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/`

- 임시 파일을 저장하는 디렉토리

- 캐시, 로그, 임시 데이터 등을 저장할 때 사용됨

[`vendor/`

- **Go Modules**에서 관리하는 외부 라이브러리 디렉토리
- 프로젝트에서 사용하는 서드파티 라이브러리들이 포함됨

[`.gitignore`

- **Git**에서 제외할 파일 목록을 정의하는 파일
- 빌드된 바이너리 파일, 로그, 환경 설정 파일 등 불필요한 파일을 제외하는 역할

[`docker-compose.yml`

- **Docker** 설정 파일
- Docker 컨테이너를 실행할 때 사용되는 설정이 포함됨

[`go.mod`

- **Go** 모듈 관리 파일
- 프로젝트에서 사용하는 Go 패키지와 버전을 정의함

[`go.sum`

- **Go** 의존성 관리 파일
- `go.mod`에 명시된 패키지의 버전 및 체크섬을 저장하는 파일

[`main.go`

- 프로그램의 진입점**(Entry Point)**
- `main()` 함수가 포함되어 있으며, 프로그램 실행 시 최초로 실행됨

[`Readme.md`

- 프로젝트 설명 파일
- 프로젝트의 개요, 사용 방법, 설정 방법 등을 문서화한 파일

[`strong-go`

- **Go** 언어로 빌드된 실행 파일
- `go build` 명령을 실행하면 생성됨

O

Choose a file

Revision #4
Created 5 March 2025 07:25:58 by □□□
Updated 5 March 2025 08:57:27 by □□