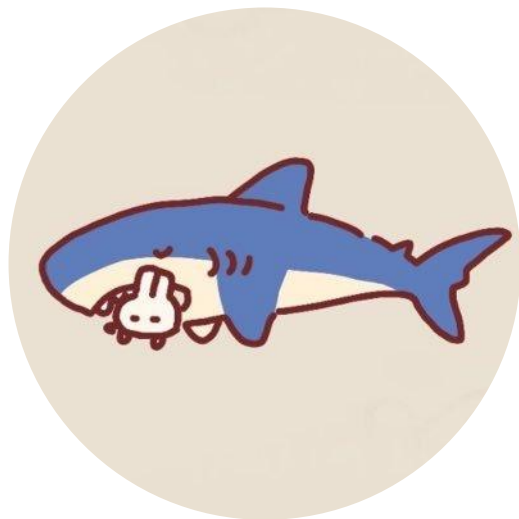


# tblsの紹介

# 自己紹介



sho44

新卒2年目エンジニア

媒体サイドの広告管理画面作ってます

普段はRails, AWS, postgresql

学生時代インターンでPHP (CodeIgniter)



(副業先探してます！)



xにとびます

# tblsの紹介



tbls

tbls ( is pronounced /'teɪb|z/. ) is a  
CI-Friendly tool for document a  
database, written in Go.

<https://github.com/k1LoW/tbls>より引用

Go製なんかい！

DBドキュメントツールなんかい！

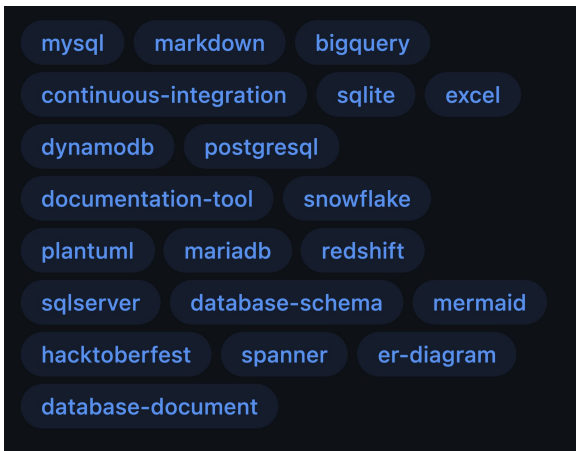
PHPの話じゃなくて  
ごめんなさい！！

寛大な気持ちで聞いてください🧚

# tblsのすごいところ

- 環境構築がとても楽
- 一瞬でER図付きのDBドキュメントを作ってくれる
- CIでDBのdiffを出してくれる
- ネットワークに接続しなくても良い(はず)
- 対応しているDBが多い

○



# デモ動画の前提条件

- 使用DB: postgresql
- docker composeで環境作っています
  - dbコンテナ、tblsコンテナのみ

<https://github.com/shosho44/tbls-introduction>にデモ環境を載せています

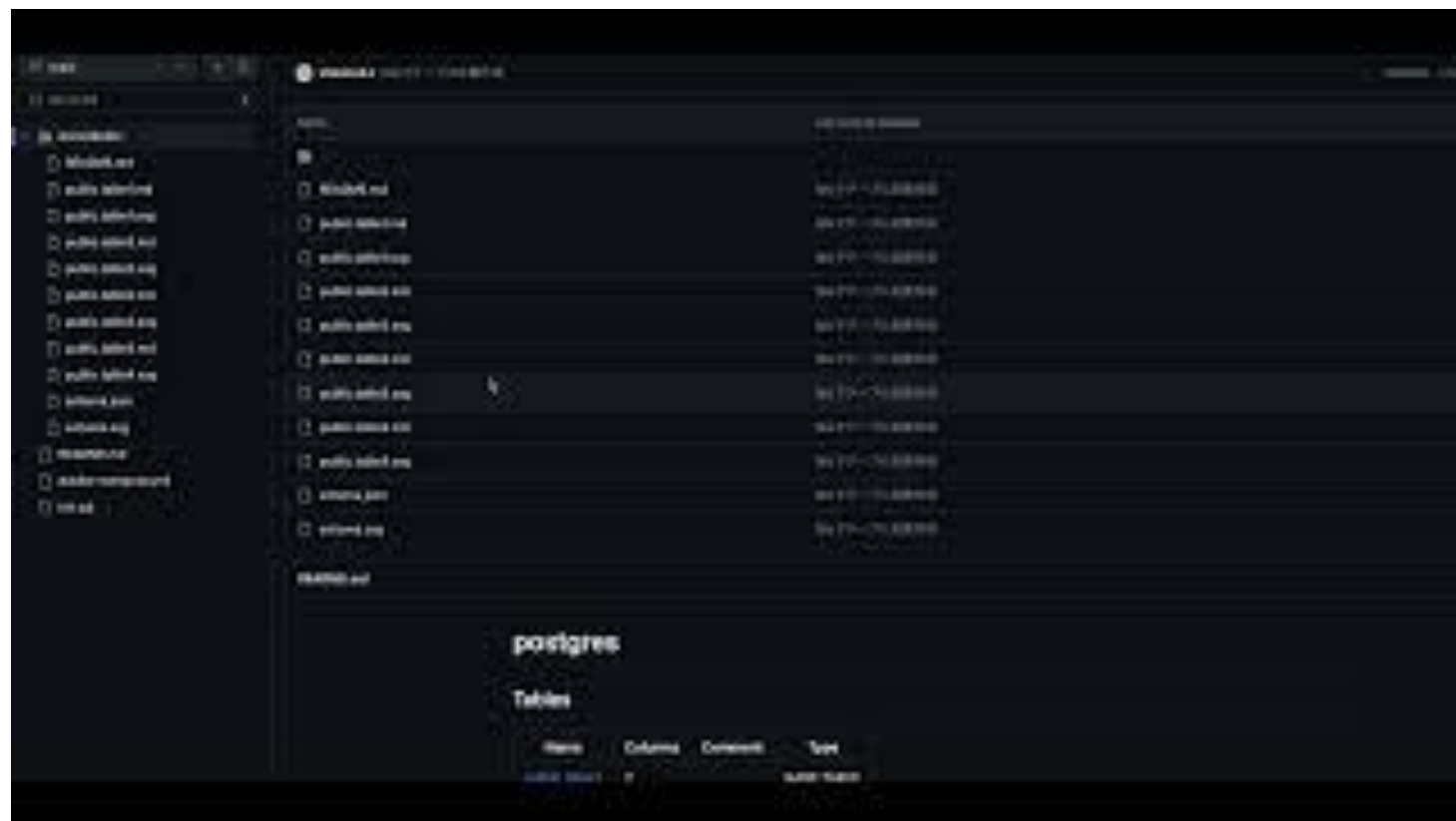




# デモ動画



# デモ動画



```
version: "3.9"

services:
  db:
    image: postgres:14.11
    volumes:
      - data:/var/lib/postgresql/data
    environment:
      POSTGRES_DB: db
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
    healthcheck:
      test: "psql -U postgres -l"
      interval: 1s
      timeout: 1s
      retries: 5
    ports:
      - "5432:5432"
    networks:
      - net_internal
      - net_external
```

```
tbls:
  image: ghcr.io/kllow/tbls:latest
  working_dir: /mnt
  tty: true
  depends_on:
    db:
      condition: service_healthy
  volumes:
    - ./docs:/mnt
  entrypoint: tbls doc --rm-dist
postgres://postgres:postgres@db:5432/postgres?sslmode=disable
  networks:
    - net_internal

volumes:
  data:

networks:
  net_internal:
    name: net_internal
    internal: true
  net_external:
    name: net_external
```

## 他にも色々できます その1

```
$ tbls lint
```

```
users.username: column comment required.
```

```
users.password: column comment required.
```

```
comments.post_id: column comment required.
```

```
posts: too many columns. [12/10]
```

```
comments: too many columns. [11/10]
```

```
5 detected
```

DBに対してlintを適用

## 他にも色々できます その2

\$ tbls coverage

Table	Coverage
All tables	16.1%
public.users	20%
public.comments	14.3%
public.post_comments	87.5%
public.post_comment_stars	0%

DBのcommentなどの  
カバレッジ計測

## 他にも色々できます その3

- yamlファイルに定義するだけで、、、
  - 外部キーがはられていない tableのリレーションを定義できる
  - カラム・DBにコメントを定義できる
  - ALTER, Migrationする必要がない！

# 利用シーン

- 新人が入ってきたけどDBのカラムの説明や関連付けについて説明するのが大変
- SQLを叩くBizメンバーからER図を欲しいと言われた
- 使っていないテーブルやカラムのお掃除をしたいけどDBの全体像を把握できない



おわり



tbis