

신발 매장 API 가이드 문서

버전: 1.0

최종 업데이트: 2025-01-XX

기본 URL: <http://127.0.0.1:8000>



목차

1. API 개요
 2. 기본 설정
 3. 인증 및 보안
 4. 공통 응답 형식
 5. 기본 CRUD API
 6. JOIN API
 7. 특수 기능 API
 8. 에러 처리
 9. 사용 예시
-

API 개요

서버 정보

- 프레임워크: FastAPI
- 데이터베이스: MySQL
- 데이터 형식: JSON (Form 데이터 방식)
- 문서: Swagger UI (<http://127.0.0.1:8000/docs>)

API 구조

- 기본 **CRUD API**: 14개 테이블에 대한 CRUD 작업
 - **JOIN API**: 복잡한 조인 쿼리를 위한 6개 API 그룹
 - 총 엔드포인트: 약 100개 이상
-

기본 설정

서버 실행

```
cd backend/app_new_form
python main.py
```

또는

```
cd backend
uvicorn app_new_form.main:app --host 127.0.0.1 --port 8000 --reload
```

헬스 체크

```
GET /health
```

응답 예시:

```
{
  "status": "healthy",
  "database": "connected"
}
```

루트 엔드포인트

```
GET /
```

응답 예시:

```
{
  "message": "Shoes Store API - 새로운 ERD 구조",
  "status": "running",
  "endpoints": {
    "branches": "/api/branches",
    "users": "/api/users",
    ...
  }
}
```

인증 및 보안

현재 버전에서는 인증이 구현되지 않았습니다. 향후 JWT 토큰 기반 인증이 추가될 예정입니다.

공통 응답 형식

성공 응답

목록 조회:

```
{
  "results": [
    {
      "id": 1,
      "name": "값"
    }
  ]
}
```

단일 조회:

```
{
  "result": {
    "id": 1,
    "name": "값"
  }
}
```

생성/수정/삭제:

```
{
  "result": "OK",
  "id": 1 // 생성 시에만 포함
}
```

에러 응답

```
{
  "result": "Error",
  "errorMsg": "에러 메시지",
  "message": "상세 메시지" // 선택적
}
```

기본 CRUD API

1. 지점 (Branch)

기본 경로: `/api/branches`

메서드	엔드포인트	설명
GET	<code>/api/branches</code>	전체 지점 조회
GET	<code>/api/branches/{br_seq}</code>	지점 상세 조회

메서드	엔드포인트	설명
POST	/api/branches	지점 추가
POST	/api/branches/{br_seq}	지점 수정
DELETE	/api/branches/{br_seq}	지점 삭제

데이터 모델:

```
{
  "br_seq": 1,
  "br_name": "강남점",
  "br_phone": "02-1234-5678",
  "br_address": "서울시 강남구 테헤란로 123",
  "br_lat": 37.5010,
  "br_lng": 127.0260
}
```

2. 고객 (User)

기본 경로: /api/users

메서드	엔드포인트	설명
GET	/api/users	전체 고객 조회
GET	/api/users/{u_seq}	고객 상세 조회
POST	/api/users	고객 추가 (이미지 필수)
POST	/api/users/{u_seq}	고객 수정
POST	/api/users/{u_seq}/with_image	고객 수정 (이미지 포함)
GET	/api/users/{u_seq}/profile_image	프로필 이미지 조회
DELETE	/api/users/{u_seq}/profile_image	프로필 이미지 삭제
DELETE	/api/users/{u_seq}	고객 삭제

데이터 모델:

```
{
  "u_seq": 1,
  "u_id": "user001",
  "u_password": "hashed_password",
  "u_name": "홍길동",
  "u_phone": "010-1111-1111",
  "u_address": "서울시 강남구",
  "created_at": "2025-01-15T10:30:00",
}
```

```
{
  "u_quit_date": null
}
```

고객 추가 예시 (Form 데이터):

```
curl -X POST "http://127.0.0.1:8000/api/users" \
  -F "u_id=user001" \
  -F "u_password=pass1234" \
  -F "u_name=홍길동" \
  -F "u_phone=010-1111-1111" \
  -F "u_address=서울시 강남구" \
  -F "file=@profile.jpg"
```

3. 직원 (Staff)

기본 경로: `/api/staffs`

메서드	엔드포인트	설명
GET	<code>/api/staffs</code>	전체 직원 조회
GET	<code>/api/staffs/{s_seq}</code>	직원 상세 조회
GET	<code>/api/staffs/by_branch/{br_seq}</code>	지점별 직원 조회
POST	<code>/api/staffs</code>	직원 추가 (이미지 필수)
POST	<code>/api/staffs/{s_seq}</code>	직원 수정
POST	<code>/api/staffs/{s_seq}/with_image</code>	직원 수정 (이미지 포함)
GET	<code>/api/staffs/{s_seq}/profile_image</code>	프로필 이미지 조회
DELETE	<code>/api/staffs/{s_seq}/profile_image</code>	프로필 이미지 삭제
DELETE	<code>/api/staffs/{s_seq}</code>	직원 삭제

데이터 모델:

```
{
  "s_seq": 1,
  "s_id": "staff001",
  "br_seq": 1,
  "s_password": "hashed_password",
  "s_name": "김점장",
  "s_phone": "010-1001-1001",
  "s_rank": "점장",
  "s_superseq": null,
  "created_at": "2025-01-15T10:30:00",
}
```

```

    "s_quit_date": null
  }

```

직원 추가 예시:

```

curl -X POST "http://127.0.0.1:8000/api/staffs" \
  -F "s_id=staff001" \
  -F "br_seq=1" \
  -F "s_password=pass1234" \
  -F "s_name=김점장" \
  -F "s_phone=010-1001-1001" \
  -F "s_rank=점장" \
  -F "file=@profile.jpg"

```

4. 제조사 (Maker)

기본 경로: `/api/makers`

메서드	엔드포인트	설명
GET	<code>/api/makers</code>	전체 제조사 조회
GET	<code>/api/makers/{m_seq}</code>	제조사 상세 조회
POST	<code>/api/makers</code>	제조사 추가
POST	<code>/api/makers/{m_seq}</code>	제조사 수정
DELETE	<code>/api/makers/{m_seq}</code>	제조사 삭제

5. 카테고리 (Categories)

5.1 종류 카테고리 (Kind Category)

기본 경로: `/api/kind_categories`

메서드	엔드포인트	설명
GET	<code>/api/kind_categories</code>	전체 조회
GET	<code>/api/kind_categories/{kc_seq}</code>	상세 조회
POST	<code>/api/kind_categories</code>	추가
POST	<code>/api/kind_categories/{kc_seq}</code>	수정
DELETE	<code>/api/kind_categories/{kc_seq}</code>	삭제

5.2 색상 카테고리 (Color Category)

기본 경로: `/api/color_categories`

5.3 사이즈 카테고리 (Size Category)

기본 경로: `/api/size_categories`

5.4 성별 카테고리 (Gender Category)

기본 경로: `/api/gender_categories`

6. 제품 (Product)

기본 경로: `/api/products`

메서드	엔드포인트	설명
메서드	엔드포인트	설명
-----	-----	-----
GET	<code>/api/products</code>	전체 제품 조회
GET	<code>/api/products/{p_seq}</code>	제품 상세 조회
GET	<code>/api/products/by_maker/{m_seq}</code>	제조사별 제품 조회
POST	<code>/api/products</code>	제품 추가
POST	<code>/api/products/{p_seq}</code>	제품 수정
POST	<code>/api/products/{p_seq}/stock</code>	제품 재고 수정
DELETE	<code>/api/products/{p_seq}</code>	제품 삭제

데이터 모델:

```
{
  "p_seq": 1,
  "kc_seq": 1,
  "cc_seq": 1,
  "sc_seq": 1,
  "gc_seq": 1,
  "m_seq": 1,
  "p_name": "에어맥스 90",
  "p_price": 150000,
  "p_stock": 50,
  "p_image": "/images/product_1.jpg",
  "p_description": "나이키 에어맥스 90 클래식",
  "created_at": "2025-01-15T10:30:00"
}
```

제품 추가 예시:

```
curl -X POST "http://127.0.0.1:8000/api/products" \
  -F "kc_seq=1" \
  -F "cc_seq=1" \
  -F "sc_seq=1" \
  -F "gc_seq=1" \
  -F "m_seq=1" \
  -F "p_name=에어맥스 90" \
  -F "p_price=150000" \
  -F "p_stock=50" \
  -F "p_image=/images/product_1.jpg" \
  -F "p_description=나이키 에어맥스 90 클래식"
```

7. 구매 내역 (Purchase Item)

기본 경로: `/api/purchase_items`

메서드	엔드포인트	설명
GET	<code>/api/purchase_items</code>	전체 구매 내역 조회
GET	<code>/api/purchase_items/{b_seq}</code>	구매 내역 상세 조회
GET	<code>/api/purchase_items/by_user/{u_seq}</code>	고객별 구매 내역 조회
GET	<code>/api/purchase_items/by_datetime</code>	분 단위 그룹화된 주문 조회
POST	<code>/api/purchase_items</code>	구매 내역 추가
POST	<code>/api/purchase_items/{b_seq}</code>	구매 내역 수정
DELETE	<code>/api/purchase_items/{b_seq}</code>	구매 내역 삭제

데이터 모델:

```
{
  "b_seq": 1,
  "br_seq": 1,
  "u_seq": 1,
  "p_seq": 1,
  "b_price": 150000,
  "b_quantity": 2,
  "b_date": "2025-01-15T14:30:00",
  "b_tnum": "TXN0001",
  "b_status": "주문완료"
}
```

구매 내역 추가 예시:


```
curl -X POST "http://127.0.0.1:8000/api/purchase_items" \
  -F "br_seq=1" \
  -F "u_seq=1" \
  -F "p_seq=1" \
  -F "b_price=150000" \
  -F "b_quantity=2" \
  -F "b_date=2025-01-15T14:30:00" \
  -F "b_status=주문완료"
```

본 단위 그룹화 조회:

```
curl "http://127.0.0.1:8000/api/purchase_items/by_datetime?
user_seq=1&order_datetime=2025-01-15%2014:30&branch_seq=1"
```

8. 수령 (Pickup)

기본 경로: `/api/pickups`

메서드	엔드포인트	설명
메서드	엔드포인트	설명
-----	-----	-----
GET	<code>/api/pickups</code>	전체 수령 내역 조회
GET	<code>/api/pickups/{pic_seq}</code>	수령 내역 상세 조회
GET	<code>/api/pickups/{b_seq}</code>	구매 내역별 수령 조회
POST	<code>/api/pickups</code>	수령 내역 추가
POST	<code>/api/pickups/{pic_seq}</code>	수령 내역 수정
POST	<code>/api/pickups/{pic_seq}/complete</code>	수령 완료 처리
DELETE	<code>/api/pickups/{pic_seq}</code>	수령 내역 삭제

데이터 모델:

```
{
  "pic_seq": 1,
  "b_seq": 1,
  "u_seq": 1,
  "created_at": "2025-01-15T15:00:00"
}
```

수령 내역 추가 예시:

```
curl -X POST "http://127.0.0.1:8000/api/pickups" \
  -F "b_seq=1" \
  -F "u_seq=1"
```

9. 반품 (Refund)

기본 경로: `/api/refunds`

메서드	엔드포인트	설명
메서드	엔드포인트	설명
-----	-----	-----
GET	<code>/api/refunds</code>	전체 반품 내역 조회
GET	<code>/api/refunds/{ref_seq}</code>	반품 내역 상세 조회
GET	<code>/api/refunds/by_user/{u_seq}</code>	고객별 반품 내역 조회
POST	<code>/api/refunds</code>	반품 내역 추가
POST	<code>/api/refunds/{ref_seq}</code>	반품 내역 수정
POST	<code>/api/refunds/{ref_seq}/process</code>	반품 처리
DELETE	<code>/api/refunds/{ref_seq}</code>	반품 내역 삭제

데이터 모델:

```
{
  "ref_seq": 1,
  "ref_date": "2025-01-15T16:00:00",
  "ref_reason": "사이즈 불일치",
  "ref_re_seq": 1,
  "ref_re_content": "260 사이즈가 너무 작습니다",
  "u_seq": 1,
  "s_seq": 1,
  "pic_seq": 1
}
```

10. 입고 (Receive)

기본 경로: `/api/receives`

메서드	엔드포인트	설명
메서드	엔드포인트	설명

메서드	엔드포인트	설명
-----	-----	-----
GET	/api/receives	전체 입고 내역 조회
GET	/api/receives/{rec_seq}	입고 내역 상세 조회
GET	/api/receives/{p_seq}	제품별 입고 내역 조회
POST	/api/receives	입고 내역 추가
POST	/api/receives/{rec_seq}	입고 내역 수정
POST	/api/receives/{rec_seq}/process	입고 처리
DELETE	/api/receives/{rec_seq}	입고 내역 삭제

11. 발주 (Request)

기본 경로: /api/requests

메서드	엔드포인트	설명
메서드	엔드포인트	설명
-----	-----	-----
GET	/api/requests	전체 발주 내역 조회
GET	/api/requests/{req_seq}	발주 내역 상세 조회
POST	/api/requests	발주 내역 추가
POST	/api/requests/{req_seq}	발주 내역 수정
POST	/api/requests/{req_seq}/approve_manager	팀장 결재 처리
POST	/api/requests/{req_seq}/approve_director	이사 결재 처리
DELETE	/api/requests/{req_seq}	발주 내역 삭제

JOIN API

1. 제품 JOIN API

기본 경로: /api/products

1.1 제품 전체 상세 조회

```
GET /api/products/{p_seq}/full_detail
```

설명: 제품 + 모든 카테고리 + 제조사 정보 (6테이블 JOIN)

응답 예시:

```
{
  "result": {
    "p_seq": 1,
    "p_name": "에어맥스 90",
    "p_price": 150000,
    "p_stock": 50,
    "p_image": "/images/product_1.jpg",
    "kind_name": "러닝화",
    "color_name": "블랙",
    "size_name": "260",
    "gender_name": "남성",
    "maker_name": "나이키",
    "maker_phone": "02-1111-1111",
    "maker_address": "서울시 강남구"
  }
}
```

1.2 제품 목록 + 카테고리 조회

GET /api/products/with_categories

설명: 모든 제품과 카테고리 정보를 함께 조회 (필터링 가능)

쿼리 파라미터:

- **maker_seq** (선택): 제조사 ID
- **kind_seq** (선택): 종류 카테고리 ID
- **color_seq** (선택): 색상 카테고리 ID
- **size_seq** (선택): 사이즈 카테고리 ID
- **gender_seq** (선택): 성별 카테고리 ID

예시:

```
# 전체 제품 조회
curl "http://127.0.0.1:8000/api/products/with_categories"

# 필터링: 나이키 제품 중 남성용
curl "http://127.0.0.1:8000/api/products/with_categories?
maker_seq=1&gender_seq=1"
```

응답 예시:

```
{
  "results": [
    {
      "p_seq": 1,
      "p_name": "에어맥스 90",
      "p_price": 150000,
      "p_stock": 50,
      "p_image": "/images/product_1.jpg",
      "kind_name": "러닝화",
      "color_name": "블랙",
      "size_name": "260",
      "gender_name": "남성",
      "maker_name": "나이키"
    }
  ]
}
```

1.3 제조사별 제품 목록

```
GET /api/products/by_maker/{maker_seq}/with_categories
```

설명: 특정 제조사의 모든 제품과 카테고리 정보 조회

1.4 카테고리별 제품 목록

```
GET /api/products/by_category
```

설명: 여러 카테고리 조건으로 제품 필터링

쿼리 파라미터:

- **maker_seq** (선택): 제조사 ID
- **kind_seq** (선택): 종류 카테고리 ID
- **color_seq** (선택): 색상 카테고리 ID
- **size_seq** (선택): 사이즈 카테고리 ID
- **gender_seq** (선택): 성별 카테고리 ID

2. 구매 내역 JOIN API

기본 경로: `/api/purchase_items`

2.1 구매 내역 상세 조회

```
GET /api/purchase_items/{b_seq}/with_details
```

설명: 구매 내역 + 고객 + 제품 + 지점 정보 (4테이블 JOIN)

2.2 구매 내역 전체 상세 조회

```
GET /api/purchase_items/{b_seq}/full_detail
```

설명: 구매 내역 + 고객 + 제품 + 지점 + 모든 카테고리 + 제조사 (9테이블 JOIN)

2.3 분 단위 그룹화된 주문 조회

```
GET /api/purchase_items/by_datetime/with_details
```

파라미터:

- **user_seq** (필수): 고객 번호
- **order_datetime** (필수): 주문 일시 (YYYY-MM-DD HH:MM 형식)
- **branch_seq** (필수): 지점 번호

예시:

```
curl "http://127.0.0.1:8000/api/purchase_items/by_datetime/with_details?
user_seq=1&order_datetime=2025-01-15%2014:30&branch_seq=1"
```

응답 예시:

```
{
  "result": {
    "order_minute": "2025-01-15 14:30",
    "branch_name": "강남점",
    "item_count": 3,
    "total_amount": 450000,
    "items": [
      {
        "b_seq": 1,
        "b_price": 150000,
        "b_quantity": 2,
        "b_status": "주문완료",
        "product_name": "에어맥스 90",
        "branch_name": "강남점"
      }
    ]
  }
}
```

2.4 고객별 주문 목록

```
GET /api/purchase_items/by_user/{user_seq}/orders
```

설명: 특정 고객의 모든 주문 목록 (분 단위 그룹화)

응답 예시:

```
{
  "results": [
    {
      "order_minute": "2025-01-15 14:30",
      "order_date_full": "2025-01-15T14:30:45",
      "branch_seq": 1,
      "item_count": 3,
      "total_amount": 450000,
      "items": [...]
    }
  ]
}
```

2.5 고객별 구매 내역 상세 목록

```
GET /api/purchase_items/by_user/{user_seq}/with_details
```

설명: 특정 고객의 모든 구매 내역과 상세 정보 조회

3. 수령 JOIN API

기본 경로: `/api/pickups`

3.1 수령 상세 조회

```
GET /api/pickups/{pic_seq}/with_details
```

설명: 수령 + 구매 내역 + 고객 + 제품 + 지점 정보 (5테이블 JOIN)

3.2 수령 전체 상세 조회

```
GET /api/pickups/{pic_seq}/full_detail
```

설명: 수령 + 구매 내역 + 고객 + 제품 + 지점 + 모든 카테고리 + 제조사 (10테이블 JOIN)

3.3 고객별 수령 목록

```
GET /api/pickups/by_user/{user_seq}/with_details
```

3.4 지점별 수령 목록

```
GET /api/pickups/by_branch/{branch_seq}/with_details
```

4. 반품 JOIN API

기본 경로: /api/refunds

4.1 반품 상세 조회

```
GET /api/refunds/{ref_seq}/with_details
```

설명: 반품 + 고객 + 직원 + 수령 + 구매 내역 + 제품 + 지점 정보 (7테이블 JOIN)

4.2 반품 전체 상세 조회

```
GET /api/refunds/{ref_seq}/full_detail
```

설명: 반품 + 고객 + 직원 + 수령 + 구매 내역 + 제품 + 지점 + 모든 카테고리 + 제조사 (12테이블 JOIN)

4.3 고객별 반품 목록

```
GET /api/refunds/by_user/{user_seq}/with_details
```

4.4 직원별 반품 목록

```
GET /api/refunds/by_staff/{staff_seq}/with_details
```

5. 입고 JOIN API

기본 경로: `/api/receives`

5.1 입고 상세 조회

```
GET /api/receives/{rec_seq}/with_details
```

설명: 입고 + 직원 + 제품 + 제조사 정보 (4테이블 JOIN)

5.2 입고 전체 상세 조회

```
GET /api/receives/{rec_seq}/full_detail
```

설명: 입고 + 직원 + 제품 + 제조사 + 모든 카테고리 정보 (9테이블 JOIN)

5.3 직원별 입고 목록

```
GET /api/receives/by_staff/{staff_seq}/with_details
```

설명: 특정 직원이 처리한 모든 입고 내역 조회

5.4 제품별 입고 목록

```
GET /api/receives/by_product/{product_seq}/with_details
```

설명: 특정 제품의 모든 입고 내역 조회

5.5 제조사별 입고 목록

```
GET /api/receives/by_maker/{maker_seq}/with_details
```

설명: 특정 제조사의 모든 입고 내역 조회

6. 발주 JOIN API

기본 경로: `/api/requests`

6.1 발주 상세 조회

```
GET /api/requests/{req_seq}/with_details
```

설명: 발주 + 직원 + 제품 + 제조사 정보 (4테이블 JOIN)

6.2 발주 전체 상세 조회

```
GET /api/requests/{req_seq}/full_detail
```

설명: 발주 + 직원 + 제품 + 제조사 + 모든 카테고리 정보 (9테이블 JOIN)

6.3 직원별 발주 목록

```
GET /api/requests/by_staff/{staff_seq}/with_details
```

설명: 특정 직원이 생성한 모든 발주 내역 조회

6.4 결제 상태별 발주 목록

```
GET /api/requests/by_status?status={status}
```

설명: 결제 상태별 발주 목록 조회

파라미터:

- **status** (필수): **pending** (대기), **manager_approved** (팀장승인), **director_approved** (이사승인), **all** (전체)

예시:

```
# 대기 중인 발주 조회
curl "http://127.0.0.1:8000/api/requests/by_status?status=pending"
```

6.5 제품별 발주 목록

```
GET /api/requests/by_product/{product_seq}/with_details
```

설명: 특정 제품의 모든 발주 내역 조회

6.6 제조사별 발주 목록

```
GET /api/requests/by_maker/{maker_seq}/with_details
```

설명: 특정 제조사의 모든 발주 내역 조회

특수 기능 API

주문 그룹화

구매 내역은 **b_date** 필드를 기준으로 분 단위(YYYY-MM-DD HH:MM)로 그룹화됩니다.

그룹화 규칙:

- 같은 분에 구매한 항목들이 하나의 주문으로 묶임
- 같은 고객(**u_seq**)과 같은 지점(**br_seq**)에서 구매한 항목만 그룹화
- 예: 2025-01-15 14:30에 구매한 모든 항목이 하나의 주문

사용 예시:

```
# 특정 분의 주문 조회
curl "http://127.0.0.1:8000/api/purchase_items/by_datetime/with_details?
user_seq=1&order_datetime=2025-01-15%2014:30&branch_seq=1"

# 고객의 모든 주문 목록 조회
curl "http://127.0.0.1:8000/api/purchase_items/by_user/1/orders"
```

에러 처리

공통 에러 코드

HTTP 상태 코드	의미	설명
200	OK	요청 성공
400	Bad Request	잘못된 요청
404	Not Found	리소스를 찾을 수 없음
500	Internal Server Error	서버 오류

에러 응답 형식

```
{
  "result": "Error",
  "errorMsg": "에러 메시지",
  "message": "상세 메시지" // 선택적
}
```

주요 에러 케이스

1. 리소스를 찾을 수 없음

```
{
  "result": "Error",
  "message": "User not found"
}
```

2. 중복 데이터

```
{
  "result": "Error",
  "errorMsg": "(1062, \"Duplicate entry 'user001' for key 'user.idx_user_id'\")"
}
```

3. 외래 키 제약 조건 위반

```
{
  "result": "Error",
  "errorMsg": "(1452, \"Cannot add or update a child row: a foreign key constraint fails\")"
}
```

사용 예시

예시 1: 고객 가입 및 주문

```
# 1. 고객 가입
curl -X POST "http://127.0.0.1:8000/api/users" \
  -F "u_id=user001" \
  -F "u_password=pass1234" \
  -F "u_name=홍길동" \
  -F "u_phone=010-1111-1111" \
  -F "u_address=서울시 강남구" \
  -F "file=@profile.jpg"

# 응답: {"result": "OK", "u_seq": 1}

# 2. 제품 조회
curl "http://127.0.0.1:8000/api/products/1/full_detail"
```

```
# 3. 구매 내역 추가
curl -X POST "http://127.0.0.1:8000/api/purchase_items" \
  -F "br_seq=1" \
  -F "u_seq=1" \
  -F "p_seq=1" \
  -F "b_price=150000" \
  -F "b_quantity=2" \
  -F "b_date=2025-01-15T14:30:00" \
  -F "b_status=주문완료"

# 4. 주문 목록 조회
curl "http://127.0.0.1:8000/api/purchase_items/by_user/1/orders"
```

예시 2: 수령 처리

```
# 1. 수령 내역 추가
curl -X POST "http://127.0.0.1:8000/api/pickups" \
  -F "b_seq=1" \
  -F "u_seq=1"

# 2. 수령 상세 조회
curl "http://127.0.0.1:8000/api/pickups/1/full_detail"
```

예시 3: 반품 처리

```
# 1. 반품 내역 추가
curl -X POST "http://127.0.0.1:8000/api/refunds" \
  -F "ref_reason=사이즈 불일치" \
  -F "ref_re_seq=1" \
  -F "ref_re_content=260 사이즈가 너무 작습니다" \
  -F "u_seq=1" \
  -F "s_seq=1" \
  -F "pic_seq=1"

# 2. 반품 상세 조회
curl "http://127.0.0.1:8000/api/refunds/1/full_detail"
```

데이터 타입 및 형식

날짜/시간 형식

- 데이터베이스: **DATETIME** (YYYY-MM-DD HH:MM:SS)
- **API 요청**: ISO 8601 형식 (2025-01-15T14:30:00) 또는 YYYY-MM-DD HH:MM
- **API 응답**: ISO 8601 형식 (2025-01-15T14:30:00.000000)

이미지 업로드

- 형식: Form 데이터 (`multipart/form-data`)
- 필드명: `file`
- 지원 형식: JPEG, PNG 등
- 저장 방식: `MEDIUMBLOB` (데이터베이스에 직접 저장)

주문 그룹화 날짜 형식

- 형식: `YYYY-MM-DD HH:MM`
- 예시: `2025-01-15 14:30`
- URL 인코딩: `2025-01-15%2014:30`

주의사항

1. 이미지 업로드: 고객/직원 추가 시 이미지는 필수입니다.
2. 외래 키 제약: 참조하는 테이블의 데이터가 먼저 존재해야 합니다.
3. UNIQUE 제약: `u_id`, `s_id`, `u_phone`, `s_phone` 등은 중복될 수 없습니다.
4. 주문 그룹화: 같은 분에 구매한 항목만 그룹화되므로, 정확한 시간 설정이 중요합니다.
5. 소프트 삭제: `u_quit_date`, `s_quit_date`를 설정하여 탈퇴 처리를 할 수 있습니다.

추가 리소스

- Swagger UI: `http://127.0.0.1:8000/docs`
- ReDoc: `http://127.0.0.1:8000/redoc`
- 비교 문서: `API_COMPARISON.md`
- 데이터베이스 스키마: `backend/database/renew/shoes_shop_db_mysql_init_improved.sql`

문서 버전: 1.0

최종 업데이트: 2025-01-XX