

# Отчёт по лабораторной работе № 1

Выполнила:

Кахнович Мария

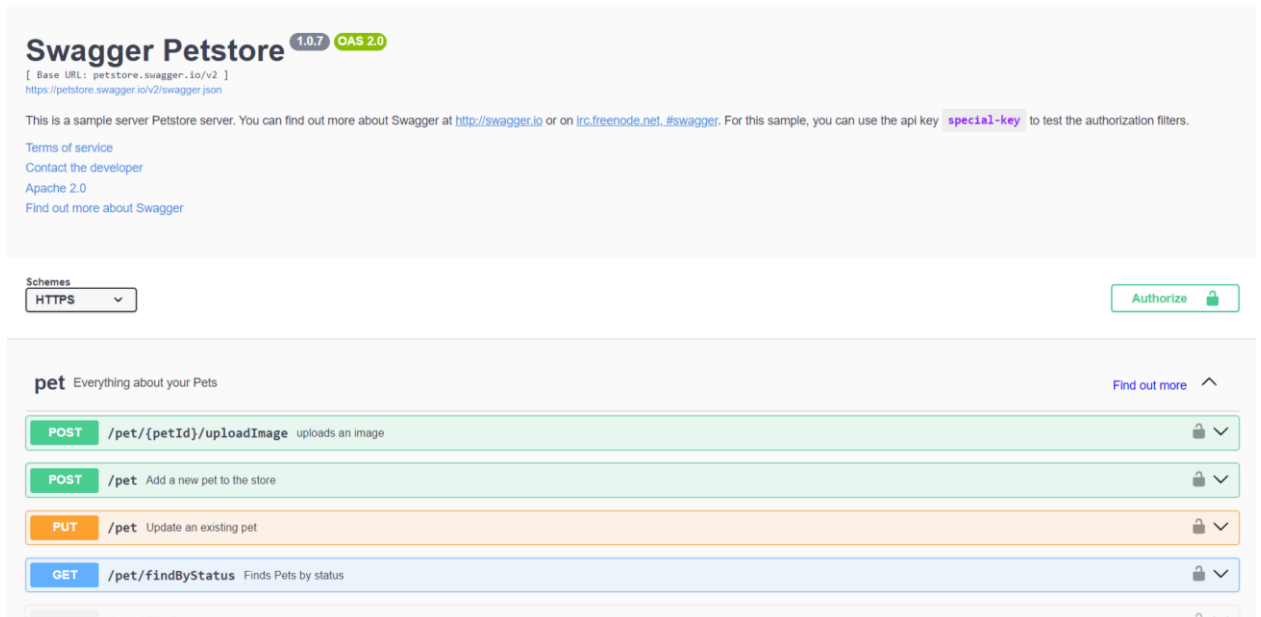
Группа: ИС221

Написано: 12.09.2024

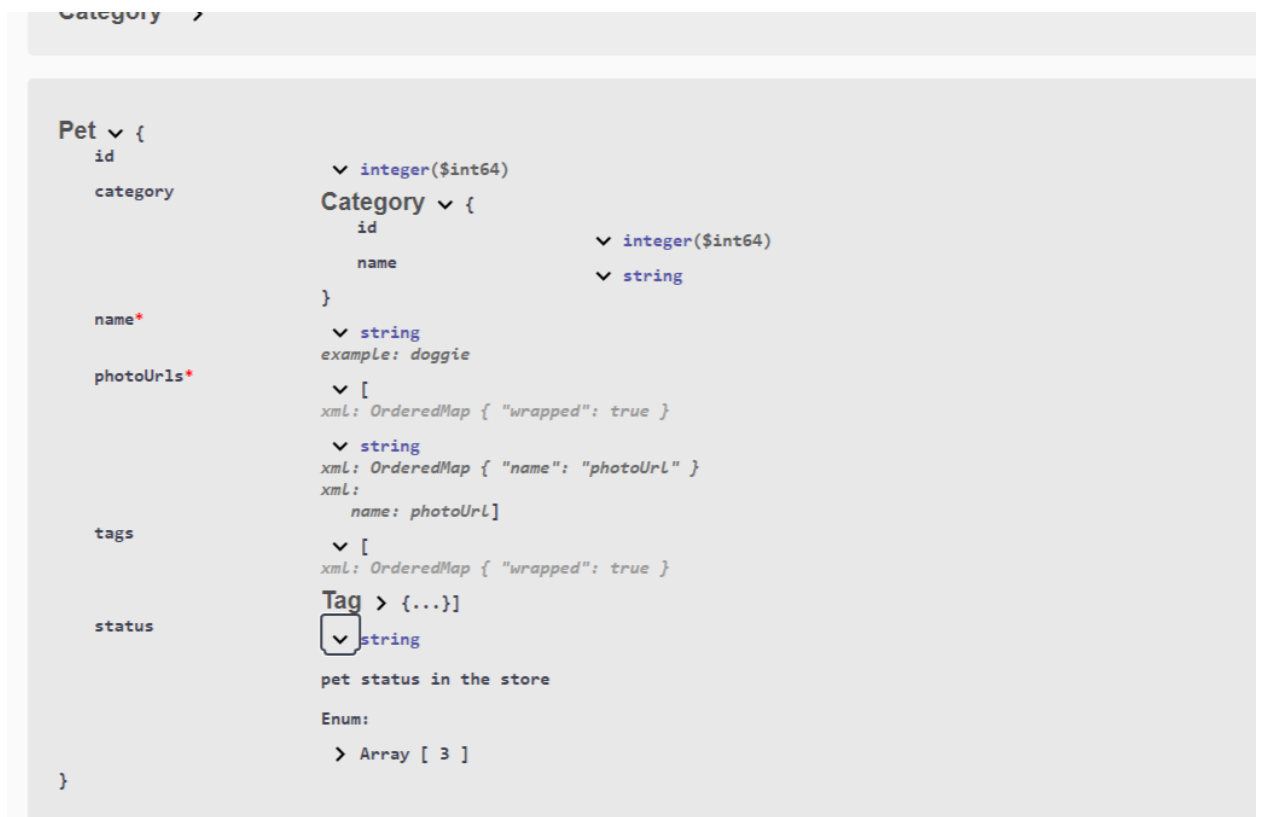
Сдано: 12.09.2024

## Ход работы

1) Переходим на сайт <https://petstore.swagger.io/>



2) Выбираем модель Pet и рассмотрим содержимое



3) Добавляем новое животное в магазин через метод POST /pet, где мы будем передавать данные в теле body при помощи json.

POST

/pet Add a new pet to the store

Parameters

Try it out

Name	Description
<b>body</b> <span style="color: red;">* required</span> object (body)	Pet object that needs to be added to the store <div> <div>Example Value</div> <div>Model</div> </div> <pre> {   "id": 0,   "category": {     "id": 0,     "name": "string"   },   "name": "doggie",   "photoUrls": [     "string"   ],   "tags": [     {       "id": 0,       "name": "string"     }   ],   "status": "available" } </pre> <div> <div>Parameter content type</div> <div>application/json</div> </div>

Responses

Response content type

application/json

4) Нажимаем на Try it out и в json указываем те данные, которые мы хотим, чтобы были у нашего животного.

Parameters

Cancel

Name	Description
<b>body</b> <span style="color: red;">* required</span> object (body)	Pet object that needs to be added to the store <div> <div>Edit Value</div> <div>Model</div> </div> <pre> {   "id": 555777,   "category": {     "id": 4444,     "name": "dogs"   },   "name": "Jack",   "photoUrls": [     "string"   ],   "tags": [     {       "id": 0,       "name": "string"     }   ],   "status": "available" } </pre> <div> <div>Cancel</div> <div>Parameter content type</div> <div>application/json</div> </div>

Execute

Clear

## 5) Получаем результат


Code

Details

200  
Undocumented

Response body

```
{
  "id": 555777,
  "category": {
    "id": 4444,
    "name": "dogs"
  },
  "name": "Jack",
  "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 11:39:49 GMT
server: Jetty(9.2.9.v20150224)
```

Responses

## 6) Нам вернулся запрос, отправленный при помощи curl.

Responses

Response content type application/json

Curl

```
curl -X 'POST' \
  'https://petstore.swagger.io/v2/pet' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 555777,
    "category": {
      "id": 4444,
      "name": "dogs"
    },
    "name": "Jack",
    "photoUrls": [
      "string"
    ],
    "tags": [
      {
        "id": 0,
        "name": "string"
      }
    ],
    "status": "available"
  }'
```

Request URL

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

Server response

Code

Details

## 7) При помощи curl создаем нового животного в командной строке

Run

Clear

US

```
curl -X 'POST' \
  'https://petstore.swagger.io/v2/pet' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 555778,
    "category": {
      "id": 4444,
      "name": "dogs"
    },
    "name": "Barsik",
    "photoUrls": [

```

Run

Clear

JS

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

8) Видим что нам вернулся json с новым животным

```
HTTP/1.1 200 OK
Date: Thu, 12 Sep 2024 12:40:11 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT
Access-Control-Allow-Headers: Content-Type, api_key, Authorization
Server: Jetty(9.2.9.v20150224)

{"id":555778,"category":{"id":4444,"name":"dogs"},"name":"Barsik","photoUrls":["string"],"tags":[{"id":0,"name":"string"}],"status":
```

9) Ищем оба животных при помощи метода get /pet/{petID}

GET /pet/{petId} Find pet by ID

Cancel

Returns a single pet

Name	Description
<b>petId</b> * required integer(\$int64) (path)	ID of pet to return

Execute

Clear

Responses

Response content type application/json

Curl

curl -X 'GET' \n'https://petstore.swagger.io/v2/pet/555778' \n-H 'accept: application/json'

Request URL

https://petstore.swagger.io/v2/pet/555778

Server response

Curl

```
curl -X 'GET' \
  'https://petstore.swagger.io/v2/pet/555778' \
  -H 'accept: application/json'
```

Request URL

https://petstore.swagger.io/v2/pet/555778

Server response

Code

Details

200

Response body

```
{
  "id": 555778,
  "category": {
    "id": 4444,
    "name": "dogs"
  },
  "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 12:45:40 GMT
server: Jetty(9.2.9.v20150224)
```

Responses

GET

/pet/{petId}

Find pet by ID

Returns a single pet

Parameters

Cancel

Name	Description
<b>petId</b> <small>required</small> integer(int64) (path)	ID of pet to return

555777

Execute

Curl

```
curl -X 'GET' \
  'https://petstore.swagger.io/v2/pet/555777' \
  -H 'accept: application/json'
```

Request URL

https://petstore.swagger.io/v2/pet/555777

Server response

Code

Details

200

Response body

```
{
  "id": 555777,
  "category": {
    "id": 4444,
    "name": "dogs"
  },
  "name": "Jack",
  "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 13:12:15 GMT
server: Jetty(9.2.9.v20150224)
```

Вывод: В данной лабораторной работе я познакомилась с инструментом Swagger.  
Научилась создавать, менять данные, описывать и тестировать