Распределенные системы и сети

Урок 4. Распределённые системы

Домашнее задание

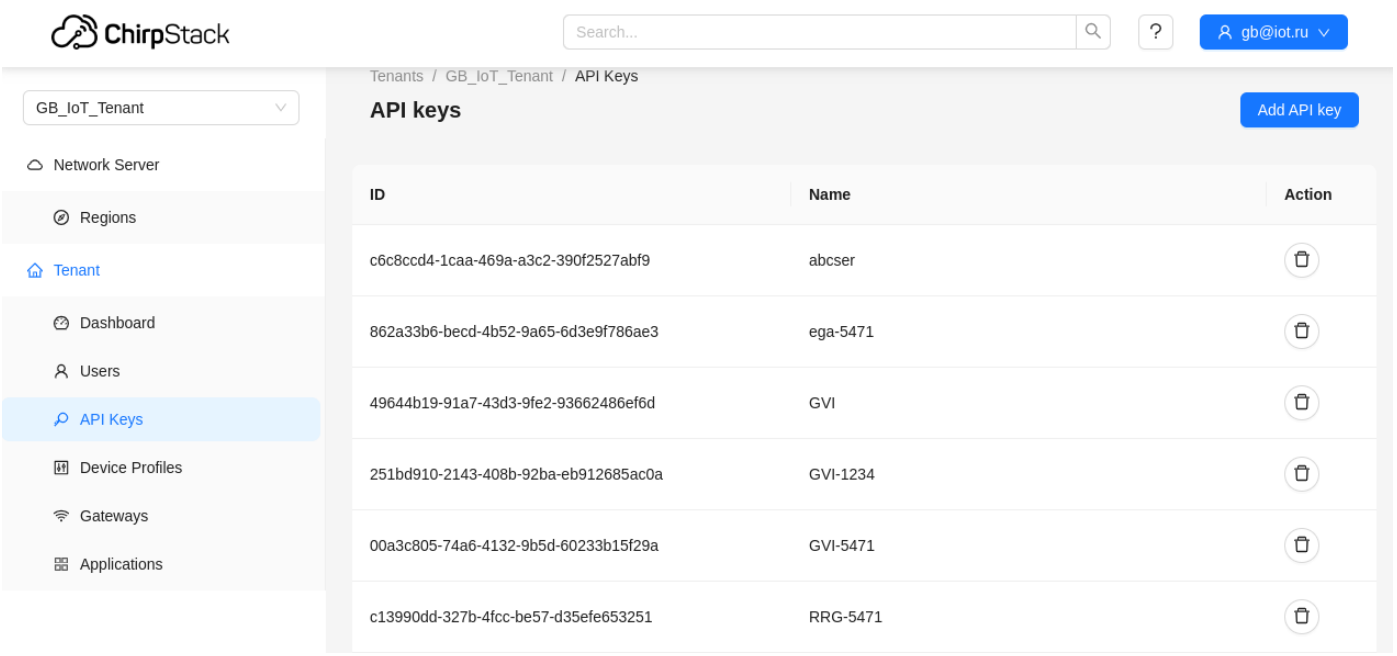
1. Прислать результаты выполненных запросов через Swagger UI
2. Postman
3. Прислать скрипт на любом удобном языке программирования, где через API будут созданы 10 устройств (все переменные должны задаваться в коде, списком или массивом, но одной строкой), будет запрошена информация о 10 устройствах, будут обновлены имена 10 устройств, будут удалены 10 устройств.

Желательно использовать Python через Google Collab. Для сдачи ДЗ студент должен предоставить преподавателю ссылку с доступом к блокноту с кодом (перед отправкой ДЗ проверьте, что ссылка открывается в приватном режиме браузера).

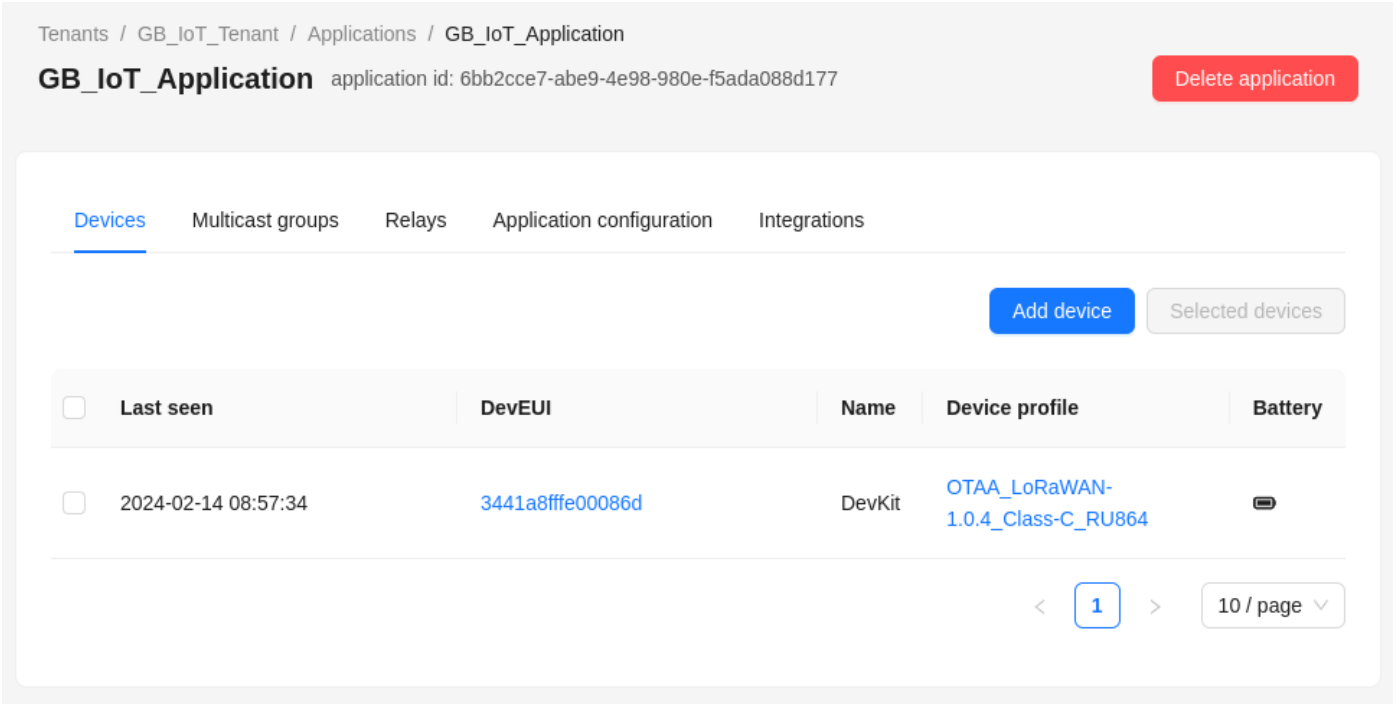
Решение

1. Swagger UI

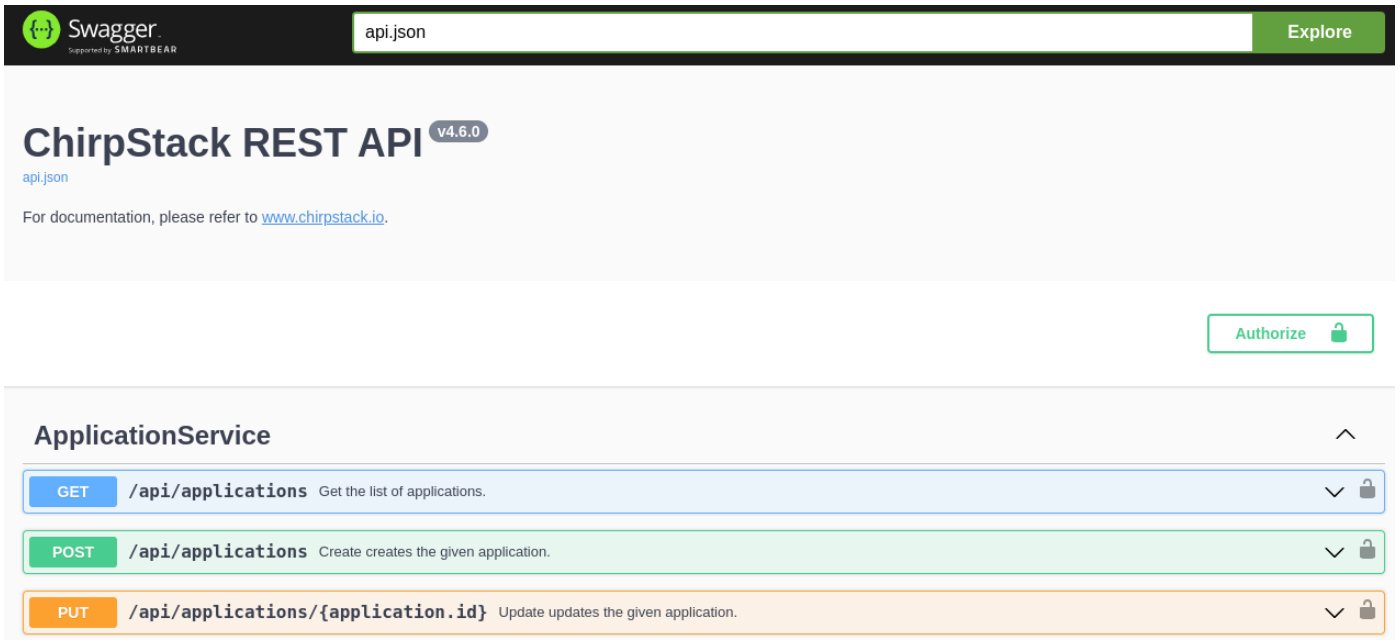
Создание API-token



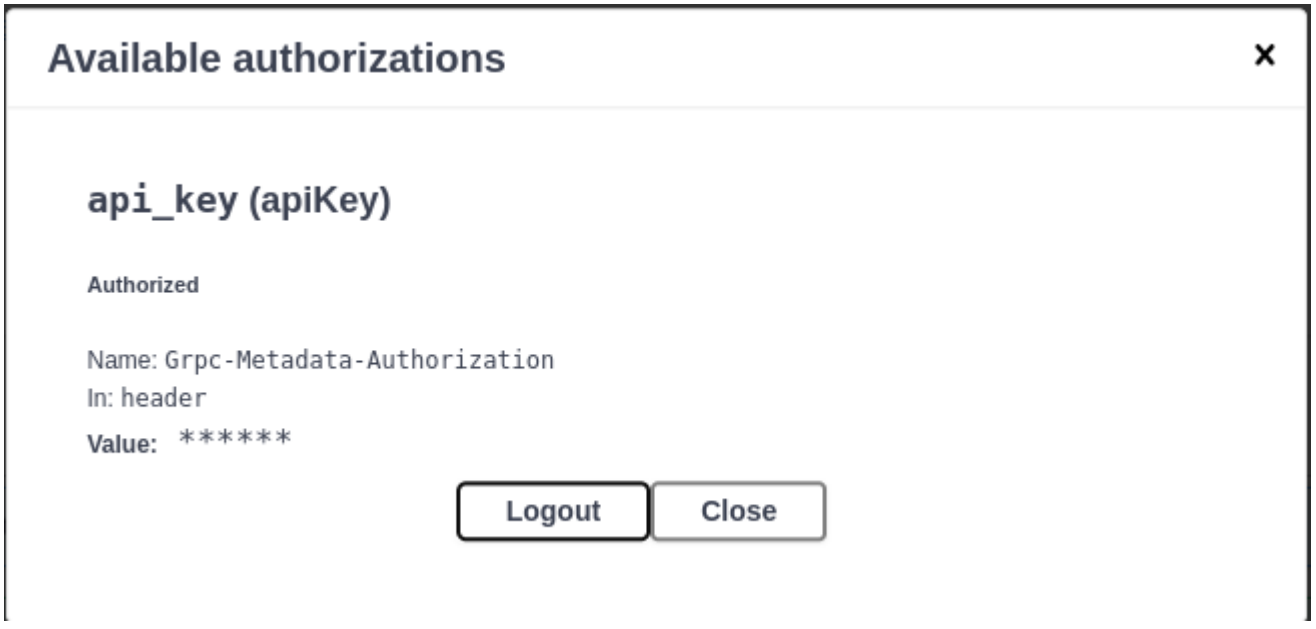
Applications



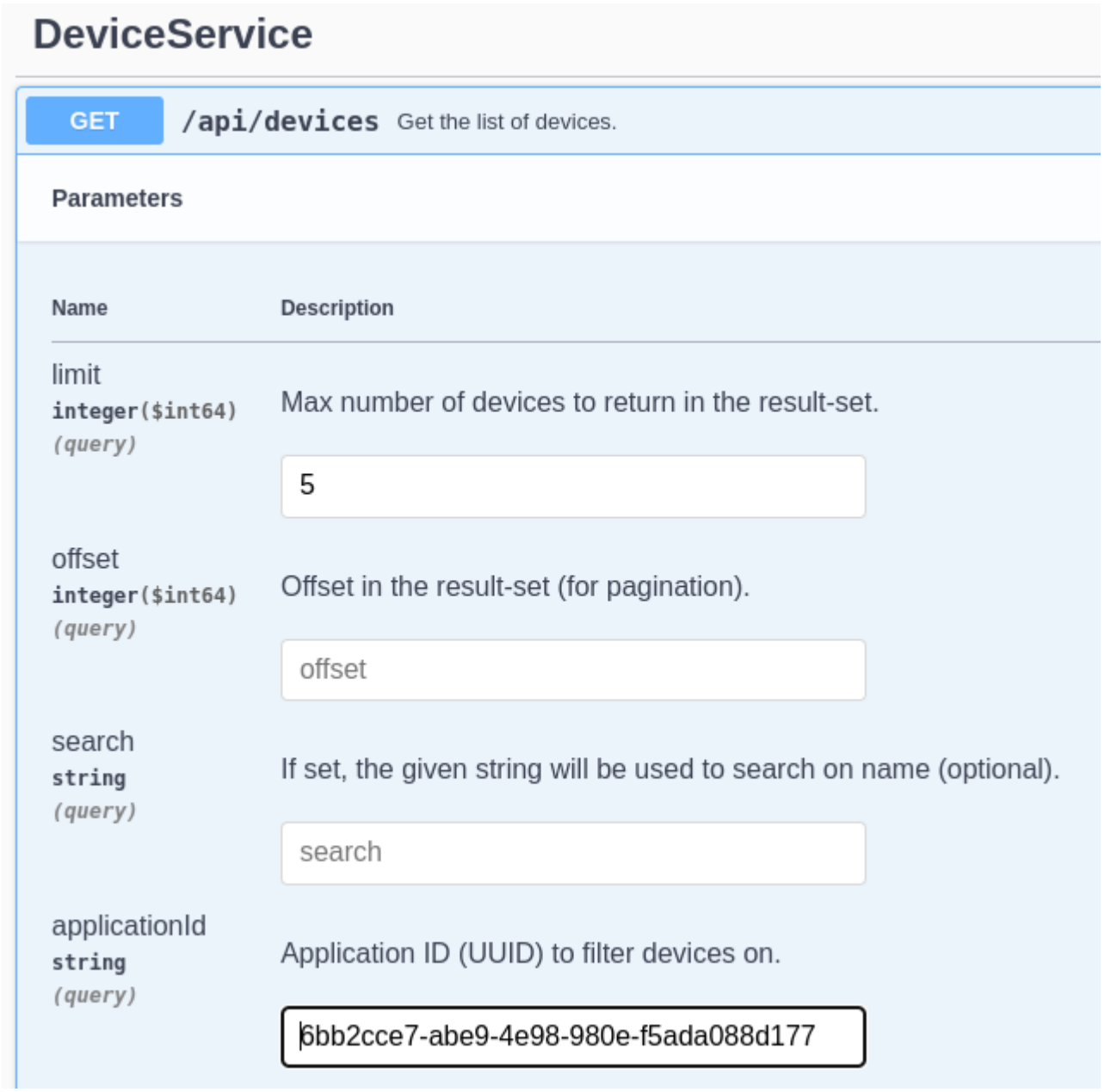
Swagger API



Bearer token



GET


DeviceService
GET
/api/devices
Get the list of devices.
Parameters
Name
Description
limit
integer($int64)
(query)
Max number of devices to return in the result-set.
5
offset
integer($int64)
(query)
Offset in the result-set (for pagination).
offset
search
string
(query)
If set, the given string will be used to search on name (optional).
search
applicationId
string
(query)
Application ID (UUID) to filter devices on.
6bb2cce7-abe9-4e98-980e-f5ada088d177

Responses

Responses

Curl

```
curl -X 'GET' \
  'https://chirpstack-api.iotserv.ru/api/devices?limit=5&applicationId=6bb2cce7-abe9-4e98-980e-f5ada088d177' \
  -H 'accept: application/json' \
  -H 'Grpc-Metadata-Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJjaGlycHN0YWNrIiwiaXNzIj
```

Request URL

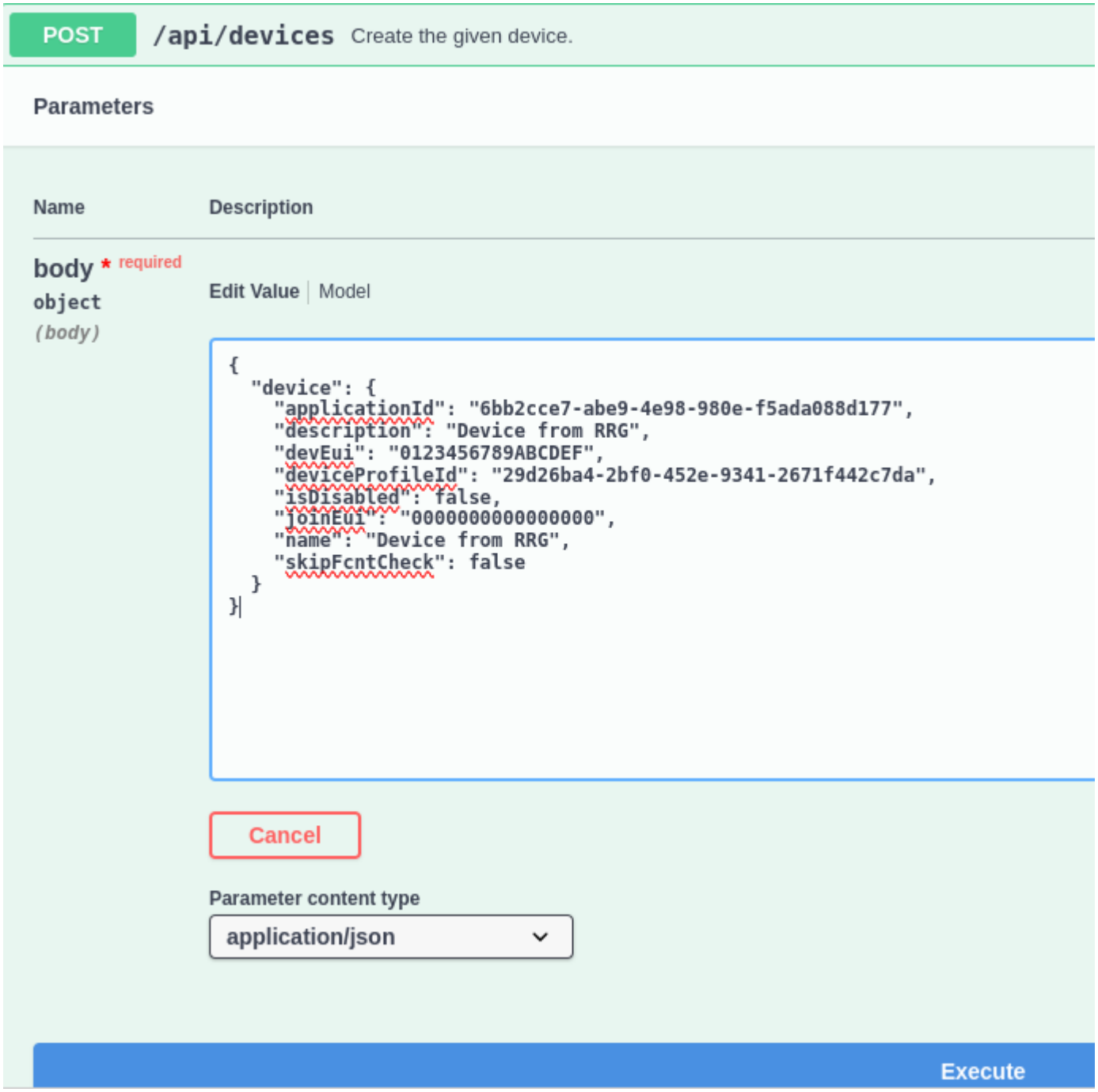
```
https://chirpstack-api.iotserv.ru/api/devices?limit=5&applicationId=6bb2cce7-abe9-4e98-980e-f5ada088d177
```

Server response

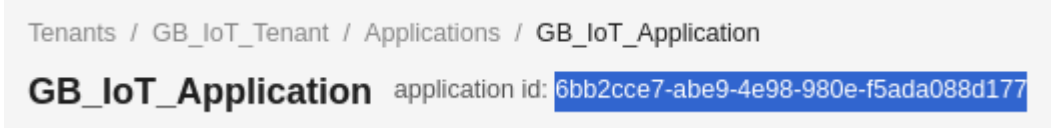
Code	Details
200	Response body


```
{
  "totalCount": 1,
  "result": [
    {
      "devEui": "3441a8fffe00086d",
      "createdAt": "2024-01-17T10:50:58.793954Z",
      "updatedAt": "2024-01-17T10:50:58.793954Z",
      "lastSeenAt": "2024-02-14T08:57:34.178648Z",
      "name": "DevKit",
      "description": "LoRaWAN DevKit",
      "deviceProfileId": "29d26ba4-2bf0-452e-9341-2671f442c7da",
      "deviceProfileName": "OTAA_LoRaWAN-1.0.4_Class-C_RU864",
      "deviceStatus": {
        "margin": 6,
        "externalPowerSource": false,
        "batteryLevel": 100
      }
    }
  ]
}
```

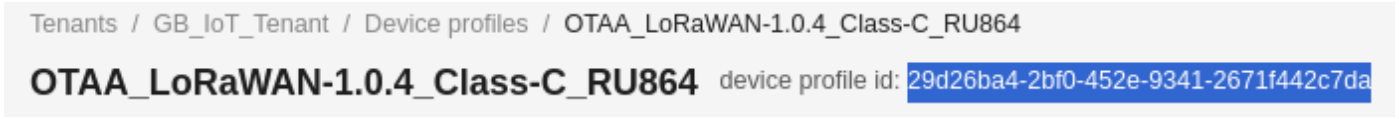

POST



Application id



Device profile id



DevEUI

Любое 64-битное значение в HEX: 0123456789abcdef

Responses

Curl

```
curl -X 'POST' \
  'https://chirpstack-api.iotserv.ru/api/devices' \
  -H 'accept: application/json' \
  -H 'Grpc-Metadata-Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJ
  -H 'Content-Type: application/json' \
  -d '{
  "device": {
    "applicationId": "6bb2cce7-abe9-4e98-980e-f5ada088d177",
    "description": "Device from RRG",
    "devEui": "0123456789ABCDEF",
    "deviceProfileId": "29d26ba4-2bf0-452e-9341-2671f442c7da",
    "isDisabled": false,
    "joinEui": "0000000000000000",
    "name": "Device from RRG",
    "skipFcntCheck": false
  }
}'
```

Request URL

```
https://chirpstack-api.iotserv.ru/api/devices
```

Server response

Code	Details
200	Response body

```
{}
```

Response headers

```
content-length: 2
content-type: application/json
date: Sat,17 Feb 2024 10:48:29 GMT
grpc-metadata-content-type: application/grpc
grpc-metadata-date: Sat,17 Feb 2024 10:48:29 GMT
grpc-metadata-x-log-dev_eui: 0123456789ABCDEF
grpc-metadata-x-log-is_disabled: false
server: nginx
x-http-version: HTTP/2.0
```

Result

Tenants / GB_IoT_Tenant / Applications / GB_IoT_Application

GB_IoT_Application application id: 6bb2cce7-abe9-4e98-980e-f5ada088d177

Devices | Multicast groups | Relays | Application configuration | Integrations

Last seen	DevEUI	Name	Device profile
Never	0123456789abcdef	Device from RRG	OTAA_LoRaWAN-

PUT

Responses

Curl

```
curl -X 'PUT' \
  'https://chirpstack-api.iotserv.ru/api/devices/0123456789abcdef' \
  -H 'accept: application/json' \
  -H 'Grpc-Metadata-Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUz
  -H 'Content-Type: application/json' \
  -d '{
  "device": {
    "applicationId": "6bb2cce7-abe9-4e98-980e-f5ada088d177",
    "deviceProfileId": "29d26ba4-2bf0-452e-9341-2671f442c7da",
    "name": "Device from RRG UPDATED"
  }
}'
```

Request URL

```
https://chirpstack-api.iotserv.ru/api/devices/0123456789abcdef
```

Server response

Code	Details
200	Response body

```
{}
```

Response headers

```
content-length: 2
content-type: application/json
date: Sat,17 Feb 2024 11:07:14 GMT
grpc-metadata-content-type: application/grpc
grpc-metadata-date: Sat,17 Feb 2024 11:07:14 GMT
grpc-metadata-x-log-dev_eui: 0123456789abcdef
grpc-metadata-x-log-is_disabled: false
server: nginx
x-http-version: HTTP/2.0
```

Responses

Responses

Curl

```
curl -X 'PUT' \
  'https://chirpstack-api.iotserv.ru/api/devices/0123456789abcdef' \
  -H 'accept: application/json' \
  -H 'Grpc-Metadata-Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUz
  -H 'Content-Type: application/json' \
  -d '{
  "device": {
    "applicationId": "6bb2cce7-abe9-4e98-980e-f5ada088d177",
    "deviceProfileId": "29d26ba4-2bf0-452e-9341-2671f442c7da",
    "name": "Device from RRG UPDATED"
  }
}'
```

Request URL

```
https://chirpstack-api.iotserv.ru/api/devices/0123456789abcdef
```

Server response

Code	Details

200	**Response body**
	`{}`
	Response headers
	`content-length: 2`
	`content-type: application/json`
	`date: Sat,17 Feb 2024 11:07:14 GMT`
	`grpc-metadata-content-type: application/grpc`
	`grpc-metadata-date: Sat,17 Feb 2024 11:07:14 GMT`
	`grpc-metadata-x-log-dev_eui: 0123456789abcdef`
	`grpc-metadata-x-log-is_disabled: false`
	`server: nginx`
	`x-http-version: HTTP/2.0`

Result

Tenants / GB_IoT_Tenant / Applications / GB_IoT_Application

GB_IoT_Application application id: 6bb2cce7-abe9-4e98-980e-f5ada088d177

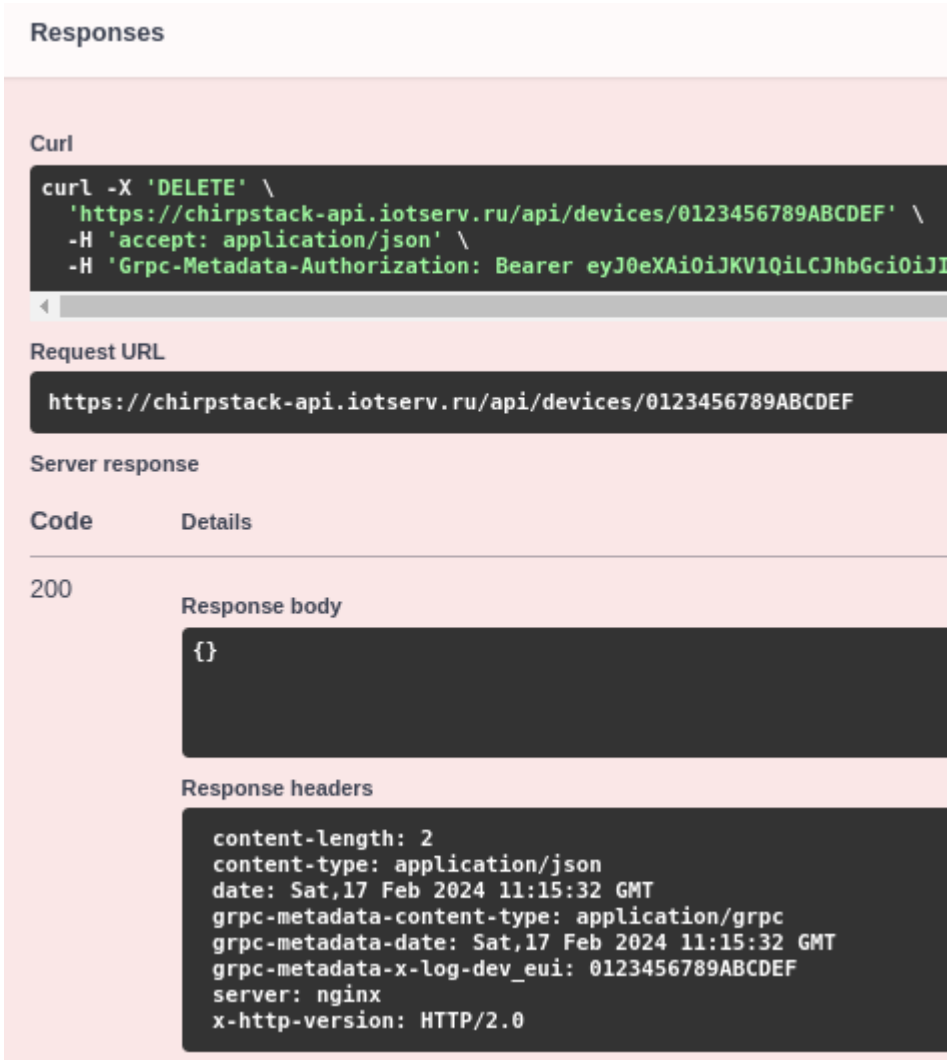
Devices | Multicast groups | Relays | Application configuration | Integrations

Ad

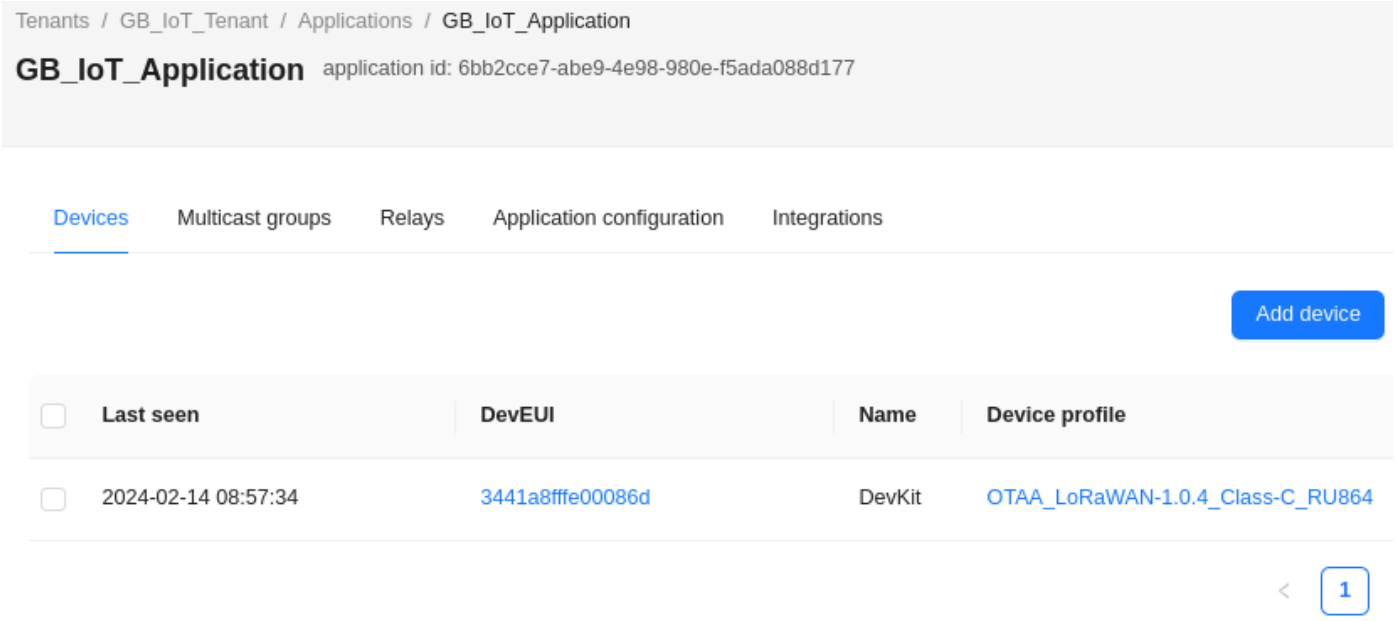
Last seen	DevEUI	Name	Device profile
Never	0123456789abcdef	Device from RRG UPDATED	OTAA_LoRaWAC_RU864

DELETE

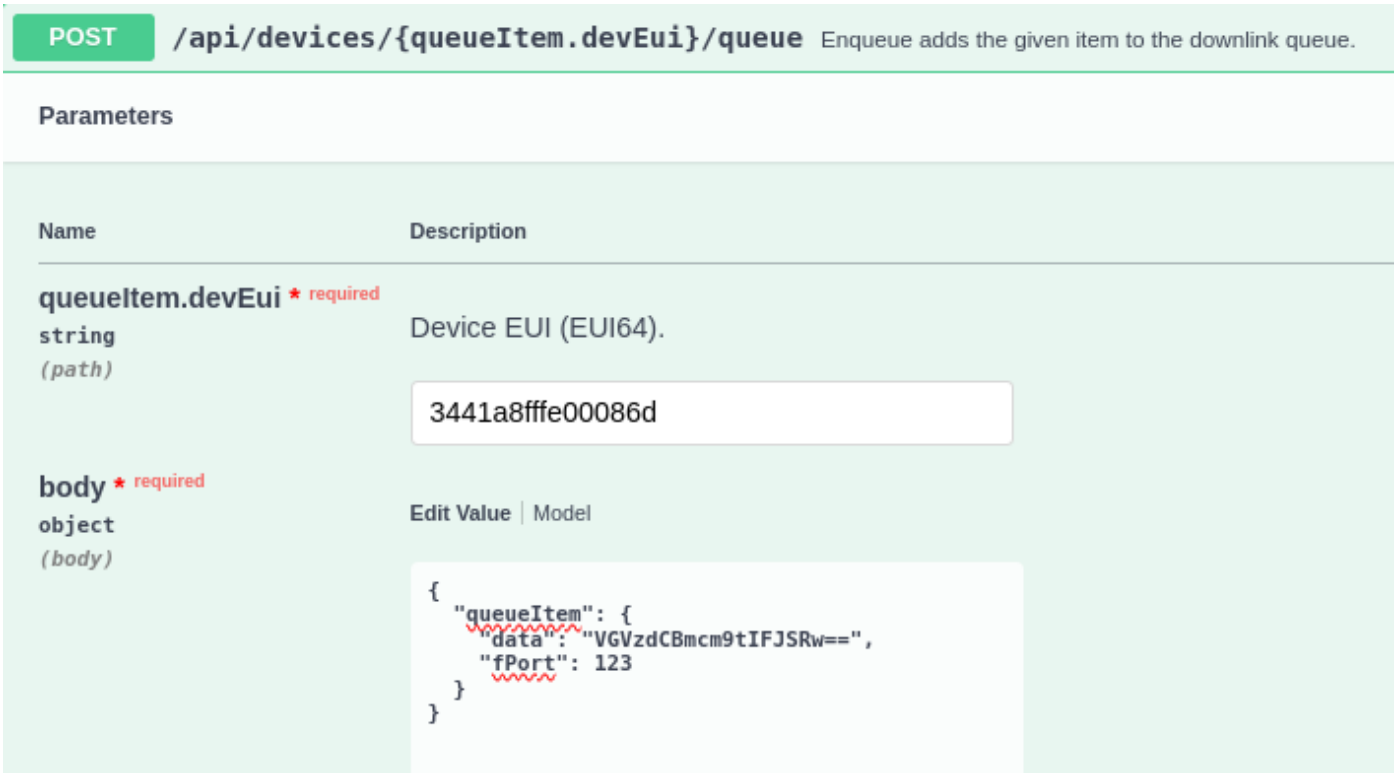
Responses



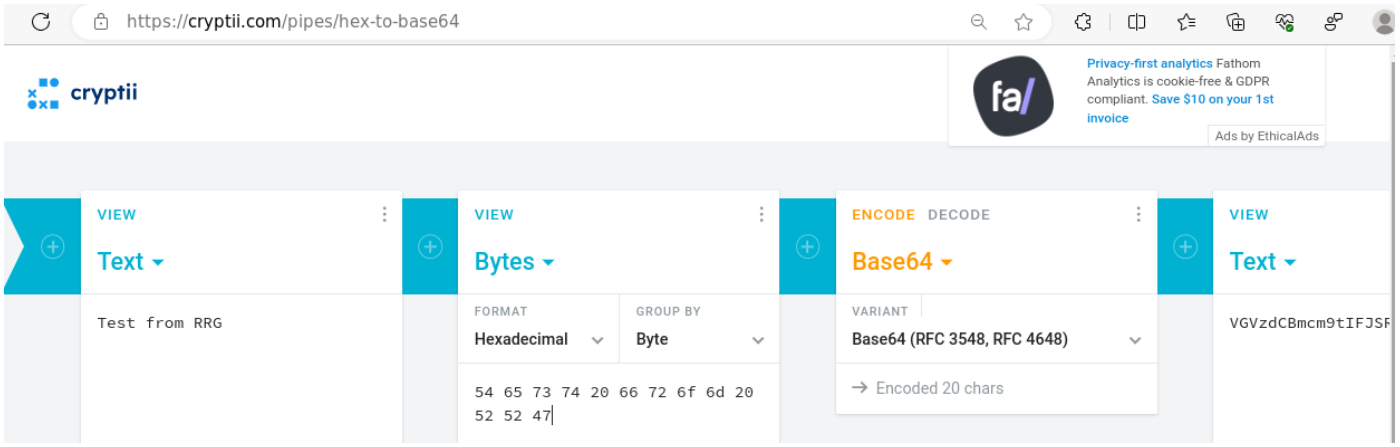
Result



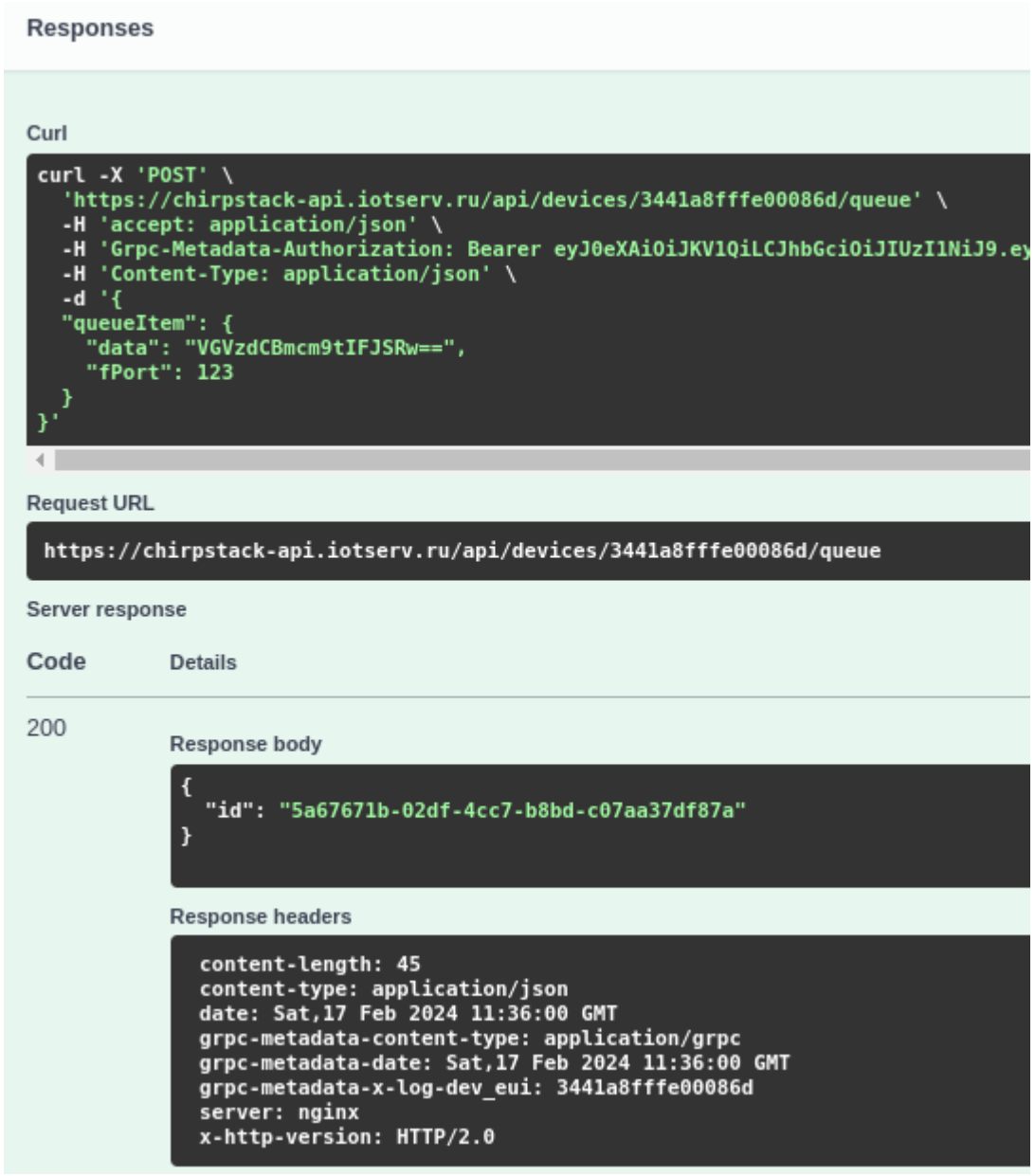
POST Queue to real device



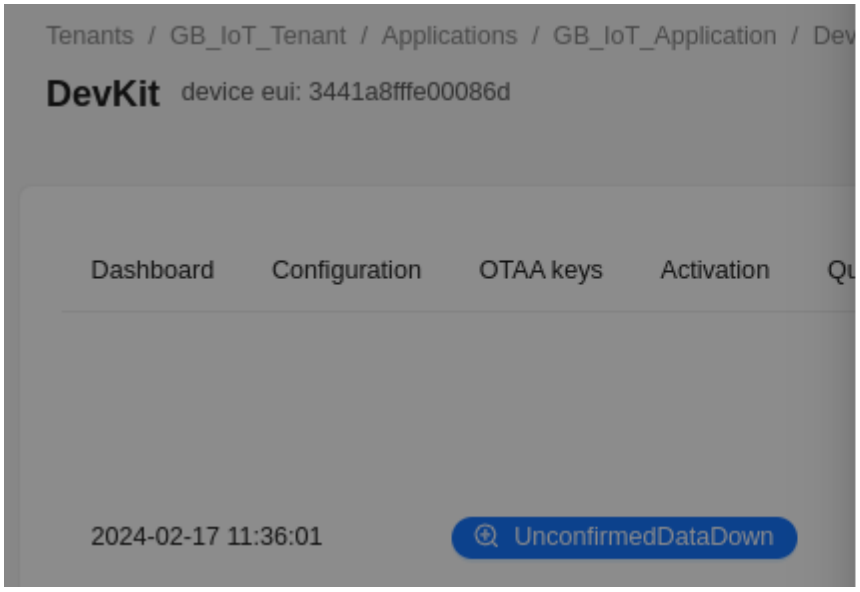
Data to Base64

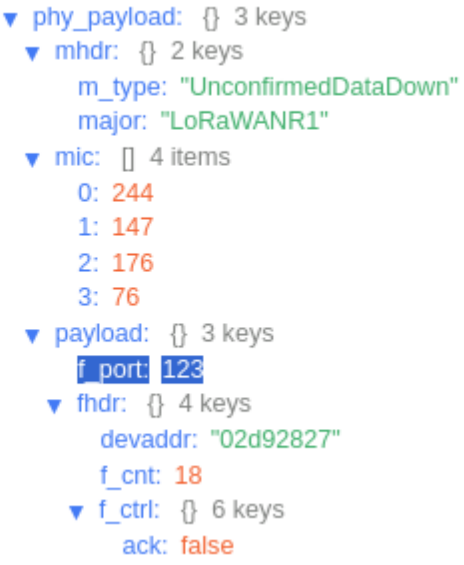


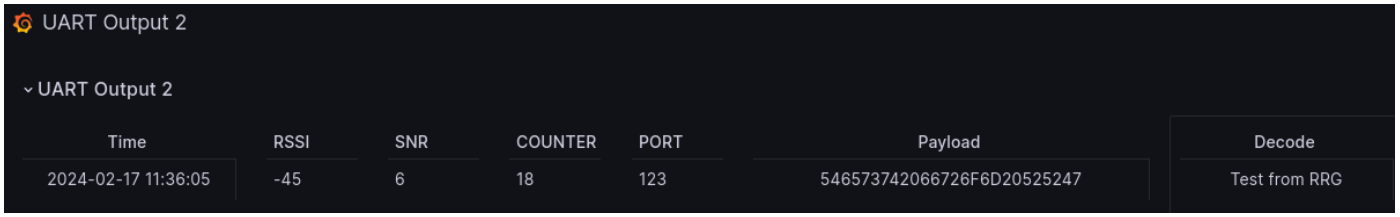
Responses



Result

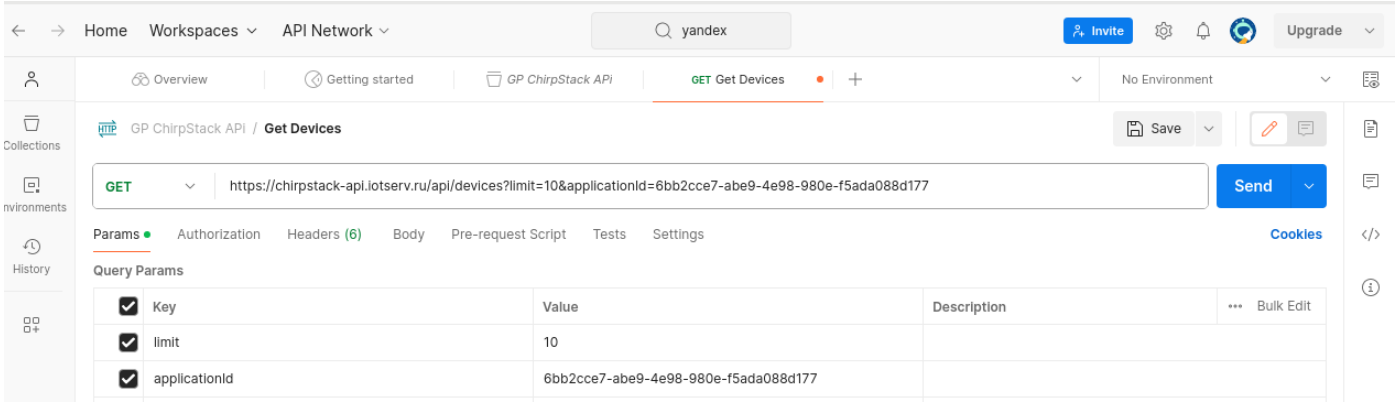




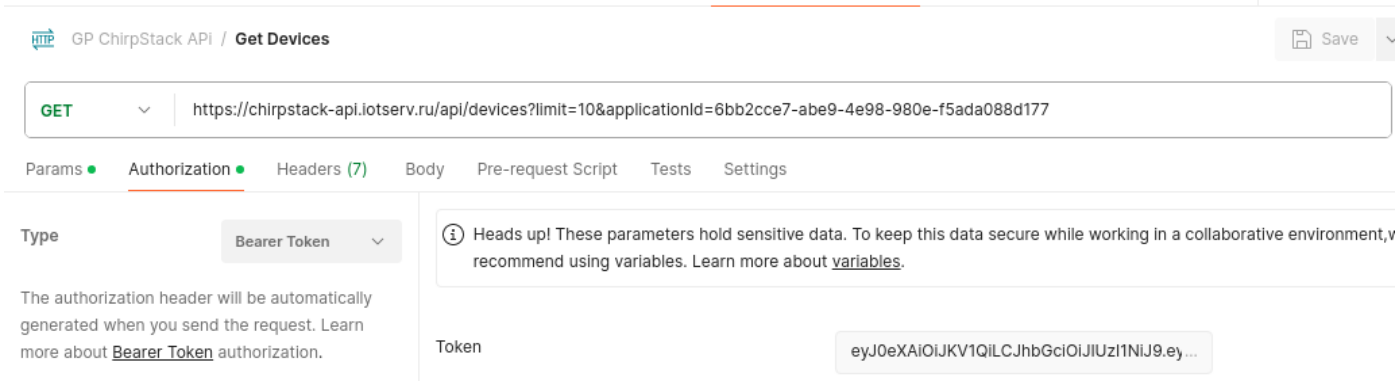


2. Postman

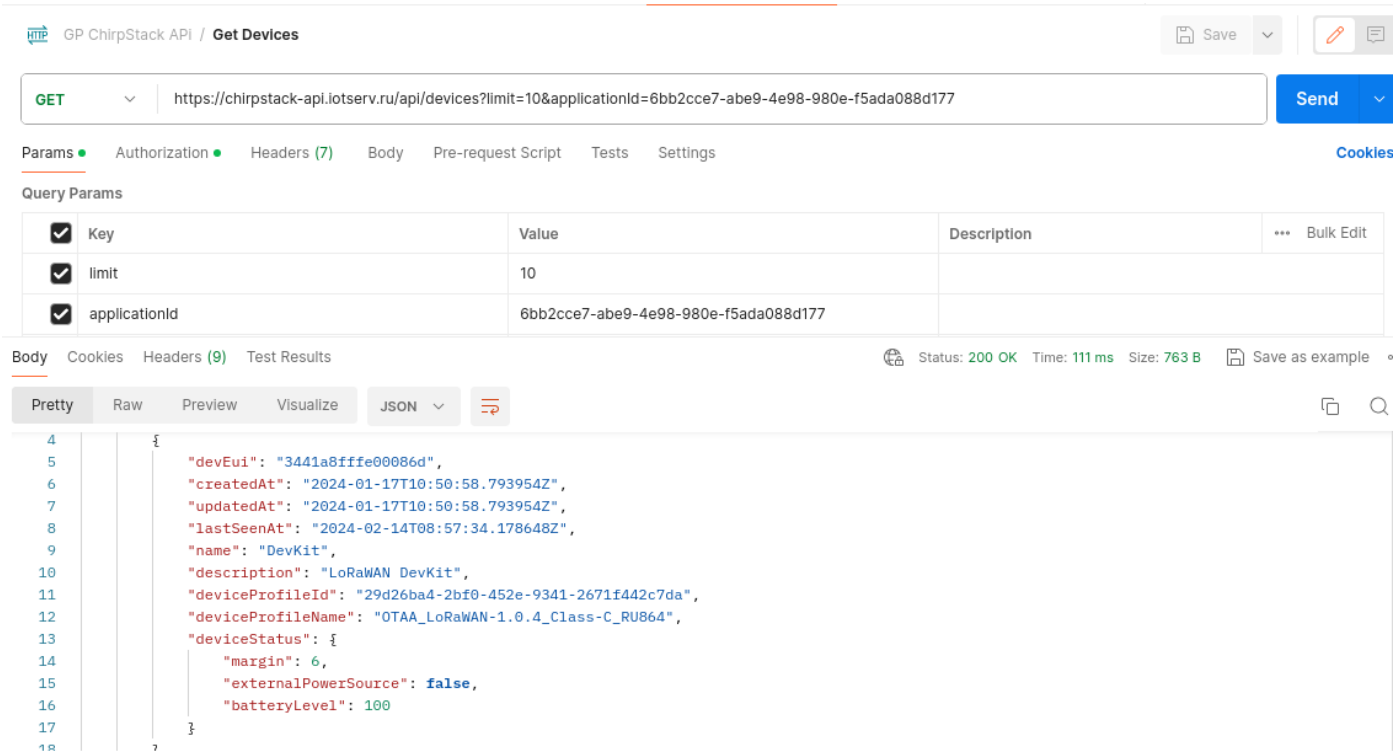
GET



Authorization



Save & Send

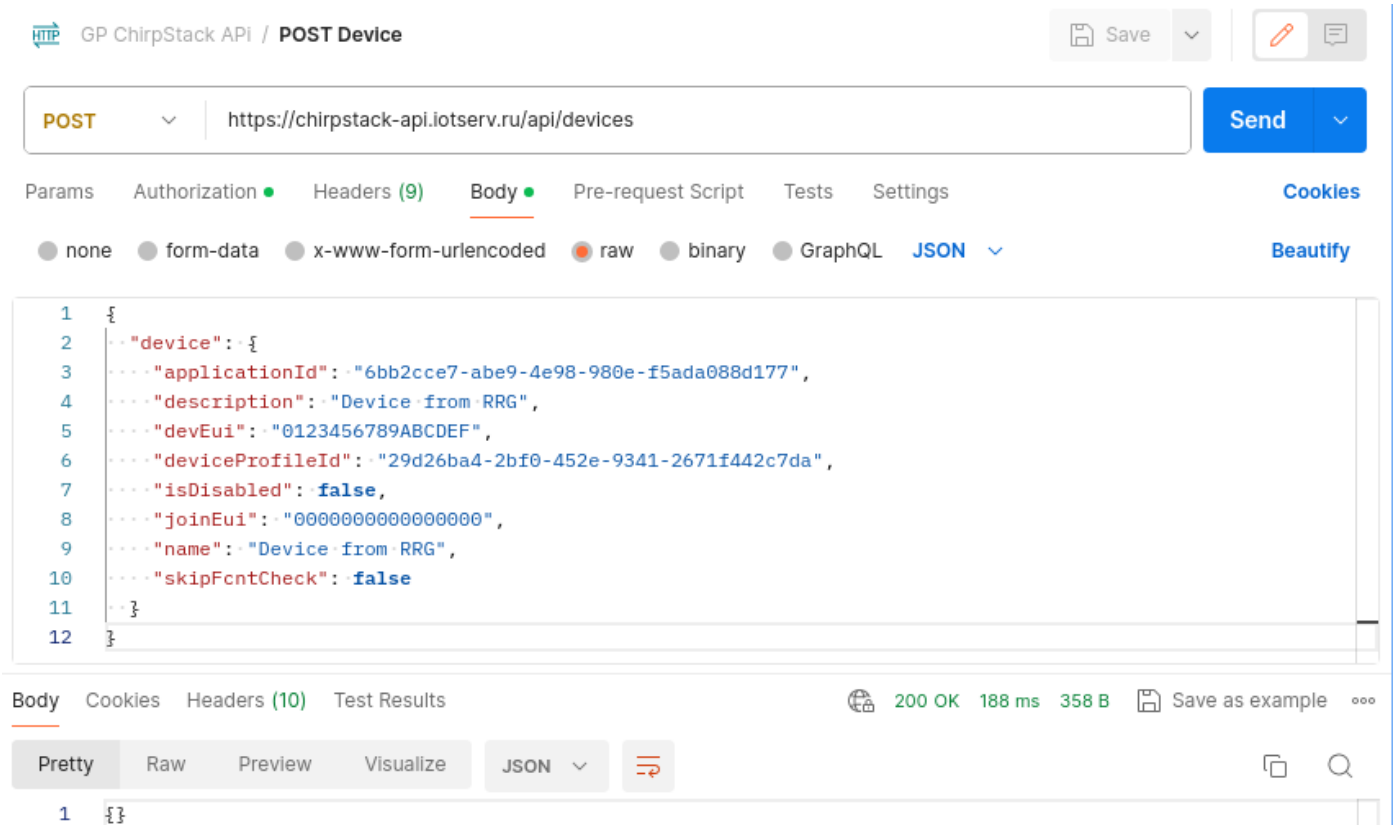


POST

Body

RAW -> JSON

Save & Send

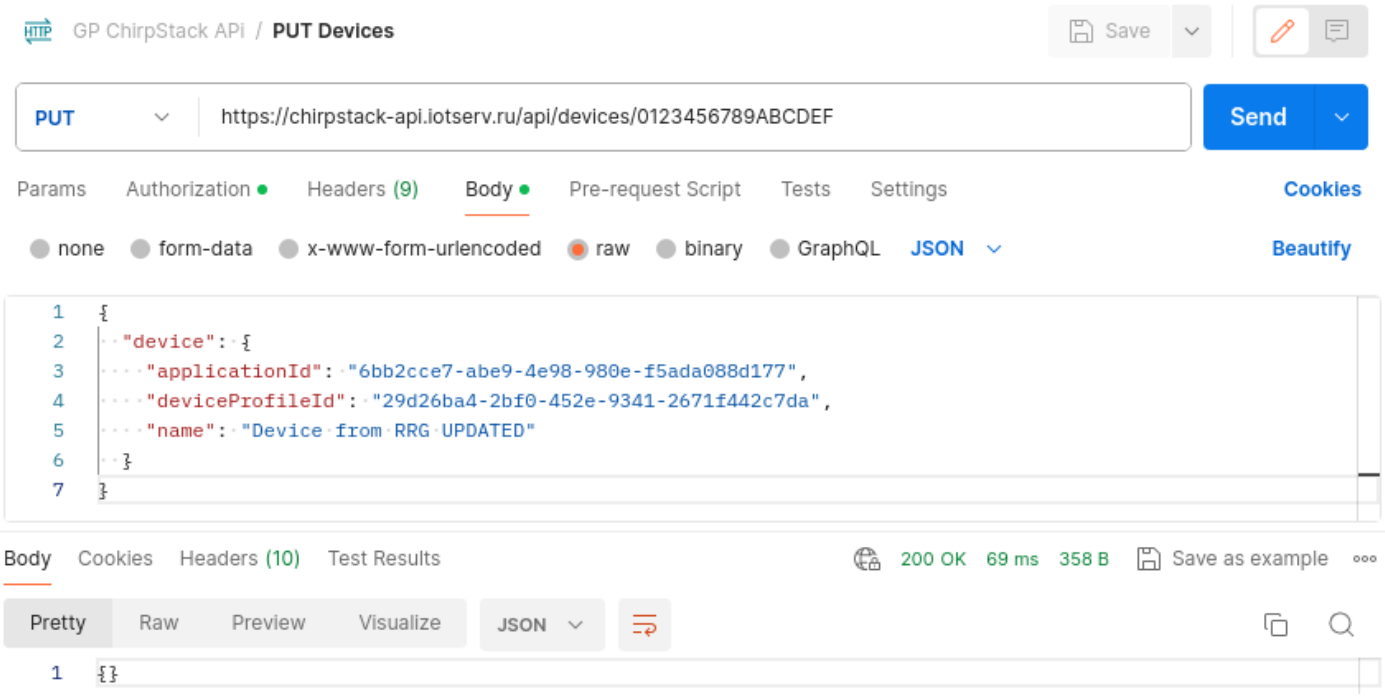


PUT

Body

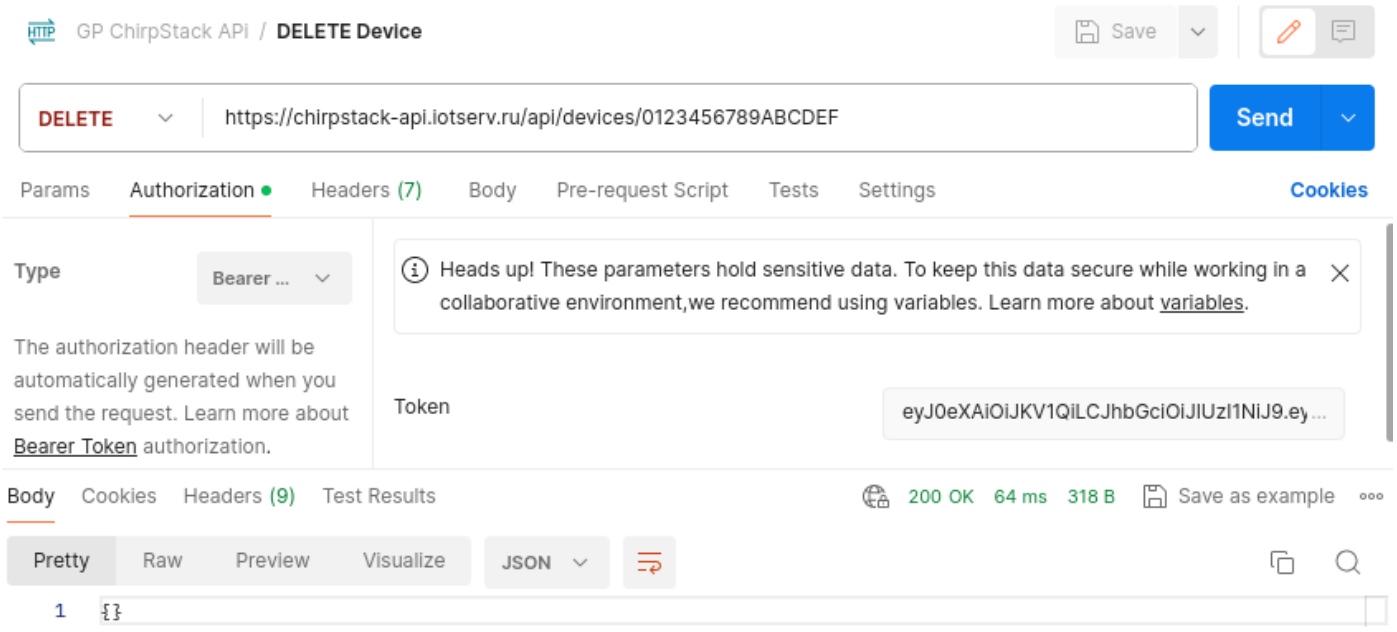
RAW -> JSON

Save & Send



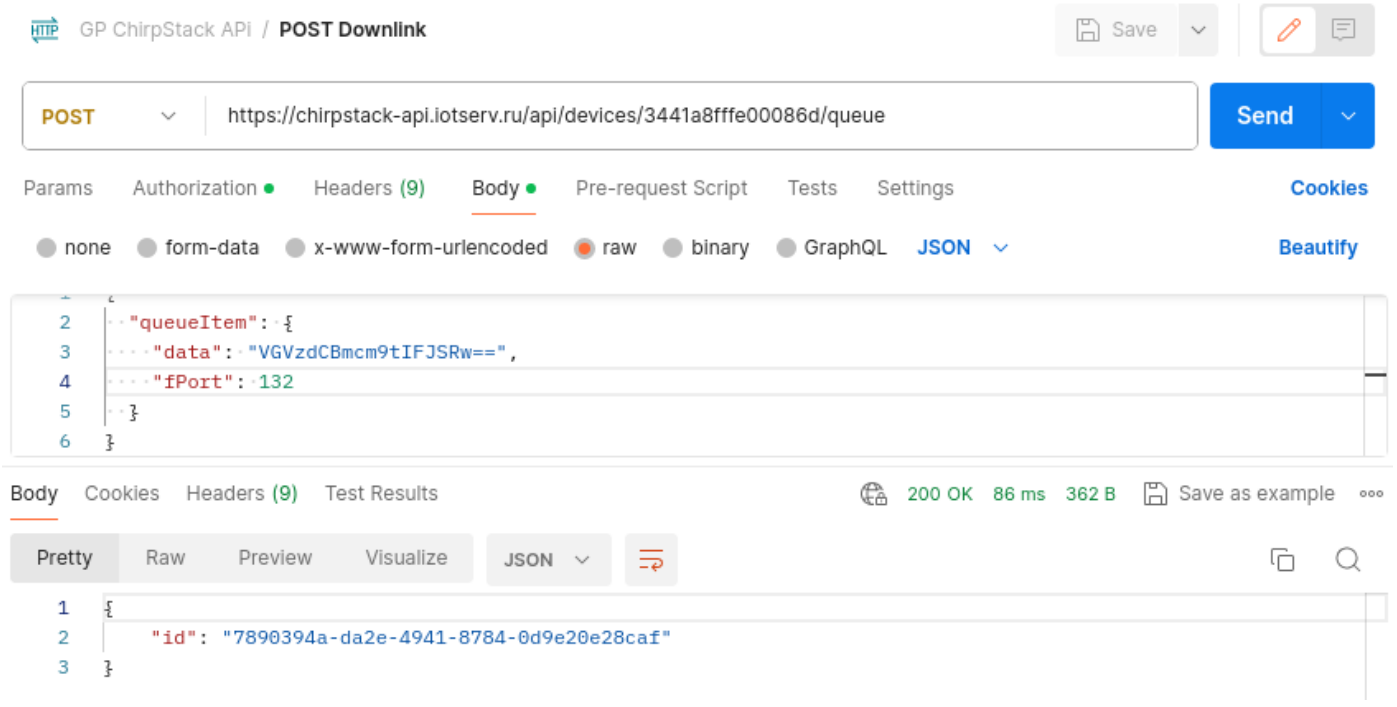
DELETE

Save & Send

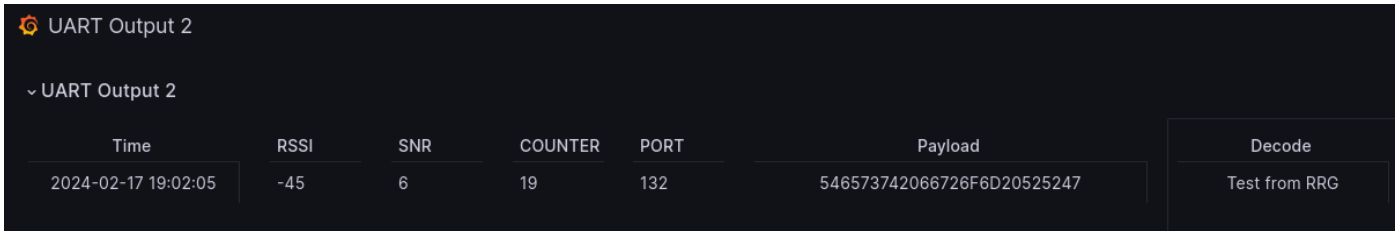


POST Downlink

Save & Send



Result in Grafana



3. Python

Создание 10 устройств

```
import requests
import json

token="eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJjaGlycHN0YWNrIiwiaXNz
IjoiY2hpcnBzdGFjayIsInN1YiI6ImMxMzk5MGRkLTMyN2ItNGZjYy1iZTU3LWQzNWVmZTY1MzI1
MSIsInR5cCI6ImtleSJ9.irLcZ138B8-UTrYhWZG-gFXwvsDMheqz8uiozaOyGzs"
url = 'https://chirpstack-api.iotserv.ru/api/devices'
applicationId = "6bb2cce7-abe9-4e98-980e-f5ada088d177"
deviceProfileId = "29d26ba4-2bf0-452e-9341-2671f442c7da"
devQuantity = 10
devicesList = [f"{i:02d}23456789ABCDEF" for i in range(devQuantity)]
```

```
def create(url, token, applicationId, deviceProfileId, name, devEui):
    headers = {
        'accept': 'application/json',
        'Grpc-Metadata-Authorization': f'Bearer {token}',
        'Content-Type': 'application/json',
    }
    data = {
        "device": {
            "applicationId": applicationId,
            "description": name,
            "devEui": devEui,
            "deviceProfileId": deviceProfileId,
            "isDisabled": False,
            "joinEui": "0000000000000000",
            "name": name,
            "skipFcntCheck": False
        }
    }
    return requests.post(url, json=data, headers=headers)


for i in range(len(devicesList)):
  print(create(url, token, applicationId, deviceProfileId, f"RRG Device
{i:02d}", devicesList[i]))
```

Запрос информации о 10 устройствах

```
def info(url, token, applicationId, limits):
    headers = {
        'accept': 'application/json',
        'Grpc-Metadata-Authorization': f'Bearer {token}',
    }
    params = {
        'limit': limits,
        'applicationId': applicationId
    }
    response = requests.get(url, headers=headers, params=params)
    return response.json()

print(json.dumps(info(url, token, applicationId, devQuantity+1), indent=4))
```

Обновить имена 10 устройств

```
def update(url, token, applicationId, deviceProfileId, newName, devEui):
    headers = {
        'accept': 'application/json',
        'Grpc-Metadata-Authorization': f'Bearer {token}',
        'Content-Type': 'application/json',
    }
    data = {
        "device": {
            "applicationId": applicationId,
            "deviceProfileId": deviceProfileId,
            "name": newName,
            "description": newName
        }
    }
    put_url = f"{url}/{devEui}"
    return requests.put(put_url, json=data, headers=headers)

for i in range(len(devicesList)):
  print(update(url, token, applicationId, deviceProfileId, f"RRG Device
{i:02d} UPDATED", devicesList[i]))

print(json.dumps(info(url, token, applicationId, devQuantity+1), indent=4))
```

Удаление 10 устройств

```
def delete(url, token, devEui):
    headers = {
        'accept': 'application/json',
        'Grpc-Metadata-Authorization': f'Bearer {token}',
    }
    return requests.delete(f"{url}/{devEui}", headers=headers)


for i in range(len(devicesList)):
  print(delete(url, token, devicesList[i]))
```