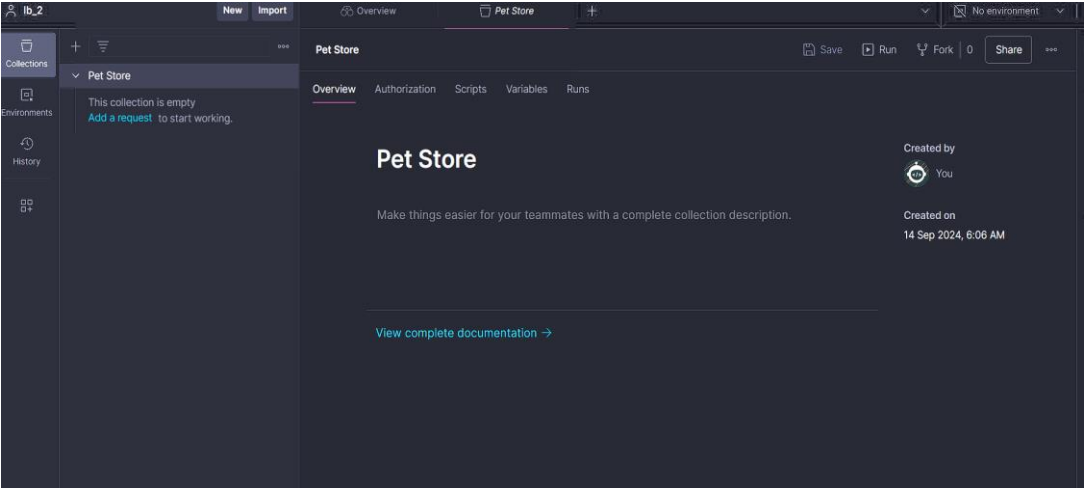
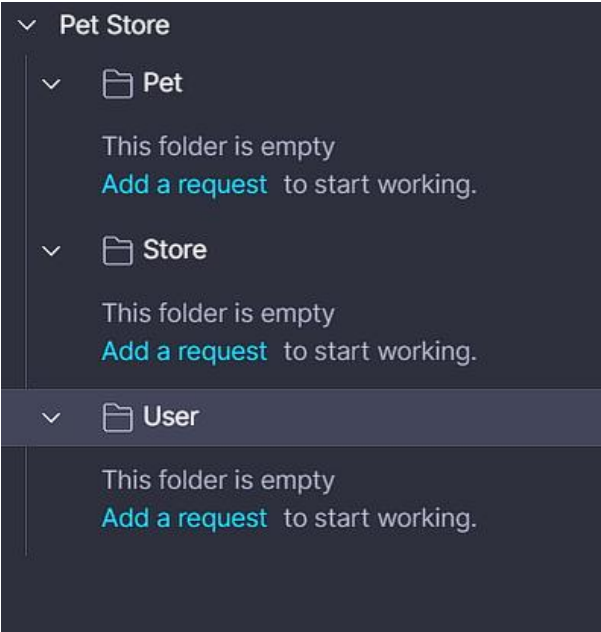
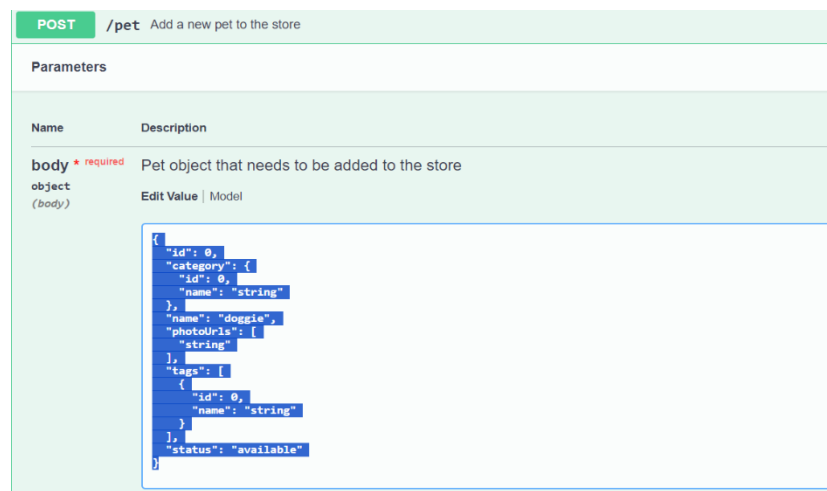


Тема лабораторной работы	lb_2
Выполняющий	Ржевский Глеб
Помогал	
Ход работы	<div>1)Создал Blank Workspace, добавил blank collection и назвал её Pet Store</div> <div></div> <div>2)Создал 3 папки Pet, Store и User</div> <div></div>

3) Создал http запрос, переименовал его, вставил код из сайта в body->raw->JSON и изменил некоторые данные

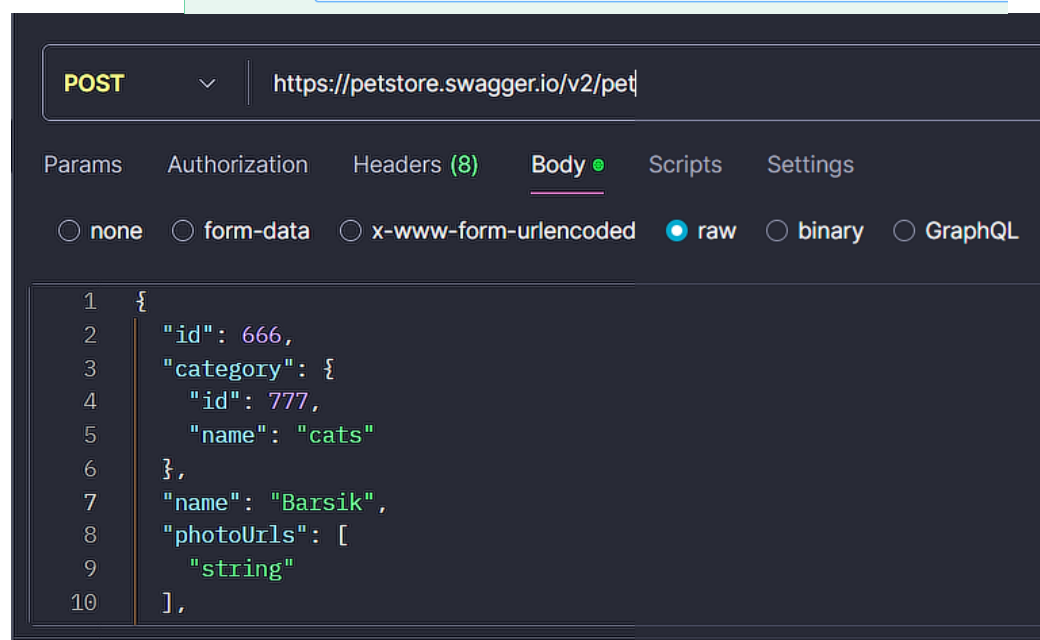


POST /pet Add a new pet to the store

Parameters

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

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



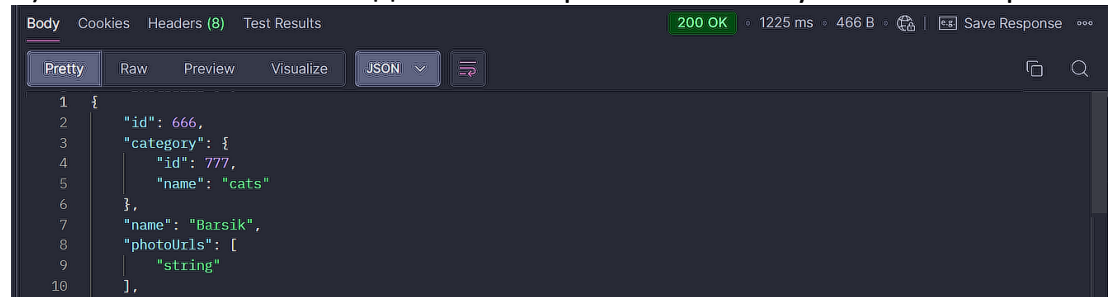
POST https://petstore.swagger.io/v2/pet

Params Authorization Headers (8) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL

```
1 {
2   "id": 666,
3   "category": {
4     "id": 777,
5     "name": "cats"
6   },
7   "name": "Barsik",
8   "photoUrls": [
9     "string"
10  ],
```

4) Нажимаем Send и видим что запрос выполнен успешно. Сохраняем.



Body Cookies Headers (8) Test Results 200 OK 1225 ms 466 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 666,
3   "category": {
4     "id": 777,
5     "name": "cats"
6   },
7   "name": "Barsik",
8   "photoUrls": [
9     "string"
10  ],
```

5) Сформировал новый curl

Curl

```
curl -X 'POST' \
  'https://petstore.swagger.io/v2/pet' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 555,
    "category": {
      "id": 777,
      "name": "cats"
    },
    "name": "Gavrik",
    "photoUrls": [
      "string"
    ],
    "tags": [
      {
        "id": 0,
        "name": "string"
      }
    ],
    "status": "available"
  }'
```

6) Импортировал curl

Import cURL into a collection

```
1 curl -X 'POST' \
2   'https://petstore.swagger.io/v2/pet' \
3   -H 'accept: application/json' \
4   -H 'Content-Type: application/json' \
5   -d '{
6     "id": 555,
```

Request name

Collection name ⓘ

Pet Store

Import Without Saving

Import Into Collection

7) Нажимаем Send. Успех

Body Cookies Headers (8) Test Results

200 OK · 24.94 s · 466 B

Pretty Raw Preview Visualize

JSON

```
1 {
2   "id": 555,
3   "category": {
4     "id": 777,
5     "name": "cats"
6   },
7   "name": "Gavrik",
8   "photoUrls": [
9     "string"
10  ],
11  "status": "available"
12 }
```

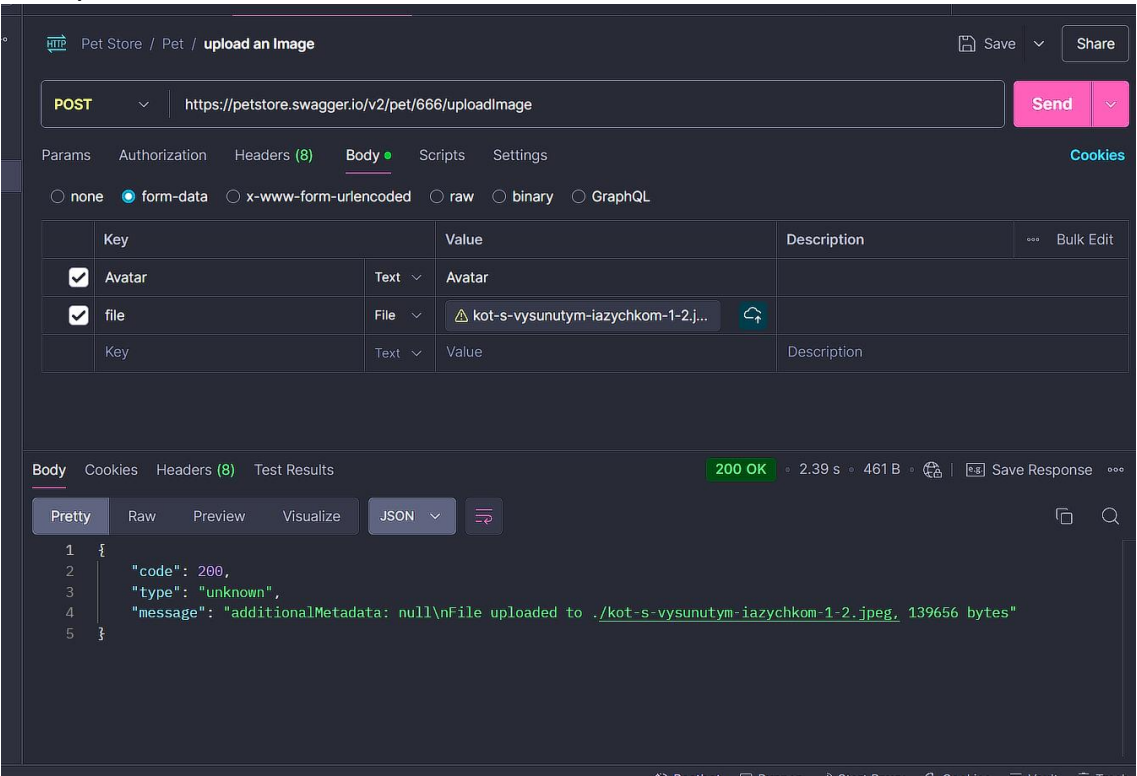
8)Сделал дубликат второго метода



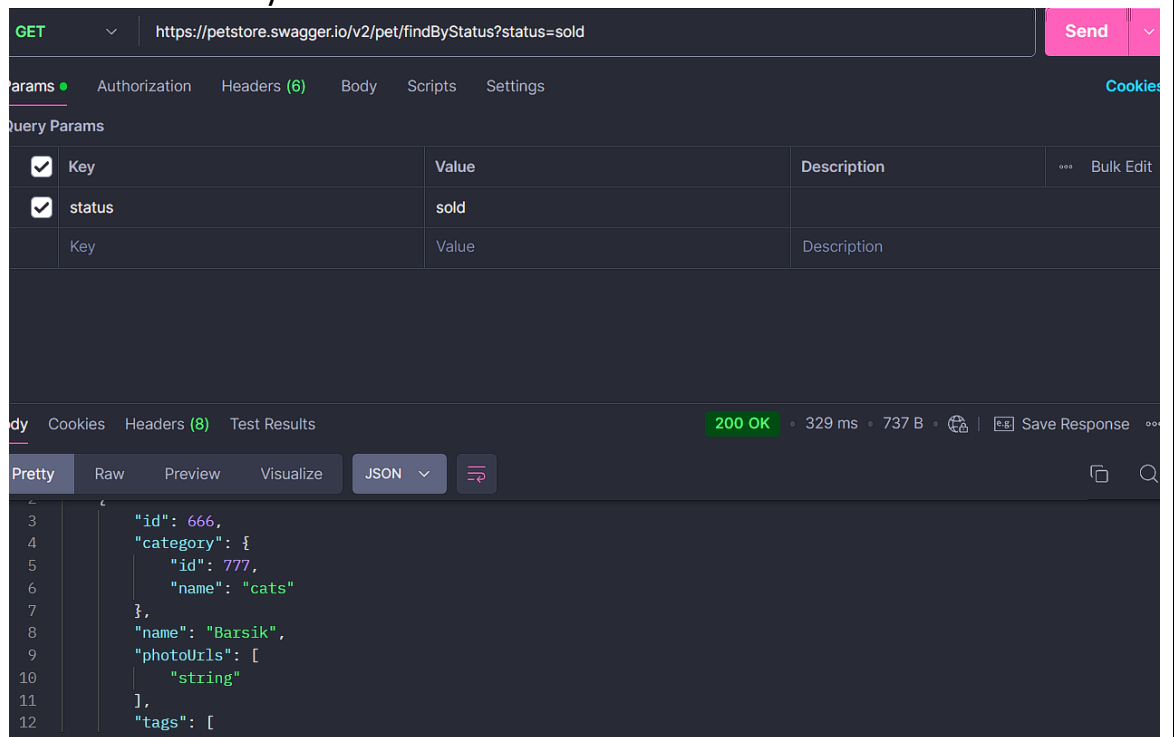
9)Указал ID моего животного и загрузил картинку

Name	Description
petId <small>★ required</small> <code>integer(\$int64)</code> <small>(path)</small>	ID of pet to update <input type="text" value="666"/>
additionalMetadata <code>string</code> <small>(formData)</small>	Additional data to pass to server <input type="text" value="Avatar"/>
file <code>file</code> <small>(formData)</small>	file to upload <div>Выберите файл <input type="text" value="kot-s-vysun...m-1-2.jpeg"/></div>

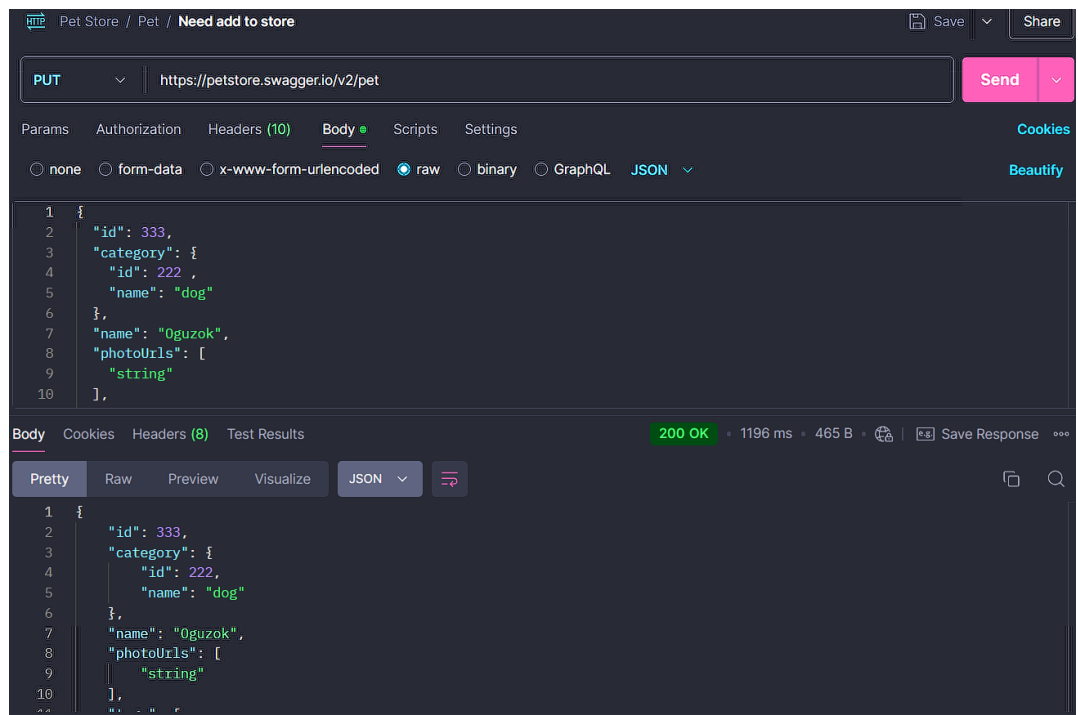
10)Добавил в URL ID животного и uploadImage для загрузки изображения. Заполнил form-data и нажал Send. Успех



11) С помощью другого метода GET создаём новый URL, который показывает статус такого или иного животного



12) Добавил PUT, чтобы узнать какое животное должно быть добавлено на сайт



13)Добавил GET, для нахождения питомца по ID

GEThttps://petstore.swagger.io/v2/pet/666Send

ParamsAuthorizationHeaders (9)BodyScriptsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQL

	Key		Value	Description		Bulk Edit
<input checked="" type="checkbox"/>	petId	Text	666			
	Key	Text	Value	Description		

bodyCookiesHeaders (8)Test Results

200 OK337 ms456 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "id": 666,
3   "category": {
4     "id": 777,
5     "name": "cats"
6   },
7   "name": "Barsik",
8   "photoUrls": [
9     "string"
10  ],
11 }
```

Результат

Я изучил методы и запросы для сайта по поиску питомцев и научился их использовать на практике.