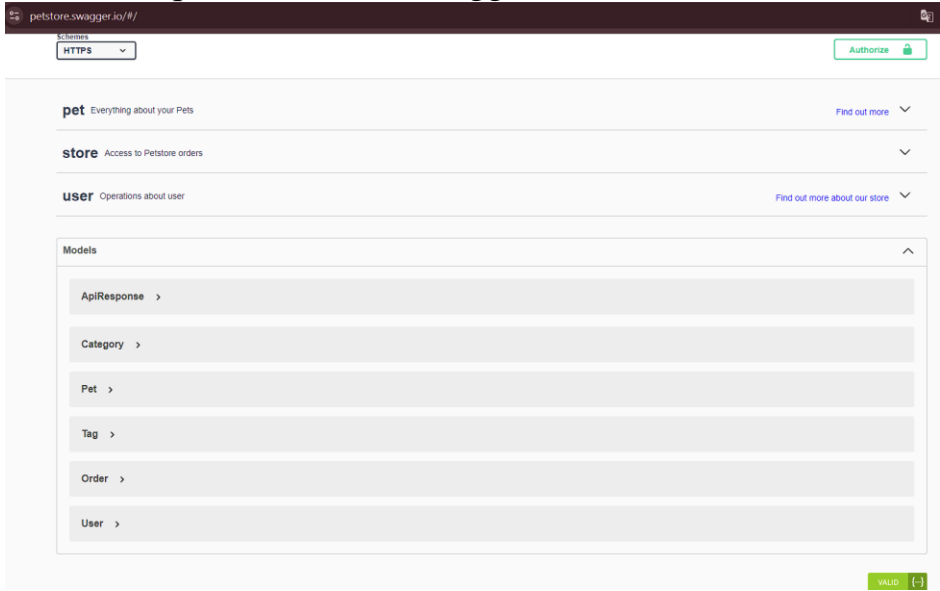


Тема лабораторной работы	lb_1
Выполняющий	Мирзоев Руслан
Помогающий	Александра Ерёмина
Ход работы	<div>1. Мы открыли сам сайт swagger.io</div> <div></div> <div>2. У сущности "pet" есть поле id типа integer (число), и она относится к категории (Category). У категории тоже есть id типа integer и название (name) типа string (строка). Также у животного есть имя (name) типа string.</div> <div><pre>Pet { id category name* photoUrls* tags status } Category { id name } integer(\$int64) integer(\$int64) string</pre></div>

3. Пробуем добавить новое животное в магазин через метод post:

POST /pet Add a new pet to the store

Parameters

Try it out

Name	Description
body * required	Pet object that needs to be added to the store
object (body)	Example Value Model

```
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "doggie",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Parameter content type
application/json

После нажатия на try it out мы изменили id, id категории и имя животного:

POST /pet Add a new pet to the store

Parameters

Cancel

Name	Description
body * required	Pet object that needs to be added to the store
object (body)	Edit Value Model

```
{
  "id": 556677,
  "category": {
    "id": 232323,
    "name": "dogs"
  },
  "name": "jackynson",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Cancel

Parameter content type
application/json

Execute Clear

При помощи curl (который позволяет отправлять запросы к серверу через командную строку) мы получили ответ. Нам вернулся Request URL, на который был отправлен запрос, код ответа 200, что означает успешное выполнение запроса, и тело ответа в формате JSON. Также был возвращён список заголовков headers: Server, Date, Content-Type

Curl

```
curl -X 'POST' \
  'https://petstore.swagger.io/v2/pet' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 556677,
    "category": {
      "id": 232323,
      "name": "dogs"
    },
    "name": "jackynson",
    "photoUrls": [
      "string"
    ],
    "tags": [
      {
        "id": 0,
        "name": "string"
      }
    ],
    "status": "available"
  }'
```

Request URL

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

Server response

Code	Details
200	Response body

Response body

```
{
  "id": 556677,
  "category": {
    "id": 232323,
    "name": "dogs"
  },
  "name": "jackynson",
  "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-origins: *
content-type: application/json
date: Fri, 13 Sep 2024 12:38:59 GMT
server: Jetty(9.2.8.v20150224)
```

Code	Description
405	Invalid input

4. Пробуем создать ещё одно животное, но через curl и cmd. Копируем curl который нам вернулся и меняем id с 556677 на 5566778 и имя животного меняем на «jackyn12». Мы использовали одну строку без экранирования, потому что с экранированием у нас была ошибка

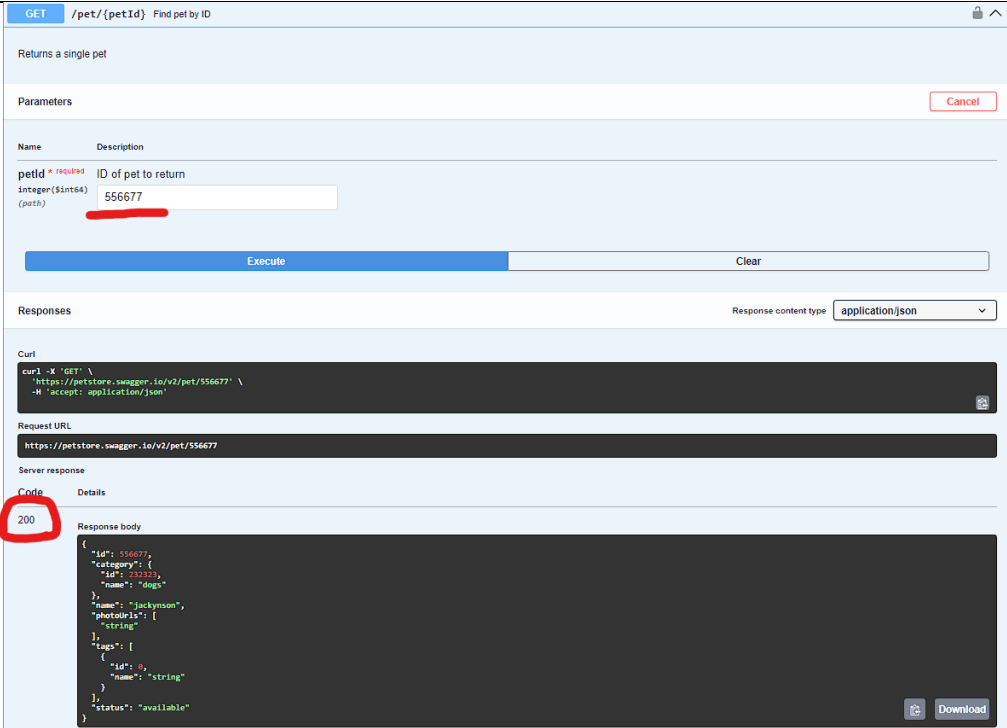
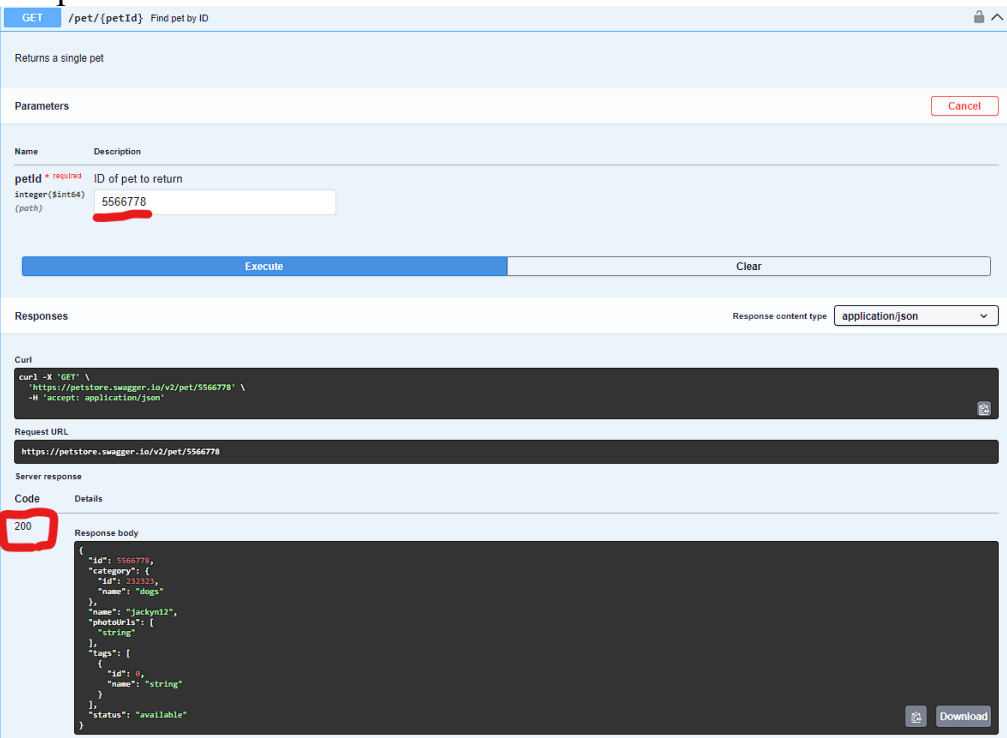
```
C:\Users\user>curl -X POST "https://petstore.swagger.io/v2/pet" -H "accept: application/json" -H "Content-Type: application/json" -d '{"id":5566778,"category":{"id":232323,"name":"dogs"},"name":"jackyn12","photoUrls":["string"],"tags":[{"id":0,"name":"string"}],"status":"available"}'
```

После нажатия на enter:

```
{
  "id": 5566778,
  "category": {
    "id": 232323,
    "name": "dogs"
  },
  "name": "jackyn12",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

нам вернулся json с новым животным

5. Теперь убеждаемся при помощи метода get /pet/{petID} Первая животина:

	 <p>Вторая:</p> 
Результат	<p>После проделанной нами лабораторной работы, по итогу мы изучили:</p> <ol style="list-style-type: none"> 1. Что такое Swagger UI и для чего он нужен 2. Из чего состоит сущность «pet» 3. Рассмотрели методы post и get и что они делают 4. Научились создавать сущности через curl с cmd
Оценка	При сдаче не забываем ставить пометку отправить в google

Githab-файлы pdf
Программный код-Readme

Критерий оценивания

Отлично	Четко, лаконично, по факту, гитхаб на ура
Хорошо	Небольшие корректировки
Удовлетворительно	С вопросами, и комментариями
Неудовлетворительно	Ничего непонятно