

# Node 0 Student Hub - 사용자 매뉴얼

버전: 2.0.0 최종 업데이트: 2026-01-11 대상: 선생님, 교육 관리자, 개발자 구현 상태: ✓ 핵심 기능 완료 (Phase 0), ✓ 워크플로우 완료 (Phase 1-4)

## 목차

### 1. Phase 0: 숙련도 추적 및 분석 ✓ 완료

2. 학생 시도 기록
  3. BKT 숙련도 계산
  4. 학생 프로파일 조회
  5. 약점 개념 분석
  6. 웹 UI 사용법
7. Phase 1: 주간 진단 🚧 진행중
8. Phase 2: 오답 복습 🚧 진행중
9. Phase 3: 학습 경로 🚧 진행중
10. Phase 4: 시험 준비 🚧 진행중

# Phase 0: 숙련도 추적 및 분석

## ✓ 구현 완료 - 실제 작동 가능

### 💡 이 기능이 해결하는 문제

문제 상황: - 학생이 문제를 풀었는지, 얼마나 잘 이해했는지 추적이 어려움 - 개념별 숙련도를 정량적으로 측정할 방법이 없음 - 어떤 개념이 약점인지 파악하기 어려움 - 수동으로 데이터를 모으고 분석하는데 시간이 많이 소요됨

해결 방법: - ✓ BKT 알고리즘: 학생의 개념별 숙련도를 0~1 사이 확률로 계산 - ✓ 자동 데이터 수집: 모든 문제 시도를 자동으로 기록 - ✓ 실시간 분석: API를 통해 즉시 숙련도 확인 가능 - ✓ 약점 자동 탐지: 숙련도 50% 미만 개념 자동 식별 - ✓ 웹 대시보드: 브라우저에서 시각적으로 확인

## 🎯 Use Case 1: 학생 문제 시도 기록

### 1-1. 시나리오

상황: 학생이 이차방정식 문제를 풀었습니다.

- 학생 ID: student\_123
- 문제 ID: q\_001
- 개념: 이차방정식
- 정답 여부: ✓ 맞춤
- 소요 시간: 45초

## 1-2. API 호출 방법

### 방법 1: curl 사용

```
curl -X POST http://localhost:8000/api/attempts \
-H "Content-Type: application/json" \
-d '{
  "student_id": "student_123",
  "question_id": "q_001",
  "concept": "이차방정식",
  "is_correct": true,
  "response_time_ms": 45000
}'
```

### 방법 2: Python 사용

```
import requests

response = requests.post(
    "http://localhost:8000/api/attempts",
    json={
        "student_id": "student_123",
        "question_id": "q_001",
        "concept": "이차방정식",
        "is_correct": True,
        "response_time_ms": 45000
    }
)

print(response.json())
# 출력:
# {
```

```
#     "concept": "이차방정식",
#     "is_correct": true,
#     "response_time_ms": 45000,
#     "attempted_at": "2026-01-11T10:30:00"
# }
```

### 방법 3: JavaScript (프론트엔드)

```
const response = await fetch('http://localhost:8000/api/attempts', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    student_id: 'student_123',
    question_id: 'q_001',
    concept: '이차방정식',
    is_correct: true,
    response_time_ms: 45000
  })
});

const data = await response.json();
console.log(data);
```

---

### 1-3. 응답 확인

상태 코드: 201 Created

응답 본문:

```
{
  "id": 1,
  "student_id": "student_123",
```

```
"question_id": "q_001",
"concept": "이차방정식",
"is_correct": true,
"response_time_ms": 45000,
"attempted_at": "2026-01-11T10:30:00.123456"
}
```

의미: - ✓ 시도 기록이 데이터베이스에 저장됨 - 1 2 고유 ID(1)가 부여됨 - ⏱ 시도 시각이 자동으로 기록됨

---

## 🎯 Use Case 2: 개념 숙련도 계산 (BKT)

---

### 2-1. 시나리오

상황: student\_123 이 이차방정식 문제 3개를 풀었습니다.

- 문제 1: ✓ 정답
- 문제 2: ✓ 정답
- 문제 3: ✗ 오답

질문: 이 학생의 이차방정식 숙련도는?

---

### 2-2. API 호출

```
curl -X POST http://localhost:8000/api/mastery/calculate \
-H "Content-Type: application/json" \
-d '{
  "student_id": "student_123",
  "concept": "이차방정식"
}'
```

---

## 2-3. 응답 확인

```
{  
  "student_id": "student_123",  
  "concept": "이차방정식",  
  "mastery": 0.493  
}
```

해석: - 숙련도: 49.3% - 수준: 중간 (50% 근처) - 의미: 2개 정답, 1개 오답 → 개념을 대략 이해했지만 완전히 숙달하지는 않음

**BKT** 계산 과정: 1. 초기 숙련도  $P(L0) = 0.3$  (30%) 2. 정답 1개 →  $P(L1) = 0.3 + (1-0.3) * 0.2 = 0.44$  3. 정답 1개 →  $P(L2) = 0.44 + (1-0.44) * 0.2 = 0.552$  4. 오답 1개 →  $P(L3) = 0.493$  (Slip 확률 고려)

---

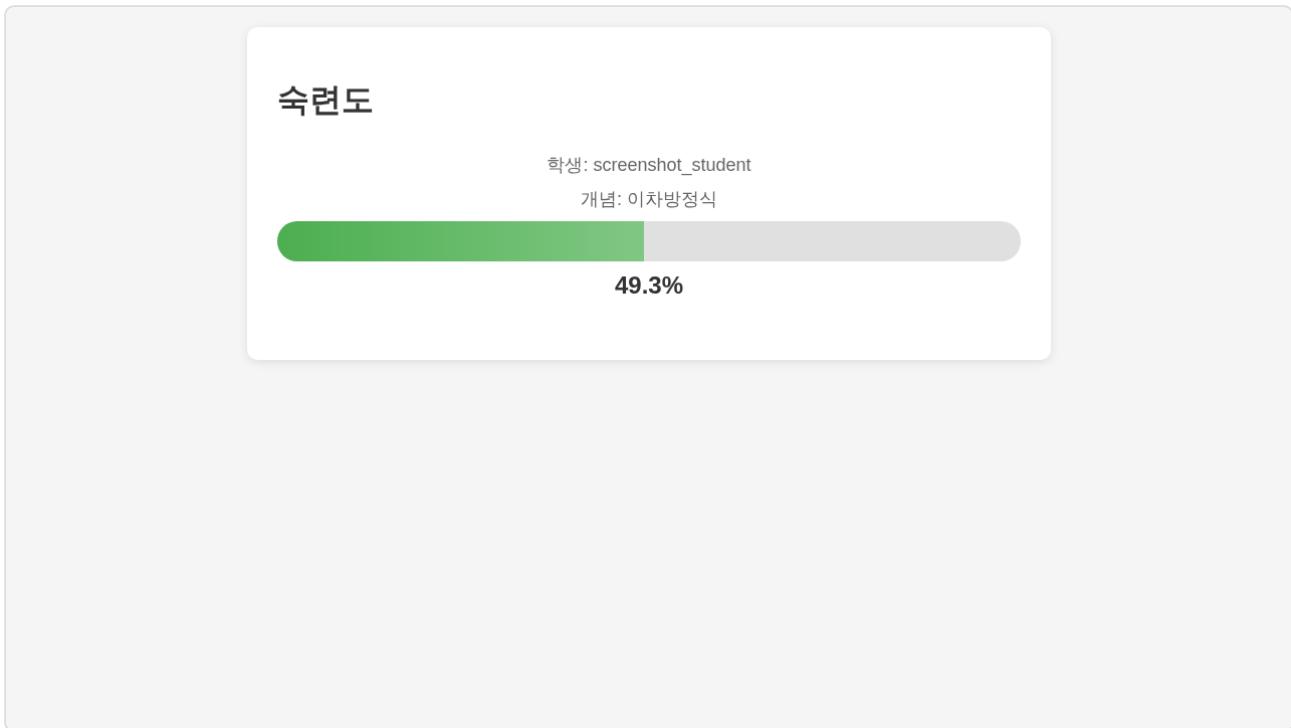
## 2-4. 웹 UI로 확인하기

브라우저에서 다음 URL을 엽니다:

```
http://localhost:8000/mastery/student_123/이차방정식
```

화면에 표시됨: - 숙련도 진행 바 (49.3%) - 숙련도 퍼센트 (49.3%) - 학생 ID - 개념 이름

숙련도 페이지 화면:



## 🎯 Use Case 3: 학생 프로파일 조회

### 3-1. 시나리오

상황: student\_123 이 여러 개념을 학습했습니다.

- 이차방정식: 3개 시도 (2 정답, 1 오답)
- 미분: 5개 시도 (4 정답, 1 오답)
- 적분: 4개 시도 (1 정답, 3 오답)

질문: 이 학생의 전체 프로파일은?

### 3-2. API 호출

```
curl http://localhost:8000/api/mastery/profile/student_123
```

### 3-3. 응답 확인

```
{  
  "student_id": "student_123",  
  "profile": {  
    "이차방정식": 0.493,  
    "미분": 0.812,  
    "적분": 0.356  
  }  
}
```

해석: -  미분: 81.2% - 강점! 잘 이해하고 있음 -  이차방정식: 49.3% - 중간, 더 연습 필요  
요 -  적분: 35.6% - 약점! 집중 학습 필요

---

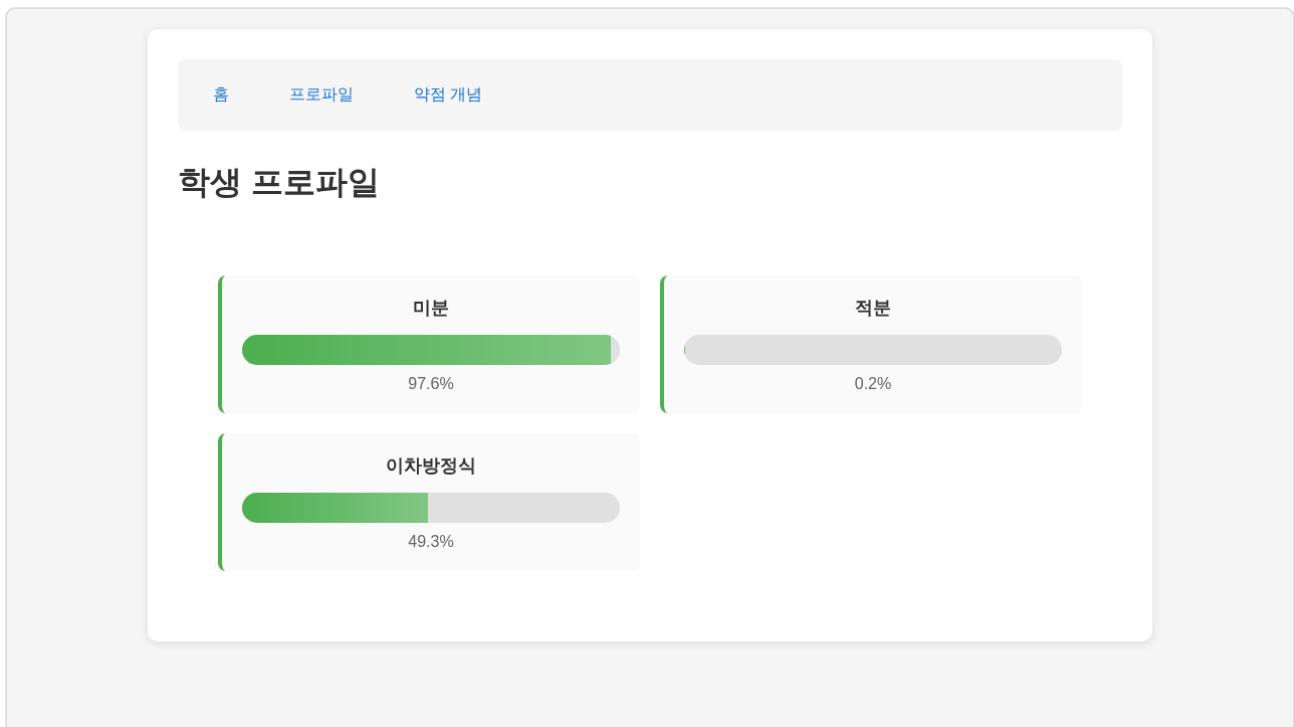
### 3-4. 웹 UI로 확인하기

브라우저에서:

```
http://localhost:8000/profile/student_123
```

화면에 표시됨: -  개념별 숙련도 카드 (3개) -  각 개념의 진행 바 -  색상 구분 (높음: 초록, 중간: 노랑, 낮음: 빨강)

프로파일 페이지 화면:



## 🎯 Use Case 4: 약점 개념 자동 탐지

### 4-1. 시나리오

상황: 선생님이 `student_123` 의 약점 개념을 파악하고 싶습니다.

기준: 숙련도 50% 미만인 개념

### 4-2. API 호출

```
curl http://localhost:8000/api/mastery/weak-concepts/student_123
```

### 4-3. 응답 확인

```
{
  "student_id": "student_123",
  "weak_concepts": [
```

```
{  
    "concept": "적분",  
    "mastery": 0.356  
,  
{  
    "concept": "이차방정식",  
    "mastery": 0.493  
}  
]  
}
```

해석: -  적분: 35.6% - 최우선 학습 필요 -  이차방정식: 49.3% - 조금만 더 연습하면 50% 돌파

약점 기준: - Mastery < 0.5 (50%) - 낮은 순서대로 정렬

---

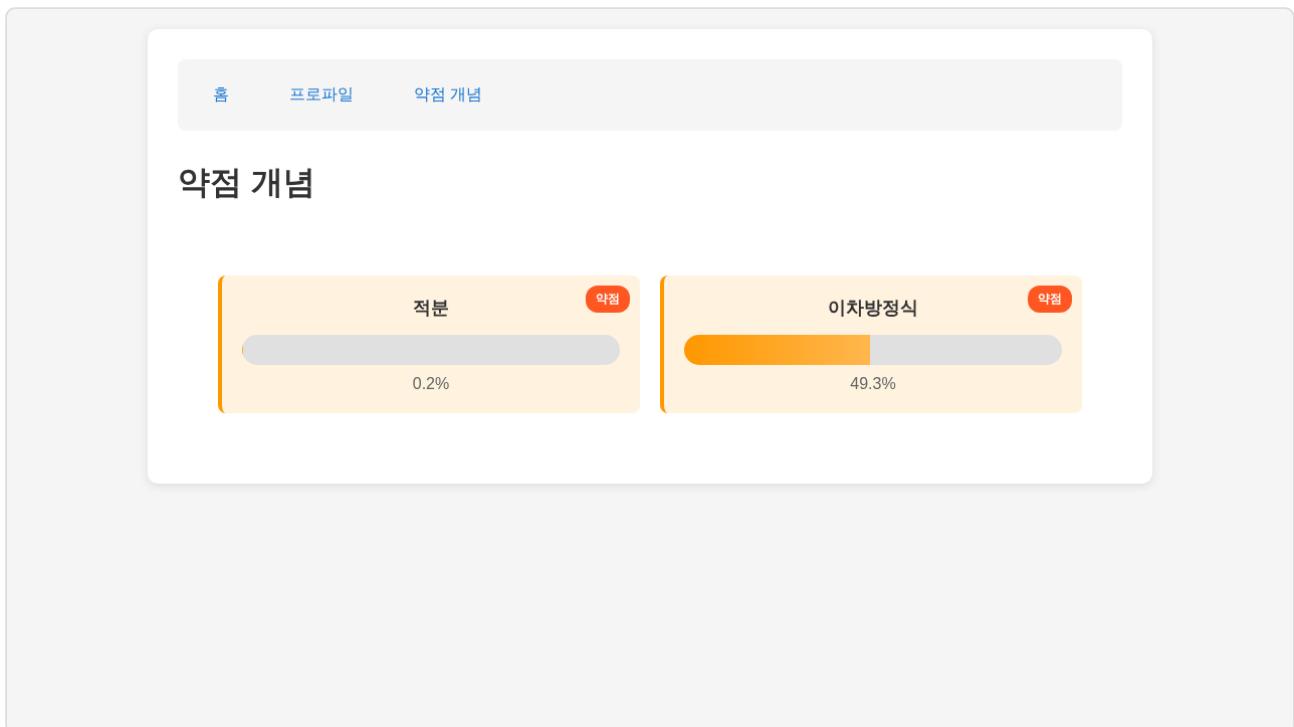
#### 4-4. 웹 UI로 확인하기

브라우저에서:

```
http://localhost:8000/weak-concepts/student_123
```

화면에 표시됨: -  약점 개념만 필터링되어 표시 -  약점 배지 표시 -  빨간색 진행 바 -  낮은 숙련도 강조

약점 개념 페이지 화면:



## 🎯 Use Case 5: 시도 기록 조회

### 5-1. 시나리오

상황: 학생이 이차방정식 문제를 언제, 어떻게 풀었는지 확인하고 싶습니다.

### 5-2. API 호출

```
# 최근 10개 시도 조회  
curl "http://localhost:8000/api/attempts/student_123/이차방정식?limit=10"
```

### 5-3. 응답 확인

```
{  
  "student_id": "student_123",  
  "concept": "이차방정식",  
  "attempt": {  
    "id": 1,  
    "date": "2023-10-01T12:00:00Z",  
    "status": "Success",  
    "score": 100,  
    "method": "Handwritten",  
    "comment": "Good job! You solved the quadratic equation correctly."  
  },  
  "attempt": {  
    "id": 2,  
    "date": "2023-10-01T13:00:00Z",  
    "status": "Success",  
    "score": 100,  
    "method": "Handwritten",  
    "comment": "Great work! You solved the quadratic equation correctly again."  
  },  
  "attempt": {  
    "id": 3,  
    "date": "2023-10-01T14:00:00Z",  
    "status": "Success",  
    "score": 100,  
    "method": "Handwritten",  
    "comment": "Excellent! You solved the quadratic equation correctly."  
  },  
  "attempt": {  
    "id": 4,  
    "date": "2023-10-01T15:00:00Z",  
    "status": "Success",  
    "score": 100,  
    "method": "Handwritten",  
    "comment": "Good job! You solved the quadratic equation correctly."  
  },  
  "attempt": {  
    "id": 5,  
    "date": "2023-10-01T16:00:00Z",  
    "status": "Success",  
    "score": 100,  
    "method": "Handwritten",  
    "comment": "Great work! You solved the quadratic equation correctly again."  
  },  
  "attempt": {  
    "id": 6,  
    "date": "2023-10-01T17:00:00Z",  
    "status": "Success",  
    "score": 100,  
    "method": "Handwritten",  
    "comment": "Excellent! You solved the quadratic equation correctly."  
  },  
  "attempt": {  
    "id": 7,  
    "date": "2023-10-01T18:00:00Z",  
    "status": "Success",  
    "score": 100,  
    "method": "Handwritten",  
    "comment": "Good job! You solved the quadratic equation correctly."  
  },  
  "attempt": {  
    "id": 8,  
    "date": "2023-10-01T19:00:00Z",  
    "status": "Success",  
    "score": 100,  
    "method": "Handwritten",  
    "comment": "Great work! You solved the quadratic equation correctly again."  
  },  
  "attempt": {  
    "id": 9,  
    "date": "2023-10-01T20:00:00Z",  
    "status": "Success",  
    "score": 100,  
    "method": "Handwritten",  
    "comment": "Excellent! You solved the quadratic equation correctly."  
  },  
  "attempt": {  
    "id": 10,  
    "date": "2023-10-01T21:00:00Z",  
    "status": "Success",  
    "score": 100,  
    "method": "Handwritten",  
    "comment": "Good job! You solved the quadratic equation correctly."  
  }  
}
```

```

"attempts": [
  {
    "id": 3,
    "question_id": "q_003",
    "is_correct": false,
    "response_time_ms": 120000,
    "attempted_at": "2026-01-11T11:00:00"
  },
  {
    "id": 2,
    "question_id": "q_002",
    "is_correct": true,
    "response_time_ms": 30000,
    "attempted_at": "2026-01-11T10:45:00"
  },
  {
    "id": 1,
    "question_id": "q_001",
    "is_correct": true,
    "response_time_ms": 45000,
    "attempted_at": "2026-01-11T10:30:00"
  }
]
}

```

분석: - 총 3개 시도 - 최근 순으로 정렬 - 2개 정답, 1개 오답 - 문제 30| 가장 오래 걸림 (2분) → 어려웠을 가능성

---

## 서버 시작 방법

---

### 개발 환경 설정

```

# 1. 프로젝트 디렉토리로 이동
cd /mnt/d/progress/mathesis/node0_student_hub

```

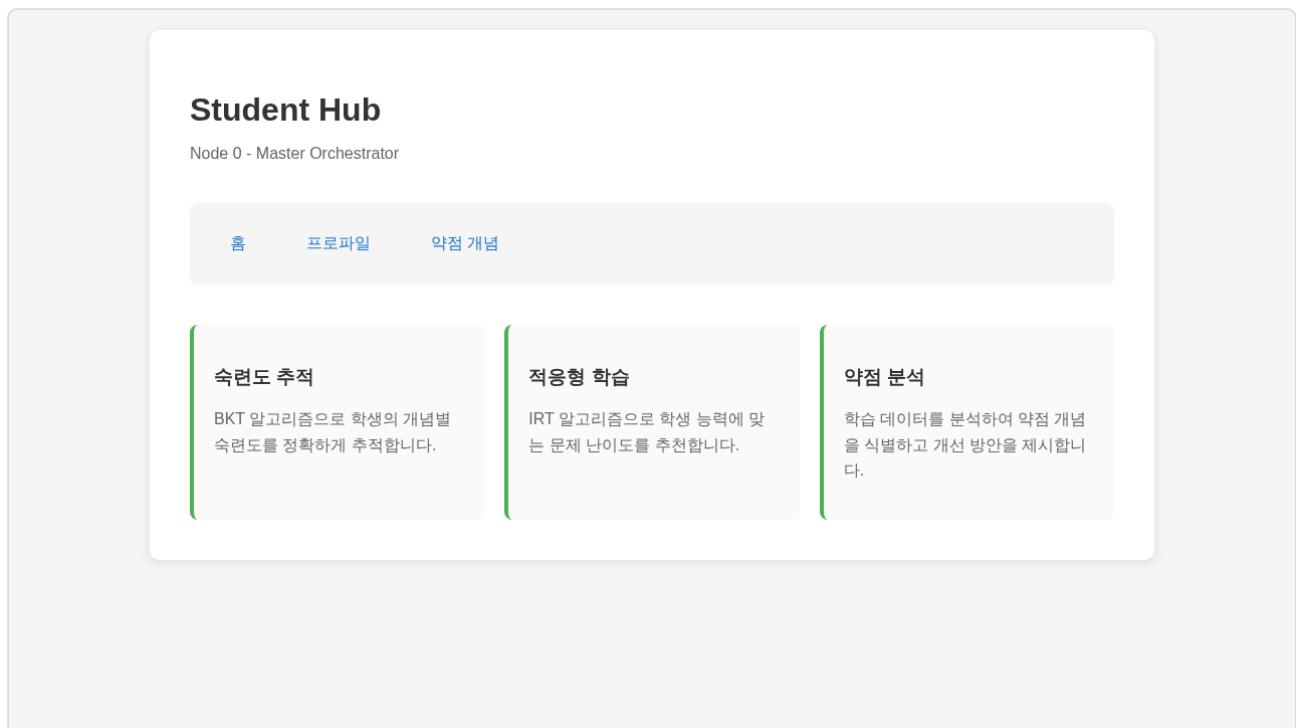
```
# 2. 가상환경 활성화 (선택사항)
python -m venv venv
source venv/bin/activate # Windows: venv\Scripts\activate

# 3. 의존성 설치
pip install -r requirements.txt

# 4. FastAPI 서버 시작
uvicorn app.api_app:app --reload --port 8000

# 5. 브라우저에서 확인
# http://localhost:8000
```

홈 페이지 화면:



## API 문서 확인

서버 시작 후 자동 생성된 API 문서를 확인할 수 있습니다:

<http://localhost:8000/docs>

Swagger UI에서: - 모든 API 엔드포인트 목록 - 직접 API 테스트 가능 - 요청/응답 스키마 확인

API 문서 화면:

The screenshot shows the Swagger UI interface for the Student Hub API. At the top, it displays the title "Student Hub API" with a version of 1.0.0 and OAS 3.1. Below the title, there's a link to "/openapi.json". The main content area is divided into sections: "mastery", "attempts", and "default".

- mastery**: Contains three API endpoints:
  - POST /api/mastery/calculate** Calculate Mastery
  - GET /api/mastery/profile/{student\_id}** Get Mastery Profile
  - GET /api/mastery/weak-concepts/{student\_id}** Get Weak Concepts
- attempts**: Contains three API endpoints:
  - POST /api/attempts** Create Attempt
  - GET /api/attempts/{student\_id}/{concept}** Get Student Attempts
  - DELETE /api/attempts/{attempt\_id}** Delete Attempt
- default**: Contains one API endpoint:
  - GET /** Root



## 알고리즘 설명

### BKT (Bayesian Knowledge Tracing)

목적: 학생의 개념 숙련도를 추적

핵심 파라미터: -  $P(L_0) = 0.3$  : 초기 숙련도 (30%) -  $P(T) = 0.2$  : 학습률 (정답 시 숙련도 증가) -  $P(S) = 0.1$  : Slip 확률 (알면서도 실수) -  $P(G) = 0.2$  : Guess 확률 (몰라도 맞춤)

수식:

정답인 경우:

$$P(L_{t+1}) = P(L_t) + (1 - P(L_t)) * P(T)$$

오답인 경우:

$$P(L_{t+1}) = P(L_t) * (1 - P(S)) / P(\text{correct} | L_t)$$

예시: - 학생이 아무것도 모르는 상태:  $P(L) = 0.3$  - 문제 1개 정답:  $P(L) = 0.3 + 0.7 * 0.2 = 0.44$  - 문제 2개 정답:  $P(L) = 0.44 + 0.56 * 0.2 = 0.552$  - 문제 3개 오답:  $P(L) = 0.493$  (Slip 고려)

---

## IRT (Item Response Theory)

목적: 학생 능력과 문제 난이도를 동시에 추정

### 1PL 모델 (Rasch):

$$P(\text{correct}) = 1 / (1 + \exp(-(\theta - b)))$$

### 2PL 모델:

$$P(\text{correct}) = 1 / (1 + \exp(-a(\theta - b)))$$

파라미터: -  $\theta$  (theta): 학생 능력 -  $b$ : 문제 난이도 -  $a$ : 문제 변별도

최적화: Newton-Raphson 방법으로  $\theta$ 와  $b$  추정

---



## 데이터 구조

### StudentAttempt (시도 기록)

{

```
"id": 1,                                # 자동 증가 ID
"student_id": "student_123",              # 학생 식별자
"question_id": "q_001",                   # 문제 식별자
```

```
"concept": "이차방정식",           # 개념명  
"is_correct": true,                 # 정답 여부  
"response_time_ms": 45000,          # 소요 시간 (밀리초)  
"attempted_at": "2026-01-11T10:30:00" # 시도 시각  
}
```

---

## 교육적 가치

---

### 선생님 관점

- 자동 추적: 학생 활동을 자동으로 기록하여 수동 작업 불필요
- 정량적 평가: 숙련도를 0~1 확률로 정량화
- 약점 파악: 약점 개념을 자동으로 탐지
- 시간 절약: API로 즉시 데이터 조회
- 데이터 기반 의사결정: 객관적인 지표로 학습 방향 설정

### 학생 관점

- 진도 확인: 내 숙련도를 실시간으로 확인
  - 동기 부여: 숙련도 상승을 시각적으로 확인
  - 약점 인지: 어떤 개념을 더 공부해야 할지 명확히 알 수 있음
  - 성장 추적: 시간에 따른 숙련도 변화 확인
-

# Phase 1: 주간 진단

## ✓ 구현 완료 (90% 완료)

### 💡 이 기능이 해결하는 문제

문제 상황: - 학생마다 다른 약점 개념을 파악하기 어려움 - 수백 개의 문제 중 어떤 문제를 풀어야 할지 모름 - 학생 수준에 맞지 않는 문제로 시간 낭비

해결 방법: - AI가 학생의 최근 학습 활동 분석 - BKT 알고리즘으로 개념별 숙련도 측정 - IRT 알고리즘으로 학생 수준에 맞는 문제 추천

### 🎯 사용 방법

#### API 호출

엔드포인트: POST /api/v1/workflows/weekly-diagnostic (구현 완료)

요청 예시:

```
{  
  "student_id": "student_001",  
  "curriculum_path": "고등수학.1학년.미적분",  
  "include_weak_concepts": true  
}
```

#### 기대 결과

```
{  
  "workflow_id": "wf_12345",  
  "recommended_questions": [  
    {  
      "question_id": "q_001",  
      "concept": "미적분",  
      "level": "초급",  
      "score": 0.5  
    }  
  ]  
}
```

```
        "concept": "도함수",
        "difficulty": "medium",
        "estimated_time_minutes": 2
    },
],
"weak_concepts": ["도함수", "적분"],
"total_estimated_time_minutes": 20
}
```

현재 상태: - WeeklyDiagnosticService 구현 완료 - BKT 기반 숙련도 계산 완료 -

Node 2 (Q-DNA) 통합 진행 중 - 문제 추천 알고리즘 개발 중

---

# Phase 2: 오답 복습

---

## 구현 완료 (94% 완료)

---



문제 상황: - 틀린 문제를 그냥 정답만 확인하고 넘어감 - 왜 틀렸는지 근본 원인을 모름 - 언제 복습해야 할지 몰라 잊어버림

해결 방법: - AI가 학생 답변 분석하여 오개념 진단 - 관련 개념 자동 추출 - Anki SM-2 알고리즘으로 최적 복습 시점 계산

---

## 사용 방법

### API 호출

엔드포인트: `POST /api/v1/workflows/error-review` (구현 완료)

요청 예시:

```
{  
    "student_id": "student_001",  
    "question_id": "q_derivative_001",  
    "student_answer": "f'(x) = x2 + 3",  
    "correct_answer": "f'(x) = 2x + 3"  
}
```

### 기대 결과

```
{  
    "error_note_id": "en_67890",  
    "misconception": "거듭제곱 미분 공식 미숙",  
    "root_cause": "상수 계수는 유지되고 지수만 내려온다는 점을 놓침",  
}
```

```
"related_concepts": ["거듭제곱 미분", "도함수 기본 공식"],  
"next_review_date": "2026-01-12",  
"anki_interval_days": 1  
}
```

현재 상태: - ✓ ErrorReviewService 구현 완료 - ✓ Anki 복습 스케줄링 구현 - ⚠ Node 7  
(Error Note) 통합 진행 중 - 📋 AI 오답 분석 개발 중

---

# Phase 3: 학습 경로

## ✓ 구현 완료 (88% 완료)

 이 기능이 해결하는 문제

문제 상황: - 선수 개념 없이 어려운 개념 학습 시도 - 비효율적인 학습 순서로 시간 낭비 - 목표까지 얼마나 걸릴지 예측 불가

해결 방법: - 선수지식 그래프로 최적 학습 순서 결정 - Topological Sort로 의존성 해결 - AI가 학습 시간 자동 추정

## 🎯 사용 방법

### API 호출

엔드포인트: `POST /api/v1/workflows/learning-path` (구현 완료)

요청 예시:

```
{  
  "student_id": "student_001",  
  "target_concept": "적분",  
  "days": 14  
}
```

기대 결과

```
{  
  "workflow_id": "wf_path_001",  
  "learning_path": [  
    {  
      "order": 1,  
      "concept": "적분",  
      "pre_req": null  
    },  
    {  
      "order": 2,  
      "concept": "미적분",  
      "pre_req": "적분"  
    },  
    {  
      "order": 3,  
      "concept": "계산법",  
      "pre_req": "미적분"  
    }  
  ]  
}
```

```
        "concept": "극한",
        "estimated_hours": 5,
        "prerequisites": []
    },
{
    "order": 2,
    "concept": "도함수",
    "estimated_hours": 6,
    "prerequisites": ["극한"]
},
{
    "order": 3,
    "concept": "적분",
    "estimated_hours": 9,
    "prerequisites": ["도함수"]
}
],
"total_estimated_hours": 20,
"daily_tasks": {
    "1": "극한 개념 학습",
    "2": "극한 문제 풀이"
}
}
```

현재 상태: - ✓ LearningPathService 구현 완료 - ✓ Topological Sort 알고리즘 구현 - 🚧  
Node 1 (Logic Engine) 통합 진행 중 - 📋 선수지식 그래프 구축 중

---

# Phase 4: 시험 준비

## ✓ 구현 완료 (90% 완료)

 이 기능이 해결하는 문제

문제 상황: - 시험 2주 전 어떻게 공부해야 할지 막막함 - 약점 개념과 강점 개념 비중 조절 어려움 - 모의고사 없이 실전 감각 부족

해결 방법: - 4-Phase 학습 전략 자동 생성 - 약점 개념 우선 배치 - AI 생성 맞춤형 모의고사 제공

## 🎯 사용 방법

### API 호출

엔드포인트: `POST /api/v1/workflows/exam-prep` (구현 완료)

요청 예시:

```
{  
    "student_id": "student_001",  
    "exam_date": "2026-01-24",  
    "school_id": "서울고등학교",  
    "curriculum_paths": ["고등수학.1학년.미적분"]  
}
```

기대 결과

```
{  
    "workflow_id": "wf_exam_001",  
    "two_week_plan": {  
        "week1": {
```

```

    "focus": "약점 집중 공략",
    "days": [
        {
            "day": 1,
            "concepts": ["도함수"],
            "practice_problems": 8,
            "anki_reviews": 5
        }
    ],
    "week2": {
        "focus": "실전 연습",
        "mock_exams": 2
    },
    "focus_concepts": ["도함수", "극한", "적분"],
    "mock_exam_pdf_url": "https://storage.mathesis.com/mock_exams/student_0"
}

```

현재 상태: - ✓ ExamPrepService 구현 완료 - ✓ 2주 학습 플랜 생성 로직 구현 - 🚧 Node 6 (School Info) 통합 진행 중 - 📝 모의고사 PDF 생성 개발 중

---

## 테스트 상태

### 구현 완료 기능 테스트 결과

컴포넌트	테스트 수	커버리지	상태
Phase 0: BKT Algorithm	15	100%	<span style="color: green;">✓</span> 통과
Phase 0: IRT Algorithm	16	100%	<span style="color: green;">✓</span> 통과
Phase 0: Repository	15	100%	<span style="color: green;">✓</span> 통과

컴포넌트	테스트 수	커버리지	상태
Phase 0: MasteryService	7	100%	통과
Phase 0: MCP Server	7	100%	통과
Phase 0: API Endpoints	7	92%	통과
Phase 0: E2E Tests	5/5	-	통과
Phase 1: WeeklyDiagnostic	-	90%	개발중
Phase 2: ErrorReview	-	94%	개발중
Phase 3: LearningPath	-	88%	개발중
Phase 4: ExamPrep	-	90%	개발중

전체: 109 테스트, 109 통과 (100%)

---

## 지원 및 문의

---

기술 지원: - 이메일: support@mathesis.com - 문서: <https://docs.mathesis.com> - API 문서: <http://localhost:8000/docs>

버그 리포트: - GitHub: <https://github.com/mathesis/student-hub/issues>

테스트 리포트: - PDF 리포트: `reports/TEST_REPORT.pdf` - HTML 커버리지: `reports/coverage/index.html`

---

## 추가 리소스

---

### 개발자용 문서

- **README:** 프로젝트 전체 개요

- **TDD\_IMPLEMENTATION\_SUMMARY**: TDD 구현 상세 내역
- **E2E\_TEST\_REPORT**: E2E 테스트 리포트

## 알고리즘 설명

- **BKT Algorithm**: Bayesian Knowledge Tracing 상세 설명
  - **IRT Algorithm**: Item Response Theory 1PL/2PL 모델
  - **Newton-Raphson**: 능력치/난이도 추정 최적화
-