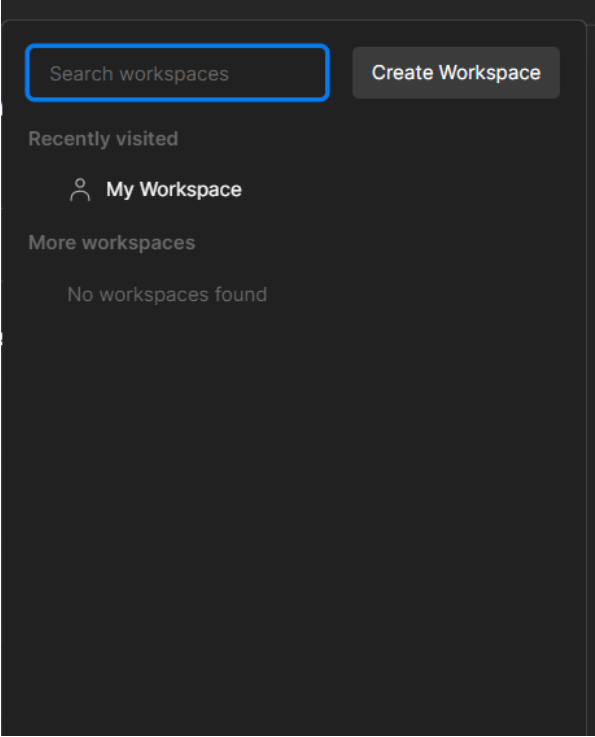
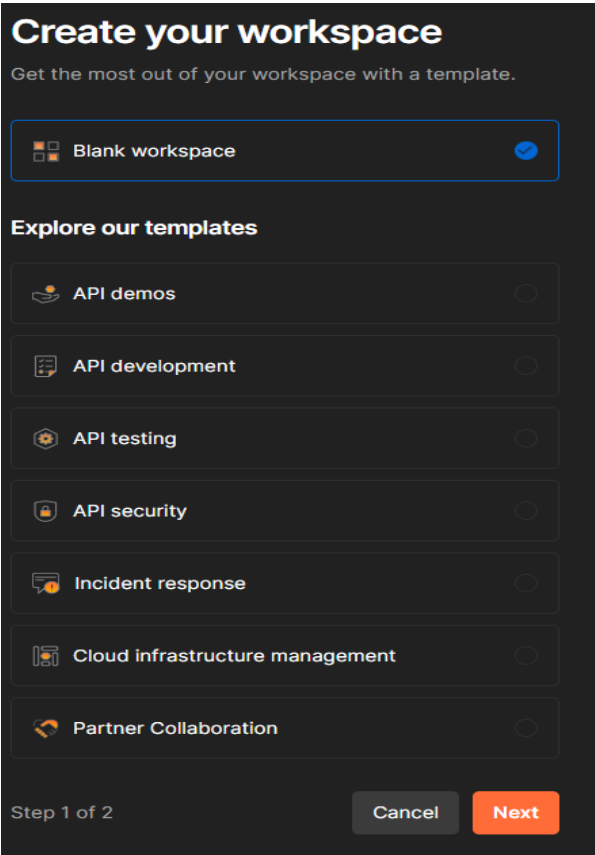


Тема лабораторной работы	lb_2
Выполняющий	Еремина Александра
Помогающий	Мирзоев Руслан
Ход работы	<div>1. В первую очередь, мы создаем workspace.</div> <div></div> <div>2. Здесь мы выбрали пустой проект.</div> <div></div>

3. Придумали название нашему проекту.

The screenshot shows the 'Create your workspace' form in Postman. The 'Name' field is filled with 'mymind'. Under 'Who can access your workspace?', the 'Only me' option is selected. Other options include 'Only invited team members', 'Everyone from your team', 'Only invited partners and team members', and 'Anyone on the internet'. At the bottom, there are 'Back' and 'Create' buttons.

Create your workspace

Name
mymind

Who can access your workspace?

- ☒ Only me
Personal
- ☐ Only invited team members
Private [UPGRADE PLAN](#)
- ☐ Everyone from your team
Team
- ☐ Only invited partners and team members
Partner [UPGRADE PLAN](#)
- ☐ Anyone on the internet
Public

Step 2 of 2 [Back](#) [Create](#)

4. Теперь создаем коллекцию.

The screenshot shows the 'New' menu in Postman. The 'Blank collection' option is highlighted. Below it, there are options to 'Create from template' with 'REST API basics', 'Integration testing basics', and 'API documentation'. A link 'View more templates' is also present.

New **Import**

- ☒ Blank collection
- Create from template
 - ☐ REST API basics
 - ☐ Integration testing basics
 - ☐ API documentation
- [View more templates](#)

Create a collection for your

5. Далее придумали название этой коллекции.

The screenshot shows the 'wasclear' collection in Postman. The collection is empty, and there is a message 'This collection is empty' with a link 'Add a request' to start working. The right sidebar shows the 'Overview' tab.

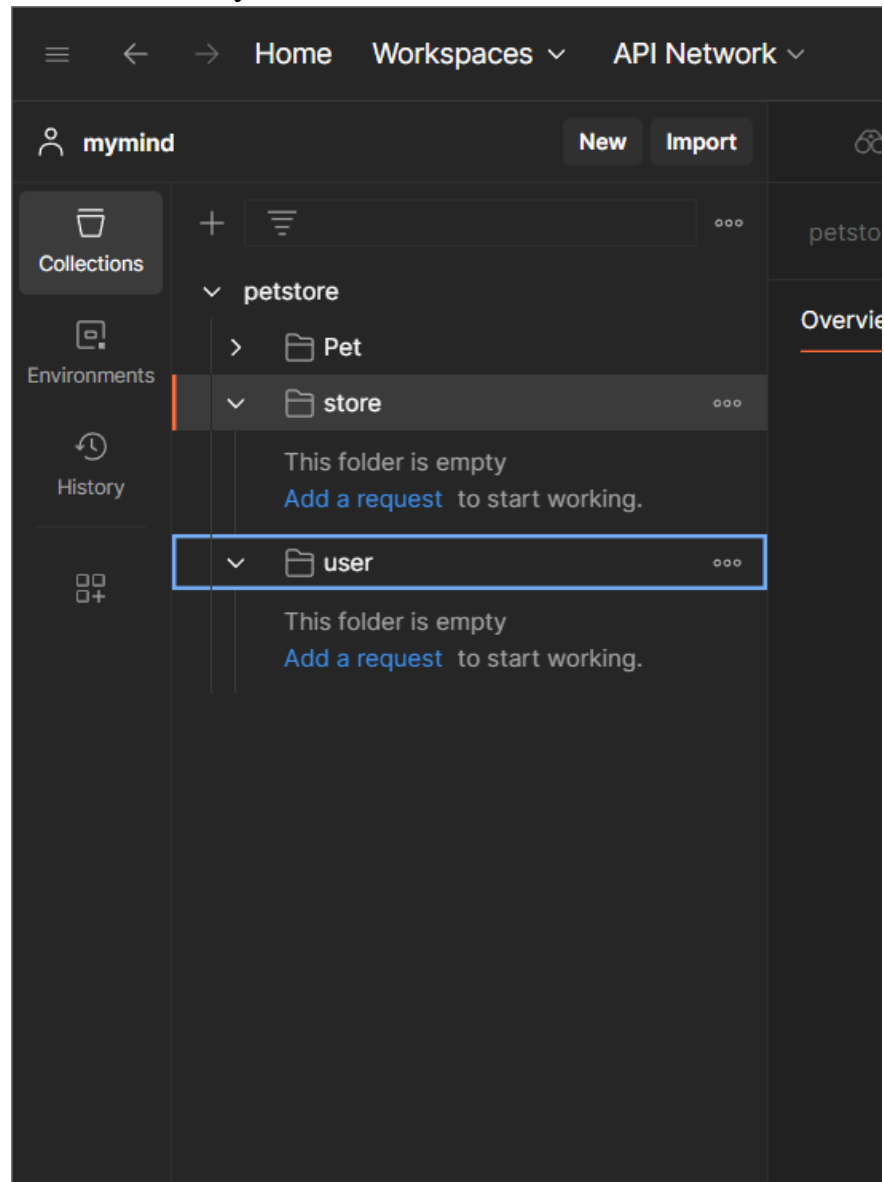
wasclear

☒ wasclear

This collection is empty
[Add a request](#) to start working.

Overview **Authorization**

6. Создали все нужные папки.



7. Создали запрос и добавили в папку Pet метод «Add a new pet to the store».

POST /pet Add a new pet to the store

Parameters

body * required Pet object that needs to be added to the store

object (body)

```

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

```

Parameter content type

application/json

Execute

Responses

Response content type application/json

Code	Description
405	Invalid input

Home Workspaces API Network

my mind

New Import

Collections

Environments

History

petstore

Pet

GET add a new pet to store

store

GET New Request

user

This folder is empty

Add a request to start working.

8. Вставляем ссылку в метод post.

my mind

New Import

petstore / pet Add a new pet to the store

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

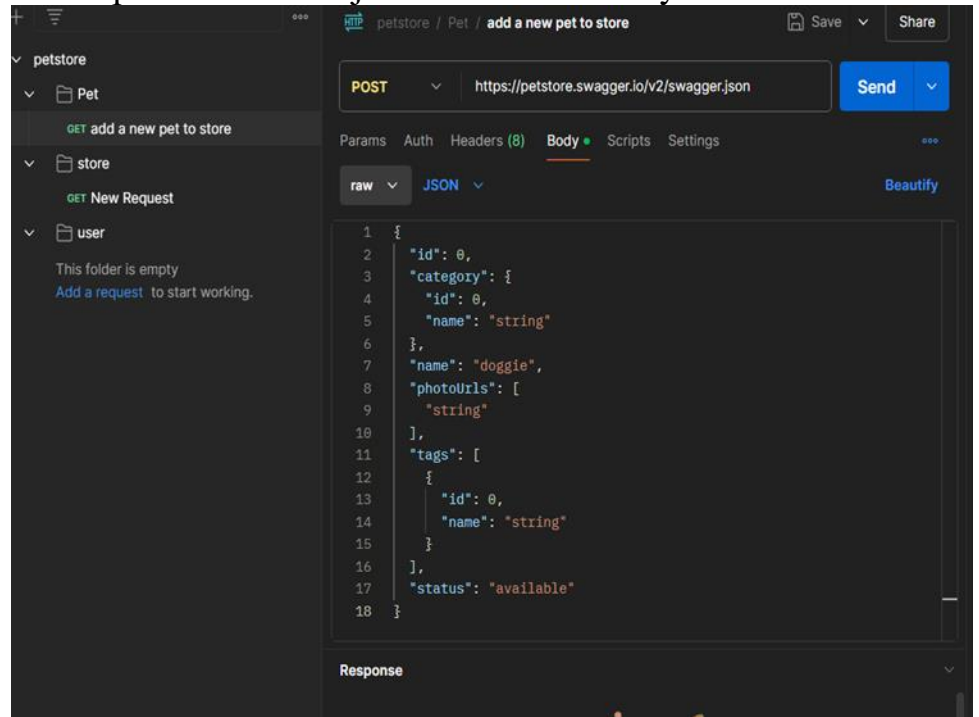
Send

Params Auth Headers (7) Body Scripts Settings

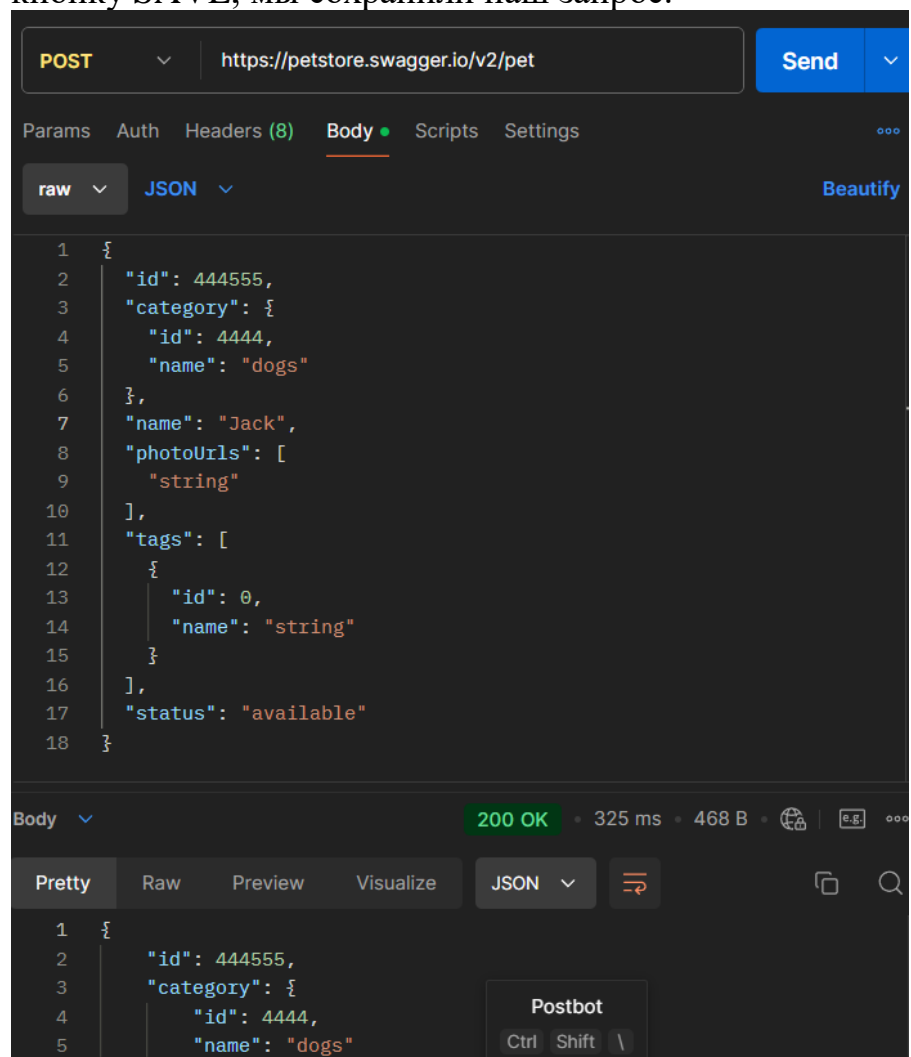
Query Params

Key	Value	Descri...	Bulk Edit
Key	Value	Description	

9. Скопировали с сайта json и встали в Body.



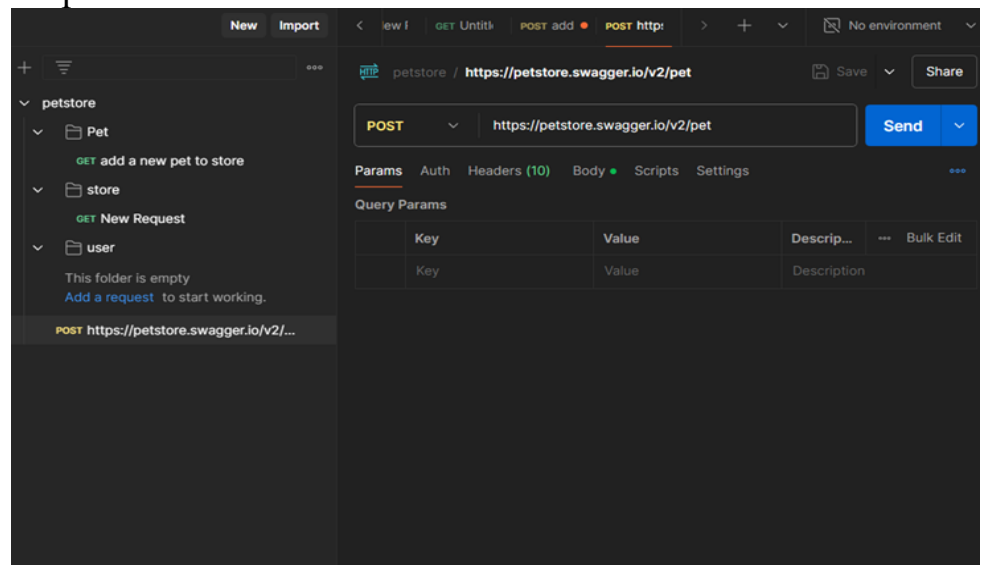
10. Ввели данные в postman, запросы были выполнены, нажав на кнопку SAVE, мы сохранили наш запрос.



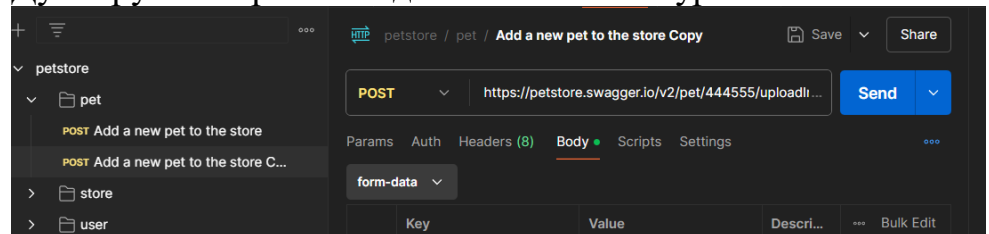
11. Выполняем запрос в postman и копируем curl

Name	Description
body * required object (body)	<p>Pet object that needs to be added to the store</p> <p>Edit Value Model</p> <pre>{ "id": 5556667, "category": { "id": 4444, "name": "dogs" }, "name": "Bobobo", "photoUrls": ["string"], "tags": [{ "id": 0, "name": "string" }], "status": "available" }</pre> <p>Cancel</p> <p>Parameter content type application/json</p> <p>Execute Clear</p>
<p>Responses</p> <p>Response content type application/json</p>	
<p>Curl</p> <pre>curl -X 'POST' \ 'https://petstore.swagger.io/v2/pet' \ -H 'accept: application/json' \ -H 'Content-Type: application/json' \ -d '{ "id": 5556667, "category": { "id": 4444, "name": "dogs" }, "name": "Bobobo", "photoUrls": ["string"], "tags": [{ "id": 0, "name": "string" }], "status": "available" }'</pre> <p>Request URL</p>	

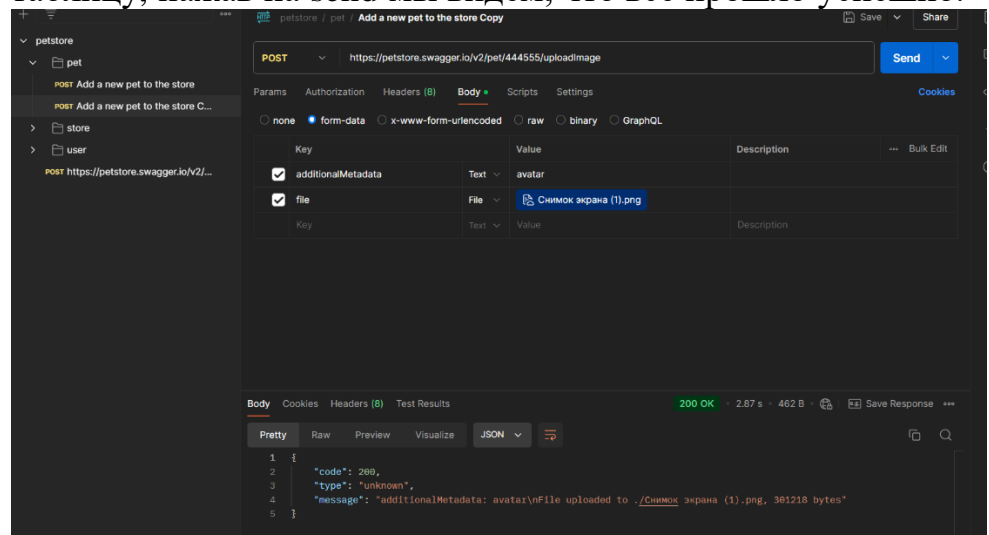
12. Мы импортировали нас автоматически сформировался запрос:

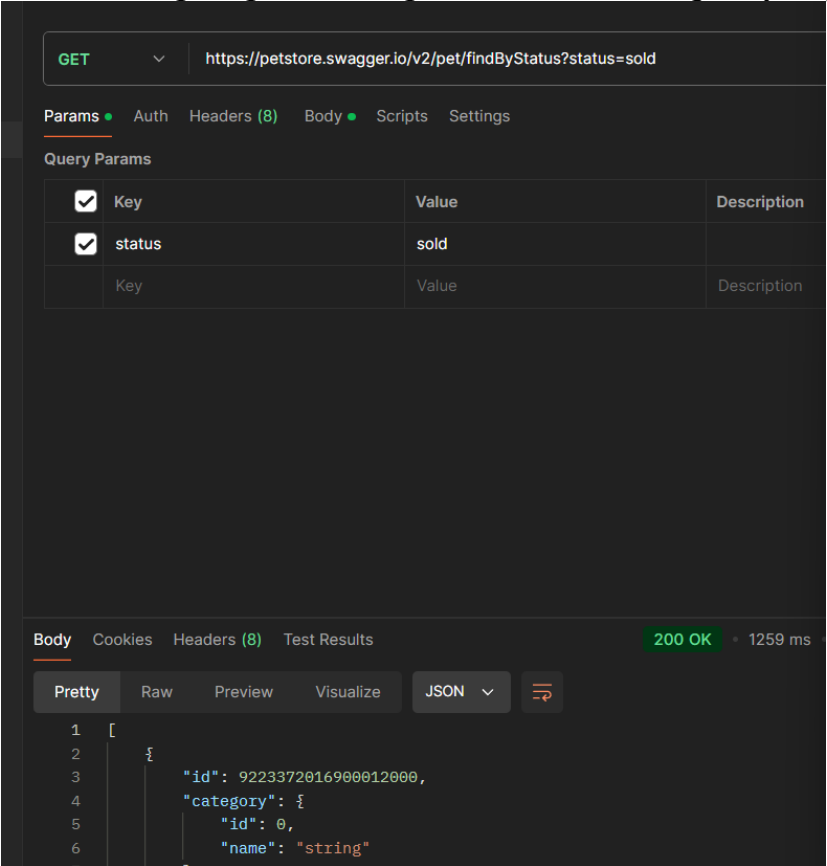


13. Дублируем второй метод и меняем в нем url.



14. Переключаем на вкладку Body и form data, далее заполняем таблицу, нажав на send мы видим, что все прошло успешно.



	<div>15.Здесь мы проверили, как работает метод с query параметром.</div> <div></div>
Результат	<div>После проделанной нами лабораторной работы, по итогу изучили:</div> <div><ol style="list-style-type: none">1. Как создавать и пользоваться workspace в postman, научились создавать в нём папки и методы get и post.2. Мы повторили как в swagger'е вызываются get и post.3. Мы научились "склеивать" вызовы API из Postman и Swagger, чтобы они работали как единое целое.4. Узнали как ещё можно передавать данные в теле запроса.</div>
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