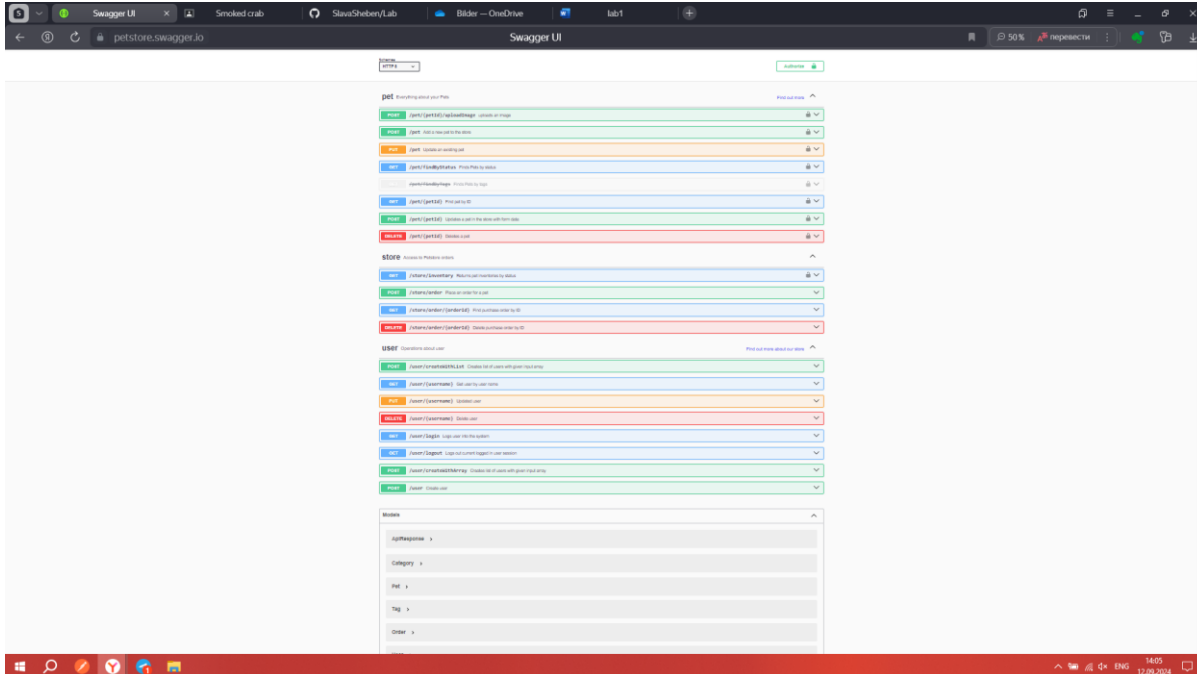


Иванчук Вячеслав Сергеевич ИС221

12.09.2024

## Лабораторная работа №1

в <https://petstore.swagger.io/> мы можем работать с животными, с заказами и с пользователями, а также у нас есть описание моделей разных сущностей, с которыми мы будем работать

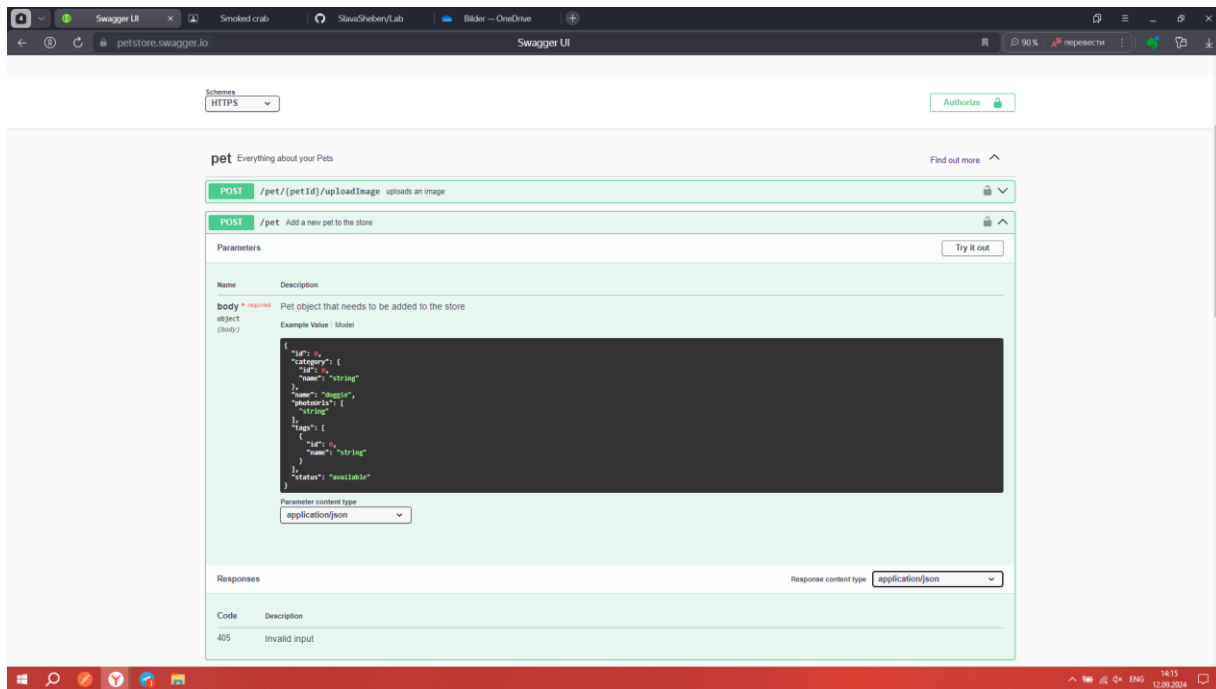


Давайте развернем модель животного (pet) и посмотрим, что там есть. У pet есть id с типом integer (число), также pet относится к какой-то категории (Category) У Category есть id с типом integer (число) и есть название категории (name) с типом string (строка). Также у животного есть имя (name) с типом String (строка)

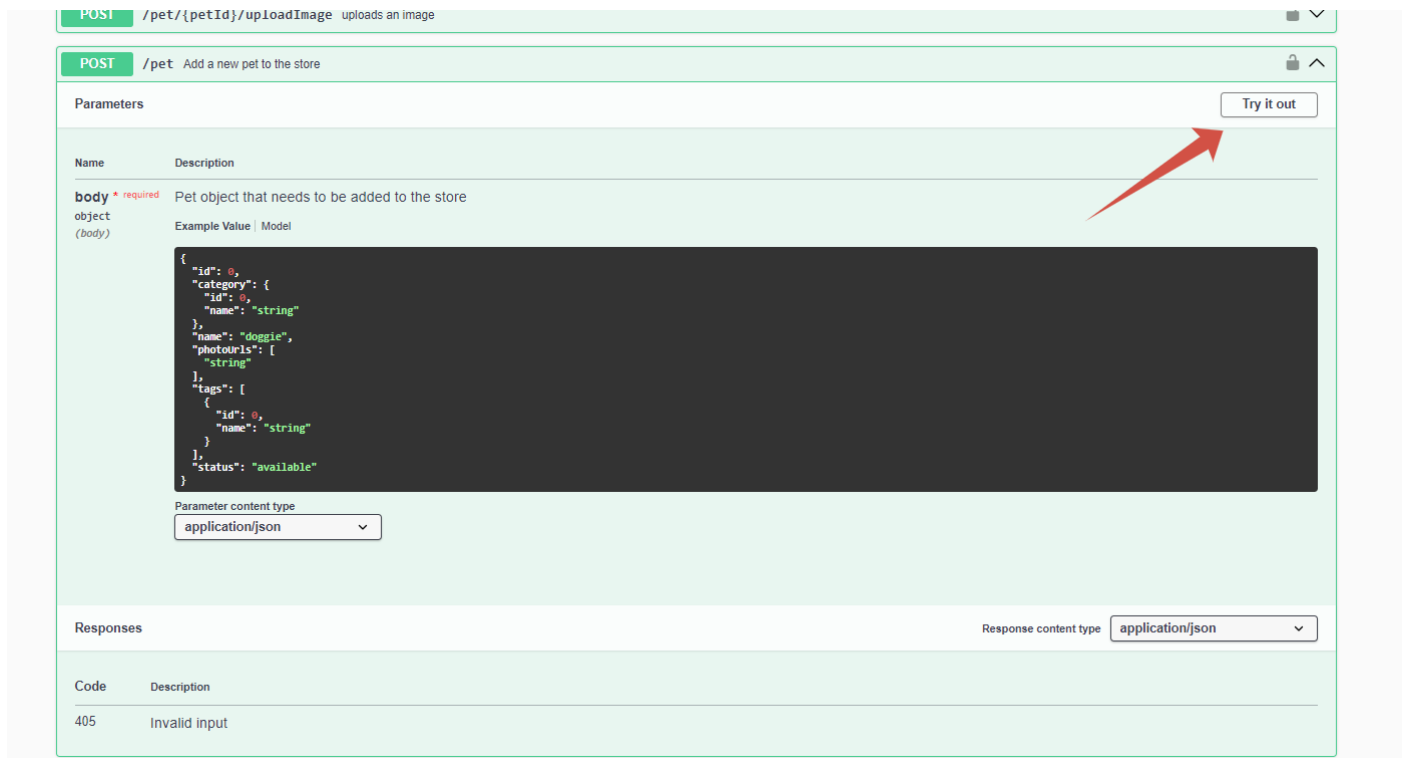
```
Pet {
  id
  category
  name*
  photoUrls*
  tags
  status
}
```

Category { ... }

Теперь давайте попробуем добавить новое животное в магазин через метод POST /pet где мы будем передавать данные в теле body при помощи json.



Нажимаем на Try it out



Теперь в json указываем те данные, которые мы хотим, чтобы были у нашего животного. Пусть id животного будет 2485342798

Id категории будет 230, название категории будет «cat». Имя животного будет «meme», остальные атрибуты оставляем без изменений и нажимаем на «Execute»



The screenshot shows a REST client interface. At the top, a JSON body is displayed with red arrows pointing to specific values: "id": 2485342798, "category": {"id": 230, "name": "cat"}, "name": "meme", and "photoUrls": [{"string"}]. Below the JSON body is a "Cancel" button. Underneath that is a "Parameter content type" dropdown menu set to "application/json". At the bottom is a large blue "Execute" button with a red arrow pointing to it.

```
{
  "id": 2485342798,
  "category": {
    "id": 230,
    "name": "cat"
  },
  "name": "meme",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Cancel

Parameter content type  
application/json

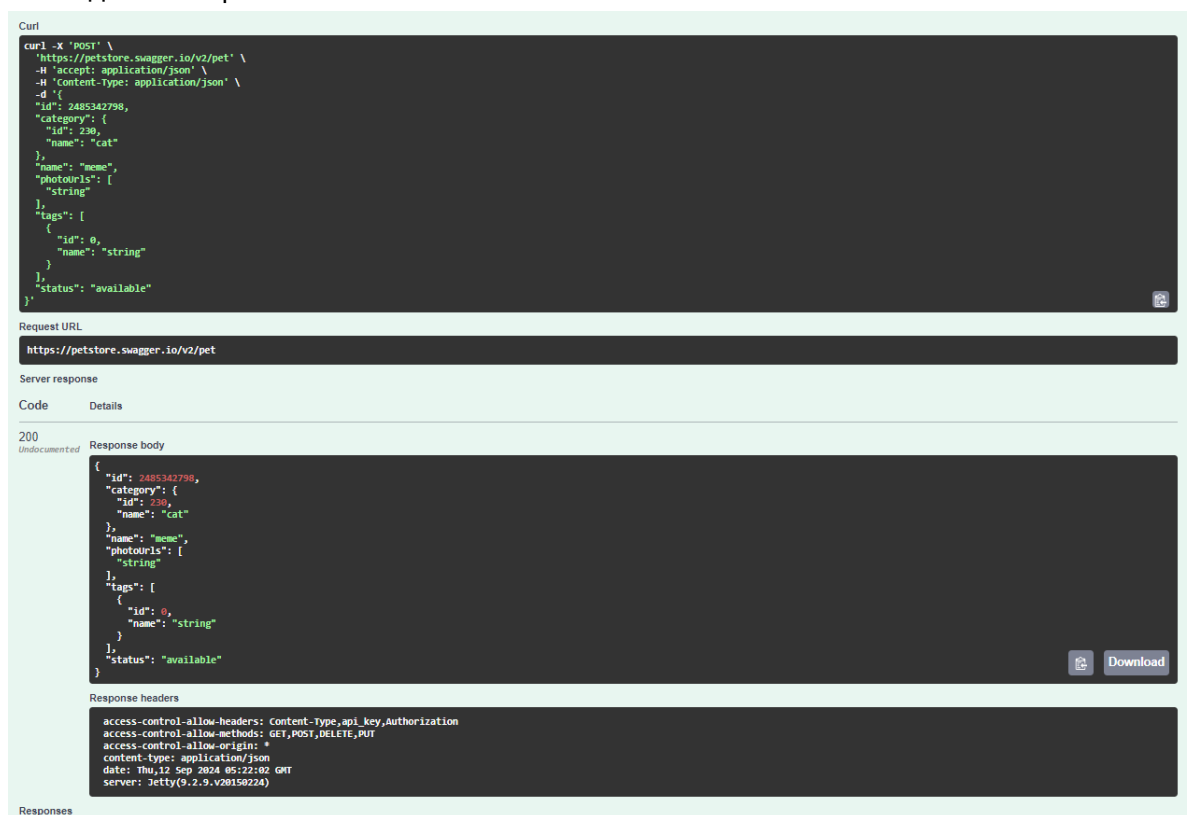
Execute

Давайте посмотрим, что нам вернулось. Нам вернулся запрос, отправленный при помощи curl (curl позволяет нам обращаться к серверу через командную строку). Нам вернулся Request URL на который мы обращались, чтобы отправить наш запрос. Нам вернулся код ответа 200 — это значит, что запрос выполнен успешно, вернулось тело ответа в виде json. И также вернулся список заголовков (headers) Server: Информация о сервере, обрабатывающем запрос.

Date: Дата и время создания запроса или ответа.

Content-Type:

Тип контента,



The screenshot shows the result of a curl command in a REST client. The "Curl" tab is active, displaying the command: `curl -X 'POST' \ 'https://petstore.swagger.io/v2/pet' \ -H 'accept: application/json' \ -H 'Content-Type: application/json' \ -d '{ "id": 2485342798, "category": { "id": 230, "name": "cat" }, "name": "meme", "photoUrls": [ "string" ], "tags": [ { "id": 0, "name": "string" } ], "status": "available" }'`. Below the command is the "Request URL" field with the value `https://petstore.swagger.io/v2/pet`. The "Server response" section shows a "Code" of 200 and a "Response body" containing the same JSON as in the previous screenshot. The "Response headers" section lists several headers: `access-control-allow-headers: Content-Type,api_key,Authorization`, `access-control-allow-methods: GET,POST,DELETE,PUT`, `access-control-allow-origin: *`, `content-type: application/json`, `date: Thu, 12 Sep 2024 02:22:02 GMT`, and `server: Jetty(9.2.9.v20150224)`.

Curl

```
curl -X 'POST' \
'https://petstore.swagger.io/v2/pet' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "id": 2485342798,
  "category": {
    "id": 230,
    "name": "cat"
  },
  "name": "meme",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}'
```

Request URL

https://petstore.swagger.io/v2/pet

Server response

Code Details

200

Response body

```
{
  "id": 2485342798,
  "category": {
    "id": 230,
    "name": "cat"
  },
  "name": "meme",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Response headers

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Thu, 12 Sep 2024 02:22:02 GMT
server: Jetty(9.2.9.v20150224)
```

Responses

передаваемого в теле запроса или ответа (например, текст, изображение, JSON)

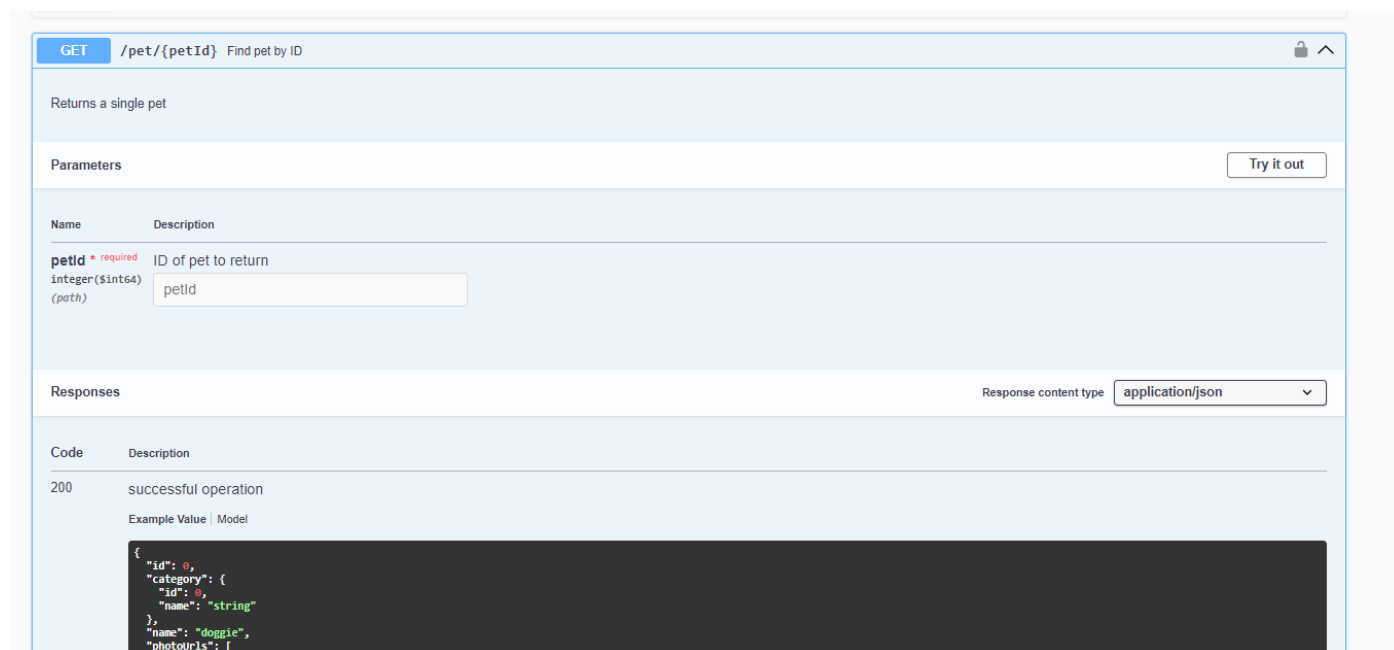
Давайте попробуем создать еще одно животное, но уже через curl. Скопируем тот curl, который нам вернулся, поменяем id животного на 2485342799, id и название категории оставим неизменными, а имя животного изменим на «Barsik» Вставляем наш запрос в командную строку (на mac это будет terminal)

```
curl -X 'POST' \
  'https://petstore.swagger.io/v2/pet' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 2485342799,
    "category": {
      "id": 230,
      "name": "cat"
    },
    "name": "Barsik",
    "photoUrls": [
```

и нажимаем на Enter и видим что нам вернулся json с новым животным

```
▼ object {6}
  id: 2485342799
  ▼ category {2}
    id: 230
    name: cat
    name: Barsik
  ▼ photoUrls [1]
    0: string
  ▼ tags [1]
    ▼ 0 {2}
```

Давайте в этом убедимся и найдем оба наших животных при помощи метода get /pet/{petID}



GET /pet/{petID} Find pet by ID

Returns a single pet

Parameters

Name	Description
<b>petid</b> * required integer(\$int64) (path)	ID of pet to return <input type="text" value="petid"/>

Responses

Response content type: application/json

Code	Description
200	successful operation Example Value   Model <pre>{   "id": 0,   "category": {     "id": 0,     "name": "string"   },   "name": "doggie",   "photoUrls": [</pre>

Для этого нажмем на «Try it out»

Введем в поле petID id нашего первого животного 2485342798, которое мы добавляли через swagger. Название ( path) означает, что это path параметр, он передается в строке. Подробнее обо всех параметрах, мы будем говорить в следующих уроках, посвященных postman. И нажимаем на execute.

Name

Description

petid \* required

ID of pet to return

integer(\$int64)

2485342798

(path)

Curl

```
curl -X 'GET' \
'https://petstore.swagger.io/v2/pet/2485342798' \
-H 'accept: application/json'
```

Request URL

https://petstore.swagger.io/v2/pet/2485342798

Server response

Code

Details

200

Response body

```
{
  "id": 2485342798,
  "category": {
    "id": 238,
    "name": "cat"
  },
  "name": "meme",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Download

Response headers

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Thu, 12 Sep 2024 05:48:22 GMT
server: Jetty(9.2.9.v20150224)
```

И как мы видим нам вернулся 200 код ответа ( это значит что запрос выполнен успешно) с телом ответа, в котором будут данные наши животного с id 2485342798

Давайте теперь аналогично сделаем для нашего животного которое мы создавали при помощи curl с id 2485342799

Curl

```
curl -X 'GET' \
'https://petstore.swagger.io/v2/pet/2485342799' \
-H 'accept: application/json'
```

Request URL

https://petstore.swagger.io/v2/pet/2485342799

Server response

Code

Details

200

Response body

```
{
  "id": 2485342799,
  "category": {
    "id": 238,
    "name": "cat"
  },
  "name": "Barsik",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Download

Response headers

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Thu, 12 Sep 2024 05:41:07 GMT
server: Jetty(9.2.9.v20150224)
```

Responses

И как мы видим и здесь нам вернулся 200 код ответа ( это значит что запрос выполнен успешно) с телом ответа, в котором будут данные нашего животного которое мы добавляли через curl

Вот так вот работает swagger.