

heroku での peewee を利用した postgresql の操作

参考1: Heroku Postgres (<https://devcenter.heroku.com/articles/heroku-postgresql#using-the-cli>)

参考2: Deploying a Flask App with Peewee to Heroku (<https://swifthorseman.com/2015/06/18/deploying-a-flask-app-with-peewee-to-heroku/>)

heroku app 作成 - postgresql アドオンの追加

heroku app の作成

```
$ heroku apps:create

Creating app... done, ● <app-name>
https://<app-name>.herokuapp.com/ | https://git.heroku.com/<app-name>.git
```

作成した heroku app に postgresql のアドオンを入れる

```
$ heroku addons:create heroku-postgresql:hobby-dev --app <app-name>

Creating heroku-postgresql:hobby-dev on ● secret-shelf-40737... free
Database has been created and is available
! This database is empty. If upgrading, you can transfer
! data from another database with pg:copy
Created postgresql-fitted-83361 as DATABASE_URL
Use heroku addons:docs heroku-postgresql to view documentation
```

アドオンが正常に導入されたことを確認する

```
$ heroku pg:info --app <app-name>
=== DATABASE_URL
Plan:          Hobby-dev
Status:        Available
Connections:   0/20
PG Version:    9.6.2
Created:       2017-06-30 08:38 UTC
Data Size:     7.2 MB
Tables:        0
```

```
Rows:          0/10000 (In compliance)
Fork/Follow:   Unsupported
Rollback:      Unsupported
Add-on:        postgresql-fitted-83361
```

heroku に導入された postgresql ヘログインする

```
$ heroku pg:psql --app <app-name>

--> Connecting to postgresql-fitted-83361
psql (9.6.3, server 9.6.2)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

secret-shelf-40737::DATABASE=> \l

                                List of databases
   Name      |  Owner   | Encoding | Collate |  Ctype  |
   |-----+-----+-----+-----+-----+
 d1036dn4sic5or | vhlyzitqexqpfh | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 8 | vhlyzitqexqpfh=CTc/vhlyzitqexqpfh
```

heroku 内において DB へのアクセスがどのような path で実現されるかを確認する

```
$ heroku run bash --app <app-name>
Running bash on secret-shelf-40737... up, run.2483 (Free)
~ $ echo $DATABASE_URL
postgres://lewcrjqpdrftsb:d35d728f021d656f735418fa07831f06b4c9e8153ba4f19
107-21-109-15.compute-1.amazonaws.com:5432/d3dp254oamild8
~ $ exit
exit
```

peewee を使った heroku 上の postgresql の操作

python 作業環境の作成

```
# 作業ディレクトリ作成
$ mkdir heroku_postgres
$ cd heroku_postgres
# python 環境の構築
$ pyenv virtualenv 3.6.0 heroku_postgres
$ pyenv local 3.6.0/envs/heroku_postgres
```

必要なパッケージのインストールと heroku 導入用のファイル作成

```
$ pip install peewee psycopg2 flask
$ pip freeze > requirements.txt
$ echo python-3.6.0 > runtime.txt
```

DB操作用の python プログラムを作成

```
models.py

from peewee import *
from urllib.parse import urlparse
import psycopg2, os

# for heroku db connection
url = urlparse(os.environ["DATABASE_URL"])
db = PostgresqlDatabase(
    database=url.path[1:],
    user=url.username,
    password=url.password,
    host=url.hostname,
    port=url.port
)

# models
## base model
class PostgresqlModel(Model):
    class Meta:
        database = db


## user defined model
class Person(PostgresqlModel):
    name = CharField()
    birthday = DateField()
```

git を通して, heroku ヘソースコードを push する

```
$ git init
$ git remote add heroku https://git.heroku.com/<app-name>.git

$ git add .
$ git commit -m "initial commit"
$ git push heroku master
```

heroku 上で コマンドを実行し, DB に Person table を作成.
また作成した table にデータを追加.

```
$ heroku run bash --app <app-name>
Running bash on  <app-name>... up, run.1446 (Free)
~ $ python
# Person table の作成
```

```
>>> from models import Person, db
>>> db.connect()
>>> Person.create_table()
# table へのデータの追加
>>> from datetime import date
>>> Bob = Person(name='Bob', birthday=date(1960, 1, 15))
>>> Bob.save()
1
```

データが追加されたことを確認する

```
$ heroku pg:psql --app <app-name>
--> Connecting to postgresql-fitted-83361
psql (9.6.3, server 9.6.2)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-
SHA384, bits: 256, compression: off)
Type "help" for help.

secret-shelf-40737::DATABASE=> \dt
      List of relations
 Schema | Name   | Type  | Owner
-----+-----+-----+-----
 public | person | table | lewcrjqpdrrftsb
(1 row)

secret-shelf-40737::DATABASE=> select * from person;
 id | name | birthday
----+-----+-----
  1 | Bob  | 1960-01-15
(1 row)
```