

Распределенные системы

ДЗ 04

Выполнил: Зеленин Е.В.

В качестве домашнего задания требуется поработать с REST API и выполнить ряд действий в SWAGGER, POSTMAN, сгенерировать WEB Token.

Сгенерируем WEB Token и посмотрим его содержимое:

Tenants / Dev_IoT_Tenant / API Keys / Add

Add API key

Use the following API token when making API requests. This token can be revoked at any time by deleting it. Please note that this token can only be retrieved once:

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJjaGlicHN0YWNrIiwiaXNzIjoiy2hpcnBzdGFjayIsInN1YiI6IjRjZjAyMGNhLTUyODEtNGUyNy1iYTc5LTQ0ZWVjYjRmYjZmYiIsInR5cCI6ImlleSJ9.t6X25tpMq78pQT72-M9EKTW7mgD67nC8aezdZVW1jcE

Back

JWT Decoder JWT Encoder

Paste a JWT below that you'd like to decode, validate, and verify.

Generate example

ENCODED VALUE

JSON WEB TOKEN (JWT) COPY CLEAR

Valid JWT

Invalid Signature

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJjaGlicHN0YWNrIiwiaXNzIjoiy2hpcnBzdGFjayIsInN1YiI6IjRjZjAyMGNhLTUyODEtNGUyNy1iYTc5LTQ0ZWVjYjRmYjZmYiIsInR5cCI6ImlleSJ9.t6X25tpMq78pQT72-M9EKTW7mgD67nC8aezdZVW1jcE

DECODED HEADER

JSON CLAIMS TABLE COPY ↗

{
 "typ": "JWT",
 "alg": "HS256"
}

DECODED PAYLOAD

JSON CLAIMS TABLE COPY ↗

{
 "aud": "chirpstack",
 "iss": "chirpstack",
 "sub": "4cf020ca-5281-4e27-ba79-44eecb4fb6fb",
 "typ": "key"
}

JWT SIGNATURE VERIFICATION (OPTIONAL)

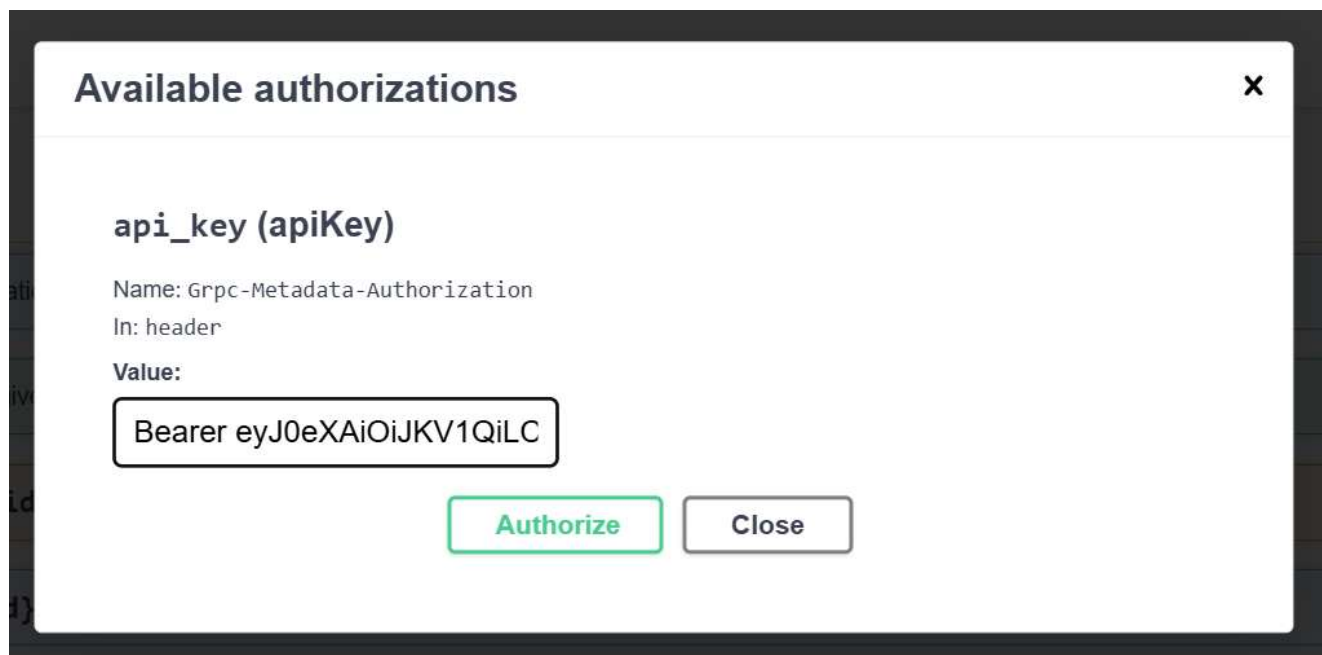
Enter the secret used to sign the JWT below:

SECRET COPY CLEAR

signature verification failed

a-string-secret-at-least-256-bits-long

Подключимся к WEB-сервисам с помощью полученного токена.



Теперь выполним несколько HTTP запросов через SWAGGER чтобы поработать с REST API.

GET запрос:

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. If not set, it will be treated as 0, and the response will only return the total_count. <div>limit</div>
offset integer(\$int64) (query)	Offset in the result-set (for pagination). <div>offset</div>
search string (query)	If set, the given string will be used to search on name (optional). <div>search</div>
applicationId string (query)	Application ID (UUID) to filter devices on. <div>applicationId</div>
multicastGroupId string (query)	Multicast-group ID (UUID) to filter devices on. <div>multicastGroupId</div>
orderBy string (query)	If set, the given value will be used to sort by (optional). Available values : NAME, DEV_EUI, LAST_SEEN_AT, DEVICE_PROFILE_NAME Default value : NAME <div></div>

Curl

```
curl -X 'GET' \
  'https://chirpstack-api.ioterv.ru/api/devices?limit=5&applicationId=9cfc547c-ccfa-4aeb-a9f1-bc73aad24dae&orderBy=NAME' \
  -H 'accept: application/json' \
  -H 'Grpc-Metadata-Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWQiOiJjaGlycHN0Ym9rIiwiaXNzIjo1V2hpcnBzdGZjaWIsInN1YiI6IjRjZjAyMGNhLTUyODEtNGUyNy1lYTc5LTQ0ZWVjYjRmYjZmYiIsInR5cCI6Imltles3J9.
```

Request URL

https://chirpstack-api.ioterv.ru/api/devices?limit=5&applicationId=9cfc547c-ccfa-4aeb-a9f1-bc73aad24dae&orderBy=NAME

Server response

Code Details

200

Response body

```
{
  "totalCount": 1,
  "result": [
    {
      "devEui": "3441a8fffe0006d",
      "createdAt": "2025-04-09T18:36:59.623068Z",
      "updatedAt": "2025-04-09T18:36:59.623068Z",
      "lastSeenAt": "2025-05-30T16:40:45.912379Z",
      "name": "DevKit",
      "description": "DevKit",
      "deviceProfileId": "a40b2348-93d8-4c8e-85f7-f4dc78f669b5",
      "deviceProfileName": "OTAA_LoRaWAN-1.0.4_Class-C_RU864",
      "deviceStatus": {
        "margin": 5,
        "externalPowerSource": false,
        "batteryLevel": 100
      },
      "tags": {}
    }
  ]
}
```



Download

Response headers

```
content-length: 420
content-type: application/json
date: Fri, 30 May 2025 17:28:00 GMT
grpc-metadata-content-type: application/grpc
grpc-metadata-date: Fri, 30 May 2025 17:28:00 GMT
grpc-metadata-x-log-application_id: 9cfc547c-ccfa-4aeb-a9f1-bc73aad24dae
server: nginx
x-http-version: HTTP/2.0
```

Responses

Curl

```
curl -X 'GET' \
  'https://chirpstack-api.ioterv.ru/api/devices/1322f3a4FFAAC00' \
  -H 'accept: application/json' \
  -H 'Grpc-Metadata-Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWQiOiJjaGlycHN0Ym9rIiwiaXNzIjo1V2hpcnBzdGZjaWIsInN1YiI6IjRjZjAyMGNhLTUyODEtNGUyNy1lYTc5LTQ0ZWVjYjRmYjZmYiIsInR5cCI6Imltles3J9.
```

Request URL

https://chirpstack-api.ioterv.ru/api/devices/1322f3a4FFAAC00

Server response

Code Details

200

Response body

```
{
  "device": {
    "devEui": "1322f3a4ffaacc00",
    "name": "Test Device from ZEV-031",
    "description": "Test Device from ZEV-031",
    "applicationId": "9cfc547c-ccfa-4aeb-a9f1-bc73aad24dae",
    "deviceProfileId": "a40b2348-93d8-4c8e-85f7-f4dc78f669b5",
    "skipFcmtCheck": false,
    "isDisabled": false,
    "variables": {},
    "tags": {},
    "joinEui": "0000000000000000"
  },
  "createdAt": "2025-05-30T17:41:15.184792Z",
  "updatedAt": "2025-05-30T17:41:15.184792Z",
  "lastSeenAt": null,
  "deviceStatus": null,
  "classEnabled": "CLASS_A"
}
```



Download

Response headers

```
content-length: 469
content-type: application/json
```

POST запрос:

body ★ required

object

(body)

Edit Value | Model

```
{
  "device": {
    "applicationId": "9cfc547c-ccfa-4aeb-a9f1-bc73aad24dae",
    "description": "Test Device From ZEV-031",
    "devEui": "af12fa45ce45bc45",
    "deviceProfileId": "a40b2348-93d8-4c8e-85f7-f4dc78f669b5",
    "isDisabled": false,
    "joinEui": "0000000000000000",
    "name": "Test Device From ZEV-031",
    "skipFcmtCheck": false,
  }
}
```

Cancel

Parameter content type

application/json

Code	Description
200	<div>A successful response.</div> <div>Example Value Model</div> <div><pre>{}</pre></div>
default	<div>An unexpected error response.</div> <div>Example Value Model</div> <div><pre>{ "code": 0, "details": [{ "@type": "string", "additionalProp1": "string", "additionalProp2": "string", "additionalProp3": "string" }], "message": "string" }</pre></div>

Допустим несколько ошибок, чтобы посмотреть какие будут ответы:

Curl

```
curl -X 'POST' \
'https://chirpstack-api.ioterv.ru/api/devices' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "device": {
    "applicationId": "9cfc547c-ccfa-4aeb-a9f1-bc73aad24dae",
    "description": "Test Device from ZEV-031",
    "devEui": "1s22f3a4FFAAC00",
    "deviceProfileId": "e40b2348-93d8-4c8e-85f7-f4dc78f669b5",
    "isDisabled": false,
    "joinEui": "0000000000000000",
    "name": "Test Device from ZEV-031",
    "skipFcmtCheck": false
  }
}'
```

Request URL

https://chirpstack-api.ioterv.ru/api/devices

Server response

Code

Details

500

Error: response status is 500

Response body

```
{
  "code": 13,
  "message": "Invalid character 's' at position 1",
  "details": []
}
```

Download

Response headers

```
content-length: 72
content-type: application/json
date: Fri,30 May 2025 17:40:05 GMT
server: nginx
```

Responses

Response content type application/json

Curl

```
curl -X 'POST' \
'https://chirpstack-api.ioterv.ru/api/devices' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "device": {
    "applicationId": "9cfc547c-ccfa-4aeb-a9f1-bc73aad24dae",
    "description": "Test Device from ZEV-031",
    "devEui": "1322f3a4FFAAC00",
    "deviceProfileId": "e40b2348-93d8-4c8e-85f7-f4dc78f669b5",
    "isDisabled": false,
    "joinEui": "0000000000000000",
    "name": "Test Device from ZEV-031",
    "skipFcmtCheck": false
  }
}'
```

Request URL

https://chirpstack-api.ioterv.ru/api/devices

Server response

Code

Details

401

Error: response status is 401

Response body

```
{
  "code": 16,
  "message": "no authorization provided",
  "details": []
}
```

Download

Response headers

В случае успеха, получаем пустой ответ {}

Responses Response content type application/json

Curl

```
curl -X 'POST' \
'https://chirpstack-api.ioterv.ru/api/devices' \
-H 'accept: application/json' \
-H 'Grpc-Metadata-Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ3dWwQI0IjjaGlycHN0YWNrIiwiaXNzIjoIY2hpcnBzdGZjYyIsInN1YiI6IjRjZjAyMGNhLTUyODEtNGUyNy1iYTc5LTQ0ZWVjYjRmYjZmYiIsInR5cCI6ImtleS39.
-H 'Content-Type: application/json' \
-d '{
  "device": {
    "applicationId": "9cfc547c-ccfa-4aeb-a9f1-bc73aad24dae",
    "description": "Test Device from ZEV-031",
    "devEui": "1322f3a4FFAAC00",
    "deviceProfileId": "a40b2348-93d8-4c8e-85f7-f4dc78f669b5",
    "isDisabled": false,
    "joinEui": "0000000000000000",
    "name": "Test Device from ZEV-031",
    "skipFcntCheck": false
  }
}
```

Request URL

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

Server response

Code	Details
200	<div>Response body<pre>{}</pre><div>Download</div></div> <div>Response headers<pre>content-length: 2 content-type: application/json date: Fri,30 May 2025 17:41:15 GMT grpc-metadata-content-type: application/grpc grpc-metadata-date: Fri,30 May 2025 17:41:15 GMT grpc-metadata-x-log-dev_eui: 1322f3a4FFAAC00 grpc-metadata-x-log-is_disabled: false server: nginx x-http-version: HTTP/2.0</pre></div>

Responses

Отправим запрос на изменение с помощью PUT

Responses

Curl

```
curl -X 'PUT' \
'https://chirpstack-api.ioterv.ru/api/devices/1322f3a4FFAAC00' \
-H 'accept: application/json' \
-H 'Grpc-Metadata-Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ3dWwQI0IjjaGlycHN0YWNrIiwiaXNzIjoIY2hpcnBzdGZjYyIsInN1YiI6IjRjZjAyMGNhLTUyODEtNGUyNy1iYTc5LTQ0ZWVjYjRmYjZmYiIsInR5cCI6ImtleS39.
-H 'Content-Type: application/json' \
-d '{
  "device": {
    "applicationId": "9cfc547c-ccfa-4aeb-a9f1-bc73aad24dae",
    "deviceProfileId": "a40b2348-93d8-4c8e-85f7-f4dc78f669b5",
    "name": "Test Device from ZEV-031-UPDATED"
  }
}
```

Request URL

```
https://chirpstack-api.ioterv.ru/api/devices/1322f3a4FFAAC00
```

Server response

Code	Details
200	<div>Response body<pre>{}</pre><div>Download</div></div> <div>Response headers<pre>content-length: 2 content-type: application/json date: Fri,30 May 2025 17:44:28 GMT grpc-metadata-content-type: application/grpc grpc-metadata-date: Fri,30 May 2025 17:44:28 GMT grpc-metadata-x-log-dev_eui: 1322f3a4FFAAC00 grpc-metadata-x-log-is_disabled: false server: nginx x-http-version: HTTP/2.0</pre></div>

Responses

И запрос на удаление с помощью DELETE

1322f3a4FFAACCC00

Execute

Clear

Responses

Response content typeapplication/json▼

Curl

```
curl -X 'DELETE' \
'https://chirpstack-api.iotserve.ru/api/devices/1322f3a4FFAACCC00' \
-H 'accept: application/json' \
-H 'Grpc-Metadata-Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGU6IjY2hpbnBzdGFjayIsInN1YXI6ImZlcnRpbmkiLCJkaWQ6ImFkbGVldGUiLCJwcm9udDp1cy5kZW5pdC51dHViZXNjaW50eDEtNGUyNy11YTcsLTQ0ZWYyIjpmYjZmYiIsInR5ccCI6ImtleS19'.
⏮ ⏭ ⏯ 🔍 🗑
```

Request URL

```
https://chirpstack-api.iotserve.ru/api/devices/1322f3a4FFAACCC00
```

Server response

Code	Details
200	<div>Response body<pre>{}</pre></div> <div>Download</div>
	<div>Response headers<pre>content-length: 2 content-type: application/json date: Fri, 30 May 2025 17:47:20 GMT grpc-metadata-content-type: application/grpc grpc-metadata-date: Fri, 30 May 2025 17:47:20 GMT grpc-metadata-x-log-dev_uid: 1322f3a4FFAACCC00 server: nginx x-http-version: HTTP/2.0</pre></div>

Теперь с помощью POST запроса отправим данные с кодированным текстом на устройство.

Cryptii

local

VIEW

Text

Luke, I am your father.

VIEW

Bytes

FORMAT

Hexadecimal

GROUP BY

Byte

4c 75 6b 65 2c 20 49 20 61 6d
20 79 6f 75 72 20 66 61 74 68
65 72 2e

ENCODE

DECODE

Base64

VARIANT

Base64 (RFC 3548, RFC 4648)

→ Encoded 32 chars

VIEW

Text

THVrZSwgSSBhbS85b3VyIGZhdGhlc14=

object
(body)

```
{
  "queueItem": {
    "data": "THVrZSwgSSBhbS85b3VyIGZhdGhlc14=",
    "fPort": 127
  }
}
```

Cancel

Parameter content type
application/json

Execute

Clear

Responses

Response content type
application/json

Curl

```
curl -X 'POST' \
  'https://chirpstack-api.ioterv.ru/api/devices/3441a8ffe00086d/queue' \
  -H 'accept: application/json' \
  -H 'Grpc-Metadata-Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ3dWQ1O13JaGlycH0WbWVrIiwiaXNzIjo1Y2hpcnBzdGfjayIsInN1YiI6IjRzJjAyaG0hLTUyODRtNGUyNy11YTRcSLTQ0ZWVjYjRmYjZmYiIsInR5cCI6ImltlesJ9.' \
  -H 'Content-Type: application/json' \
  -d '{
    "queueItem": {
      "data": "THVrZSwgSSBhbS85b3VyIGZhdGhlc14=",
      "fPort": 127
    }
  }'
```

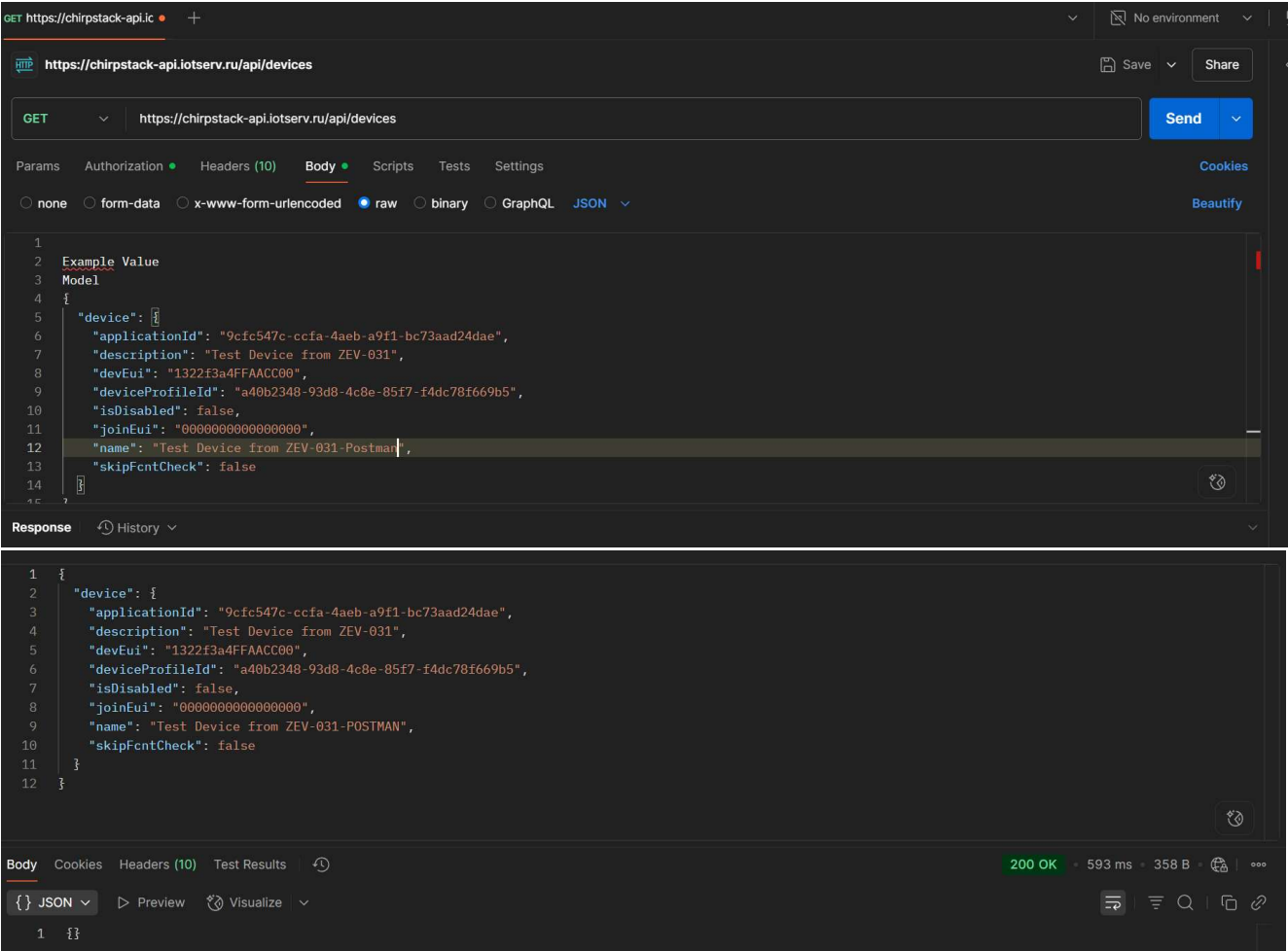
Request URL
https://chirpstack-api.ioterv.ru/api/devices/3441a8ffe00086d/queue

Как видим, все работает успешно:

Time	RSSI	SNR	COUNTER	PORT	Payload	Decode
2025-05-30 21:02:00	-62	7	5	255	546865206265737420646179	The best day
2025-05-30 21:01:56	-62	7	3	127	546657374206D6573736167652066726F6D204756492D30333121	Test message from GVI-03!
2025-05-30 20:59:44					4C7568652C204920616D20796F7572206661746865722E	Luke, I am your father.

Выполним аналогичные операции с помощью POSTMAN:

GET запрос:



<input type="checkbox"/>	Last seen	DevEUI	Name	Device profile
<input type="checkbox"/>	2025-05-30 20:48:58	3441a8ffe00086d	DevKit	OTAA_LoRaWAN-1.0.4_Class-C_RU864
<input type="checkbox"/>	Never	11223344ffaacc00	Test Device from GVI-031	OTAA_LoRaWAN-1.0.4_Class-C_RU864
<input type="checkbox"/>	Never	1322f3a4ffaacc00	Test Device from ZEV-031-POSTMAN	OTAA_LoRaWAN-1.0.4_Class-C_RU864

DELETE запрос:

POST https://chirpstack-api.i

DEL https://chirpstack-api.ic

+

▼

No environment ▼

https://chirpstack-api.ioterv.ru/api/devices/1322f3a4FFAAC00

Save ▼

Share

DELETE ▼

https://chirpstack-api.ioterv.ru/api/devices/1322f3a4FFAAC00

Send ▼

Params

Authorization ●

Headers (8)

Body

Scripts

Tests

Settings

Cookies

☒ none

☐ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL

This request does not have a body

Body

Cookies

Headers (9)

Test Results

🕒

200 OK

30 ms

318 B

🌐

⋮

{ } JSON ▼

▶ Preview

🔗 Visualize ▼

🔍

📄

🔗

1

{ }

Теперь отправим закодированный текст на устройство:

VIEW
Text

The samurai has no goal, only path. Postman

VIEW
Bytes

FORMAT: Hexadecimal

GROUP BY: Byte

54 68 65 20 73 61 6d 75 72 61
69 20 68 61 73 20 6e 6f 20 67
6f 61 6c 2c 20 6f 6e 6c 79 20
70 61 74 68 2e 20 50 6f 73 74
6d 61 6e

ENCODE DECODE
Base64

VARIANT: Base64 (RFC 3548, RFC 4648)

→ Encoded 60 chars

VIEW
Text

VGhLIHNhbXVyYWkgZGFzIG5vIGdvYWwsIG9ubHkgcGF0aC4gUG9zdG1hbg==

POST https://chirpstack-api.iot...

POST https://chirpstack-api.iot...

POST https://chirpstack-api.iot...

https://chirpstack-api.iot...

Save Share

POST https://chirpstack-api.iot...

Send

Params Authorization Headers (10) Body Scripts Settings

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

1 {
2 "queueItem": {
3 "data": "VGhLIHNhbXVyYWkgZGFzIG5vIGdvYWwsIG9ubHkgcGF0aC4gUG9zdG1hbg==",
4 "fPort": 127
5 }
6 }

Body Cookies Headers (9) Test Results

200 OK 631 ms 362 B

{ JSON Preview Visualize

1 {
2 "id": "e82154d9-c6c0-4c31-99c7-d398ab61e540"
3 }

UART Output 2

Last 15 minutes Refresh

Time	RSSI	SNR	COUNTER	PORT	Payload	Decode
2025-05-30 21:49:46	-61	6	9	127	72616920686173206E6F20676F616C2C206F6E6C792070674682E	samurai has no goal, only path. Postr
2025-05-30 21:46:52	-63	6	8	127	506F73746D616E2E	Postman.

Все работает исправно.

Отчет закончен.