

Database structure:

user:

- id = Integer, primary_key=True
- public_id = String(max size = 36), unique=True, default=str(random)
- username = String(max size = 128), unique=True
- password_hash = String(max size = 256)
- is_login = Boolean, default=True

activity:

- id = Integer, primary_key=True
- description = String(max size = 400)
- spot = String(max size = 100)
- policy_type = Integer -> 0 = no, 1 = modertet, 2 = strict
- penalty_price = Integer
- penalty_time = Integer
- days = JSON -> {"sunday": bool, "monday": bool, ...} or {"1": bool, "2": bool, ...}
- time_step = JSON -> {1: {"0": hour, "1": hour}, 2...}
- weekly = Boolean
- user_id = Integer, ForeignKey('user.id'), nullable=False)
- user = relationship('User', foreign_keys=user_id)

Note you can use the JSON how you want.

Routes:

Note data is json object

Without token:

1. <http://uripeled23.pythonanywhere.com/user> :
 methodes: POST, PUT
 - POST:
 create new user
 data = {'username': " ", 'password': " "}
 - PUT:
 login
 data = {'username': " ", 'password': " "}

With token:

Note token must be included inside the headers, Authorization = Bearer token

2. <http://uripeled23.pythonanywhere.com/logout> :
 methods: GET, DELETE
 - GET:
 logout
 - DELETE:
 delete user

3. <http://uripeled23.pythonanywhere.com/activity>:

methods: GET, POST

- GET:
get all user's the activities
- POST:
create new activity
data = {"description": " ", "spot": " ", ...} (Note no need user_id or user)

4. http://uripeled23.pythonanywhere.com/activity/<activity_ids>

methods: GET, PUT, DELETE

- GET:
get activity details
- PUT:
update activity details
data = {"name of attribute you went to update": " ", ...} (Note no need user_id or user)
- DELETE:
delete activity