

[!WARNING]

ルートディレクトリであるgohango配下で作業することを前提としています

[!NOTE]

説明内の@/はルートディレクトリを意味します。

0. フォルダ構成

```
gohango/
├── app/
│   ├── __init__.py    # Flaskアプリケーションの初期化とBlueprint登録
│   ├── db.py          # データベース接続関連の処理はここ
│   ├── services/      # APIはここに設置
│   │   ├── __init__.py # 空ファイル
│   │   └── users.py    # テーブルごとにエンドポイントを作成
│   ├── sample/        # サンプル機能のモジュール
│   │   ├── __init__.py # sampleブループリントの定義
│   │   └── routes.py   # sample機能のルーティング設定
│   ├── templates/     # HTMLはここ
│   │   ├── components/ # 使いまわす要素はここで管理
│   │   │   └── header.html
│   │   ├── layout.html # 全体レイアウトはこのファイルで定義
│   │   └── sample.html # 各ページを作成していく
│   └── static/         # 静的ファイルはここ
│       ├── css/        # HTMLファイル名.cssとかかな
│       │   └── index.css
│       └── js/         # HTMLファイル名.jsとかかな
│           └── sample.js
├── docker/            # Docker関連
│   ├── compose.yml    # Docker Composeの設定ファイル
│   └── migration/      # DBのマイグレーション
│       └── changelogs/ # マイグレーション。編集不可。必要に応じて追加
│           └── 01_create_tables.sql
├── .env               # 環境変数
├── run.py             # Flaskアプリを起動するスクリプト
├── readme.md          # このファイル
├── pyproject.toml     # パッケージリスト
└── .gitignore         # .envなどのコミットしたくないフォルダ・ファイルを設定
```

1. 参加する前に

1. [UVのサイト](#)にアクセスしてuvをインストール。(uv -Vでパスが通ってることを確認)
2. .envファイルを作成(以下はサンプル)

```
DB_HOST=localhost  
DB_PORT=65435  
DB_NAME=gohango  
DB_USER=root  
DB_PASS=password
```

3. pythonバージョンを確認

```
uv python list
```

4. python3.10.12が無かったら...

```
uv python install 3.10.12
```

5. ポスグレサーバを作成

```
cd docker  
docker compose up -d  
cd ../
```

2. 開発を始める前に

1. パッケージを同期する

```
uv sync
```

2. Dockerを立ち上げてポスグレサーバを起動&DB構造を同期

```
cd docker  
docker compose up -d  
docker compose run liquibase  
cd ../
```

3. 実行方法

```
uv run run.py
```

4. パッケージ追加方法

```
uv add パッケージ名
```

5. 新しいページを作成する

1. `@/app/__init__.py` にBlueprint登録（ここにファイルがあるから読み込んでね！！ってやつ）を行う。

```
from .フォルダ名.ファイル名 import B P 名
app.register_blueprint(B P 名, url_prefix='/ルーティング')
```

2. `@/app/フォルダ名` フォルダを作成。他の(sampleフォルダ)とかと同じ構成にする。(`__init__.py`, `routes.py`とかもイイ感じに作成)
3. 作成した`routes.py`に下記を追加

```
@B P 名.route("/") # ルーティング/の後にどのようにルーティングするか
def 分かりやすい関数名():
    return render_template("templateフォルダ内のhtmlファイル", 渡す変数)
```

4. `@/app/template` フォルダ内にhtmlファイルを作成。 `include`, `macro` を多用しよう。

6. 参考文献

[Webアプリ初心者のFlaskチュートリアル | Qiita](#)

参照不要archive

venv 作成方法

```
uv venv --python 3.10 venv
```

venv 起動方法

```
source ./venv/Scripts/activate
```

[!WARNING] Linuxはまた別です。 `activate` の前には半角スペースです

以下でシャットダウン

```
deactivate
```

パッケージ管理

なんかpip install でインストールしたときは下記コマンドでチーム共有 入れる前にチームに連絡

```
pip freeze > requirements.txt
```

開発始める前に下記コマンドで同期

```
pip install -r requirements.txt
```