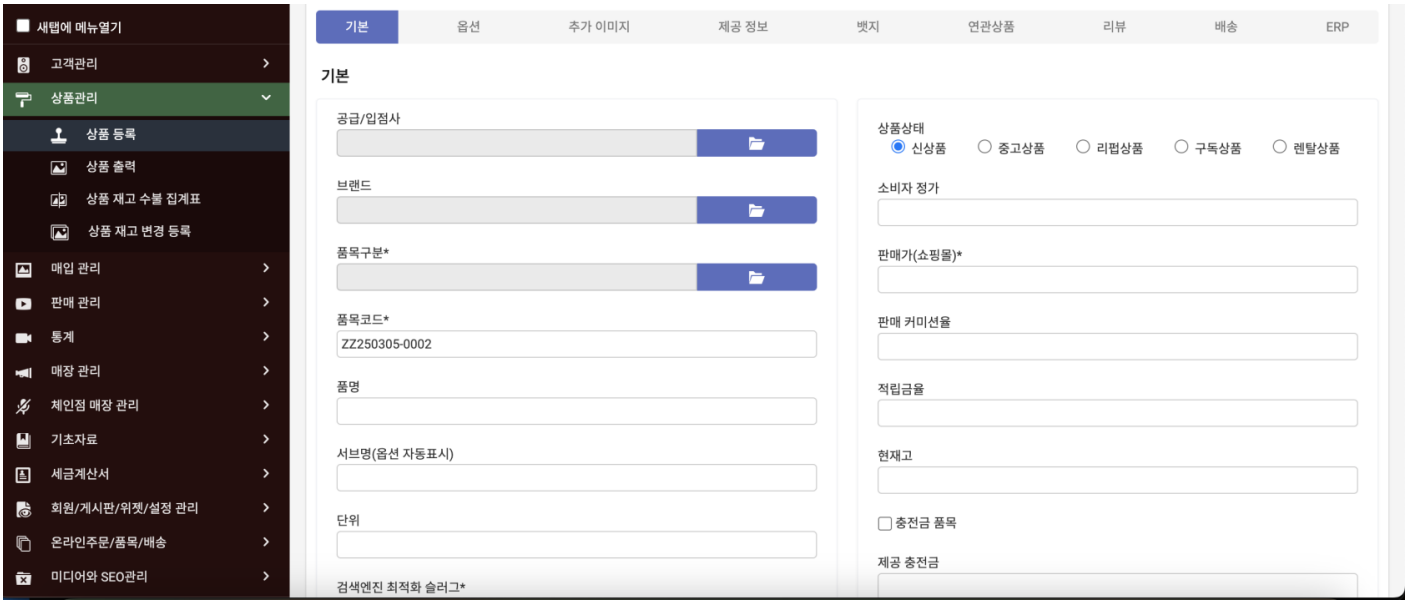
STRONG-API

- □□
- STRONG-API□ □□
- STRONG-API □□□□



개요

□□□ API (main_api)란?

표준 **API(main_api)**는 여러 회사에서 공통적으로 사용하는 핵심 **API**입니다.

예를 들어, 주문 관리, 재고 관리, 결제 처리 등의 기능을 표준 **API**가 제공할 수 있습니다.

하지만 모든 회사가 동일한 방식으로 운영되지는 않기 때문에, 개별적인 기능 확장이 필요할 수 있습니다.

새탭에 메뉴열기
고객관리
상품관리
상품 등록
상품 출력
상품 재고 수불 집계표
상품 재고 변경 등록
매입 관리
판매 관리
통계
매장 관리
체인점 매장 관리
기초자료
세금계산서
회원/게시판/위젯/설정 관리
온라인주문/품목/배송
미디어와 SEO관리
기본
옵션
추가 이미지
제공 정보
뱃지
연관상품
리뷰
배송
ERP

기본

공급/입점사

브랜드

품목구분*

품목코드* ZZ250305-0002

품명

서브명(옵션 자동표시)

단위

검색엔진 최적화 슬러그*

상품상태 ● 신상품 ○ 중고상품 ○ 리펍상품 ○ 구독상품 ○ 렌탈상품

소비자 정가

판매가(쇼핑몰)*

판매 커미션율

적립금율

현재고

☐ 충전금 품목

제공 충전금

□□ Strong API란?

□ Strong API는 클라이언트 요청을 받아 컨트롤러에서 처리 후 응답을 반환하는 기본적인 API입니다.

(예: 클라이언트가 주문 정보를 요청하면, 컨트롤러가 데이터를 가공하여 응답을 반환함)

□ □□ API(main_api)를 변경하지 않고도 회사의 요구에 맞춰 새로운 기능을 쉽게 추가할 수 있도록 설계되었습니다

□ □□□ □□□□□□ □□□ □□□ □□ □□

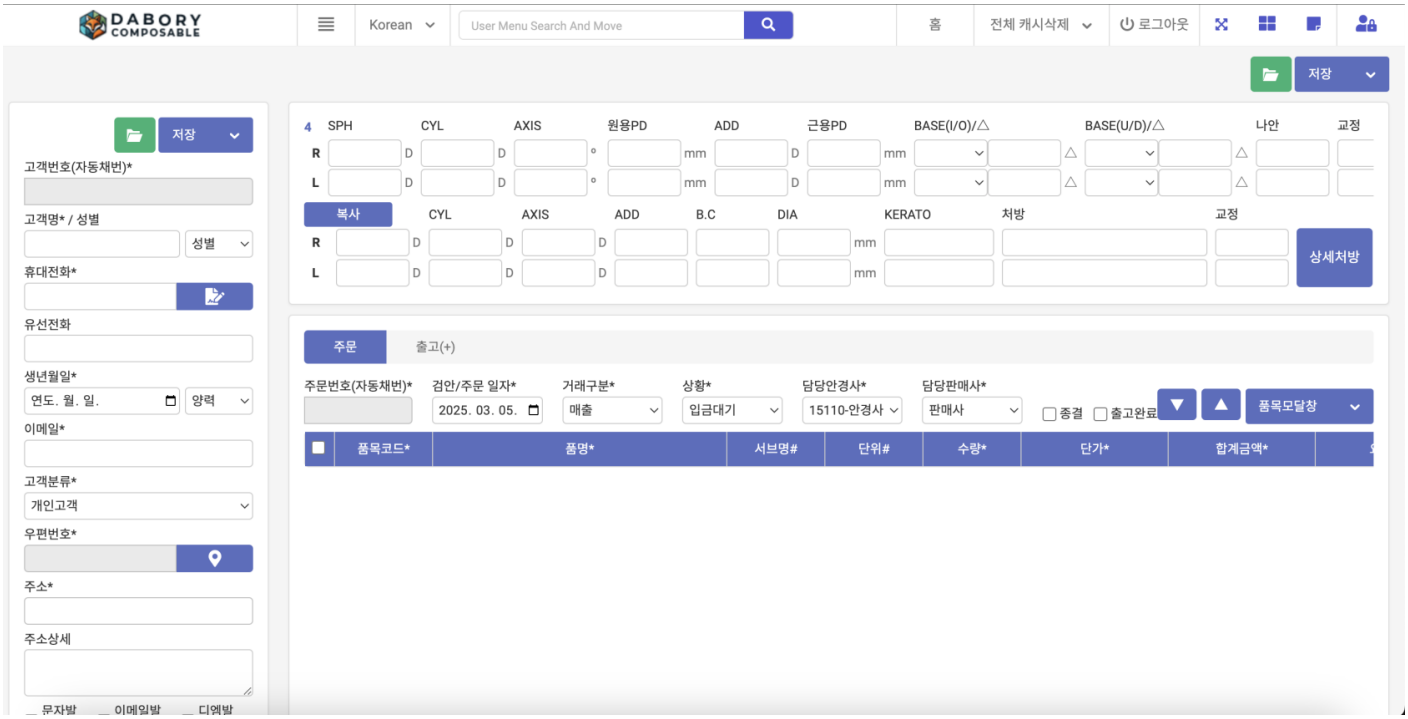
□□ 유지보수가 쉬운 구조 & 빠른 커스터마이징 가능!

□□ 왜 **Strong API**를 사용할까?

□□ 표준 백엔드를 직접 수정하면 표준을 사용하는 모든 사용자에게 영향을 미치므로 각 회사마다 원하는 기능을 추가하기 어렵습니다

□□ 하지만 Strong API를 사용하면 표준을 유지하면서도 원하는 기능을 쉽게 추가 가능!

ex) 안경사에 맞춘 검안/판매 기능 Strong 커스터마이징


DABORY COMPOSABLE
Korean
User Menu Search And Move
홈
전체 캐시삭제
로그아웃
저장
고객번호(자동채번)*
고객명* / 성별
성별
휴대전화*
유선전화
생년월일*
양력
이메일*
고객분류*
개인고객
우편번호*
주소*
주소상세
SPH
CYL
AXIS
원용PD
ADD
근용PD
BASE(I/O)/△
BASE(U/D)/△
나안
교정
복사
B.C
DIA
KERATO
처방
상세처방
주문
출고(+)
주문번호(자동채번)*
검안/주문 일자*
2025. 03. 05.
거래구분*
매출
상황*
입금대기
담당안경사*
15110-안경사
담당판매사*
판매사
종결
출고완료
품목모달창
품목코드*
품명*
서브명#
단위#
수량*
단가*
합계금액*

□□ □□□□ □□□ STRONG-API□ □□

STRONG-API의 구조

Dabory의 Strong-API 구조 (strong-go)

```
|── dabory/                 # Dabory 관련 코드
│   ├── themes/opticalpos/    # OpticalPOS 테마 관련 코드
│   │   ├── erp/            # ERP 관련 코드
│   │   ├── pro/            # Pro 관련 코드
│   │   ├── qry/            # Query를 분리하여 코드
│   │   ├── strong/           # Strong API Directory
│   │   │   ├── backend/strong-go/  # Strong API 백엔드 코드
│   │   │   │   ├── conf/      # 설정 파일
│   │   │   │   ├── controllers/  # 컨트롤러 로직
│   │   │   │   ├── handlers/ # 컨트롤러와 모델 역할을 둘 다 하는 로직
│   │   │   │   ├── locals/   # 유틸리티 기능을 제공하는 패키지
│   │   │   │   ├── models/   # 데이터 처리 로직 (Table 구조체 정의)
│   │   │   │   ├── routers/  # routers 패키지는 클라이언트 요청을 받아 해당 요청을 적절한 컨트롤러로 전달하는 역할
│   │   │   │   ├── tmp/       # 임시 파일
│   │   │   │   ├── vendor/   # 외부 라이브러리 (Go Modules)
│   │   │   │   ├── .gitignore  # Git에서 제외할 파일 목록
│   │   │   │   ├── docker-compose.yml  # Docker 설정 파일
│   │   │   │   ├── go.mod      # Go 모듈 관리 파일
│   │   │   │   ├── go.sum      # Go 의존성 관리 파일
│   │   │   │   ├── main.go     # 프로그램 진입점 (Entry Point)
│   │   │   │   ├── Readme.md  # 프로젝트 설명 파일
│   │   │   │   ├── strong-go  # 빌드된 실행 파일
```

▯▯ 디렉토리 상세 설명

▯ **`dabory/`**

- Dabory 프로젝트 관련 전체 코드가 포함된 최상위 디렉토리

▯ **`themes/opticalpos/`**

- OpticalPOS 테마 관련 코드가 들어 있는 디렉토리

[`qry/`

- Query가 포함된 디렉토리 (Query를 따로 분리함으로써 유연성 증가)

[`strong/backend/strong-go/`

- **Strong API**의 백엔드 코드가 포함된 디렉토리.

[`conf/`

- 설정**(Configuration)** 관련 파일이 포함된 디렉토리
- DB 연결 설정, 환경 변수, 서비스 설정 파일 등이 위치함

[`controllers/`

- 클라이언트 요청을 처리하는 컨트롤러 로직이 포함된 디렉토리
- 요청을 받아 적절한 로직을 실행한 후 응답을 반환하는 역할을 수행함

[`handlers/`

- 컨트롤러와 모델 역할을 둘 다 하는 로직이 포함된 디렉토리
- 특정 API 엔드포인트에서 직접 데이터 처리 및 응답을 수행하는 기능을 가짐

[`locals/`

- 유틸리티 기능을 제공하는 패키지
- 해싱, 암호화, 설정값 로드 등의 공통 기능을 포함함

[`models/`

- 데이터베이스 테이블과 관련된 구조체 정의 및 데이터 처리 로직이 포함된 디렉토리
- ORM(Object Relational Mapping)을 활용하여 DB와 상호작용함

[`routers/`

- 클라이언트 요청을 적절한 컨트롤러로 전달하는 역할을 수행하는 디렉토리
- API 엔드포인트 정의 및 요청을 라우팅함

[`tmp/`

- [illegible]
- [illegible], [illegible], [illegible] [illegible] [illegible] [illegible]

[**vendor/**

- **Go Modules**[illegible] [illegible] [illegible] [illegible]
- [illegible] [illegible] [illegible] [illegible]

[**.gitignore**

- **Git**[illegible] [illegible] [illegible] [illegible]
- [illegible] [illegible] [illegible], [illegible], [illegible] [illegible] [illegible] [illegible] [illegible]

[**docker-compose.yml**

- **Docker** [illegible] [illegible]
- Docker [illegible] [illegible] [illegible] [illegible] [illegible]

[**go.mod**

- **Go** [illegible] [illegible] [illegible]
- [illegible] [illegible] Go [illegible] [illegible] [illegible]

[**go.sum**

- **Go** [illegible] [illegible] [illegible]
- `go.mod` [illegible] [illegible] [illegible] [illegible] [illegible] [illegible]

[**main.go**

- [illegible] [illegible]**(Entry Point)**
- `main()` [illegible] [illegible] [illegible], [illegible] [illegible] [illegible] [illegible]

[**Readme.md**

- [illegible] [illegible] [illegible]
- [illegible] [illegible], [illegible] [illegible], [illegible] [illegible] [illegible] [illegible] [illegible]

[**strong-go**

- **Go** [illegible] [illegible] [illegible] [illegible]
- `go build` [illegible] [illegible] [illegible]

O

Choose a file

STRONG-API □□□