

# 뉴스 요약 서비스 테스트 케이스 설계서

---

## 1. 개요

뉴스 요약 서비스의 기능 검증을 위한 테스트 케이스를 정의합니다. 단위 테스트, 통합 테스트, API 테스트를 포함합니다.

## 2. 테스트 환경 구성

### 2.1 테스트 데이터베이스

- H2 인메모리 데이터베이스 사용
- 테스트용 뉴스 데이터 및 요약 캐시 데이터 준비

### 2.2 Mock 서비스

- Flask API Mock 서버 구성
- 외부 의존성 격리를 위한 Mock 객체 사용

## 3. 단위 테스트 케이스

### 3.1 NewsSummaryController 테스트

#### TC-001: 뉴스 ID 기반 요약 성공

**목적:** 뉴스 ID로 요약 요청 시 정상 처리 확인

**전제조건:**

- 유효한 뉴스 ID가 존재
- 해당 뉴스의 요약이 캐시에 없음

**테스트 데이터:**

```
{
  "newsId": 12345,
  "type": "DEFAULT",
  "lines": 3
}
```

**예상 결과:**

```
{
  "newsId": 12345,
  "resolvedType": "DEFAULT",
  "lines": 3,
  "summary": "요약된 텍스트...",
  "cached": false,
}
```

```
"createAt": "2024-01-01T00:00:00"
}
```

### TC-002: 캐시된 요약 반환

목적: 이미 캐시된 요약이 있을 때 캐시 반환 확인

전제조건:

- 요청한 뉴스 ID와 타입의 요약이 DB에 캐시되어 있음

테스트 데이터:

```
{
  "newsId": 12345,
  "type": "DEFAULT",
  "lines": 3
}
```

예상 결과:

```
{
  "newsId": 12345,
  "resolvedType": "DEFAULT",
  "lines": 3,
  "summary": "캐시된 요약 텍스트...",
  "cached": true,
  "createAt": "2024-01-01T00:00:00"
}
```

### TC-003: 강제 재생성 요청

목적: force=true로 캐시 무시하고 재생성 확인

전제조건:

- 요청한 뉴스 ID와 타입의 요약이 DB에 캐시되어 있음

테스트 데이터:

```
{
  "newsId": 12345,
  "type": "DEFAULT",
  "lines": 3,
  "force": true
}
```

**예상 결과:**

```
{
  "newsId": 12345,
  "resolvedType": "DEFAULT",
  "lines": 3,
  "summary": "새로 생성된 요약 텍스트...",
  "cached": false,
  "createAt": "2024-01-01T00:00:00"
}
```

**TC-004: 텍스트 기반 임시 요약**

**목적:** 텍스트로 임시 요약 생성 확인

**테스트 데이터:**

```
{
  "text": "요약할 뉴스 내용입니다. 이 텍스트는 요약되어야 합니다.",
  "type": "DEFAULT",
  "lines": 2
}
```

**예상 결과:**

```
{
  "newsId": 0,
  "resolvedType": "DEFAULT",
  "lines": 2,
  "summary": "요약된 텍스트...",
  "cached": false,
  "createAt": "2024-01-01T00:00:00"
}
```

**TC-005: newsId와 text 모두 제공 시 newsId 우선**

**목적:** newsId와 text가 모두 있을 때 newsId 우선 처리 확인

**테스트 데이터:**

```
{
  "newsId": 12345,
  "text": "이 텍스트는 무시되어야 합니다.",
  "type": "DEFAULT",
}
```

```
"lines": 3
}
```

**예상 결과:** newsId 기반 요약 처리됨

#### TC-006: 필수 파라미터 누락 에러

**목적:** newsId와 text 모두 없을 때 에러 반환 확인

**테스트 데이터:**

```
{
  "type": "DEFAULT",
  "lines": 3
}
```

**예상 결과:**

```
{
  "timestamp": "2024-01-01T00:00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "newsId 또는 text 중 하나는 필수입니다."
}
```

#### TC-007: 존재하지 않는 뉴스 ID 에러

**목적:** 존재하지 않는 뉴스 ID 요청 시 에러 반환 확인

**테스트 데이터:**

```
{
  "newsId": 99999,
  "type": "DEFAULT",
  "lines": 3
}
```

**예상 결과:**

```
{
  "timestamp": "2024-01-01T00:00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "뉴스를 찾을 수 없습니다: id=99999"
}
```

### 3.2 유효성 검증 테스트

#### TC-008: lines 범위 검증 (최소값)

목적: lines가 1 미만일 때 검증 에러 확인

테스트 데이터:

```
{
  "newsId": 12345,
  "lines": 0
}
```

예상 결과: 400 Bad Request

#### TC-009: lines 범위 검증 (최대값)

목적: lines가 10 초과일 때 검증 에러 확인

테스트 데이터:

```
{
  "newsId": 12345,
  "lines": 11
}
```

예상 결과: 400 Bad Request

#### TC-010: 빈 텍스트 요청

목적: 빈 텍스트로 요청 시 처리 확인

테스트 데이터:

```
{
  "text": "",
  "type": "DEFAULT",
  "lines": 3
}
```

예상 결과: 빈 요약 반환

## 4. 통합 테스트 케이스

## 4.1 Flask API 연동 테스트

### TC-011: Flask API 정상 호출

목적: Flask API와의 정상적인 연동 확인

테스트 시나리오:

1. 뉴스 ID로 요약 요청
2. Flask API 호출 확인
3. 응답 데이터 검증

### TC-012: Flask API 타임아웃 처리

목적: Flask API 응답 지연 시 타임아웃 처리 확인

테스트 시나리오:

1. Flask API 응답 지연 시뮬레이션
2. 타임아웃 발생 확인
3. 적절한 에러 응답 반환 확인

### TC-013: Flask API 에러 처리

목적: Flask API 에러 시 적절한 처리 확인

테스트 시나리오:

1. Flask API 500 에러 시뮬레이션
2. 에러 응답 처리 확인
3. 클라이언트에게 적절한 에러 메시지 전달

## 4.2 데이터베이스 연동 테스트

### TC-014: 요약 캐시 저장

목적: 생성된 요약이 DB에 정상 저장되는지 확인

테스트 시나리오:

1. 새로운 뉴스 ID로 요약 요청
2. DB에 요약 데이터 저장 확인
3. 캐시 키 (newsId, summaryType) 확인

### TC-015: 요약 캐시 업데이트

목적: 기존 캐시 데이터 업데이트 확인

테스트 시나리오:

1. 기존 캐시가 있는 뉴스로 force=true 요청
2. DB 데이터 업데이트 확인

### 3. 새로운 요약 텍스트 반영 확인

#### TC-016: 동시 요청 처리

목적: 동일한 뉴스에 대한 동시 요청 처리 확인

테스트 시나리오:

1. 동일한 뉴스 ID로 여러 요청 동시 발생
2. DB 락 처리 확인
3. 데이터 일관성 유지 확인

## 5. 성능 테스트 케이스

### 5.1 응답 시간 테스트

#### TC-017: 캐시 히트 응답 시간

목적: 캐시된 요약 반환 시 응답 시간 측정

테스트 조건:

- 캐시된 요약이 있는 상태
- 목표 응답 시간: 100ms 이하

#### TC-018: 캐시 미스 응답 시간

목적: 새로운 요약 생성 시 응답 시간 측정

테스트 조건:

- 캐시되지 않은 뉴스 요청
- Flask API 호출 포함
- 목표 응답 시간: 2초 이하

### 5.2 부하 테스트

#### TC-019: 동시 사용자 부하

목적: 다중 사용자 동시 요청 처리 성능 확인

테스트 조건:

- 100명 동시 사용자
- 각 사용자 10회 요청
- 목표 성공률: 95% 이상

#### TC-020: 대용량 텍스트 처리

목적: 긴 텍스트 요약 처리 성능 확인

테스트 조건:

- 10,000자 이상 텍스트
- 목표 응답 시간: 5초 이하

## 6. 보안 테스트 케이스

### 6.1 입력 검증 테스트

#### TC-021: SQL 인젝션 방지

목적: 악의적인 SQL 인젝션 시도 차단 확인

테스트 데이터:

```
{
  "newsId": "12345; DROP TABLE news_summary; --",
  "type": "DEFAULT"
}
```

#### TC-022: XSS 방지

목적: 악의적인 스크립트 삽입 시도 차단 확인

테스트 데이터:

```
{
  "text": "<script>alert('XSS')</script>",
  "type": "DEFAULT"
}
```

## 7. 테스트 실행 방법

### 7.1 단위 테스트 실행

```
./gradlew test --tests NewsSummaryControllerTest
./gradlew test --tests NewsSummaryServiceTest
```

### 7.2 통합 테스트 실행

```
./gradlew integrationTest
```

### 7.3 API 테스트 실행



```
./gradlew apiTest
```

## 8. 테스트 데이터 준비

### 8.1 테스트용 뉴스 데이터

```
INSERT INTO news (id, title, content, created_at) VALUES  
(12345, '테스트 뉴스 제목', '테스트 뉴스 내용입니다.', NOW()),  
(12346, '테스트 뉴스 제목 2', '테스트 뉴스 내용 2입니다.', NOW());
```

### 8.2 테스트용 요약 캐시 데이터

```
INSERT INTO news_summary (news_id, summary_type, lines, summary_text)  
VALUES  
(12345, 'DEFAULT', 3, '캐시된 요약 텍스트입니다.');
```

## 9. 테스트 결과 검증 기준

### 9.1 성공 기준

- 모든 단위 테스트 통과
- 통합 테스트 성공률 95% 이상
- API 응답 시간 목표 달성
- 에러 처리 정상 동작

### 9.2 실패 기준

- 단위 테스트 실패
- 통합 테스트 성공률 95% 미만
- API 응답 시간 목표 미달성
- 보안 취약점 발견