

{

"codeValue": "머신 러닝",

"isActive": false

}

* **응답값 예시**:

json

복사편집

{

"message": "코드가 수정되었습니다."

}

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Config Service

### **❌ API-CONFIG-004 | 시스템 코드 삭제**

* **URI**: /admin/system-codes/{codeId}
* **Method**: DELETE
* **설명**: 시스템 코드 삭제 처리
* **응답값 예시**:

json

복사편집

{

"message": "코드가 삭제되었습니다."

}

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자
* **대상 서비스**: Config Service

### **👥 API-CONFIG-005 | 관리자 목록 및 권한 설정**

* **URI**: /admin/users
* **Method**: GET / PATCH
* **설명**: 전체 사용자 목록 중 관리자 권한이 있는 계정을 조회하거나, 권한을 수정
* **요청값 예시 (권한 수정)**:

json

복사편집

{

"userId": "u-3001",

"roles": ["ROLE\_ADMIN", "ROLE\_INSTRUCTOR"]

}

* **응답값 예시**:

json

복사편집

{

"message": "권한이 업데이트되었습니다."

}

* **인증 방식**: JWT + ROLE\_ADMIN (슈퍼 관리자)
* **호출 주체**: 관리자
* **대상 서비스**: User Service or Config Service

## **[10] 결제 및 환불**

### **💳 API-PAY-001 | 결제 요청**

* **URI**: /payments
* **Method**: POST
* **설명**: 교육생이 유료 교육 과정에 대해 결제를 진행. 외부 결제 모듈(PG) 연동 예정
* **요청값 예시**:

json

복사편집

{

"courseId": "c-2001",

"paymentMethod": "CARD",

"amount": 120000

}

* **응답값 예시**:

json

복사편집

{

"paymentId": "p-501",

"pgRedirectUrl": "https://pg.example.com/checkout?id=pg123456"

}

* **인증 방식**: JWT
* **호출 주체**: 교육생 (클라이언트)
* **대상 서비스**: Payment Service

### **✅ API-PAY-002 | 결제 완료 처리 (Webhook / 콜백)**

* **URI**: /payments/confirm
* **Method**: POST
* **설명**: PG사에서 결제 완료 후 시스템에 결과를 통보하는 콜백 엔드포인트
* **요청값 예시**:

json

복사편집

{

"paymentId": "p-501",

"transactionId": "pg-tx-1234",

"status": "PAID",

"paidAt": "2025-05-01T12:00:00"

}

* **응답값 예시**:

json

복사편집

{

"message": "결제 완료가 확인되었습니다."

}

* **인증 방식**: 내부 통신 (PG 서버)
* **호출 주체**: PG 시스템
* **대상 서비스**: Payment Service

### **📄 API-PAY-003 | 결제 이력 조회**

* **URI**: /payments
* **Method**: GET
* **설명**: 사용자의 결제 내역 조회 (페이징 포함)
* **요청값 예시 (Query)**:

복사편집

/payments?page=1&size=10

* **응답값 예시**:

json

복사편집

{

"totalElements": 2,

"content": [

{

"paymentId": "p-501",

"courseTitle": "AI 실습 강의",

"amount": 120000,

"status": "PAID",

"paidAt": "2025-05-01"

},

{

"paymentId": "p-502",

"courseTitle": "React 입문",

"amount": 90000,

"status": "REFUNDED"

}

]

}

* **인증 방식**: JWT
* **호출 주체**: 교육생
* **대상 서비스**: Payment Service

### **🔁 API-REFUND-001 | 환불 요청**

* **URI**: /refunds
* **Method**: POST
* **설명**: 결제 건에 대해 환불 신청 요청
* **요청값 예시**:

json

복사편집

{

"paymentId": "p-501",

"reason": "개인 사정으로 인한 환불 요청"

}

* **응답값 예시**:

json

복사편집

{

"refundId": "r-901",

"status": "REQUESTED",

"message": "환불 요청이 접수되었습니다."

}

* **인증 방식**: JWT
* **호출 주체**: 교육생
* **대상 서비스**: Payment Service

### **✅ API-REFUND-002 | 환불 처리 완료 (관리자)**

* **URI**: /refunds/{refundId}/confirm
* **Method**: POST
* **설명**: 환불 요청이 승인되어 PG를 통해 환불 처리된 후 최종 완료 상태 반영
* **요청값 예시**:

json

복사편집

{

"status": "COMPLETED",

"refundedAmount": 120000

}

* **응답값 예시**:

json

복사편집

{

"message": "환불이 완료되었습니다."

}

* **인증 방식**: JWT + ROLE\_ADMIN
* **호출 주체**: 관리자 or 자동 처리 시스템
* **대상 서비스**: Payment Service