

# Проект "Клиника домашних животных"

11 мая 2023 г. 14:52

## ! Приложение еще в разработке поэтому:

1. Не используется gunicorn
2. Используется сразу SQLite
3. Не покрыто тестами pytest
4. Не скрыт config.py и выложен на github.
5. И другие нюансы

## Сценарий:

- На текущий момент клиника одна (Как на вырост можно рассмотреть расширение до сети клиник)

## Описание данных по умолчанию:

Дано:

**5 котов** = sima, marsik, barsik1, barsik2, barsik3

**3 собаки** = jena, rex1, rex2

**2 обычных пользователя** = ivanov, petrov

**1 администратор** = maksimov

**За ivanov числется** следующие животные: **jena, rex1, barsik1**

**За petrov числется** следующие животные: **rex2, barsik2**

**За maksimov числется** следующие животные: **barsik3**

## Учетные данные пользователей по умолчанию:

```
ivanov = UsersModel(email="ivanov@mail.com", password="password1",
    firstname="Ivan", lastname="Ivanov", phone=81231231212,
    sex="male", birthdate=date(2000, 3, 21))
```

```
petrov = UsersModel(email="petrov@mail.com", password="password2",
    firstname="Petr", lastname="Petrov", phone=81231231211,
    sex="male", birthdate=date(1998, 3, 21))
```

```
maksimov = UsersModel(email="maksimov@mail.com", password="password3",
    firstname="Maksim", lastname="Maksimov", phone=81231231213,
    sex="male", birthdate=date(1991, 3, 21), is_admin=True)
```

## 1. Пользователи без аутентификации могут пользоваться только Auth API.

Для регистрации, входа и выхода из системы.

При входе в систему добавляется access\_token, в cookies.

И дополнительно возвращается в ответе, для возможности его поместить в X-API-KEY.

Главным считается ключ в X-API-KEY, при их отличии.

## Пример попытки воспользоваться эндпоинтом без аутентификации

**Pets**

GET /pets Get all pets

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://192.168.1.60:5000/pets' \
  -H 'accept: application/json'
```

Request URL

http://192.168.1.60:5000/pets

Server response

Code	Details
401	Error: UNAUTHORIZED

Response body

```
{
  "message": "Требуется пройти аутентификацию. Токен не определен"
}
```

## 2. Пользователь регистрируется на нашей платформе по почте и паролю.

С этих пор это его учетные данные для логина в систему и получения токена.

### Пример создания нового пользователя.

**Auth**

POST /register Registering a new user

Parameters

No parameters

Request body

application/json

```
{
  "email": "test@test.com",
  "password": "test",
  "firstname": "test",
  "lastname": "test",
  "phone": "80000000000",
  "sex": "male",
  "birthdate": "2020-01-25"
}
```

Execute Clear

Responses

Curl

```
curl -X 'POST' \
  'http://192.168.1.60:5000/register' \
  -H 'accept: application/json' \
  -H 'content-type: application/json' \
  -d '{
    "email": "test@test.com",
    "password": "test",
    "firstname": "test",
    "lastname": "test",
    "phone": "80000000000",
    "sex": "male",
    "birthdate": "2020-01-25"
  }'
```

Request URL

http://192.168.1.60:5000/register

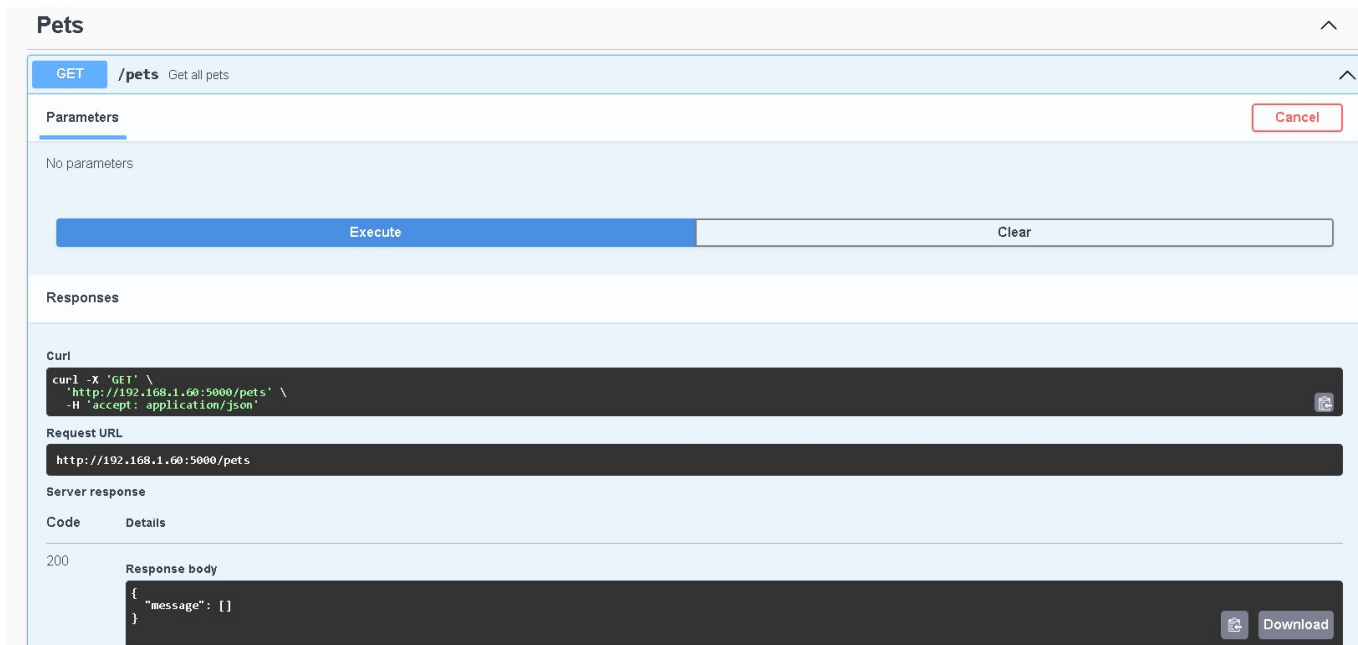
Server response

Code	Details
201	

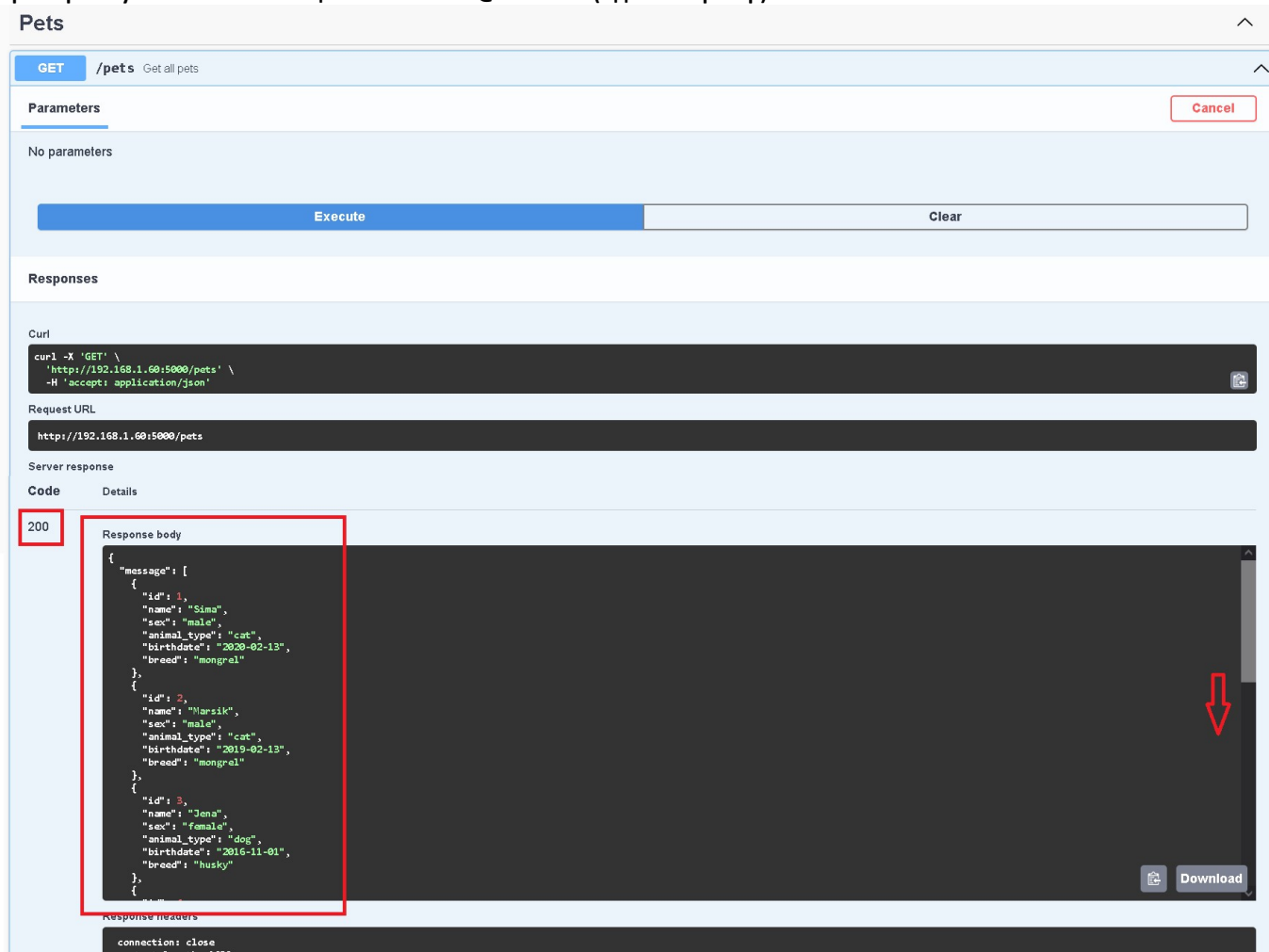
Response body

```
{
  "message": {
    "id": 4,
    "sex": "male",
    "lastname": "test",
    "birthdate": "2020-01-25",
    "firstname": "test",
    "phone": "80000000000",
    "email": "test@test.com",
    "password": "test"
  }
}
```





Пример получения всех питомцев от maksimov@mail.com (Администратор):



Далее будут перечислены возможности администратора.

4. Когда пользователь обладает правами администратора, то у него добавляется доступ:
  - а. Получить список всех пользователей.

- b. Разрешить добавление новых пользователей.
- c. Разрешить удалять пользователей.
- d. Разрешить редактирование параметров всех пользователя.  
В том числе закрепление и открепление питомцев.  
Назначать их администраторами.
- e. Разрешить добавление новых питомцев.
- f. Разрешить редактирование параметров всех питомцев.
- g. Разрешить удаление всех питомцев.

Рассмотрим пример закрепления питомцев за пользователем.

Добавим новому пользователю test@test.com питомца с id = 4.

У самого нового пользователя id сейчас 4.

PUT /users/bind Assigning a pet to a user

Parameters

No parameters

Request body

application/json

```
{
  "user_id": "4",
  "pet_id": "4"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'PUT' \
'http://192.168.1.60:5000/users/bind' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "user_id": "4",
  "pet_id": "4"
}'
```

Request URL

http://192.168.1.60:5000/users/bind

Server response

Code

Details

200

Response body

```
{
  "message": "Пользователю Test добавлен питомец 'Rex'"
}
```

Download

Проверим еще рас список животных у пользователя test@test.com

Новый раздел 2 Стр.5

# Pets

GET

/pets

Get all pets

Parameters

Cancel

No parameters

ExecuteClear

Responses

Curl

```
curl -X 'GET' \
  'http://192.168.1.60:5000/pets' \
  -H 'accept: application/json'
```

Request URL

```
http://192.168.1.60:5000/pets
```

Server response

Code	Details
200	<div>Response body<pre>{   "message": [     {       "id": 4,       "name": "Rex",       "sex": "male",       "animal_type": "dog",       "birthdate": "2016-05-01",       "breed": "shepherd"     }   ] }</pre></div> <div>Download</div>

Видим что связка прошла успешно.