

인증 방식

- **OAuth 2.0 Bearer Token**

엔드포인트 목록

1. 사용자 인증

POST /api/auth/login

- **설명:** 사용자의 로그인 정보를 검증하고 접근 토큰을 발급합니다.
- **요청 예시:**

```
{
  "username": "user@example.com",
  "password": "SecurePassword123!"
}
```

- **응답 예시:**

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6I..",
  "token_type": "Bearer",
  "expires_in": 3600
}
```

2. 교육과정 관리

GET /api/courses

- 설명: 모든 교육과정의 목록을 조회합니다.
- 요청 예시:

```
GET /api/courses HTTP/1.1
Host: example.com
Authorization: Bearer {access_token}
```

- 응답 예시:

$$\left[\begin{array}{c} \{ \end{array} \right.$$

```

    "id": 101,
    "title": "스포츠 리더십 기초",
    "description": "스포츠 리더십의 기본 개념을 학습합니다.",
    "duration": "4주",
    "instructor": "홍길동"
  },
  {
    "id": 102,
    "title": "고급 경기 분석",
    "description": "경기 데이터를 분석하여 전략을 수립합니다.",
    "duration": "6주",
    "instructor": "김철수"
  }
]

```

GET /api/courses/{id}

- **설명:** 특정 교육과정의 상세 정보를 조회합니다.
- **요청 예시:**

```

GET /api/courses/101 HTTP/1.1
Host: example.com
Authorization: Bearer {access_token}

```

- **응답 예시:**

```

{
  "id": 101,
  "title": "스포츠 리더십 기초",
  "description": "스포츠 리더십의 기본 개념을 학습합니다.",
  "duration": "4주",
  "instructor": "홍길동",
  "modules": [
    {
      "module_id": 1,
      "title": "리더십 이론",
      "content": "리더십의 다양한 이론을 소개합니다."
    },
    {
      "module_id": 2,
      "title": "실습 세션",
      "content": "리더십 스킬을 실제로 적용해 봅니다."
    }
  ]
}

```

3. 교육 신청 및 이수 관리

POST /api/enroll

- 설명: 사용자가 특정 교육과정에 등록합니다.
- 요청 예시:

```
{
  "course_id": 101,
  "user_id": 55
}
```

- 응답 예시:

```
{
  "message": "교육과정에 성공적으로 등록되었습니다.",
  "enrollment_id": 2021
}
```

GET /api/users/{id}/progress

- 설명: 사용자의 학습 진행 상황을 조회합니다.
- 요청 예시:

```
GET /api/users/55/progress HTTP/1.1
Host: example.com
Authorization: Bearer {access_token}
```

- 응답 예시:

```
{
  "user_id": 55,
  "courses": [
    {
      "course_id": 101,
      "title": "스포츠 리더십 기초",
      "progress": "75%",
      "completed_modules": 3,
      "total_modules": 4
    }
  ]
}
```

4. 수료증 발급 및 관리

POST /api/certificate/request

- **설명:** 사용자가 교육 수료증을 요청합니다.
- **요청 예시:**

```
{
  "user_id": 55,
  "course_id": 101
}
```

- **응답 예시:**

```
{
  "message": "수료증 발급 요청이 접수되었습니다.",
  "certificate_id": 3050,
  "status": "Pending"
}
```

GET /api/certificate/{id}

- **설명:** 특정 수료증의 상태 및 상세 정보를 조회합니다.
- **요청 예시:**

```
GET /api/certificate/3050 HTTP/1.1
Host: example.com
Authorization: Bearer {access_token}
```

- **응답 예시:**

```
{
  "certificate_id": 3050,
  "user_id": 55,
  "course_id": 101,
  "issue_date": "2024-09-15",
  "status": "Issued",
  "certificate_url": "https://example.com/certificates/3050.pdf"
}
```

5. 시스템 연계 API

POST /api/integration/sync

- **설명:** 다른 시스템과의 데이터 동기화를 수행합니다.
- **요청 예시:**

```
{
  "source_system": "스포츠지원포털",
  "target_system": "체육정보시스템",
  "data_type": "교육과정",
  "sync_type": "real-time"
}
```

- 응답 예시:

```
{
  "message": "데이터 동기화가 성공적으로 완료되었습니다.",
  "sync_id": 4001,
  "status": "Completed"
}
```

인증 방식 상세

- **OAuth 2.0 Bearer Token**
 - 클라이언트는 `/api/auth/login` 엔드포인트를 통해 사용자 인증을 수행하고, 서버는 JWT(JSON Web Token) 형식의 토큰을 반환합니다.
 - 모든 API 요청에는 `Authorization: Bearer {access_token}` 헤더를 포함해야 합니다.
 - 토큰은 `expires_in` 필드에 명시된 시간(초) 후 만료됩니다. 만료된 토큰은 다시 인증을 통해 갱신해야 합니다.

보안 요구사항

- **HTTPS**를 통해 모든 API 요청이 전송되어야 합니다.
- 입력 검증을 통해 SQL 인젝션, XSS 등의 공격을 방지합니다.
- 접근 제어를 통해 사용자 권한에 따라 API 접근을 제한합니다.
- 로그 및 모니터링을 통해 API 사용 현황 및 이상 징후를 실시간으로 감시합니다.

데이터 형식

- 요청 및 응답 데이터는 모두 JSON 형식을 사용합니다.
- 날짜 및 시간 필드는 ISO 8601 형식을 따릅니다. (예: `2024-09-15T13:45:30Z`)
- ID 필드는 숫자형으로 고유하게 식별됩니다.

버전 관리

- 모든 API는 버전 정보를 URL에 포함하여 관리합니다. (예: `/api/v1/courses`)
- 주요 변경 사항 발생 시 새로운 버전을 출시하여 기존 클라이언트의 호환성을 유지합니다.

오류 처리

- **HTTP 상태 코드**를 사용하여 오류를 명확하게 전달합니다.
 - 예: `400 Bad Request`, `401 Unauthorized`, `404 Not Found`, `500 Internal Server Error`

- 오류 응답은 다음과 같은 구조를 따릅니다:

```
{
  "error_code": "INVALID_REQUEST",
  "message": "요청 파라미터가 올바르지 않습니다.",
  "details": {
    "field": "username",
    "issue": "필수 입력 항목입니다."
  }
}
```

제한 사항

- API 호출 횟수는 분당 **60회**로 제한됩니다. 초과 시 **429 Too Many Requests** 오류를 반환합니다.
- 대기업 참여 제한 사업의 경우, 인증된 외부 기업만 API 접근이 가능합니다.

본 API 명세서는 RFP 문서의 내용을 기반으로 작성되었으며, 상세 구현 사항은 프로젝트 진행 중 추가 협의에 따라 변경될 수 있습니다.