



HOMEWORK #2.2

ОТКРЫТАЯ ШКОЛА
СИСТЕМНЫХ АНАЛИТИКОВ



[CLICK HERE](#)



TASKS

1.

Научиться создавать, отправлять и тестировать API-запросы с помощью Postman.

2.

Научиться добавлять различные типы авторизации к API-запросам в Postman.

3.

Работа с GitHub API, используя Bearer Token.





TASK #1.1

Научиться создавать, отправлять и тестировать API-запросы с помощью Postman.

Создана новая коллекция

Мой первый API

GET Получение информации о выбранном посте

POST Создание поста

PUT Обновление информации о выбранном посте

DEL Удаление выбранного поста

Создан GET-метод

HTTP Мой первый API / Получение информации о выбранном посте

GET https://jsonplaceholder.typicode.com/posts/:postID

Path Variables

Key	Value	Description
postID	5	Description

Body Cookies Headers (25) Test Results 200 OK

{ } JSON Preview Visualize

```
1 {  
2   "userId": 1,  
3   "id": 5,  
4   "title": "nesciunt quas odio",  
5   "body": "reprehenderit veniam quaerat sunt sed\\nalias aut fugiat sit autem sed est\\nvolutpatem omnis possimus esse voluptatibus quis\\nest aut tenetur dolor neque"  
6 }
```



TASK #1.2

Научиться создавать, отправлять и тестировать API-запросы с помощью Postman.

Создан POST-метод

The screenshot shows the Postman interface with a POST method selected and the URL <https://jsonplaceholder.typicode.com/posts>. The title bar says "HTTP Мой первый API / Создание поста".

Добавлен JSON-файл
в тело запроса

The screenshot shows the Postman Body tab with the "raw" option selected. The JSON payload is:

```
1 {  
2   "title": "foo",  
3   "body": "bar",  
4   "userId": 1  
5 }
```

The screenshot shows the Postman Test Results tab with a successful 201 Created response. The JSON response body is identical to the request body:

```
1 {  
2   "title": "foo",  
3   "body": "bar",  
4   "userId": 1,  
5   "id": 101  
6 }
```



TASK #1.3

Научиться создавать, отправлять и тестировать API-запросы с помощью Postman.

Создан PUT-метод

HTTP Мой первый API / Обновление информации о выбранном посте

PUT https://jsonplaceholder.typicode.com/posts/:postID

Добавлен JSON-файл
в тело запроса

Params • Authorization Headers (9) Body • Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "id": 1,  
3   "title": "updated title",  
4   "body": "updated body",  
5   "userId": 1  
6 }
```

Body Cookies Headers (24) Test Results 200 OK

{ } JSON Preview Visualize

```
1 {  
2   "id": 5,  
3   "title": "updated title",  
4   "body": "updated body",  
5   "userId": 1  
6 }
```



TASK #1.4

Научиться создавать, отправлять и тестировать API-запросы с помощью Postman.

Создан DELETE-метод

HTTP Мой первый API / Удаление выбранного поста

DELETE | <https://jsonplaceholder.typicode.com/posts/:postID>

Path Variables

	Key	Value	Description
	postID	2	Description

Body Cookies Headers (24) Test Results | ⚙️ 200 OK

[{} JSON] ▶ Preview ⚡ Visualize | ▾

1 [{}]



TASK #2.1

Научиться добавлять различные типы авторизации к API-запросам в Postman.

Создана новая коллекция

API с авторизацией

- POST POST-запрос с использованием API
- POST POST-запрос с использованием Bearer Token
- POST POST-запрос с использованием Basic Auth

Создан POST-запрос с использованием API авторизации

HTTP API с авторизацией / POST-запрос с использованием API

POST https://jsonplaceholder.typicode.com/posts

Params Authorization • Headers (10) Body • Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "title": "API",  
3   "body": "brrrr",  
4   "userId": 1  
5 }
```

Auth Type

API Key

The authorization header will be automatically generated when you send the request. Learn more about [API Key](#) authorization.

Key: X-API-KEY

Value: (redacted)

Add to: Header



TASK #2.2

Научиться добавлять различные типы авторизации к API-запросам в Postman.

Создан POST-запрос с использованием Bearer Token

HTTP API с авторизацией / POST-запрос с использованием Bearer Token

POST <https://jsonplaceholder.typicode.com/posts>

Params Authorization Headers (10) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "title": "Bearer Token",  
3   "body": "brrrrr",  
4   "userId": 1  
5 }
```

Auth Type

Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token

.....



TASK #2.3

Научиться добавлять различные типы авторизации к API-запросам в Postman.

Создан POST-запрос с использованием Basic Auth

HTTP API с авторизацией / POST-запрос с использованием Basic Auth

POST <https://jsonplaceholder.typicode.com/posts>

Params Authorization Headers (10) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "title": "Basic Auth",  
3   "body": "brrrr",  
4   "userId": 1  
5 }
```

Auth Type: Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about [Basic Auth](#) authorization.

Username: rexandel

Password:  



TASK #2.4

Научиться добавлять различные типы авторизации к API-запросам в Postman.

Протестирован POST-метод с использованием API

Body Cookies Headers (25) Test Results | ⏱ 201 Created

{ } JSON ▾ ▷ Preview ⚡ Visualize | ▾

```
1 {  
2   "title": "API",  
3   "body": "brrrr",  
4   "userId": 1,  
5   "id": 101  
6 }
```

Протестирован POST-метод с использованием Bearer Token

Body Cookies Headers (25) Test Results | ⏱ 201 Created

{ } JSON ▾ ▷ Preview ⚡ Visualize | ▾

```
1 {  
2   "title": "Bearer Token",  
3   "body": "brrrr",  
4   "userId": 1,  
5   "id": 101  
6 }
```

Протестирован POST-метод с использованием Basic Auth

Body Cookies Headers (25) Test Results | ⏱ 201 Created

{ } JSON ▾ ▷ Preview ⚡ Visualize | ▾

```
1 {  
2   "title": "Basic Auth",  
3   "body": "brrrr",  
4   "userId": 1,  
5   "id": 101  
6 }
```



TASK #2.5

Научиться добавлять различные типы авторизации к API-запросам в Postman.

Создана новая коллекция

OpenWeather API

- GET Moscow
- GET Krasnodar
- GET Saint Petersburg

Созданы GET-запросы с использованием API авторизации к сервису OpenWeatherMap

GET https://api.openweathermap.org/data/2.5/weather?q=Moscow&units=metric&lang=ru

Params • Authorization • Headers (7) Body Scripts Settings

Query Params

Key	Value
appid
q	Moscow
units	metric
lang	ru
Key	Value

HTTP OpenWeather API / Saint Petersburg

GET https://api.openweathermap.org/data/2.5/weather?q=Saint Petersburg&units=metric&lang=ru

HTTP OpenWeather API / Krasnodar

GET https://api.openweathermap.org/data/2.5/weather?q=Krasnodar&units=metric&lang=ru



TASK #2.6

Научиться добавлять различные типы авторизации к API-запросам в Postman.

Получен и добавлен API-ключ авторизации

The screenshot shows the 'Auth Type' dropdown set to 'API Key'. Below it, a note states: 'The authorization header will be automatically generated when you send the request. Learn more about [API Key](#) authorization.' To the right, there are fields for 'Key' (set to 'appid') and 'Value' (containing a redacted string). A dropdown 'Add to' is set to 'Query Params'.

Созданные запросы протестированы

The screenshot shows a successful test result with a status of '200 OK'. The 'Body' tab displays a JSON response representing weather data for a specific location. The response includes coordinates and a single weather condition entry.

```
1 {  
2   "coord": {  
3     "lon": 37.6156,  
4     "lat": 55.7522  
5   },  
6   "weather": [  
7     {  
8       "id": 804,  
9       "main": "Clouds",  
10      "description": "пасмурно",  
11      "icon": "04d"  
12    }  
13  ],  
14}
```

The screenshot shows another successful test result with a status of '200 OK'. The 'Body' tab displays a JSON response representing weather data for a different location. The response includes coordinates and a single weather condition entry.

```
1 {  
2   "coord": {  
3     "lon": 30.2642,  
4     "lat": 59.8944  
5   },  
6   "weather": [  
7     {  
8       "id": 803,  
9       "main": "Clouds",  
10      "description": "облачно с прояснениями",  
11      "icon": "04d"  
12    }  
13  ],  
14}
```



TASK #3.1

Работа с GitHub API, используя Bearer Token.

Создана новая коллекция

GitHub API

GET Получить информацию о профиле

GET Получение информации о всех репозиториях

GET Получение информации об определенном репо...

Созданы GET-запросы с Bearer Token

HTTP GitHub API / Получить информацию о профиле

GET

<https://api.github.com/user>

HTTP GitHub API / Получение информации о всех репозиториях

GET

<https://api.github.com/user/repos>

HTTP GitHub API / Получение информации об определенном репозитории

GET

<https://api.github.com/repos/:username/:repo>



TASK #3.2

Работа с GitHub API, используя Bearer Token.

Получен и добавлен Bearer Token

Auth Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token: [REDACTED]

Созданные запросы протестированы

Body Cookies Headers (28) Test Results 200 OK

{ } JSON ▾ Preview Visualize ▾

```
1 {  
2   "login": "rexandel",  
3   "id": 112493375,  
4   "node_id": "U_kgDOBzSDPw",  
5   "avatar_url": "https://avatars.githubusercontent.com/u/112493375?v=4",  
6   "gravatar_id": "",  
7   "url": "https://api.github.com/users/rexandel",
```

Body Cookies Headers (27) Test Results 200 OK

{ } JSON ▾ Preview Visualize ▾

```
1 [  
2   {  
3     "id": 929046032,  
4     "node_id": "R_kgD0N2AeEA",  
5     "name": "big_data_research",  
6     "full_name": "rexandel/big_data_research",  
7     "private": false,  
8     "owner": {  
9       "login": "rexandel",  
10      "id": 112493375,  
11      "node_id": "U_kgDOBzSDPw",  
12      "avatar_url": "https://avatars.githubusercontent.com/u/112493375?v=4",  
13      "gravatar_id": "",
```

Body Cookies Headers (28) Test Results 200 OK

{ } JSON ▾ Preview Visualize ▾

```
1 {  
2   "id": 1028306778,  
3   "node_id": "R_kgDOPUq3Wg",  
4   "name": "rexandel",  
5   "full_name": "rexandel/rexandel",  
6   "private": false,
```



API SPECIFICATION

Для получения всех разработанных API спецификаций
можно перейти по ссылке в GitHub репозиторий:
[GitHub Link](#).





**СПАСИБО
ЗА ВНИМАНИЕ
И ВАШ ТРУД**

CLICK HERE