3□. Frontend □□ □□

- .env □ □□
- .env.dabory □ Gate Token
- □□□ Frontend □□ □□
- □□□ Theme □□□ □□

.env □ □□

> “
>
> .env □□ □□ □□

□□□□

Lavavel .env □□ □□□(https://laravel.kr/docs/5.7/configuration)

Lavavel .env Example(https://zetawiki.com/wiki/□□□_.env,_.env.example)

```
# [Dabory]
ERP_THEMES=□□□□□ (□□ □□ □□□ □□)
LOCALE_SEQUENCE=□□□ □□ □□
APP_MOBILE_NO=□□ □□X
TAG_LINE=□□ □□X
CDN_TYPE=□□□ CDN (aws-s3 or local)
MEDIA_URL=□□□ base url □□(□□□□ local or a3)
FAVICON_PATH=□□□ □□ path
IS_SKIP_DBUPDATE=DBUPDATE □□ or □□
IS_ON_MEMBER_SIGNUP=□□ □□X

## [MetaMask]
CHAIN_ID=0xE67B
CHAIN_NAME='Dang Smart Chain'
RPC_URLS=http://3.38.62.211:8545
NATIVE_CURRENCY_NAME=DGT
NATIVE_CURRENCY_SYMBOL=DGT
NATIVE_CURRENCY_DECIMALS=18
BLOCK_EXPLORER_URLS=https://explorer-staging.dangnn.co.kr

# [Strong Api]
STRGO_API_URL= strong api □□□ □□□□□.
### [asset & cache]
BROWSER_CACHE=□□□□ □□□□  (□□□□□□ false)
```

CSSJS_URL=local or asset url □□ (□□□□□□ local)

[Printer]
REPORT_SERVER_URL=□□□ □□ base url

[Laravel]
APP_NAME=□ □□
APP_ENV=.env □□ □□□ □□
APP_KEY=base64:.env □□ □□□ □□
APP_DEBUG=.env □□ □□□ □□
APP_URL=□ url
LOG_CHANNEL=stack
LOG_LEVEL=debug
DB_CONNECTION=□□x
DB_HOST=□□x
DB_PORT=□□x
DB_DATABASE=□□x
DB_USERNAME=□□x
DB_PASSWORD=□□x

BROADCAST_DRIVER=□□x
CACHE_DRIVER=□□x
FILESYSTEM_DRIVER=□□x
QUEUE_CONNECTION=□□x
SESSION_DRIVER=□□x
SESSION_LIFETIME=□□x
MEMCACHED_HOST=□□x
REDIS_HOST=□□x
REDIS_PASSWORD=□□x
REDIS_PORT=□□x

MAIL_MAILER=.env □□ □□□ □□
MAIL_HOST=.env □□ □□□ □□
MAIL_PORT=.env □□ □□□ □□
MAIL_USERNAME=.env □□ □□□ □□
MAIL_PASSWORD=.env □□ □□□ □□
MAIL_ENCRYPTION=.env □□ □□□ □□
MAIL_FROM_ADDRESS=.env □□ □□□ □□
MAIL_FROM_NAME=.env □□ □□□ □□

AWS_ACCESS_KEY_ID=.env □□ □□□ □□
AWS_SECRET_ACCESS_KEY=.env □□ □□□ □□
AWS_DEFAULT_REGION=.env □□ □□□ □□
AWS_BUCKET=.env □□ □□□ □□
AWS_USE_PATH_STYLE_ENDPOINT=.env □□ □□□ □□

PUSHER_APP_ID=□□x
PUSHER_APP_KEY=□□x
PUSHER_APP_SECRET=□□x
PUSHER_APP_CLUSTER=□□x

MIX_PUSHER_APP_KEY=□□x
MIX_PUSHER_APP_CLUSTER=□□x

ANALYTICS_UA_ID=
ANALYTICS_VIEW_ID=

ANALYTICS_G_ID=
ANALYTICS_PROPERTY_ID=

.env.dabory 와 Gate Token

제 2장 Rest Ap[https://manual.dabory.com/books/frontend-tutorial/chapter/restfull-api)에서 Gate Token에 [

이해하는데 도움이 됩니다.

참고사항

`- [ KeyPair/PublicKey 등을 를 생성하는 일반적인`]PHP Code

https://www.php.net/manual/en/function.sodium-crypto-box-seal.php

`- [Sodium 사용 Ajax 제공 (260~262 Lin`es)]

http://git.daboryhost.com:10880/dabory/dbrerp/src/master/routes/web.php

`- [ DB 정보 암호화 how to use it in PHP (254~263 Lin`(es)]

http://git.daboryhost.com:10880/dabory/dbrerp/src/master/resources/views/front/dabory/pro/my-app/popup/popup-form1/form-a/client-app-form.blade.php).

dabory Gate Token(DGT)

- 로그인한 유저가 유저페이지에 접근합니다.
- 백`secret_key`[`key_pair`를 통해서 해당 업체의 db를 생성합니다. (erp_center_db, erp_memory_db, erp_individual_dbs)
- 프론트엔드서버에 생성된 db를 제공합니다.(driver, host, username, database, passwd, public_key) -> client config
- 유저가 메뉴나 링크를 클릭합니다.

[DGT 생성과정 시작]

- 백엔드 api 서버에 DGT를 요청합니다.
- api 서버에서 프론트엔드에서 보낸 사용자의 client config를 통해 랜덤한 20자의 문자열을 생성하고 Sodium을 통해 암호화합니다.
- 백엔드에서 DGT를 생성하여 저장하고 status 200, DGT를 반환합니다.
- 프론트엔드에서 반환받은 DGT를 세션에 저장합니다.
- 만약 DGT가 존재하지 않는다면 DGT를 받아오는 과정이 생략됩니다.

[Main Request]

- 이제 프론트엔드에서 보낸 모든 요청에 DGT가 포함되어 요청합니다.

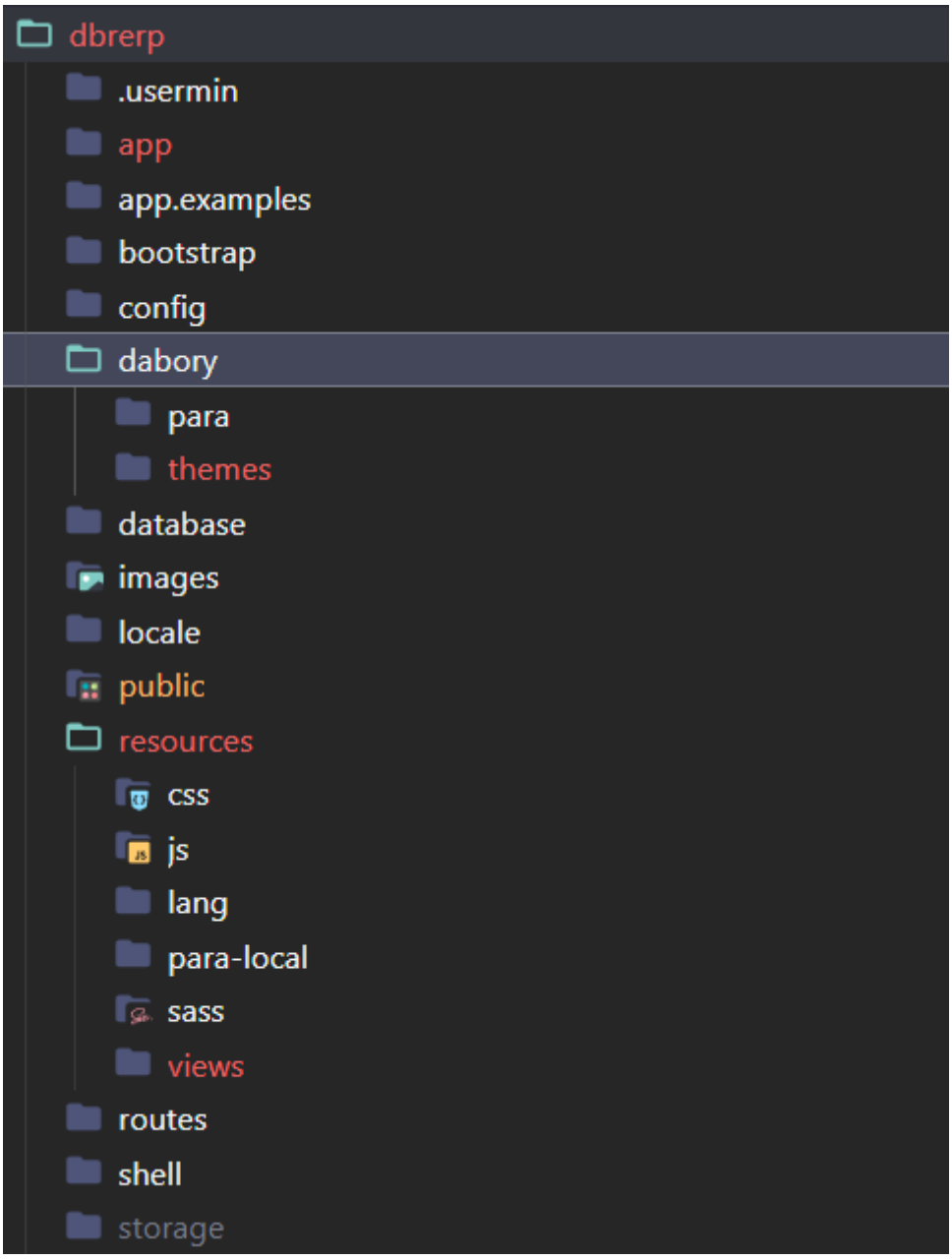
[DCS 생성과정 시작]

- 백엔드에서는 DGT를 통해 DCS를 생성하여 반환합니다.
- 만약 memory_db에 저장된 DGT가 존재하지 않는다면 DCS를 받아오는 과정이 생략됩니다.
- 검증된 DCS를 통해 사용자 DB(erp databases)의 모델에 접근합니다.
- 사용자 db로부터 요청한 data를 얻고 반환합니다.

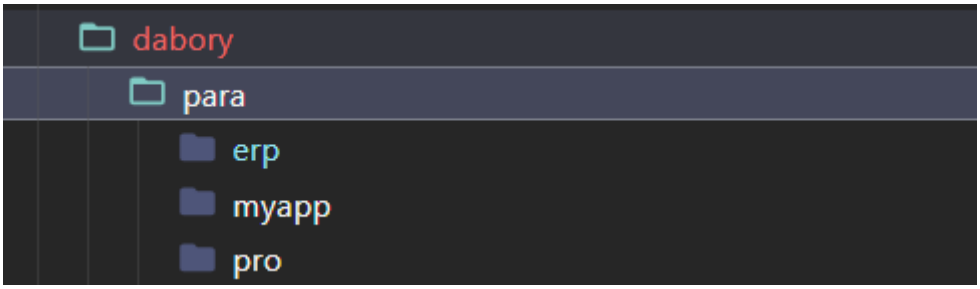
[□□ □]

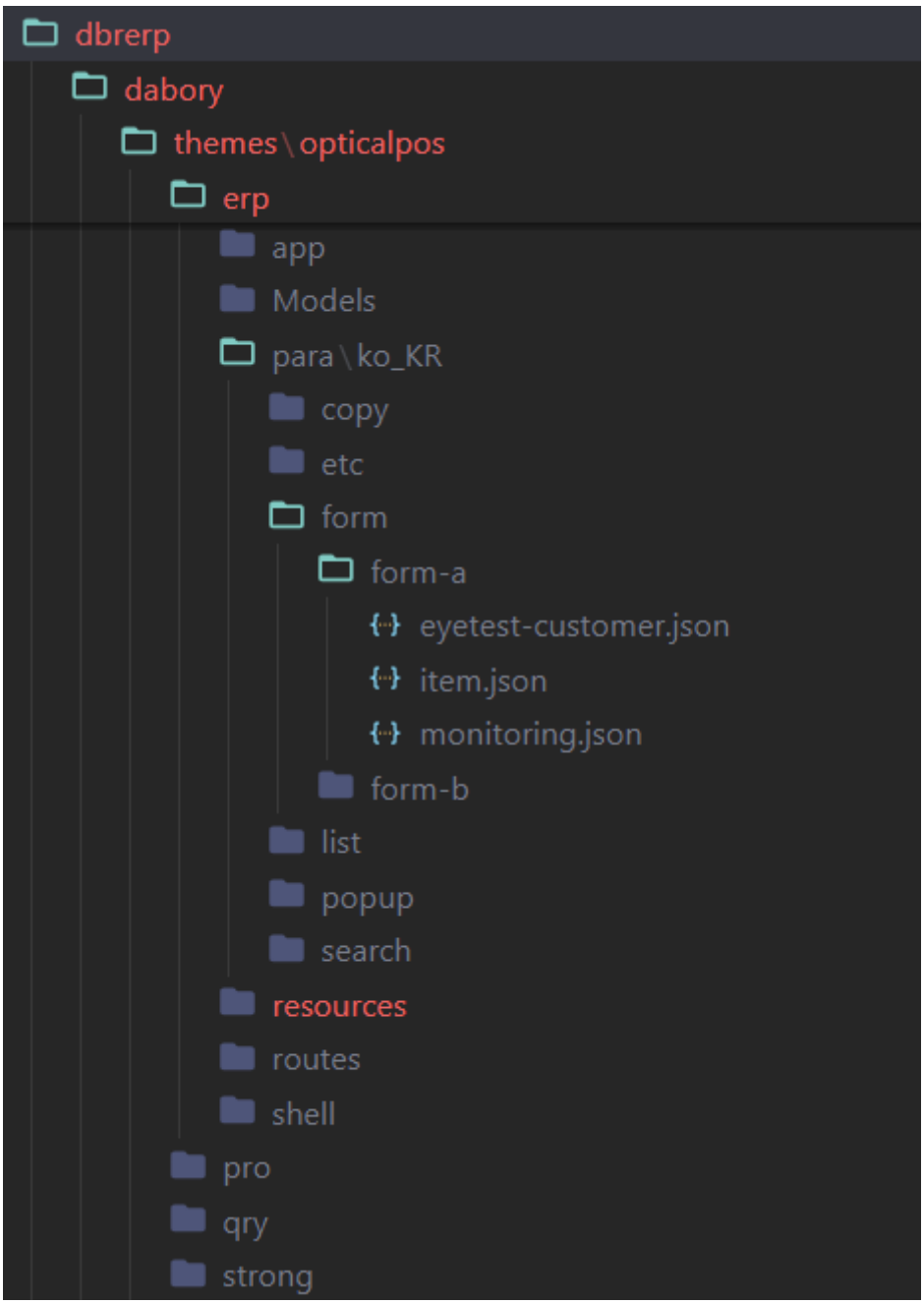
□□□ Frontend □□ □□

> “
>
> **Frontend □□(□□)**



□□ □□□□ □□





`dabory` 디렉토리는 다보리의 중요한 구성 요소로, 관리자 및 ERP 사용자 메뉴를 커스터마이징 할 수 있도록 설정하는 디렉토리입니다.

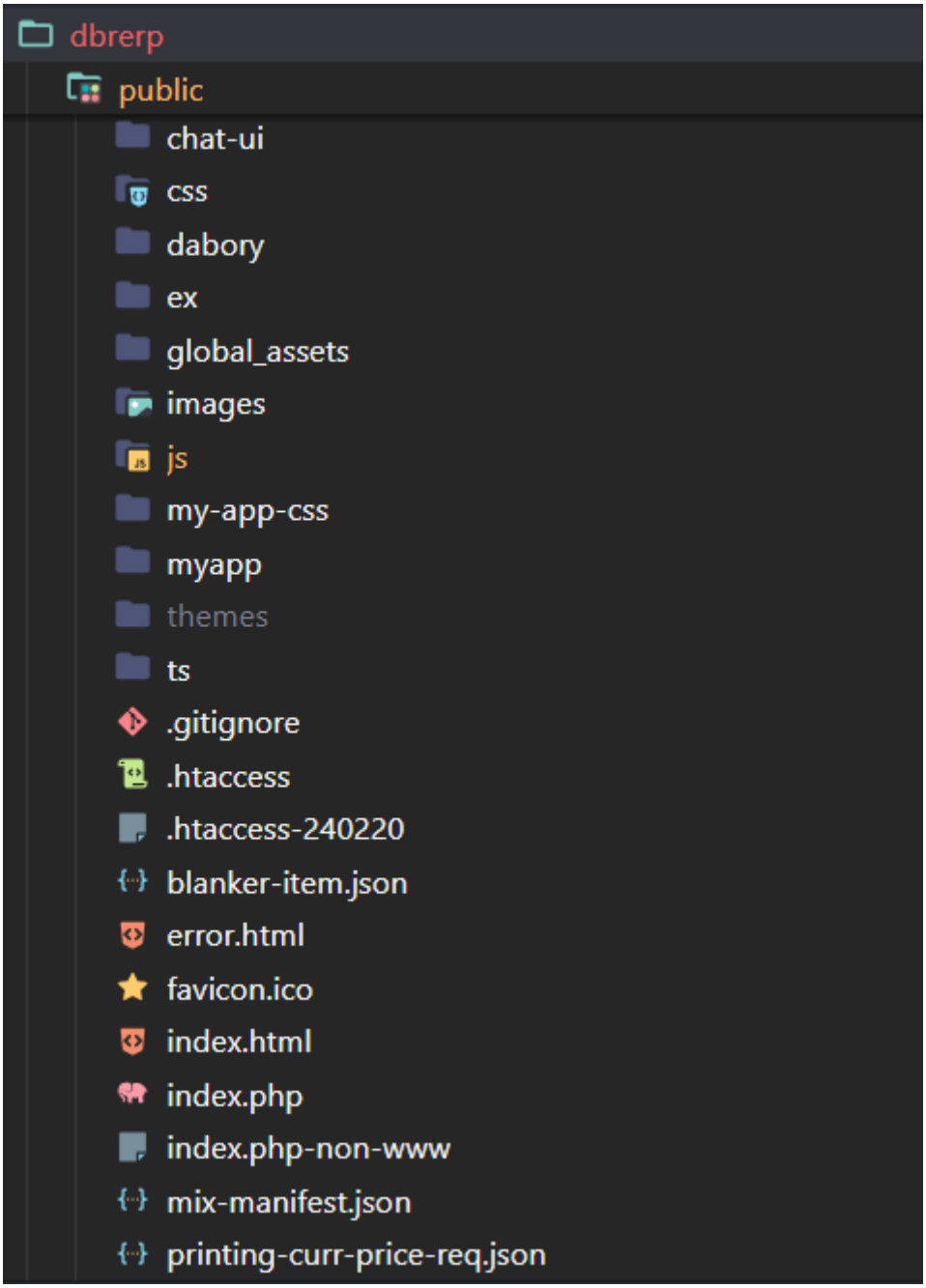
`para` : 이 디렉토ERP 관리자 메뉴를 설정할 JSON 형식으로 저장됩니다. 이러한 파일들을 통해 ERP 시스템의 메뉴 구성, 권한 관리, 기타 관리 설정 등을 유동적으로 수정이 가능합니다.


```
dabory
  para
    erp
    myapp
    pro
```

`themes` : 다보리의 프론트엔드 테마를 커스터마이징 할 수 있는 디렉토리입니다. 각 사이트별로 제공되는 테마를 개별적으로 수정하거나 새로 구축할 수 있습니다. 각 테마의 템플릿, 스타일, J


```
dbrerp
  dabory
    themes\opticalpos
      erp
        app
        Models
        para\ko_KR
          copy
          etc
          form
            form-a
              eyetest-customer.json
              item.json
              monitoring.json
            form-b
          list
          popup
          search
        resources
        routes
        shell
      pro
      qry
      strong
```

`public` : public 폴더 안에 웹에서 접근해야 하는 파일들을 넣어야 하는데 dabory 폴더 안에 있는 public 폴더를 dbrerp 폴더 안에 있는 public 폴더에 심볼릭 링크(Symbolic link)를 걸어 놓았습니다.



*심볼릭 링크 (Symbolic Link)
`dabory` 폴더 안에 있는 `public` 폴더를 심볼릭 링크(Symbolic link)하여, `dabory` 폴더를 `public` 폴더 안에 `dabory` 폴더로 연결하였습니다. 이렇게 하면 웹에서 접근 가능한 폴더 안에 있게 됩니다.

폴더구조

폴더구조 중 `dabory` 폴더안에 있는 `public` 폴더만 웹에서 접근 가능하도록 하였습니다. 이렇게 하는 이유는, 웹에서 접근 가능한 파일을 최소화 하기 위해서 입니다.

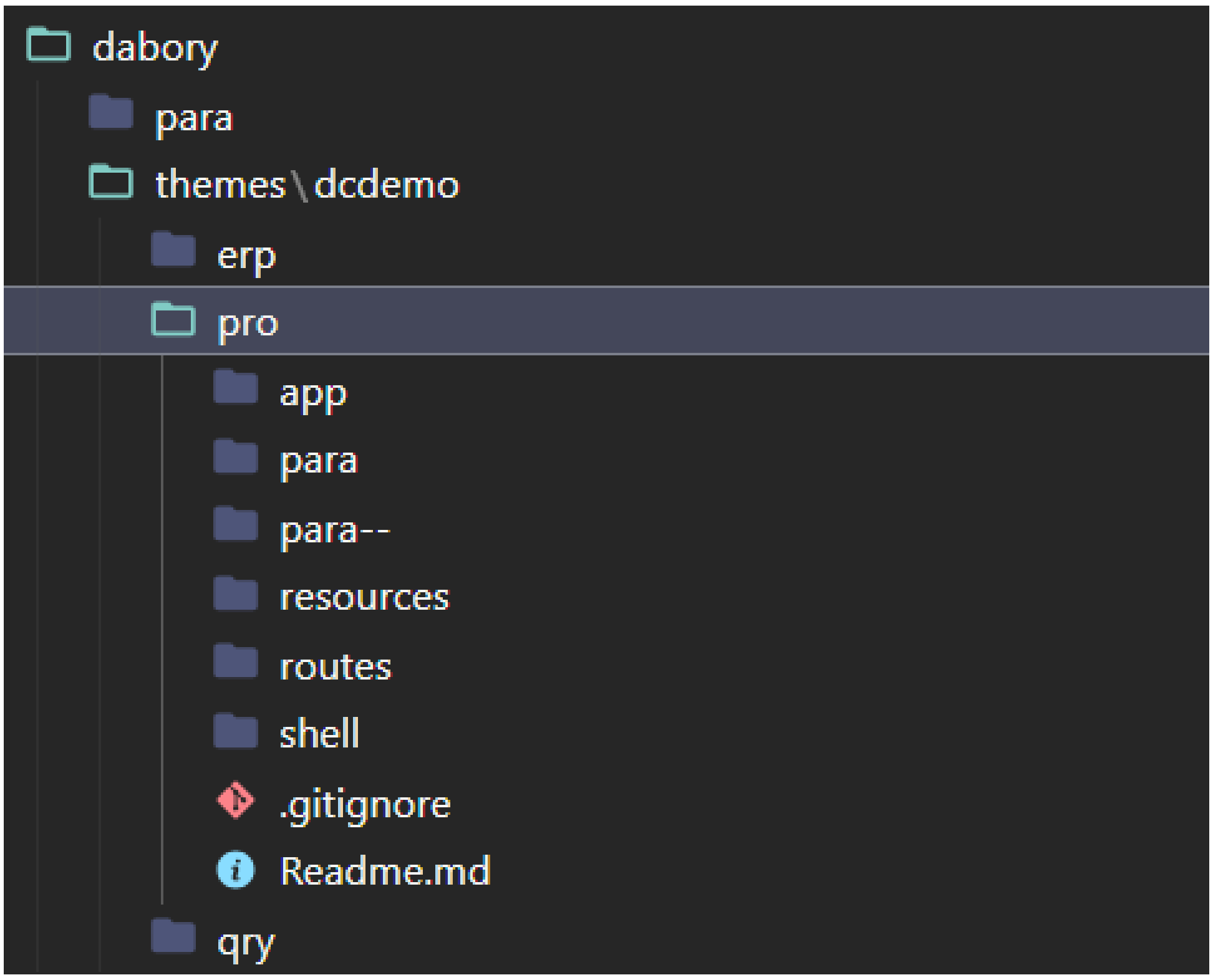
[illegible] Theme [illegible] [illegible]

> “
>
> # Theme [illegible] [illegible]

pro **(Service Page)**

pro [illegible] `app` , [illegible] `resources` [illegible] `routes` [illegible]. [illegible] ERP [illegible], [illegible] erp [illegible]. [illegible]

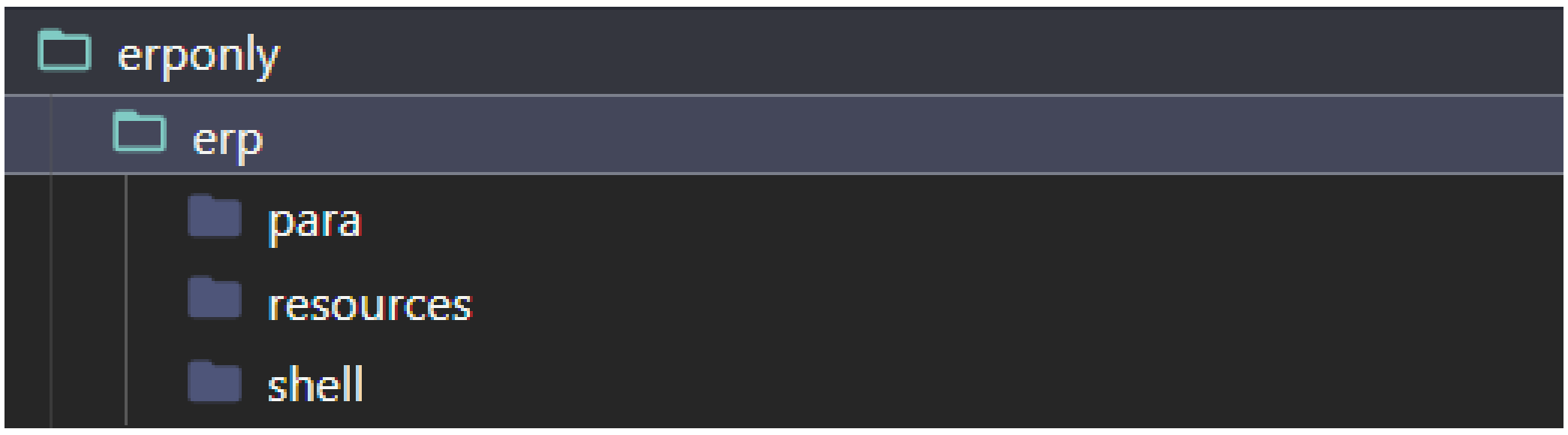
- pro [illegible]



app

[illegible] Controller[illegible], [illegible]. Controller[illegible] API [illegible] view [illegible]

para



pro 디렉토리에서 사용되는 view 페이지의 구성요소, api에 대한 명시 등등 pro 페이지를 구성하는 전반적인 요소들이 파라메터로 구성되어 있습니다.

resources

pro 디렉토리에서 작동하는 정적인 요소들이 포함된 디렉토리입니다. 보통 Controller와 동일한 네이밍 규칙을 따르며, Controller에서 받아온 데이터를 활용해 view 페이지가 구

erp (Back Office)

ERP 디렉토리는 기업이나 조직 내 직원 및 관리자를 위한 백오 `erponly` [illegible] redirect하여 표

- erp 디렉토리 예시


```
erponly
  erp
    para
    resources
    shell
```