MINFARMTECH

AN INTRODUCTION TO THE MINFARM API

Version 1.2 November 2021

CONTENTS

1.0 INTRODUCTION

This guide is an introduction to the MinFarm API. It explains what is an API, and ways of interacting with one. The user interface of the MinFarm API is called the MinFarm Dashboard. The MinFarm API can be accessed using this user interface, and also using an application called Postman. This guide presumes some basic knowledge of Postman.

2.0 WHY USE AN API?

An API allows a program to interact with the MinFarm system.

3.0 WHAT IS AN API?

The acronym API stands for Application Programming Interface. An API allows different applications communicate with one another. The MinFarm API is located on the MinFarm Bridge Server. The MinFarm Bridger Server is hosted on the Cloud.

4.0 WHAT IS A REST API?

The MinFarm API is a REST API, sometimes called RESTful API. REST stands for Representational State Transfer. It describes your style of interaction with, and how the API is set up. It employs standard HTTP methods of interaction with functions such as GET (view), POST (create), PUT (edit) and DELETE (delete).

5.0 REQUEST AND RESPONSE

When interacting with the MinFarm API, a Request is first sent to the API, and a Response is received back. The transfer protocol HTTP is used during this transfer. The Request contains 4 parts:

- The start line: contains the HTTP version number, the method *e.g.* GET/POST/PUT/DELETE, folder location in the API, and any parameters.

- Header(s): the MinFarm API URL, Authorization Token, file type *e.g.* application/json.
- A blank line: Separates the header from the body.
- Body: Sometimes called payload. Contains further information associated with the Request. Not all Requests contain a Body.

The Response also contains 4 parts:

- The start line: contains the HTTP version number and a Status Code response, *e.g.* 200 or 201 indicates a successful Request, Status Code 403 indicates an error.
- Header(s): Date, file type *e.g.* application/json, *etc*.
- A blank line: Separates the header from the body.
- Body: What was requested from the API.

6.0 WHAT SECURITY OR AUTHENTICATION METHODS ARE USED WITH THE MINFARM API?

The MinFarm API uses OAuth 2.0 authentication. OAuth 2.0 does not share password information, but gives the user a token, known as a Personal Access Token (also called a Bearer Token), which allows access to a Scope, or access capability, within the API. The MinFarm API supports four types of Scope: Read, Write, Manager, and Firmware. Read is the most basic Scope allowing for Read only access, and no editing. The Write Scope allows for editing. While the Manager Scope contains full range of access. The Firmware scope is for more advanced access and will not be discussed in this document. Contact MinFarm for further information.

Each Personal Access Token is limited to one year and is in the form of a string which must be copied and saved somewhere. The Personal Access Token is generated via the MinFarm API user interface, *i.e.* in the MinFarm Dashboard. See Section 8.3 below on how to do this.

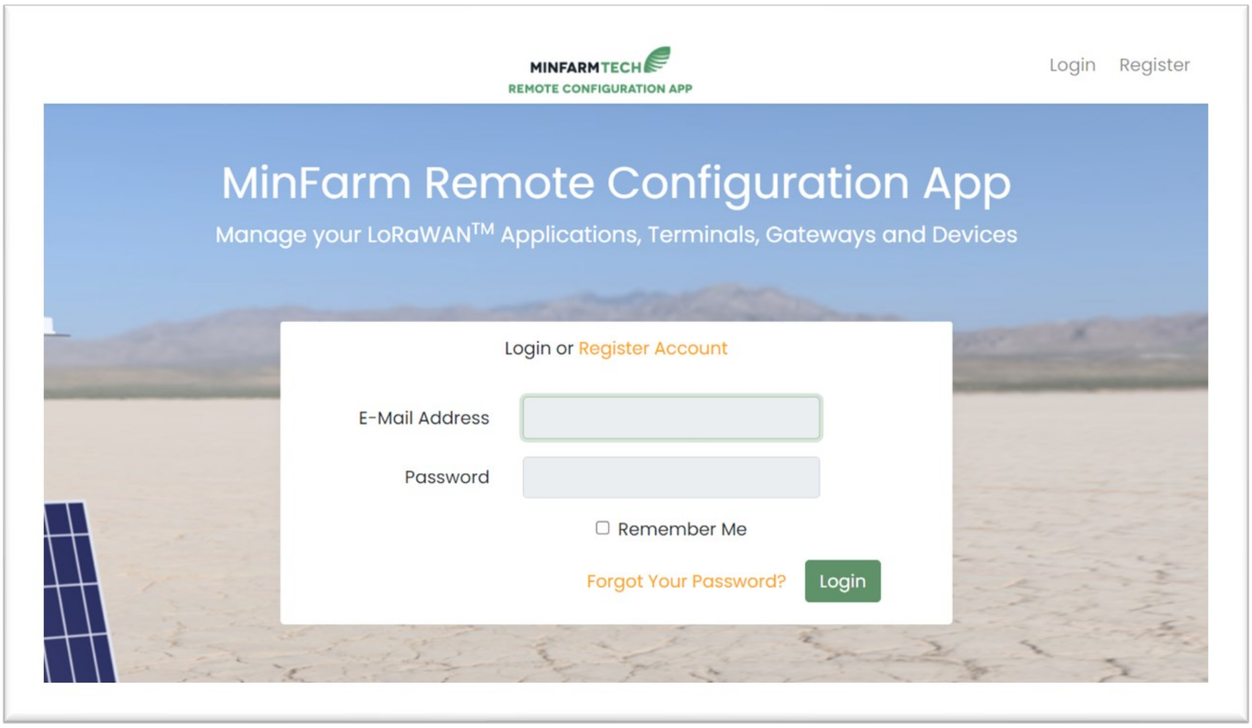
The MinFarm API also supports another type of OAuth 2.0 authentication called Authorization Code Grant. This is for more advanced access and will not be discussed in this white paper. Please contact MinFarm for further information.

7.0 STEP 1: THE MINFARM DASHBOARD

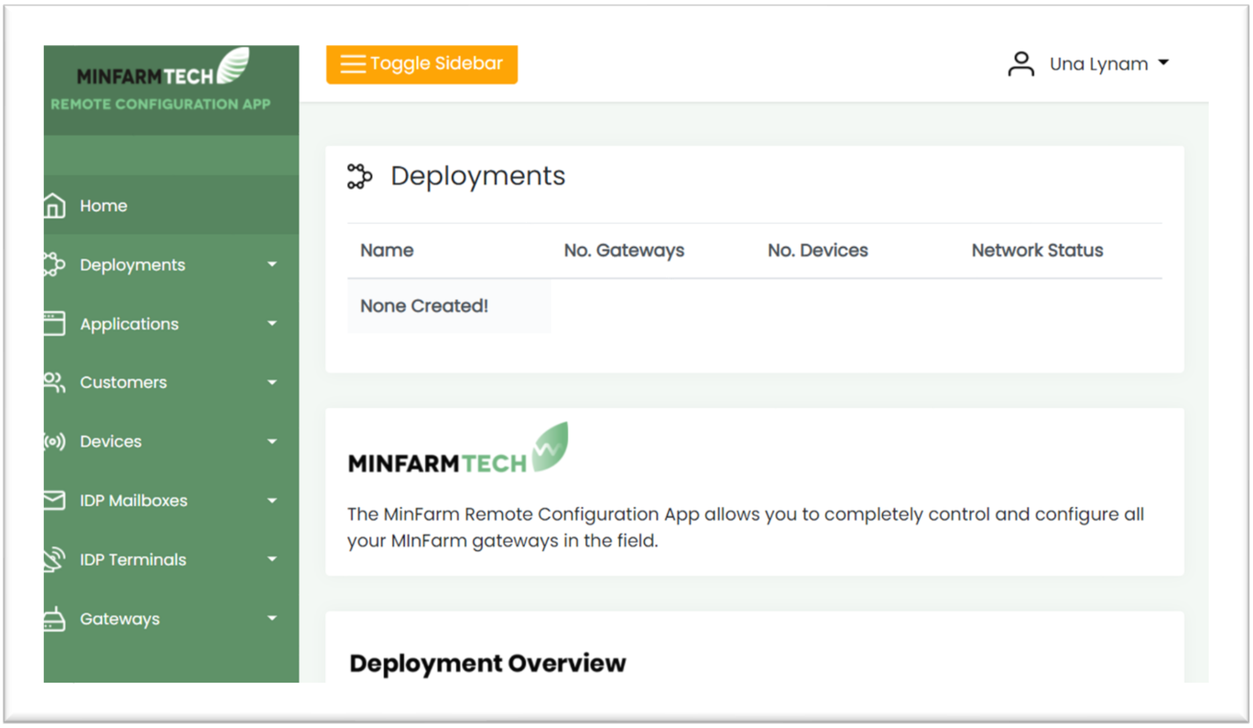
The user interface of the MinFarm API is called the MinFarm Dashboard. Accessing this user interface is the required first step in setting up access to your device / sensor. To complete this section, you should have to hand all necessary passwords and keys as provided by your satellite service provider and device / sensor supplier. The user creates a Deployment by following the steps below:

7.1 REGISTER

Go to the MinFarm Dashboard and select **<Register Account>**.


MINFARMTECH
REMOTE CONFIGURATION APP
Login
Register
MinFarm Remote Configuration App
Manage your LoRaWAN™ Applications, Terminals, Gateways and Devices
Login or Register Account
E-Mail Address
Password
Remember Me
Forgot Your Password?
Login

The homepage is displayed.



Deployments

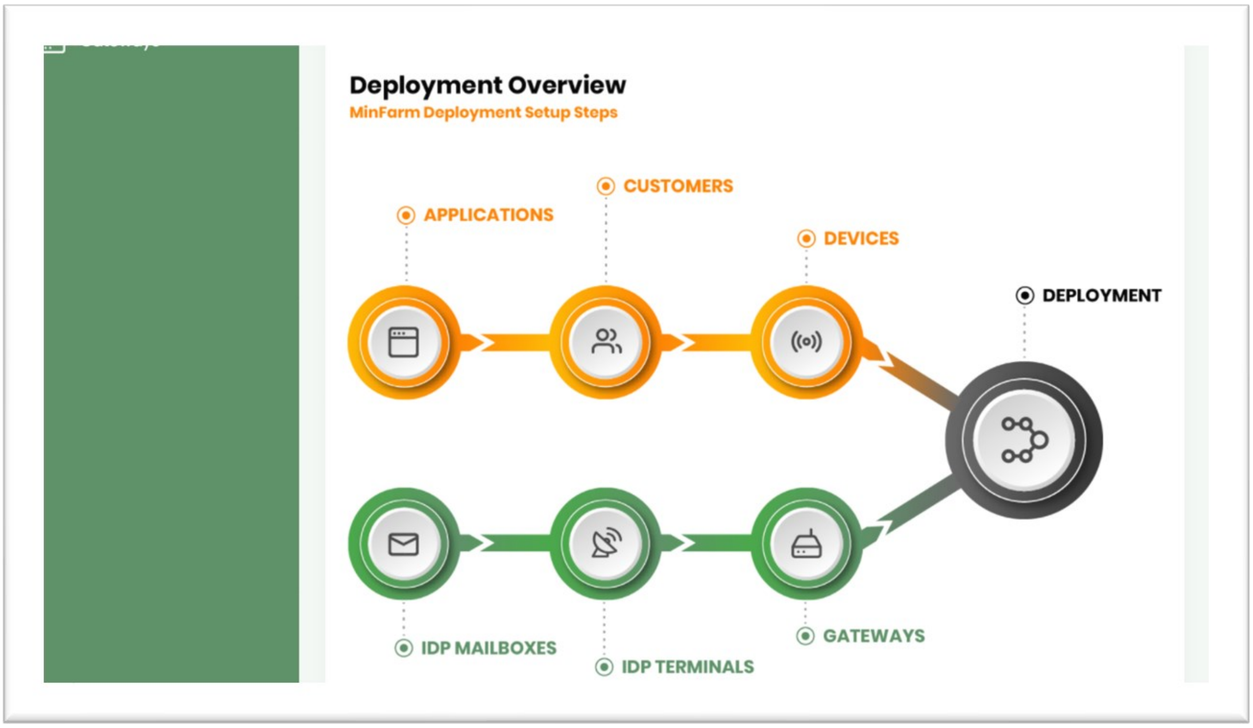
Name	No. Gateways	No. Devices	Network Status
None Created!			

MINFARMTECH

The MinFarm Remote Configuration App allows you to completely control and configure all your MInFarm gateways in the field.

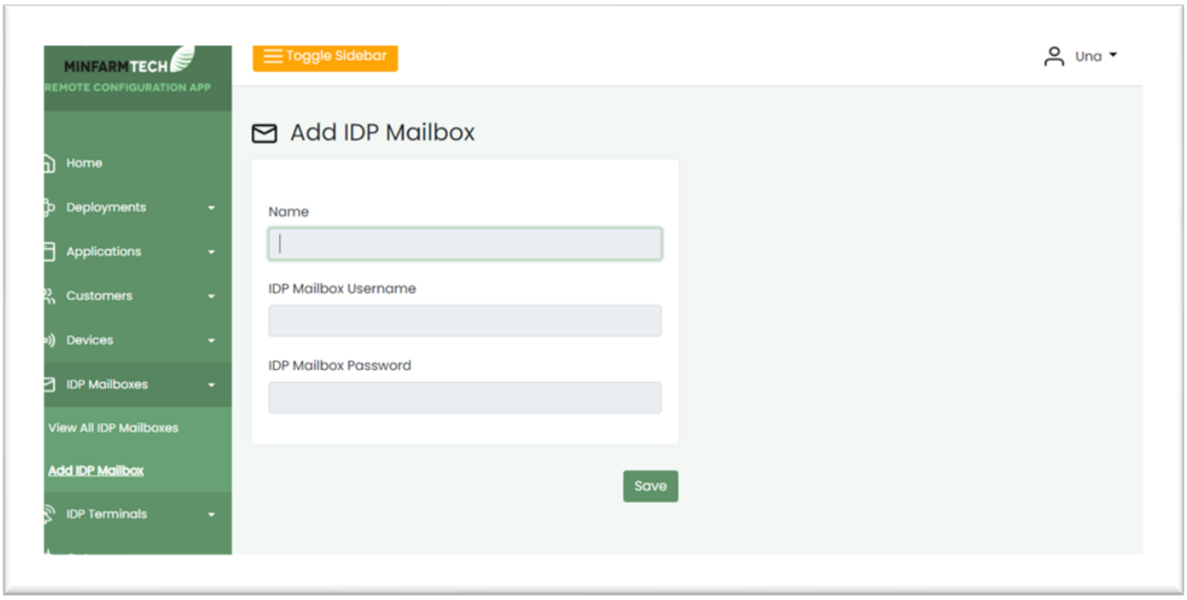
Deployment Overview

The various objects required in the Deployment are shown in the schematic below.



7.2 CREATE A DEPLOYMENT

Add an IDP Mailbox by selecting **<Add IDP Mailbox>** in the drop-down menu on the left-hand side. Enter the following: IDP Mailbox Username, IDP Mailbox Password. Select **<Save>**.



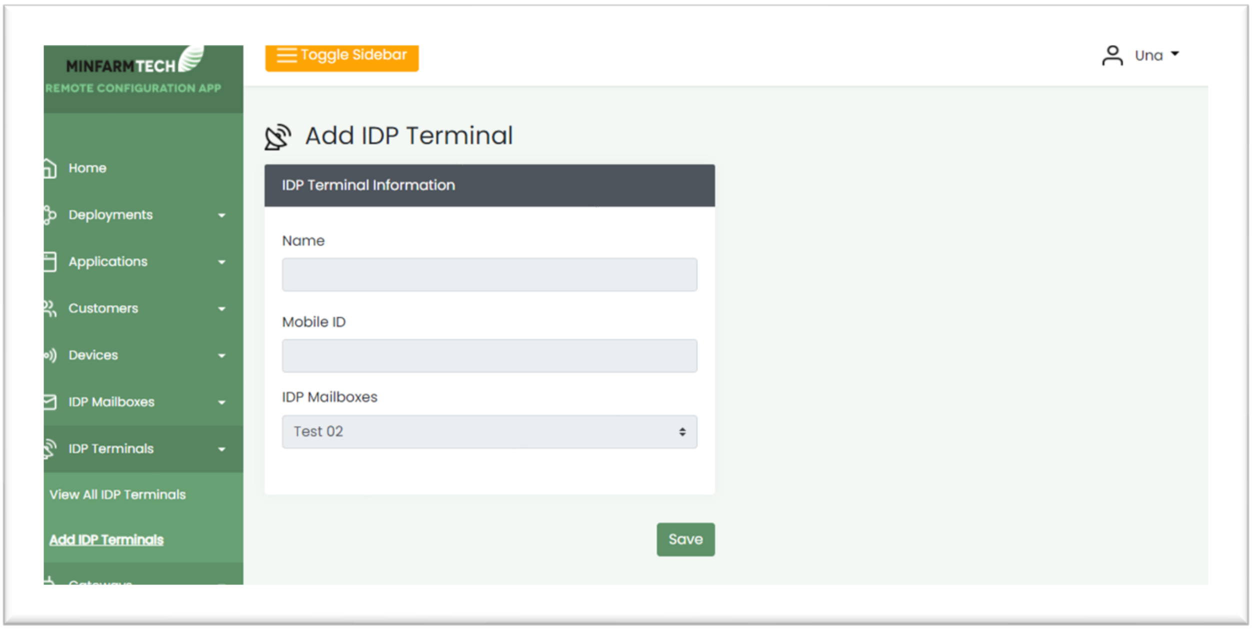
Add IDP Mailbox

Name

IDP Mailbox Username

IDP Mailbox Password

Add an IDP Terminal by selecting **<Add IDP Terminal>**. Enter the following: Name, Mobile ID, and the IDP Mailbox you have created. Select **<Save>**.



Add IDP Terminal

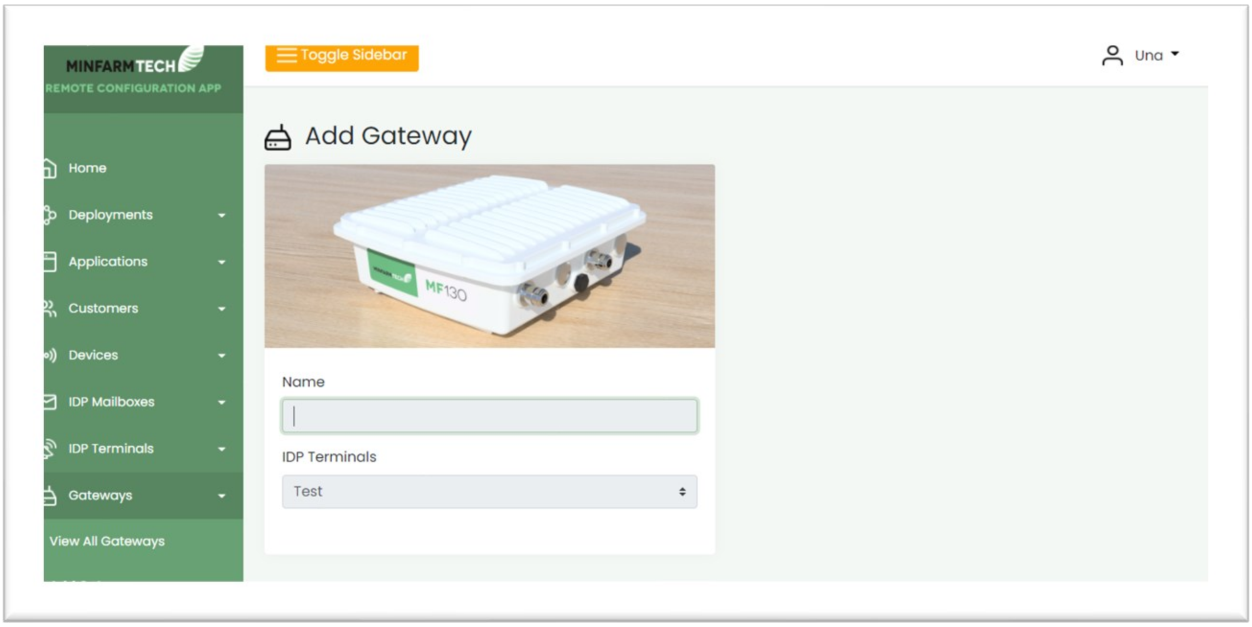
IDP Terminal Information

Name

Mobile ID

IDP Mailboxes: Test 02

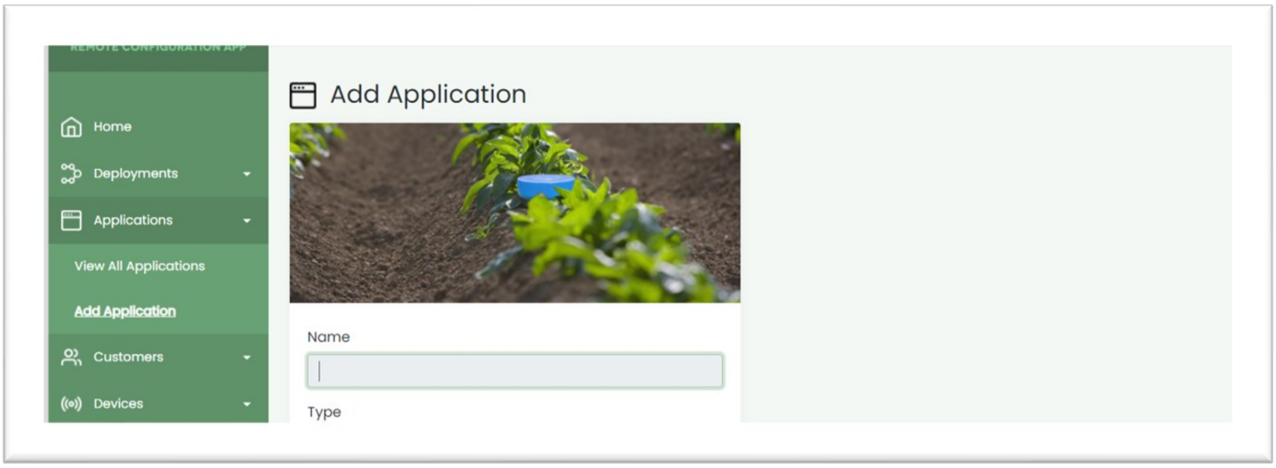
Add a Gateway by selecting **<Add Gateway>**. Select the IDP Terminal that you have created. Select **<Save>**.



Name

IDP Terminals: Test

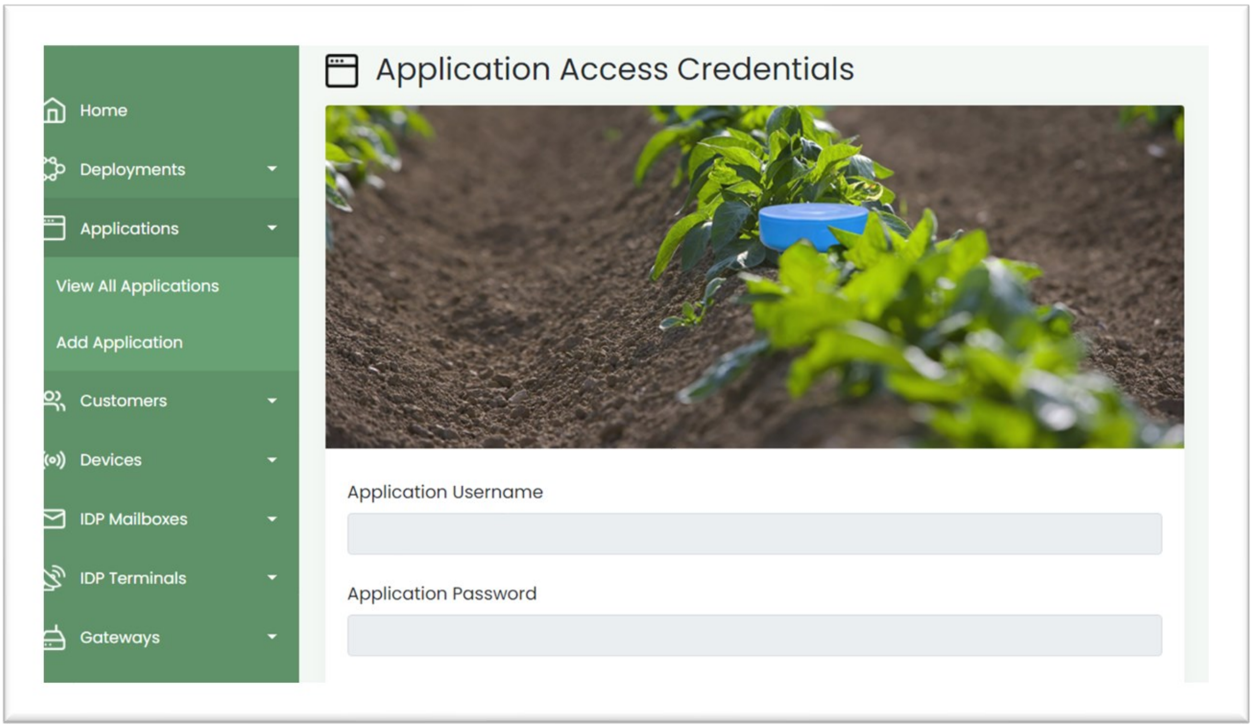
Add an Application by selecting **<Add Application>**. Adding an Application allows you decide where to forward the data to. Select your particular device / sensor in the **<Type>** drop-down menu.



Name

Type

Select **<Next>** and you are prompted to enter the Application Username and Application Password. These credentials have been given to you by the device / sensor provider. Select **<Save>**.



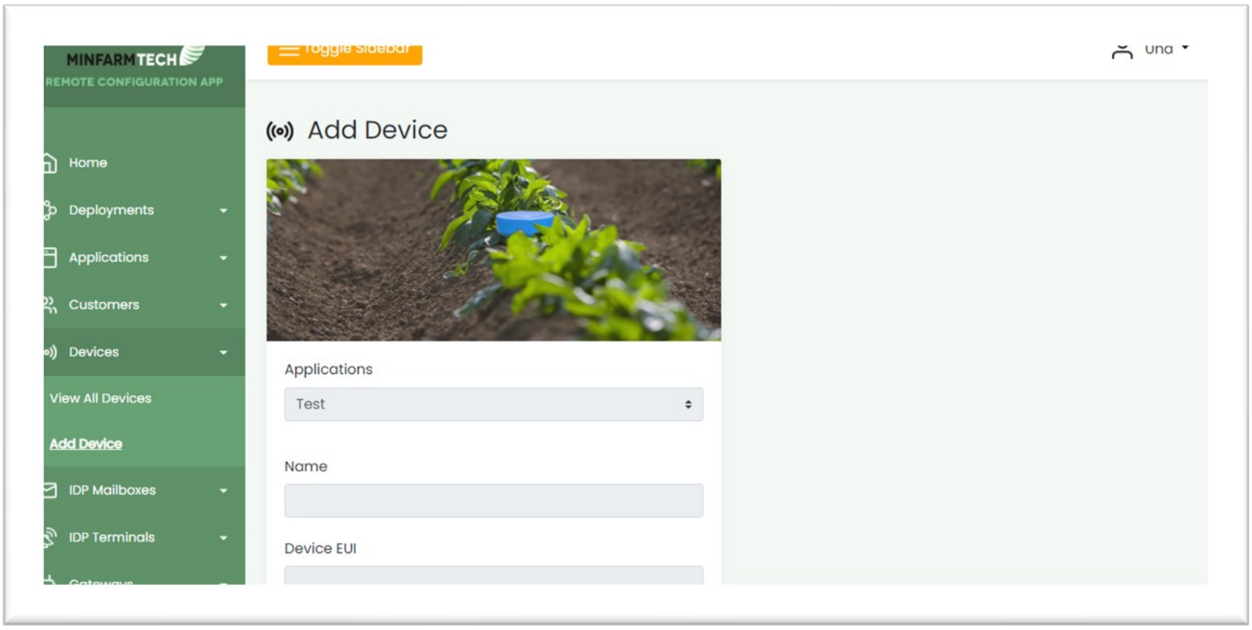
Application Access Credentials

Application Username

Application Password

If when adding an application, and your particular device / sensor is not listed under **<Type>**, then select **<Basic HTTP / HTTPS>**. This will send the data to a HTTP endpoint. Enter the URL you would like to direct the data to. Note Pipedream allows you to setup a public HTTP URL and see all traffic from your device(s).

Add a Device by selecting **<Add Device>**. Select the Application that you have just created. Device EUI, App EUI, and App Key are obtained from the device provider. Class, Device Profile and Network Profile can be left at the default settings for now. Note the Name of your device(s) could be the serial number of the device, to allow for easy identification if you have a number of devices in the field. Select **<Save>**.



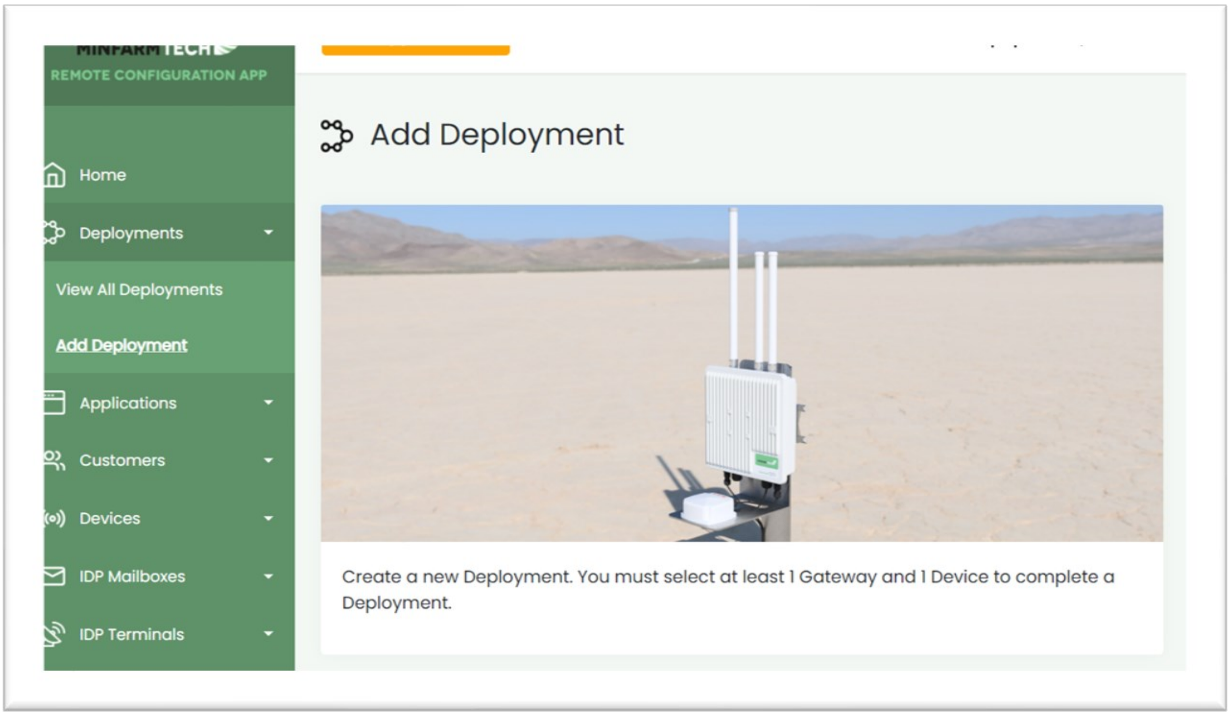
Add Device

Applications: Test

Name

Device EUI

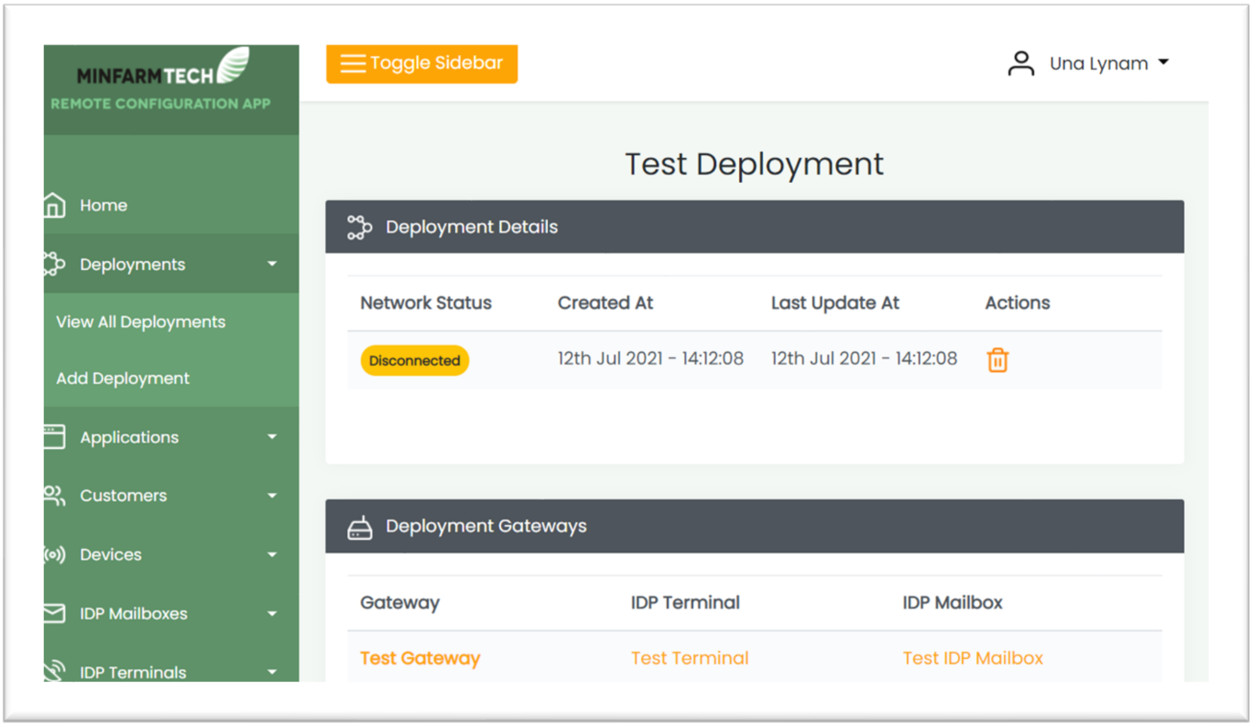
Finally, create a Deployment by selecting **<Add Deployment>**. This brings all the entered information together. Select a Gateway and a Device to add to the Deployment. Select **<Save>**.



Add Deployment

Create a new Deployment. You must select at least 1 Gateway and 1 Device to complete a Deployment.

Note the network status of the newly created Deployment is shown as a Disconnected yellow icon.



Test Deployment

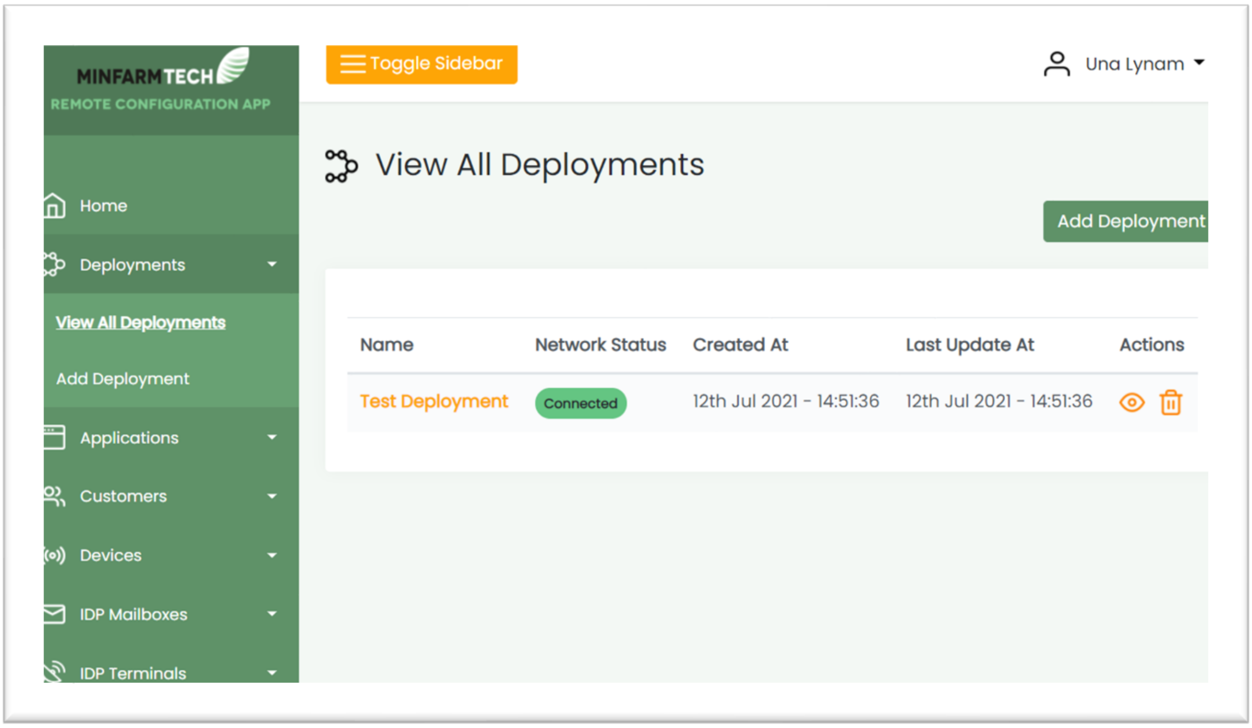
Deployment Details

Network Status	Created At	Last Update At	Actions
Disconnected	12th Jul 2021 - 14:12:08	12th Jul 2021 - 14:12:08	

Deployment Gateways

Gateway	IDP Terminal	IDP Mailbox
Test Gateway	Test Terminal	Test IDP Mailbox

Once the gateway comes online, the Network Status changes to a Connected green icon. This process takes a few minutes. Once established, all future data exchange is encrypted.



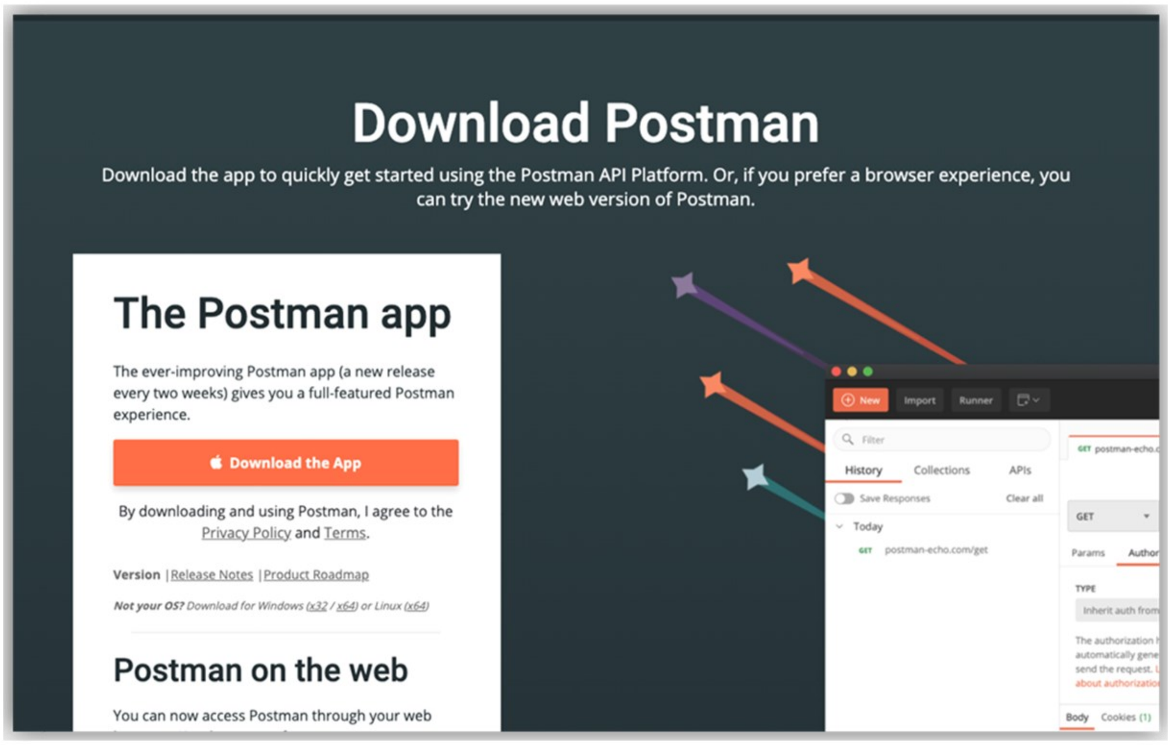
Name	Network Status	Created At	Last Update At	Actions
Test Deployment	Connected	12th Jul 2021 - 14:51:36	12th Jul 2021 - 14:51:36	

Well done, you have completed the first part of connecting to the MinFarm API.

! Note that once a Deployment has been created, it cannot be edited. If you want to change a Device in the Deployment, it is advised to delete the Deployment, and then recreate the Deployment again with the new Device.

8.0 STEP 2: POSTMAN

Postman is a great tool for interacting with a REST API. It has an easy to use interface that helps you to construct a HTTP Request. It can be used in your web browser. An account will need to be created.



Download Postman

Download the app to quickly get started using the Postman API Platform. Or, if you prefer a browser experience, you can try the new web version of Postman.

The Postman app

The ever-improving Postman app (a new release every two weeks) gives you a full-featured Postman experience.

Download the App

By downloading and using Postman, I agree to the Privacy Policy and Terms.

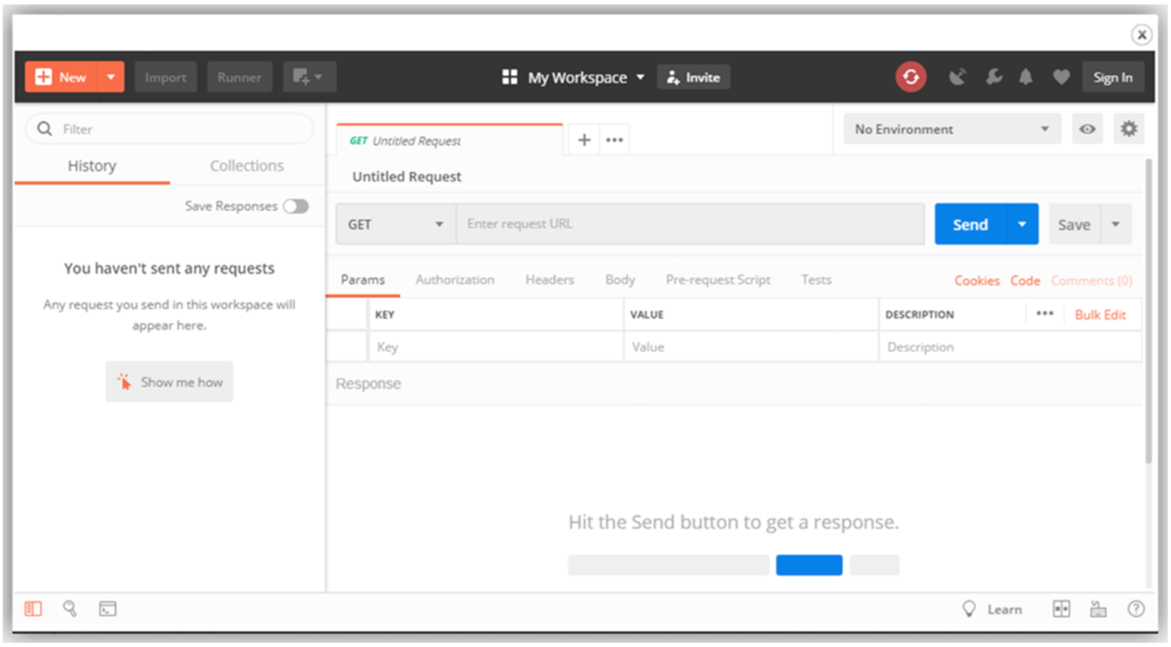
Version | Release Notes | Product Roadmap

Not your OS? Download for Windows (x32 / x64) or Linux (x64)

Postman on the web

You can now access Postman through your web

Postman streamlines the whole process of calling an API. The Postman dashboard is shown below.



You haven't sent any requests

Any request you send in this workspace will appear here.

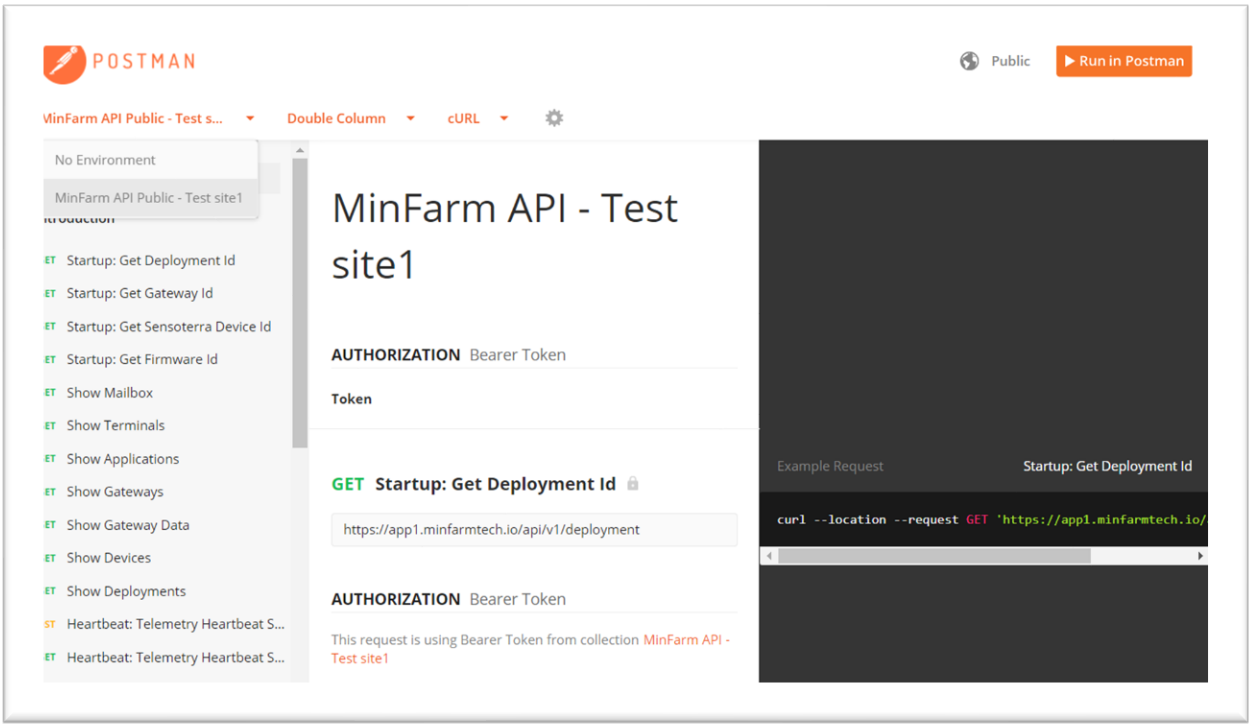
Hit the Send button to get a response.

Some knowledge of Postman and writing Requests is assumed in this user guide. Please refer to the Postman documentation for more information. Refer also to Appendix 1 for a short recap on some Postman functionality.

8.1 IMPORTING THE MINFARM POSTMAN COLLECTION

MinFarm has emailed you a link to the MinFarm API Postman Collection and associated Environment. The Requests in this Collection allow you to access the MinFarm API and enable you send and receive data to and from your device(s) and gateways in the field.

Click on the Postman link that you have received from MinFarm. A Postman page similar to that shown below is displayed. The associated Environment can be seen in the orange tab at the top left of the screen.


POSTMAN
Public
Run in Postman
MinFarm API Public - Test s...
Double Column
cURL
No Environment
MinFarm API Public - Test site1
ET Startup: Get Deployment Id
ET Startup: Get Gateway Id
ET Startup: Get Sensoterra Device Id
ET Startup: Get Firmware Id
ET Show Mailbox
ET Show Terminals
ET Show Applications
ET Show Gateways
ET Show Gateway Data
ET Show Devices
ET Show Deployments
ST Heartbeat: Telemetry Heartbeat S...
ET Heartbeat: Telemetry Heartbeat S...

MinFarm API - Test site1

AUTHORIZATION Bearer Token

Token

GET Startup: Get Deployment Id

https://app1.minfarmtech.io/api/v1/deployment

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection MinFarm API - Test site1

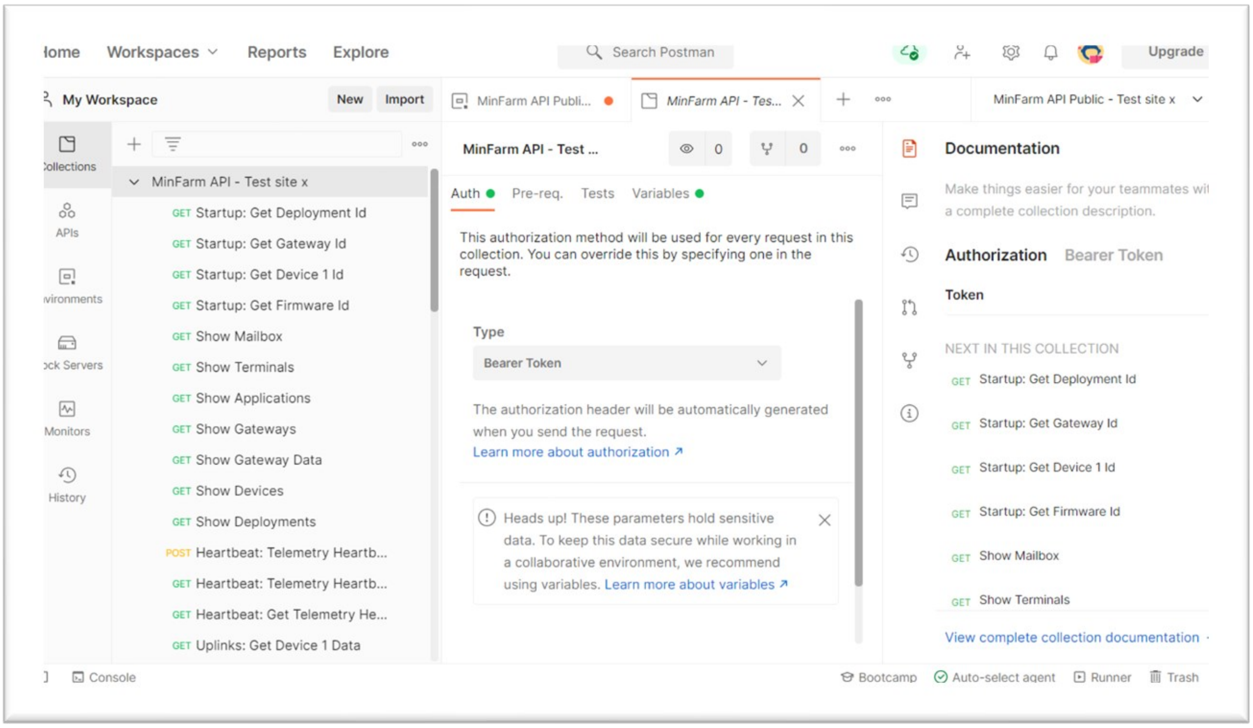
Example Request — Startup: Get Deployment Id

```
curl --location --request GET 'https://app1.minfarmtech.io/
```

Select the orange **<Run in Postman>** button at the top right of the screen. Select **<Run in - Postman for Web>**. Note that you will prompted to setup a Postman account if you have not already done so. You do not need a Postman account to view the API commands, but you will need an account to interact with the API. If this is your first time accessing Postman, you may be asked some questions. You can continue 'without a team', select the default when prompted to create a workspace, and you do not need a desktop agent.

! Note some web browser settings may need to be changed so the **<Run in Postman>** button can be activated. In Settings turn off 'Block popup windows'. If this does not work then contact MinFarm for two JSON files, one for the Collection and one for the Environment. When in Postman, select the <Import> button.

The screen below shows an example of an imported MinFarm API Collection (**MinFarm API - Test site x**) on the left-hand side tool-bar.



MinFarm API - Test site x
- GET Startup: Get Deployment Id
- GET Startup: Get Gateway Id
- GET Startup: Get Device 1 Id
- GET Startup: Get Firmware Id
- GET Show Mailbox
- GET Show Terminals
- GET Show Applications
- GET Show Gateways
- GET Show Gateway Data
- GET Show Devices
- GET Show Deployments
- POST Heartbeat: Telemetry Heartb...
- GET Heartbeat: Telemetry Heartb...
- GET Heartbeat: Get Telemetry He...
- GET Uplinks: Get Device 1 Data

This authorization method will be used for every request in this collection. You can override this by specifying one in the request.

The authorization header will be automatically generated when you send the request.
Learn more about authorization ↗

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

Make things easier for your teammates with a complete collection description.

Authorization Bearer Token

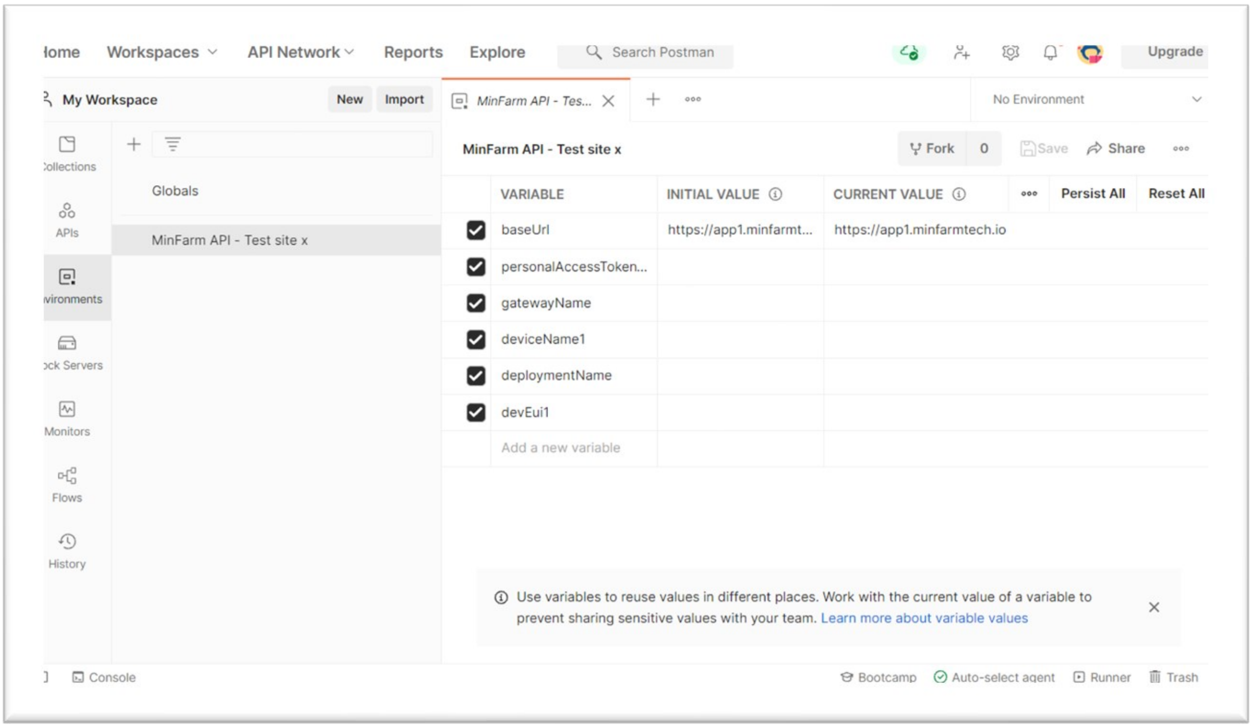
Token

NEXT IN THIS COLLECTION
- GET Startup: Get Deployment Id
- GET Startup: Get Gateway Id
- GET Startup: Get Device 1 Id
- GET Startup: Get Firmware Id
- GET Show Mailbox
- GET Show Terminals

View complete collection documentation

8.2 EDITING THE ENVIRONMENT VARIABLES

Select Environments on the left-hand side of the screen and select the required Environment. In this example the Environment is called **MinFarm API - Test site x**.



MinFarm API - Test site x

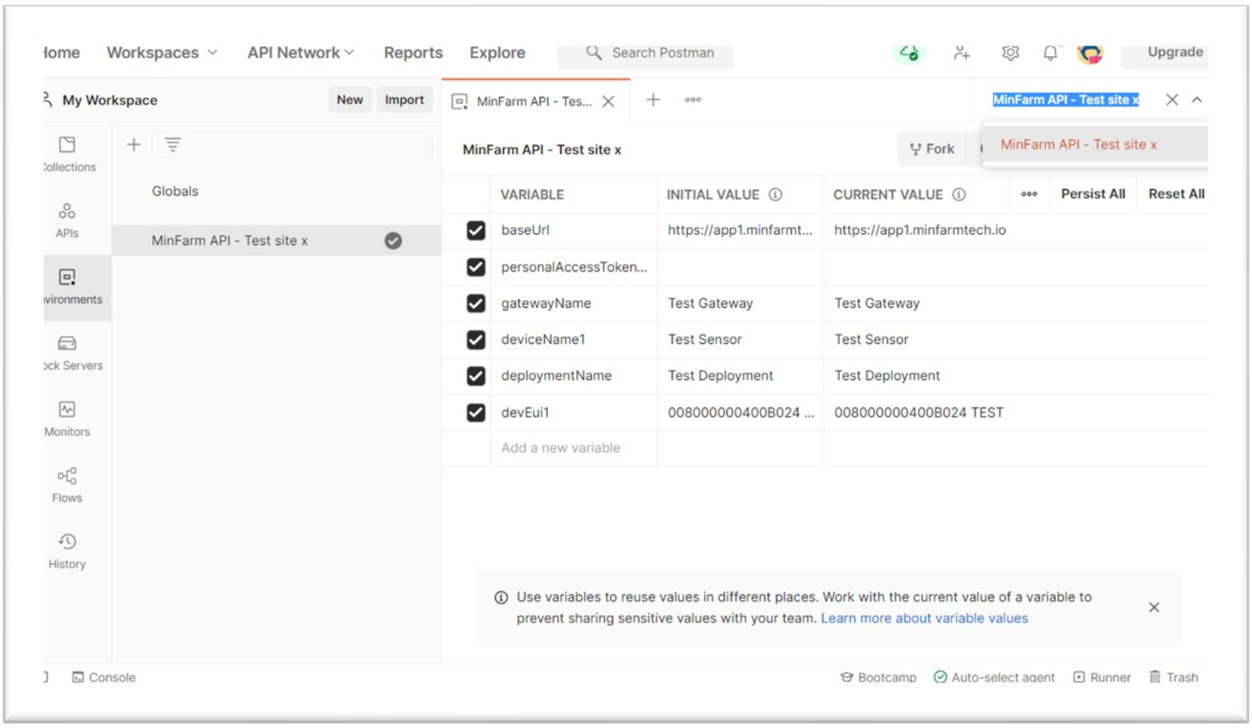
	VARIABLE	INITIAL VALUE	CURRENT VALUE
✓	baseUrl	https://app1.minfarmt...	https://app1.minfarmtech.io
✓	personalAccessToken...		
✓	gatewayName		
✓	deviceName1		
✓	deploymentName		
✓	devEui1		
	Add a new variable		

Use variables to reuse values in different places. Work with the current value of a variable to prevent sharing sensitive values with your team. Learn more about variable values

Some Environment variables need to be filled in with information that has come from the Deployment you have setup in Section 7.2. Find the information you have entered in the MinFarm Dashboard, and add each variable value to both the **<Initial Value>** and **<Current Value>**. Select **<Save>**.

- gatewayName
- deviceName1
- deploymentName
- devEui1

In this example these are populated as shown below.



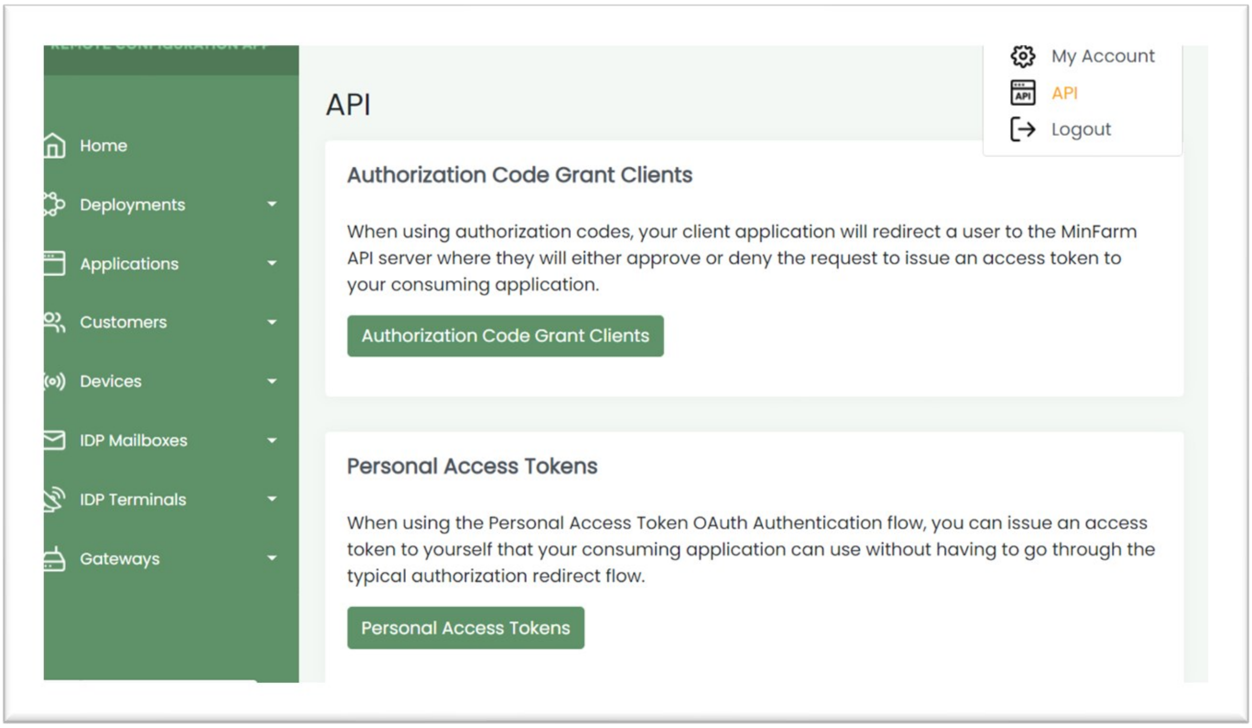
VARIABLE	INITIAL VALUE	CURRENT VALUE
✓ baseUrl	https://app1.minfarmt...	https://app1.minfarmtech.io
✓ personalAccessToken...		
✓ gatewayName	Test Gateway	Test Gateway
✓ deviceName1	Test Sensor	Test Sensor
✓ deploymentName	Test Deployment	Test Deployment
✓ devEui1	008000000400B024 ...	008000000400B024 TEST
Add a new variable		

Use variables to reuse values in different places. Work with the current value of a variable to prevent sharing sensitive values with your team. Learn more about variable values

! Note the correct Environment also needs to be correctly selected from the drop-down Environment menu at the top right of the dashboard. See above screenshot.

8.3 CREATING PERSONAL ACCESS TOKENS

Security in the form of Personal Access Tokens (PATs) was discussed in Section 6.0. To obtain the required PATs, go to API in the drop-down menu at the top right of the MinFarm Dashboard.



API

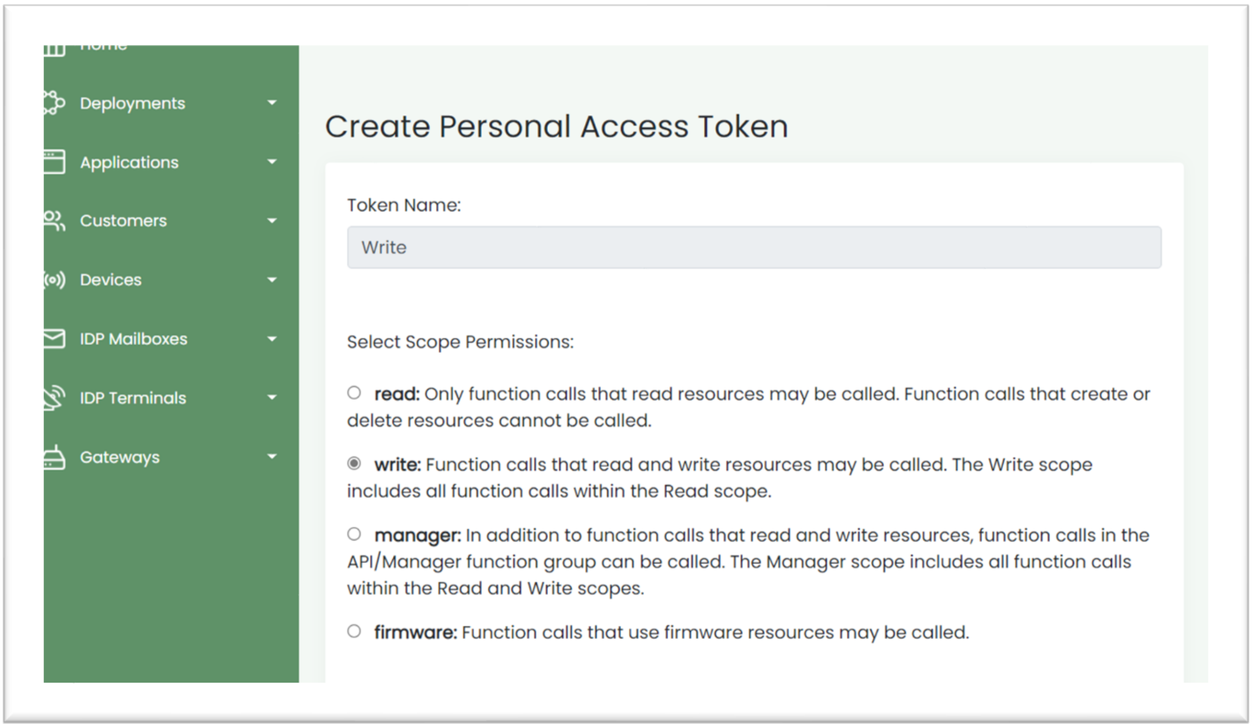
Authorization Code Grant Clients

When using authorization codes, your client application will redirect a user to the MinFarm API server where they will either approve or deny the request to issue an access token to your consuming application.

Personal Access Tokens

When using the Personal Access Token OAuth Authentication flow, you can issue an access token to yourself that your consuming application can use without having to go through the typical authorization redirect flow.

Select **<Personal Access Tokens>** and **<Create Token>**. Call the token name 'Write', and select the write scope as shown below.



Create Personal Access Token

Token Name: Write

Select Scope Permissions:

○ **read:** Only function calls that read resources may be called. Function calls that create or delete resources cannot be called.

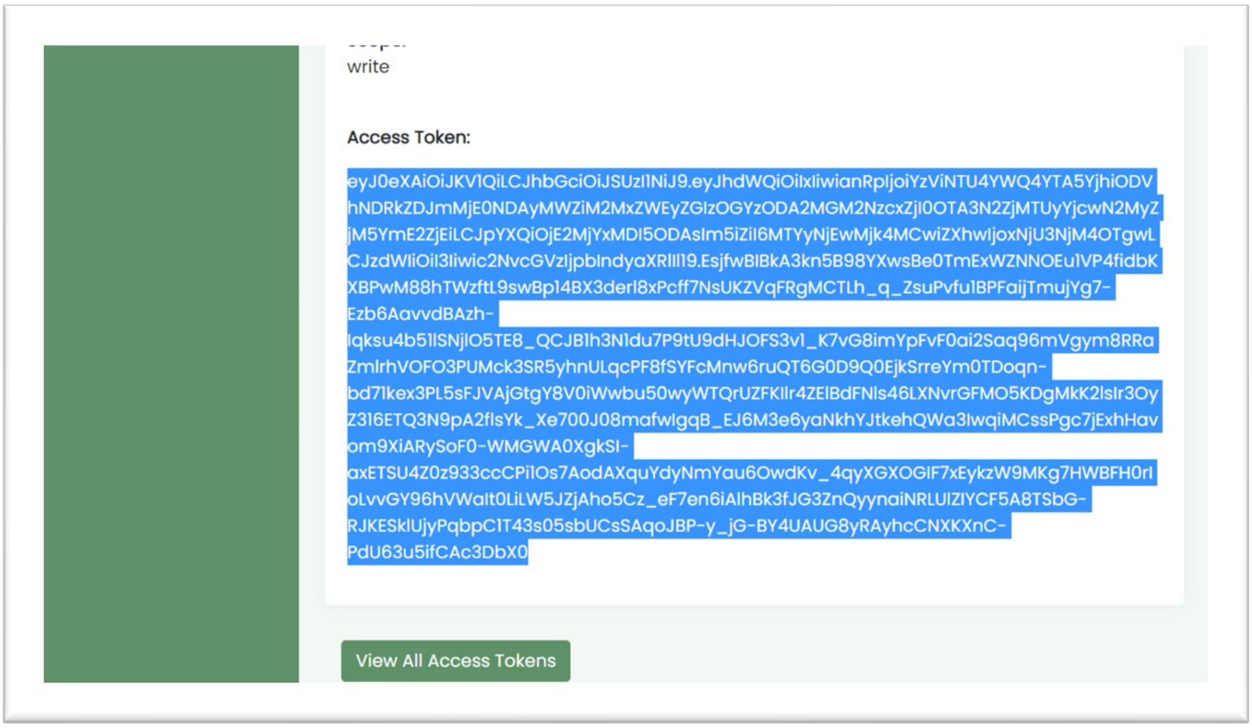
[✓ **write:** Function calls that read and write resources may be called. The Write scope includes all function calls within the Read scope.]

○ **manager:** In addition to function calls that read and write resources, function calls in the API/Manager function group can be called. The Manager scope includes all function calls within the Read and Write scopes.

○ **firmware:** Function calls that use firmware resources may be called.

Select **<Save>**. The PAT has been created. Copy all of the access token. It is important to select all of the access token and ensure that the start and end of the selected area contains no white space.

Note that it must be copied at this point and saved somewhere, as it will no longer be accessible after exiting this screen. Should you forget to save the token, simply delete the token and create a new one.

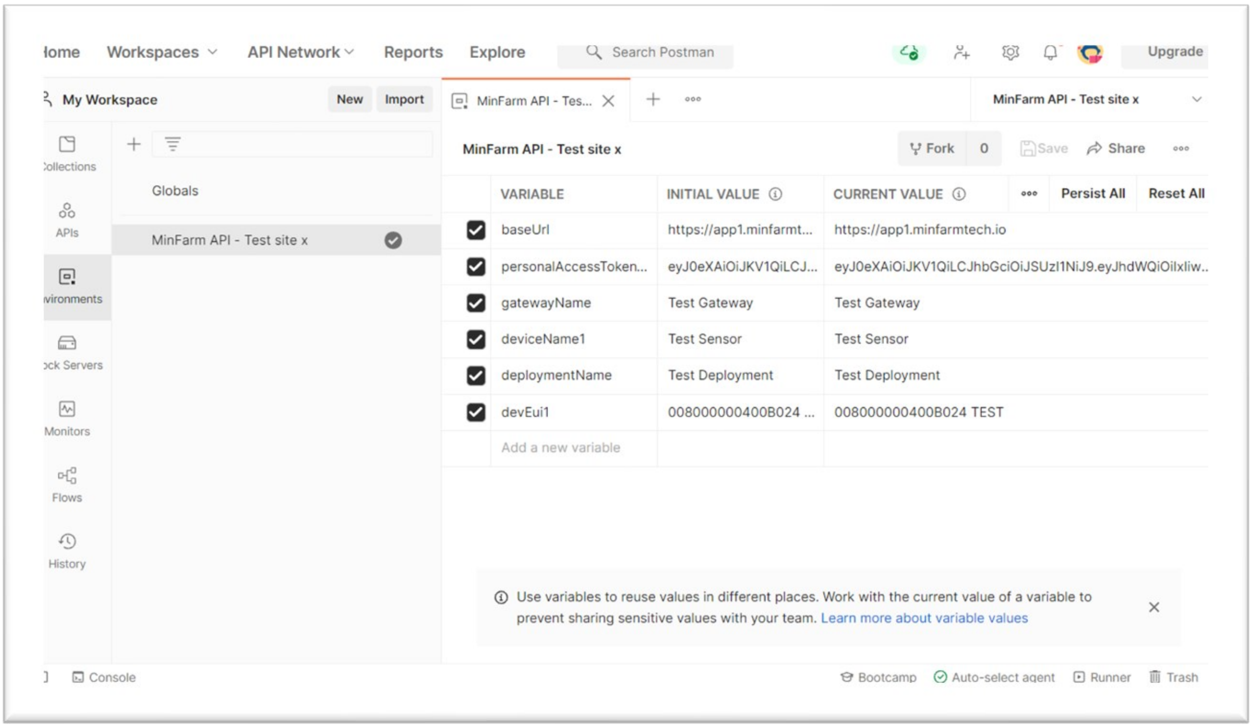


write

Access Token:

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIxliwianRpIjoiYzViNTU4YWQ4YTA5YjhiODV
hNDRkZDJmMjE0NDAyMWZiM2MxZWEyZGIzOGYzODA2MGM2NzcxZjI0OTA3N2ZjMTUyYjcwN2MyZ
jM5YmE2ZjEiLCJpYXQiOjE2MjYxMDI5ODAsIm5iZiI6MTYyNjEwMjk4MCwiZXhwIjoxNjU3NjM4OTgwL
CJzdWIiOiI3Iiwic2NvcGVzIjpbIndyaXRlIl19.EsjfwBIBkA3kn5B98YXwsBe0TmExWZNNOEu1VP4fidbK
XBPwM88hTWzftL9swBp14BX3derl8xPcff7NsUKZVqFRgMCTLh_q_ZsuPvfu1BPFaijTmujYg7-
Ezb6AavvdBAzh-
Iqksu4b51lSNjIO5TE8_QCJB1h3N1du7P9tU9dHJOFS3v1_K7vG8imYpFvF0ai2Saq96mVgym8RRa
ZmIrhVOFO3PUMck3SR5yhnULqcPF8fSYFcMnw6ruQT6G0D9Q0EjkSrreYm0TDoqn-
bd71kex3PL5sFJVAjGtgY8V0iWwbu50wyWTQrUZFKIlr4ZElBdFNls46LXNvrGFMO5KDgMkK2lsIr3Oy
Z316ETQ3N9pA2flsYk_Xe700J08mafwIgqB_EJ6M3e6yaNkhYJtkehQWa3lwqiMCssPgc7jExhHav
om9XiARySoF0-WMGWA0XgkSI-
axETSU4Z0z933ccCPi1Os7AodAXquYdyNmYau6OwdKv_4qyXGXOGIF7xEykzW9MKg7HWBFH0rl
oLvvGY96hVWalt0LiLW5JZjAho5Cz_eF7en6iAlhBk3fJG3ZnQyynaiNRLUlZIYCF5A8TSbG-
RJKESklUjyPqbpC1T43s05sbUCsSAqoJBP-y_jG-BY4UAUG8yRAyhcCNXKXnC-
PdU63u5ifCAc3DbX0
```

Paste into the **<Initial Value>** and **<Current Value>** of the personalAccessTokenWrite variable in your Environment. Select **<Save>**.

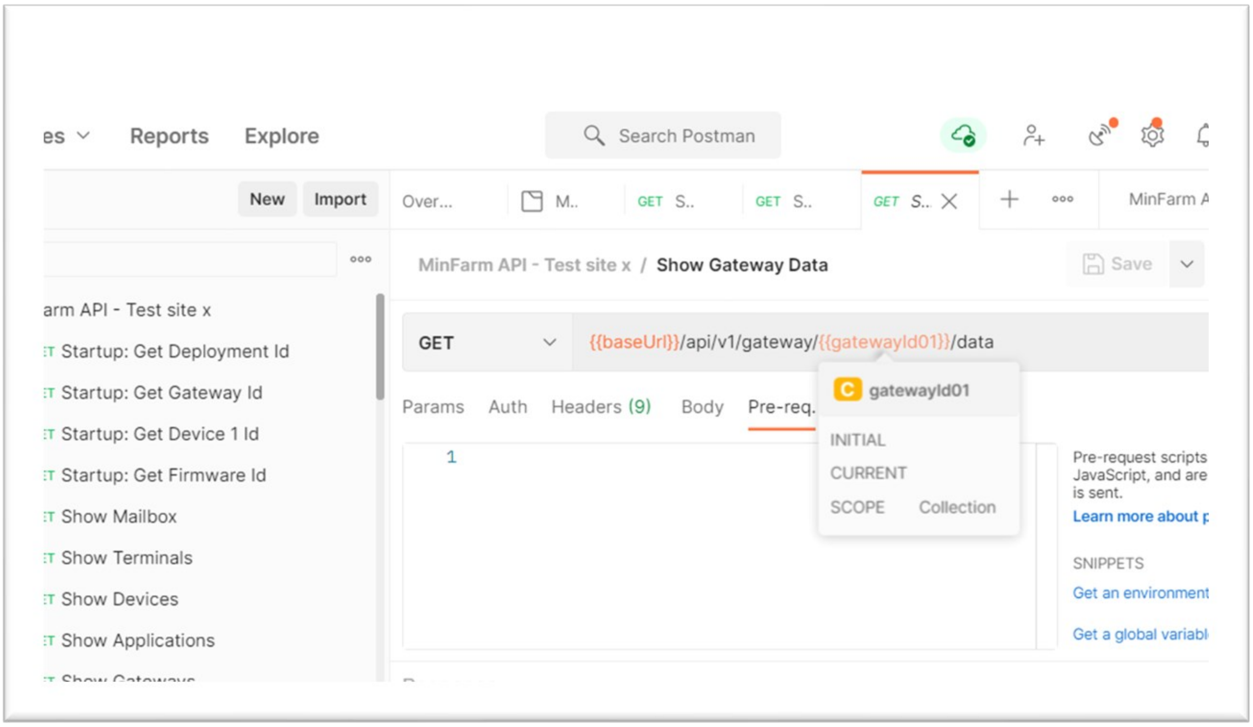


	VARIABLE	INITIAL VALUE	CURRENT VALUE
✓	baseUrl	https://app1.minfarmt...	https://app1.minfarmtech.io
✓	personalAccessToken...	eyJ0eXAiOiJKV1QiLCJ...	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIxliw..
✓	gatewayName	Test Gateway	Test Gateway
✓	deviceName1	Test Sensor	Test Sensor
✓	deploymentName	Test Deployment	Test Deployment
✓	devEui1	008000000400B024 ...	008000000400B024 TEST
	Add a new variable		

Use variables to reuse values in different places. Work with the current value of a variable to prevent sharing sensitive values with your team. Learn more about variable values

8.4 CREATING THE COLLECTION VARIABLES

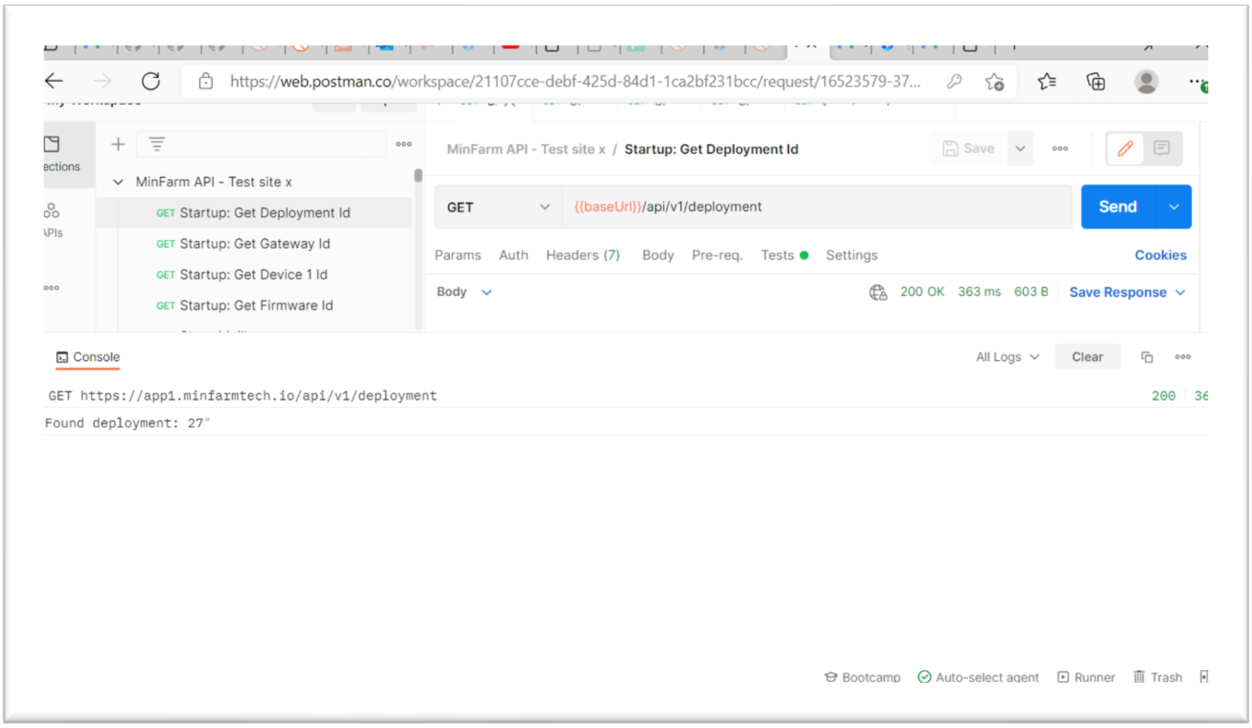
As well as Environment variables, Postman also uses some local Collection variables. These need to be set the first time you use the Collection, and also anytime you log out of Postman and log back in again. In the screenshot below, no Collection variables can be seen if you hover over the gatewayId01 variable in the URL, *i.e.* the initial and current values are blank.



To set these Collection variables it is necessary to run the following series of Postman Requests:

GET Startup: Get Deployment Id

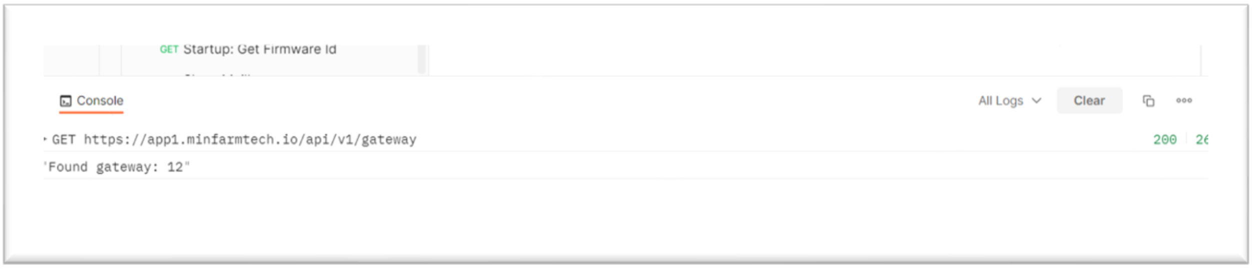
Select **<Send>**. Expect a successful response of 200 OK. Select the Console on the bottom left-hand side to confirm ‘Found deployment’.


https://web.postman.co/workspace/21107cce-debf-425d-84d1-1ca2bf231bcc/request/16523579-37...
MinFarm API - Test site x
GET Startup: Get Deployment Id
GET Startup: Get Gateway Id
GET Startup: Get Device 1 Id
GET Startup: Get Firmware Id
MinFarm API - Test site x / Startup: Get Deployment Id
Save
GET
{{baseUrl}}/api/v1/deployment
Send
Params
Auth
Headers (7)
Body
Pre-req.
Tests
Settings
Cookies
Body
200 OK
363 ms
603 B
Save Response
Console
All Logs
Clear
Bootcamp
Auto-select agent
Runner
Trash

```
GET https://app1.minfarmtech.io/api/v1/deployment                200  36
Found deployment: 27"
```

GET Startup: Get Gateway Id

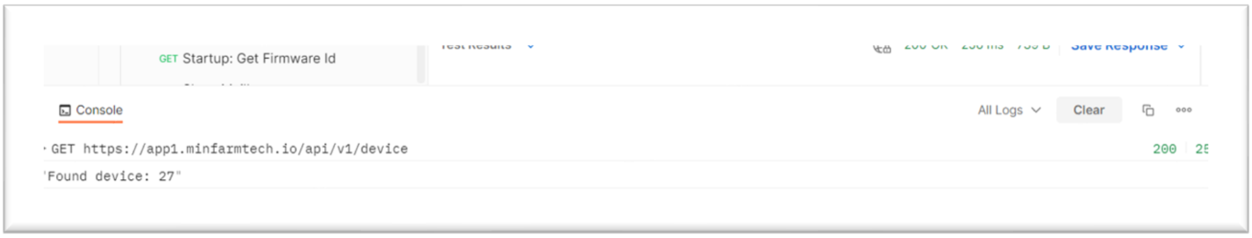
Confirm that a successful response of 200 OK is received for the Gateway, and the Console shows ‘Found gateway’.



```
GET https://app1.minfarmtech.io/api/v1/gateway                   200  26
'Found gateway: 12"
```

GET Startup: Get Device 1 Id

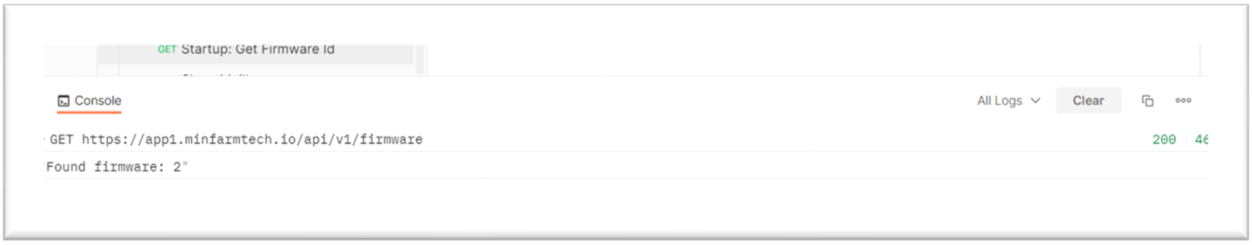
Confirm that a successful response of 200 OK is received for the Device, and the Console shows ‘Found device’.



```
GET https://app1.minfarmtech.io/api/v1/device    200
"Found device: 27"
```

GET Startup: Get Firmware Id

Only if you are using the Firmware feature is it necessary to set the Firmware Id variable. Contact MinFarm if you need further information on this, and need it added to your Collection. Confirm that a successful response is received for the Firmware, and the Console shows 'Found firmware'.



```
GET https://app1.minfarmtech.io/api/v1/firmware    200
Found firmware: 2"
```

8.5 SYNC DEVICE LIST

Finally, it is necessary to push to the Gateway the Device(s) that you have created in the MinFarm Dashboard. Refer to Section 9.2.

See Section 9.1 below for an explanation of the types of Request used by the MinFarm API, and what polling of Requests means.

9.0 THE MINFARM API COMMANDS

The Collection that you have received contains various Postman Requests. These Requests allow you to view the information flow between your Gateway, Terminal, and Device(s). The Requests and expected Responses are detailed in the Sections below.

Note 1: Times are in UTC.

Note 2: Screenshots from Postman are shown below. Where the Postman screen is not large enough to show a screenshot of all data in the Postman Response, the data has been copied from the Response, and shown here in table format.

9.1 MINFARM API COMMAND TERMINOLOGY

The MinFarm API support two types of commands, Instant and Queued.

- Instant commands are processed immediately by the MinFarm API. A Response is immediately returned which includes the command result. The IDP satellite link does not need to be active for this type of command.
- Queued commands can take several minutes to process, and use a polling approach which runs the command as a background task. The command result is made available at a later stage. The MinFarm API immediately returns a Response indicating whether the command has been queued for processing. If the command was queued successfully, the Response includes an id which can be used later to poll the result of the command. Initially, the command result will be PENDING. When the command processing is complete (less than 5 minutes for most queued commands), the processing result is made available in the command result. If the command processing does not complete within a certain time, the command result is set to TIMEOUT and the command processing is cancelled.

The result field can take the following values:

0: PENDING. Description: Keep polling for the result.

1: SUCCESS. Description: The command was successful.

2: TIMEOUT. Description: The command timed out.

The processing of a queued command involves several steps: the command is forwarded by the MinFarm API to the Gateway; the Gateway runs the command; the Gateway sends the command result back to the MinFarm API; the MinFarm API saves the command result. The IDP satellite link must be active for this type of command.

9.2 SYNC DEVICE LIST

Request: **Post Device List: Sync Device List**

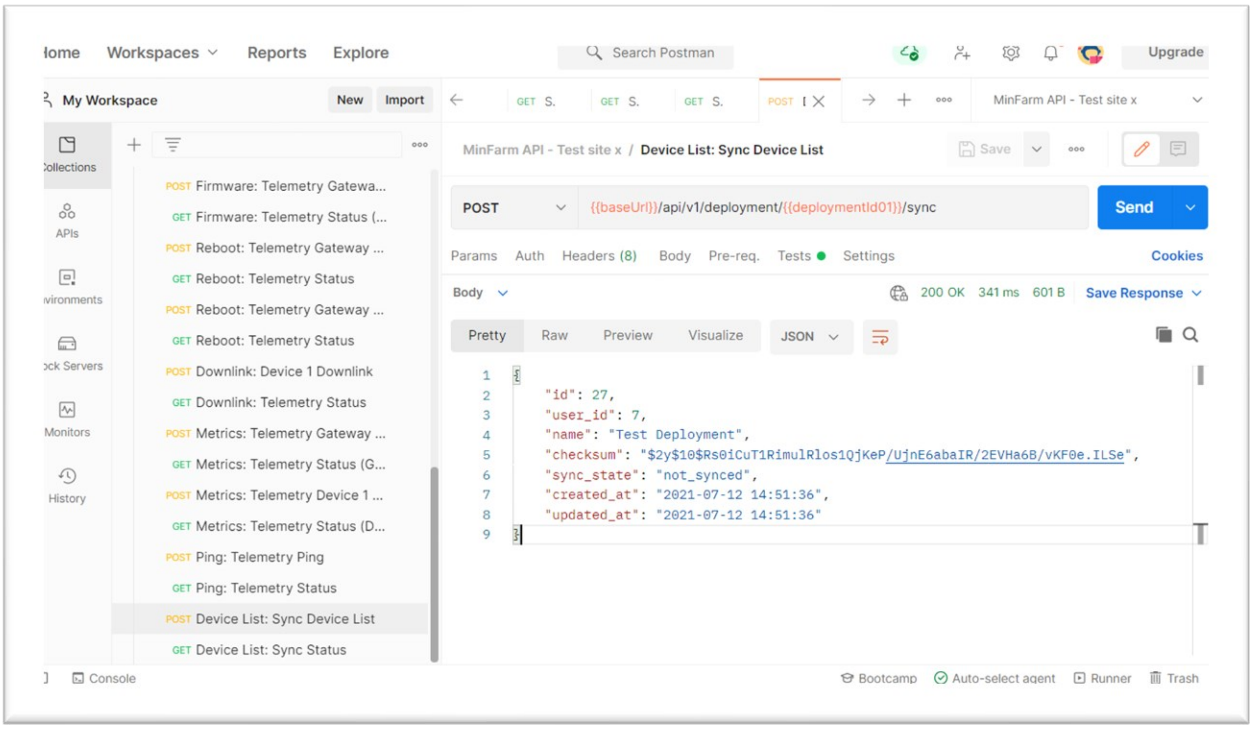
Description: Initiates a push to the Gateway of the Device(s) you have created.

Type of Command: Queued.

URL: {{baseUrl}}/api/v1/deployment/{{deploymentId01}}/sync

Expected Response: 200 OK

Example Response: The Deployment name and creation time is displayed in the Response.


Home
Workspaces
Reports
Explore
Search Postman
Upgrade
My Workspace
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
POST Firmware: Telemetry Gatewa...
GET Firmware: Telemetry Status (...
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Downlink: Device 1 Downlink
GET Downlink: Telemetry Status
POST Metrics: Telemetry Gateway ...
GET Metrics: Telemetry Status (G...
POST Metrics: Telemetry Device 1 ...
GET Metrics: Telemetry Status (D...
POST Ping: Telemetry Ping
GET Ping: Telemetry Status
POST Device List: Sync Device List
GET Device List: Sync Status
MinFarm API - Test site x
MinFarm API - Test site x / Device List: Sync Device List
Save
Send
Params
Auth
Headers (8)
Body
Pre-req.
Tests
Settings
Cookies
Body
Save Response
Pretty
Raw
Preview
Visualize
JSON
Console
Bootcamp
Auto-select agent
Runner
Trash

POST {{baseUrl}}/api/v1/deployment/{{deploymentId01}}/sync

200 OK 341 ms 601 B

```
{
    "id": 27,
    "user_id": 7,
    "name": "Test Deployment",
    "checksum": "$2y$10$Rs0iCuT1RimulRlos1QjKeP/UjnE6abaIR/2EVHa6B/vKF0e.ILSe",
    "sync_state": "not_synced",
    "created_at": "2021-07-12 14:51:36",
    "updated_at": "2021-07-12 14:51:36"
}
```

Request: **<u>GET Device List: Sync Status</u>**

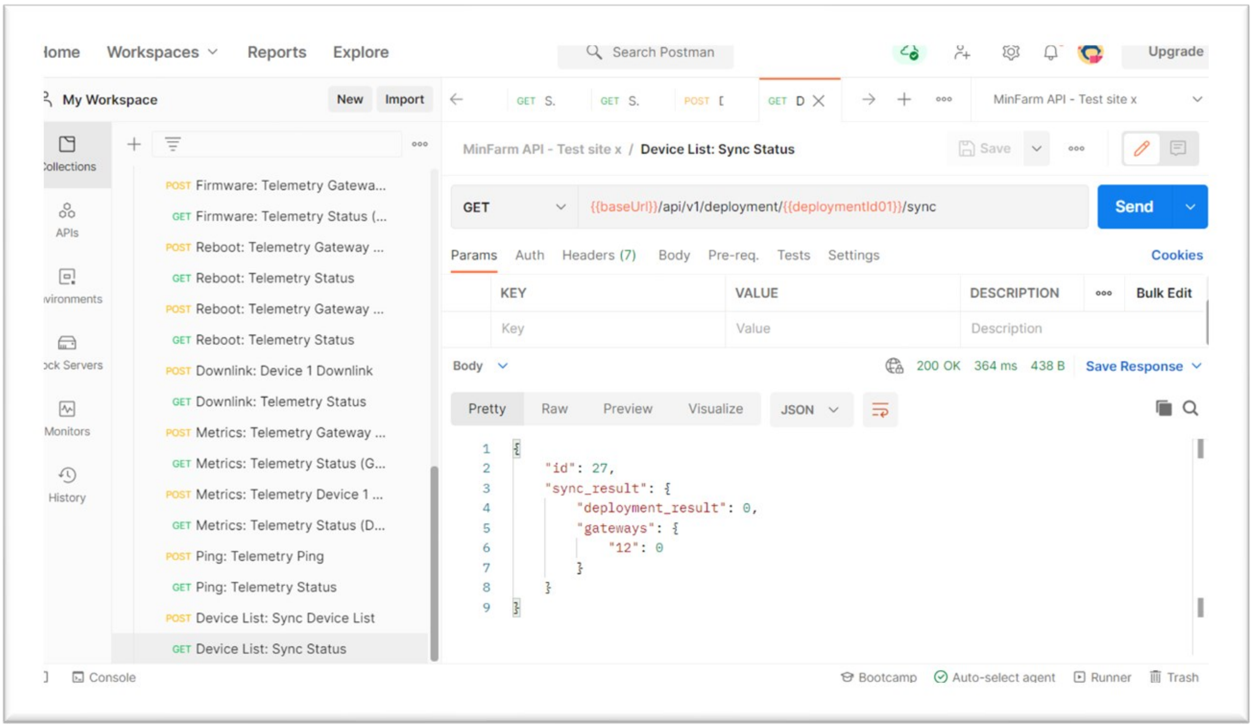
Description: Poll status of queued command.

Type of Command: Queued.

URL: {{baseUrl}}/api/v1/deployment/{{deploymentId01}}/sync

Expected Response: 200 OK

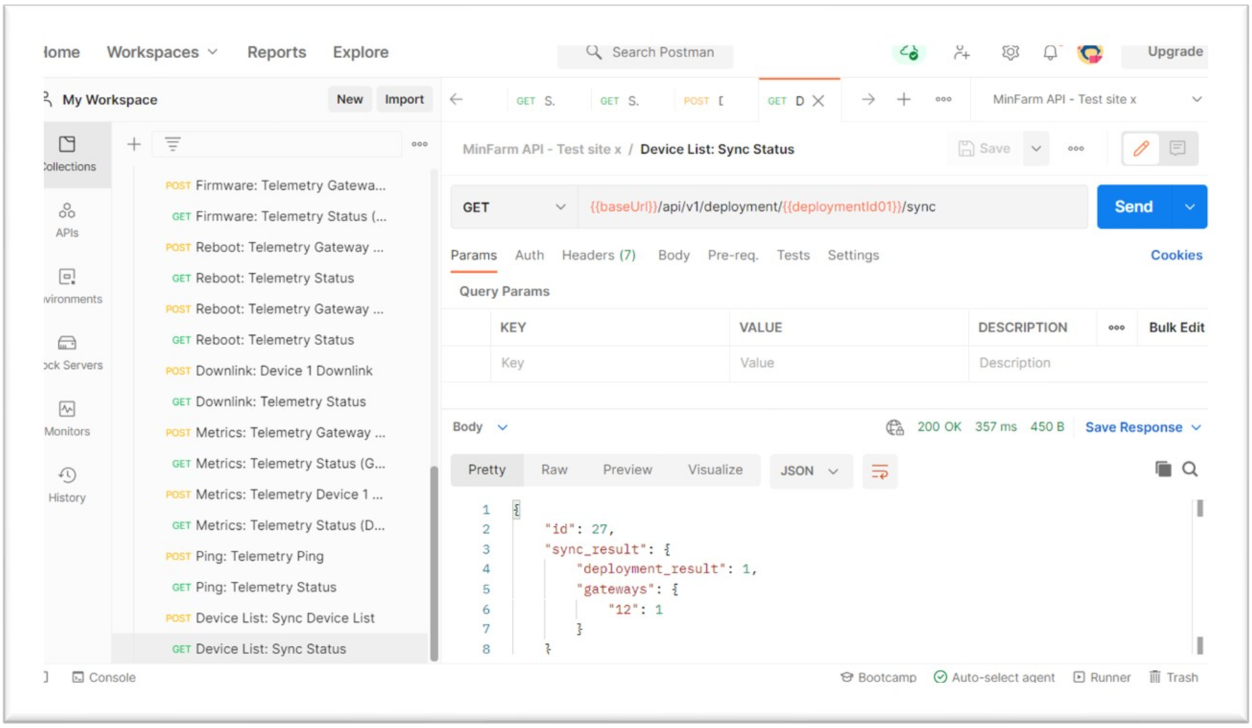
Example Immediate Response: The same id is returned. "deployment_result" of 0 to show it is pending.


Home
Workspaces
Reports
Explore
Search Postman
Upgrade
My Workspace
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
POST Firmware: Telemetry Gatewa...
GET Firmware: Telemetry Status (...
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Downlink: Device 1 Downlink
GET Downlink: Telemetry Status
POST Metrics: Telemetry Gateway ...
GET Metrics: Telemetry Status (G...
POST Metrics: Telemetry Device 1 ...
GET Metrics: Telemetry Status (D...
POST Ping: Telemetry Ping
GET Ping: Telemetry Status
POST Device List: Sync Device List
GET Device List: Sync Status
MinFarm API - Test site x
MinFarm API - Test site x / Device List: Sync Status
Save
Send
Params
Auth
Headers (7)
Body
Pre-req.
Tests
Settings
Cookies
KEY
VALUE
DESCRIPTION
Bulk Edit
Body
200 OK 364 ms 438 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
Console
Bootcamp
Auto-select agent
Runner
Trash

GET {{baseUrl}}/api/v1/deployment/{{deploymentId01}}/sync

```
{
    "id": 27,
    "sync_result": {
        "deployment_result": 0,
        "gateways": {
            "12": 0
        }
    }
}
```

Example Response after approximately 5 minutes: The same id is returned. “deployment_result” of 1 to show it is successful. A sync can take 5 minutes or more depending on the number of devices.


Home
Workspaces
Reports
Explore
Search Postman
Upgrade
My Workspace
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
POST Firmware: Telemetry Gatewa...
GET Firmware: Telemetry Status (...
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Downlink: Device 1 Downlink
GET Downlink: Telemetry Status
POST Metrics: Telemetry Gateway ...
GET Metrics: Telemetry Status (G...
POST Metrics: Telemetry Device 1 ...
GET Metrics: Telemetry Status (D...
POST Ping: Telemetry Ping
GET Ping: Telemetry Status
POST Device List: Sync Device List
GET Device List: Sync Status
MinFarm API - Test site x
MinFarm API - Test site x / Device List: Sync Status
Save
Send
Params
Auth
Headers (7)
Body
Pre-req.
Tests
Settings
Cookies
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
Key
Value
Description
Body
Save Response
Pretty
Raw
Preview
Visualize
JSON
Console
Bootcamp
Auto-select agent
Runner
Trash

GET {{baseUrl}}/api/v1/deployment/{{deploymentId01}}/sync

200 OK 357 ms 450 B

```
{
    "id": 27,
    "sync_result": {
        "deployment_result": 1,
        "gateways": {
            "12": 1
        }
    }
```

9.3 SHOW MAILBOXES

Request: **GET Show Mailboxes**

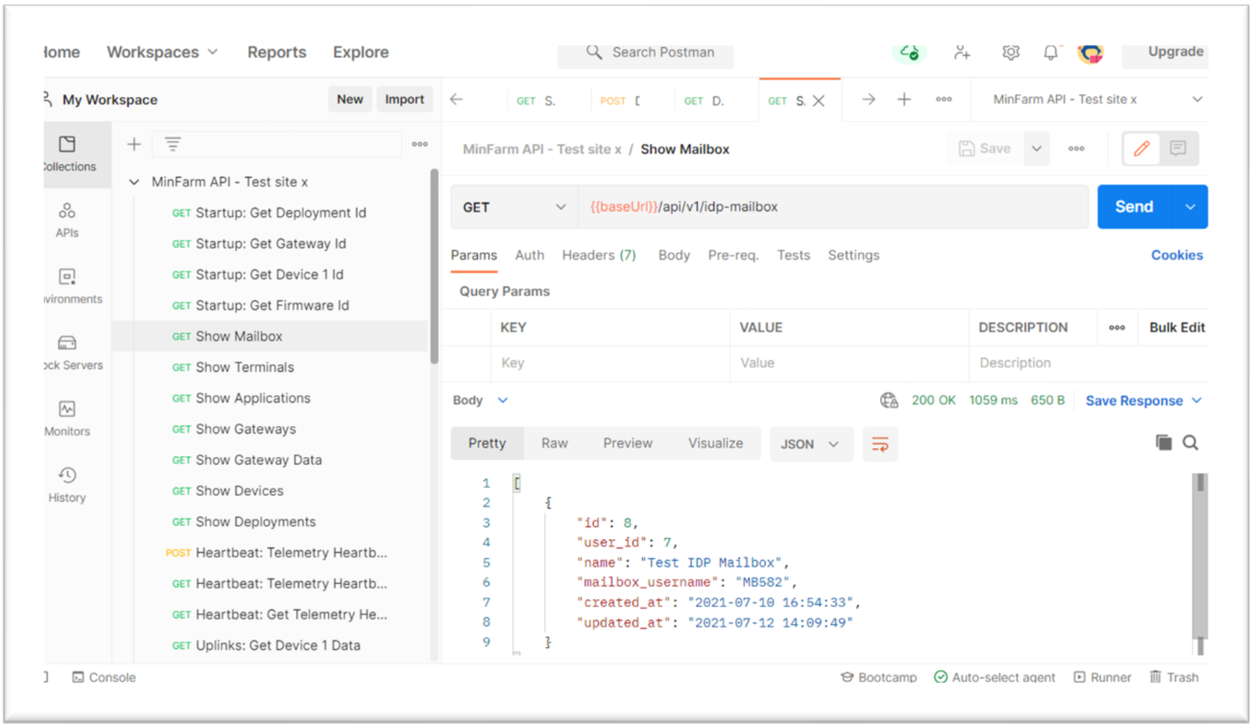
Description: Shows a list of all IDP Mailboxes created in the MinFarm Dashboard.

Type of Command: Instant.

URL: {{baseUrl}}/api/v1/idp-mailbox

Expected Response: 200 OK

Example Response:



MinFarm API - Test site x
- GET Startup: Get Deployment Id
- GET Startup: Get Gateway Id
- GET Startup: Get Device 1 Id
- GET Startup: Get Firmware Id
- GET Show Mailbox
- GET Show Terminals
- GET Show Applications
- GET Show Gateways
- GET Show Gateway Data
- GET Show Devices
- GET Show Deployments
- POST Heartbeat: Telemetry Heartb...
- GET Heartbeat: Telemetry Heartb...
- GET Heartbeat: Get Telemetry He...
- GET Uplinks: Get Device 1 Data

MinFarm API - Test site x / Show Mailbox

GET {{baseUrl}}/api/v1/idp-mailbox

Params | Auth | Headers (7) | Body | Pre-req. | Tests | Settings | Cookies

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body 200 OK 1059 ms 650 B Save Response

Pretty | Raw | Preview | Visualize | JSON

```
[
    {
        "id": 8,
        "user_id": 7,
        "name": "Test IDP Mailbox",
        "mailbox_username": "MB582",
        "created_at": "2021-07-10 16:54:33",
        "updated_at": "2021-07-12 14:09:49"
    }
```

9.4 SHOW TERMINALS

Request: **GET Show Terminals**

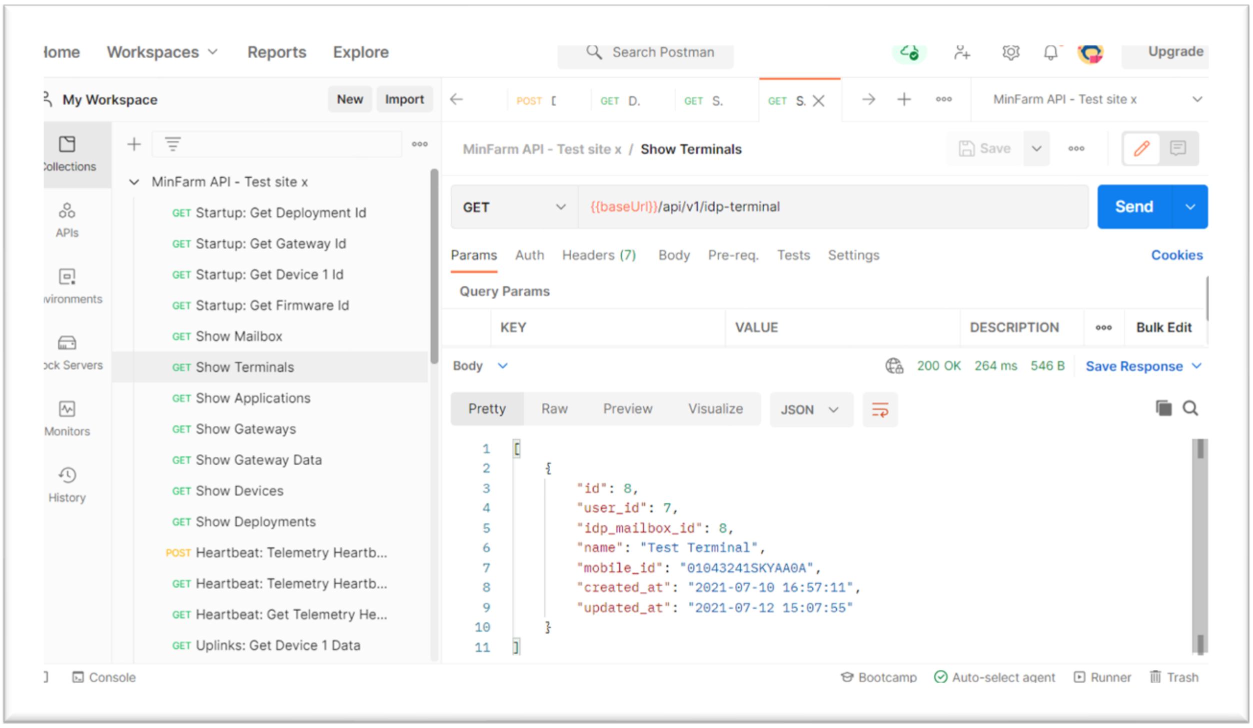
Description: Shows a list of all Terminals created in the MinFarm Dashboard.

Type of Command: Instant.

URL: {{baseUrl}}/api/v1/idp-terminal

Expected Response: 200 OK

Example Response:



GET {{baseUrl}}/api/v1/idp-terminal

200 OK 264 ms 546 B

```
[
    {
        "id": 8,
        "user_id": 7,
        "idp_mailbox_id": 8,
        "name": "Test Terminal",
        "mobile_id": "01043241SKYAA0A",
        "created_at": "2021-07-10 16:57:11",
        "updated_at": "2021-07-12 15:07:55"
    }
]
```

9.5 SHOW APPLICATIONS

Request: **GET Show Applications**

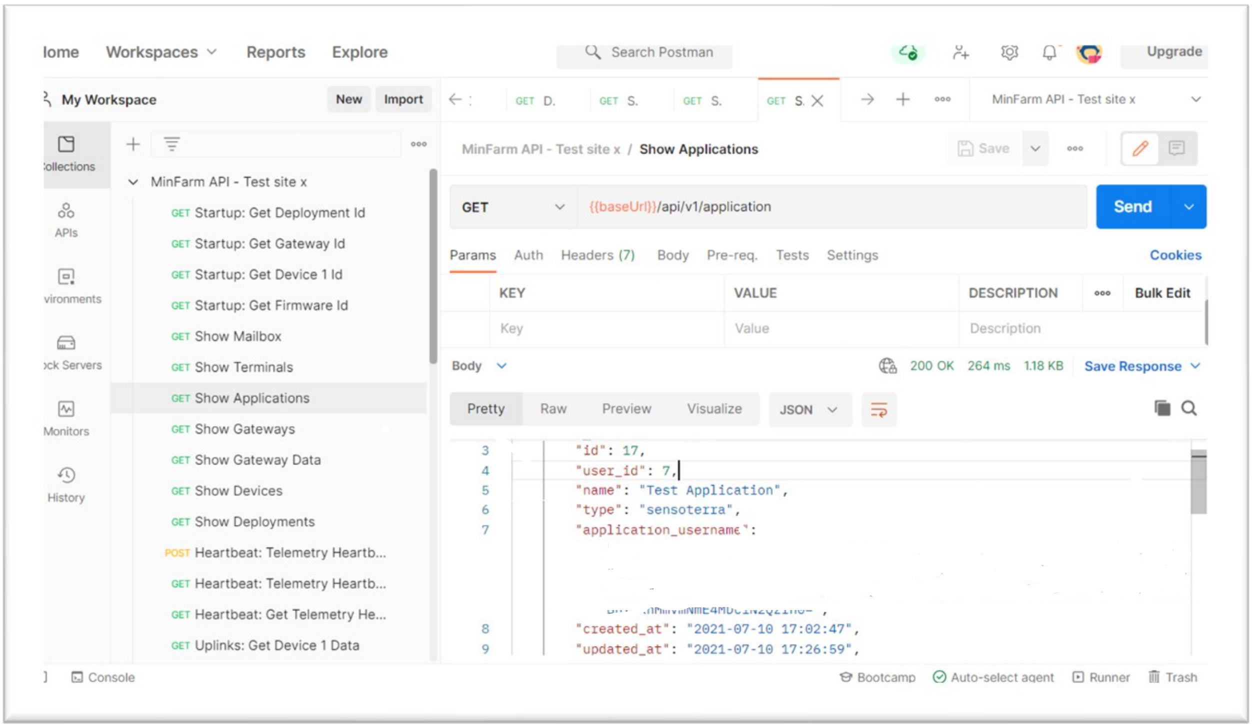
Description: Shows a list of all Applications created in the MinFarm Dashboard.

Type of Command: Instant.

URL: {{baseUrl}}/api/v1/application

Expected Response: 200 OK

Example Response: Note some sensitive information has been blurred out from the screenshot below.



```
"id": 17,
"user_id": 7,
"name": "Test Application",
"type": "sensoterra",
"application_username":

[illegible]
"created_at": "2021-07-10 17:02:47",
"updated_at": "2021-07-10 17:26:59",
```

9.6 SHOW GATEWAY

Request: <u>**GET Show Gateways**</u>

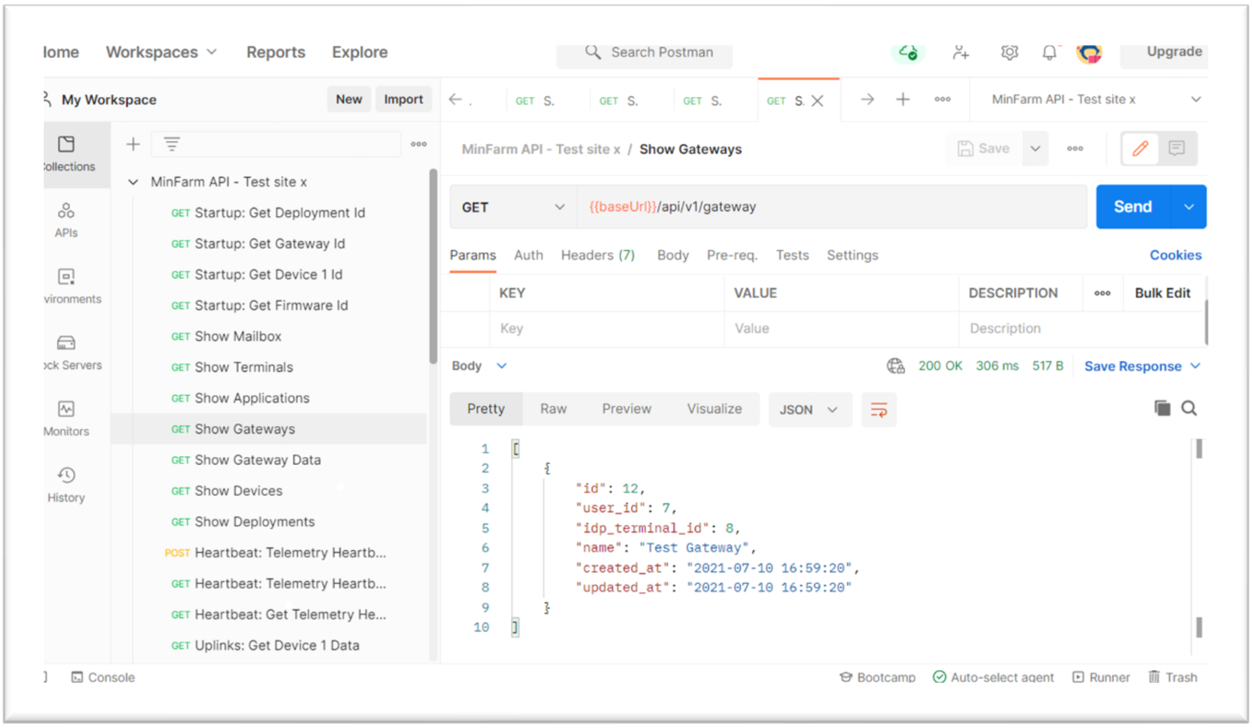
Description: Shows a list of all Gateways created in the MinFarm Dashboard.

Type of Command: Instant.

URL: {{baseUrl}}/api/v1/gateway

Expected Response: 200 OK

Example Response:



MinFarm API - Test site x

- GET Startup: Get Deployment Id
- GET Startup: Get Gateway Id
- GET Startup: Get Device 1 Id
- GET Startup: Get Firmware Id
- GET Show Mailbox
- GET Show Terminals
- GET Show Applications
- GET Show Gateways
- GET Show Gateway Data
- GET Show Devices
- GET Show Deployments
- POST Heartbeat: Telemetry Heartb...
- GET Heartbeat: Telemetry Heartb...
- GET Heartbeat: Get Telemetry He...
- GET Uplinks: Get Device 1 Data

MinFarm API - Test site x / Show Gateways

GET {{baseUrl}}/api/v1/gateway

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body 200 OK 306 ms 517 B

```
[
    {
        "id": 12,
        "user_id": 7,
        "idp_terminal_id": 8,
        "name": "Test Gateway",
        "created_at": "2021-07-10 16:59:20",
        "updated_at": "2021-07-10 16:59:20"
    }
]
```

9.7 SHOW GATEWAY DATA

Request: **GET Show Gateway Data**

Description: Shows the most recent messages sent by the Gateway over the satellite link.

Type of Command: Instant.

URL: {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/data

Expected Response: 200 OK

Example Response:

```
[
    {
        "id": 9371,
        "created_at": "2021-07-13 17:26:03",
        "updated_at": "2021-07-13 17:26:03",
        "message_id": "59463268",
        "message_utc": "2021-07-13 17:25:59",
        "receive_utc": "2021-07-13 17:25:59",
        "mobile_id": "01043241SKYAA0A",
        "region_name": "EMEARB7",
        "sin": "144",
        "payload": "\"\"",
        "ota_message_size": 24,
        "custom_data":
"{\"frame_count_client\":0,\"frame_version\":1,\"tel
emetry_command\":4,\"telemetry_data\":{\"version_maj
or\":2,\"version_minor\":1,\"version_patch\":0,\"cpu
_usage_since_boot\":32,\"cpu_usage_since_last_call\"
:21,\"last_power_on\":\"2021-07-13
17:07:07\"},\"req_frame_count_server_short\":0,\"res
ult\":1,\"mic\":\"4A2E7F44\",\"command_type\":144}"
    }
]
```

9.8 SHOW DEVICES

Request: **<u>GET Show Devices</u>**

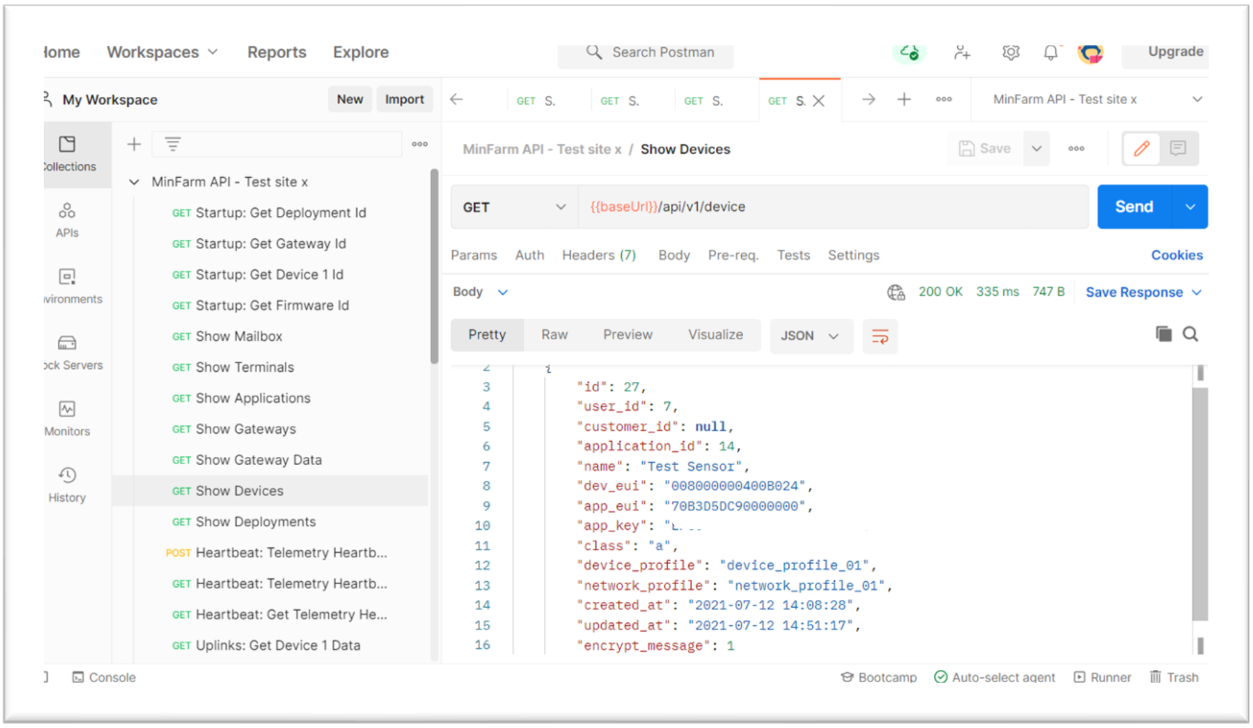
Description: Shows a list of all Devices created in the MinFarm Dashboard.

Type of Command: Instant.

URL: {{baseUrl}}/api/v1/device

Expected Response: 200 OK

Example Response: Note some sensitive information has been blurred out from the screenshot below.



```
GET {{baseUrl}}/api/v1/device
```

200 OK 335 ms 747 B

```
    {
        "id": 27,
        "user_id": 7,
        "customer_id": null,
        "application_id": 14,
        "name": "Test Sensor",
        "dev_eui": "008000000400B024",
        "app_eui": "70B3D5DC90000000",
        "app_key": "[redacted]
        "class": "a",
        "device_profile": "device_profile_01",
        "network_profile": "network_profile_01",
        "created_at": "2021-07-12 14:08:28",
        "updated_at": "2021-07-12 14:51:17",
        "encrypt_message": 1
```

9.9 SHOW DEPLOYMENTS

Request: **<u>GET Show Deployments</u>**

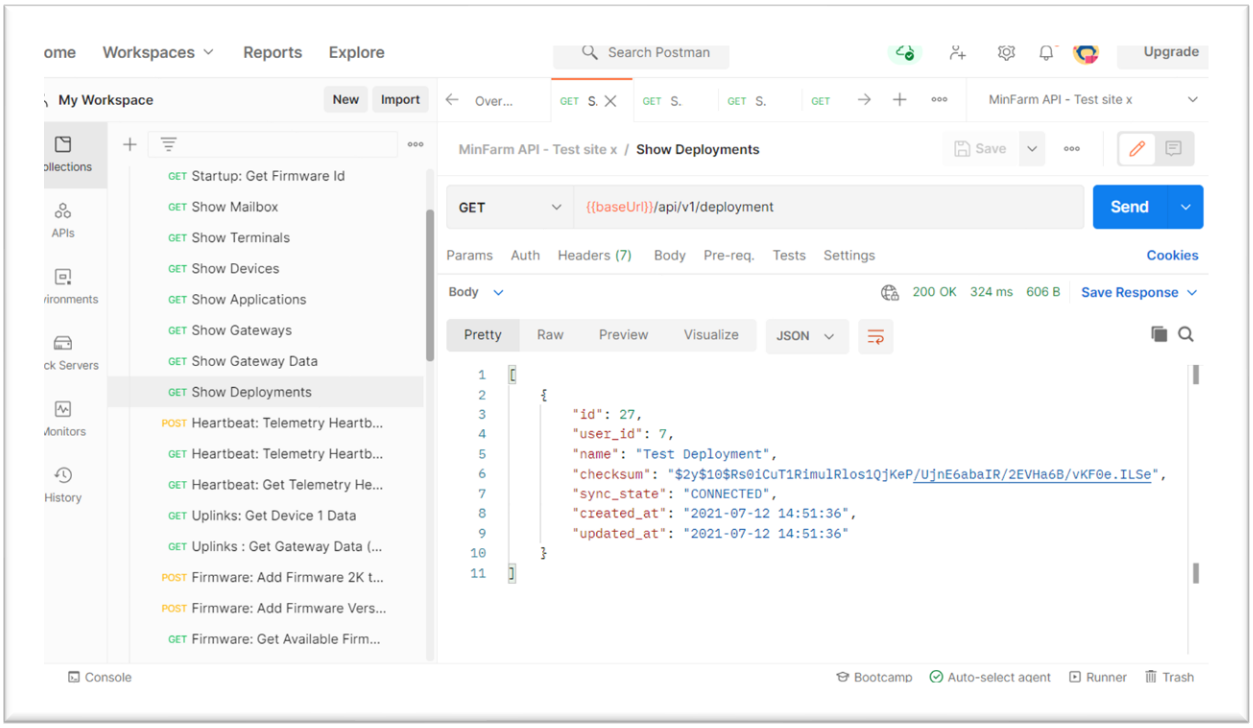
Description: Shows a list of all Deployments created in the MinFarm Dashboard.

Type of Command: Instant.

URL: {{baseUrl}}/api/v1/deployment

Expected Response: 200 OK

Example Response:


GET {{baseUrl}}/api/v1/deployment

200 OK 324 ms 606 B

```
[
    {
        "id": 27,
        "user_id": 7,
        "name": "Test Deployment",
        "checksum": "$2y$10$Rs0iCuT1RimulRlos1QjKeP/UjnE6abaIR/2EVHa6B/vKF0e.ILSe",
        "sync_state": "CONNECTED",
        "created_at": "2021-07-12 14:51:36",
        "updated_at": "2021-07-12 14:51:36"
    }
]
```

9.10 SHOW CUSTOMERS

Request: **GET Show Customers**

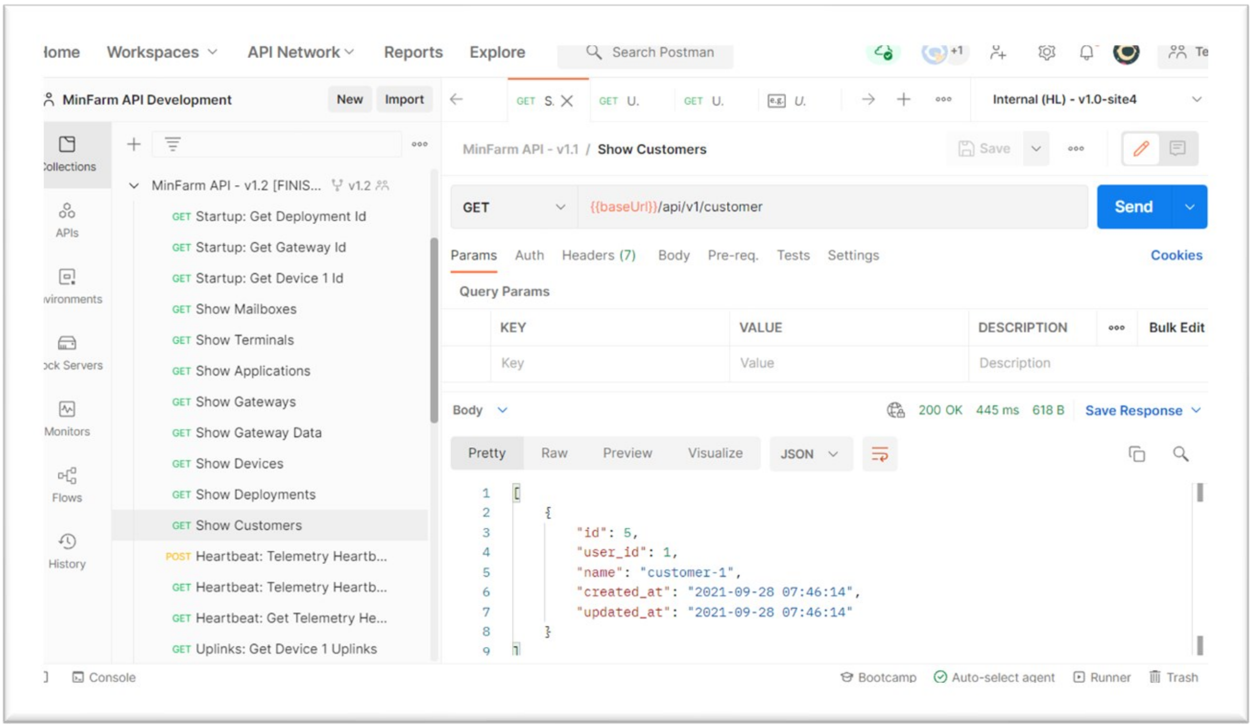
Description: Shows a list of all Customers created in the MinFarm Dashboard.

Type of Command: Instant.

URL: {{baseUrl}}/api/v1/customer

Expected Response: 200 OK

Example Response:



MinFarm API - v1.2 [FINIS... v1.2

- GET Startup: Get Deployment Id
- GET Startup: Get Gateway Id
- GET Startup: Get Device 1 Id
- GET Show Mailboxes
- GET Show Terminals
- GET Show Applications
- GET Show Gateways
- GET Show Gateway Data
- GET Show Devices
- GET Show Deployments
- GET Show Customers
- POST Heartbeat: Telemetry Heartb...
- GET Heartbeat: Telemetry Heartb...
- GET Heartbeat: Get Telemetry He...
- GET Uplinks: Get Device 1 Uplinks

MinFarm API - v1.1 / Show Customers

GET {{baseUrl}}/api/v1/customer

Params Auth Headers (7) Body Pre-req. Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body 200 OK 445 ms 618 B

Pretty Raw Preview Visualize JSON

```
[
    {
        "id": 5,
        "user_id": 1,
        "name": "customer-1",
        "created_at": "2021-09-28 07:46:14",
        "updated_at": "2021-09-28 07:46:14"
    }
]
```

9.11 HEARTBEAT MESSAGE FROM GATEWAY

Request: **POST Heartbeat: Telemetry Heartbeat Start Request**

Description: Initiates a heartbeat message from the Gateway. A heartbeat is a packet of data sent from the Gateway on a regular basis, in this example every 3600 seconds. There are two aspects to the heartbeat: the heartbeat command and subsequent response; and the actual heartbeat pulses. The heartbeat is one way of checking an open communication pathway to the Gateway.

Type of Command: Queued.

URL: {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry

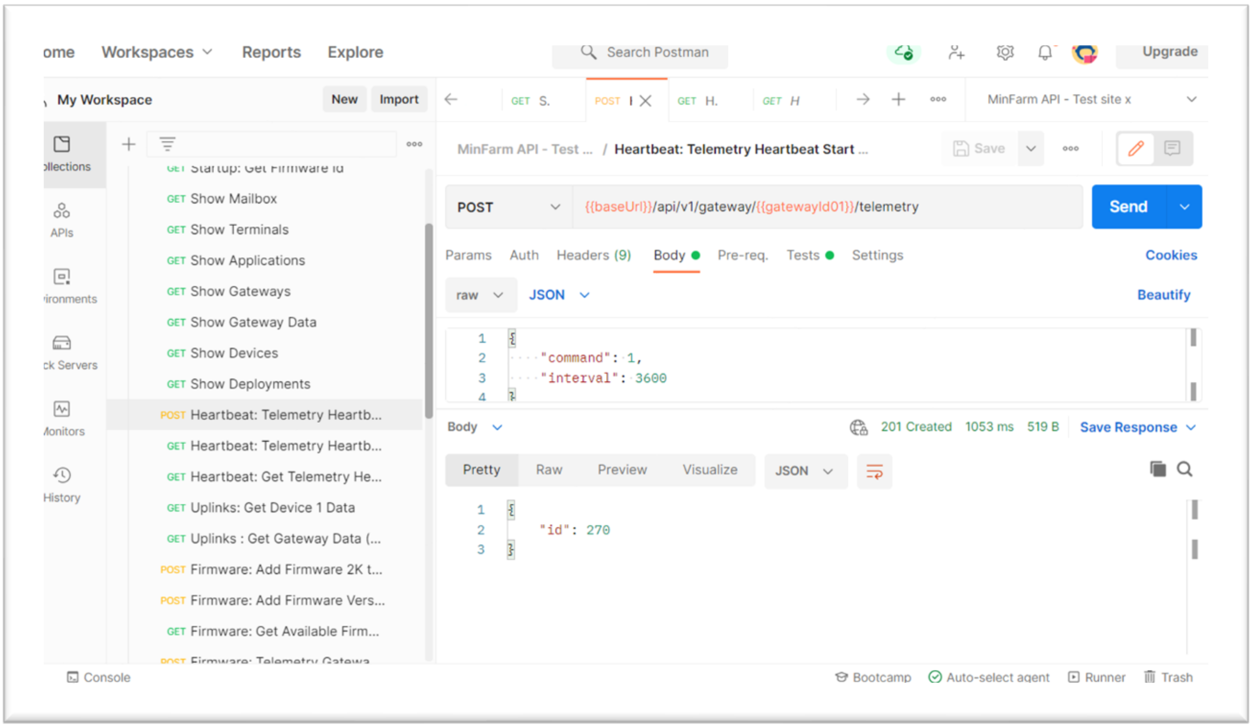
Body:

```
{
    "command": 1,
    "interval": 3600
}
```

A telemetry "command" of 1 initiates a heartbeat command. Interval of 3600 seconds, *i.e.* 1 hour. This heartbeat interval can be changed as required.

Expected Response: 201 Created

Example Response: Returns an id.


ome
Workspaces
Reports
Explore
Search Postman
Upgrade
My Workspace
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
GET Startup: Get Firmware Id
GET Show Mailbox
GET Show Terminals
GET Show Applications
GET Show Gateways
GET Show Gateway Data
GET Show Devices
GET Show Deployments
POST Heartbeat: Telemetry Heartb...
GET Heartbeat: Telemetry Heartb...
GET Heartbeat: Get Telemetry He...
GET Uplinks: Get Device 1 Data
GET Uplinks : Get Gateway Data (...
POST Firmware: Add Firmware 2K t...
POST Firmware: Add Firmware Vers...
GET Firmware: Get Available Firm...
POST Firmware: Telemetry Gatewa...
MinFarm API - Test site x
MinFarm API - Test ... / Heartbeat: Telemetry Heartbeat Start ...
Save
Send
Params
Auth
Headers (9)
Body
Pre-req.
Tests
Settings
Cookies
raw
JSON
Beautify
Body
Save Response
Pretty
Raw
Preview
Visualize
JSON
Console
Bootcamp
Auto-select agent
Runner
Trash

```
POST {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry
```

```
{
    "command": 1,
    "interval": 3600
}
```

201 Created 1053 ms 519 B

```
{
    "id": 270
}
```

Request: **<u>GET Heartbeat: Telemetry Heartbeat Start Status</u>**

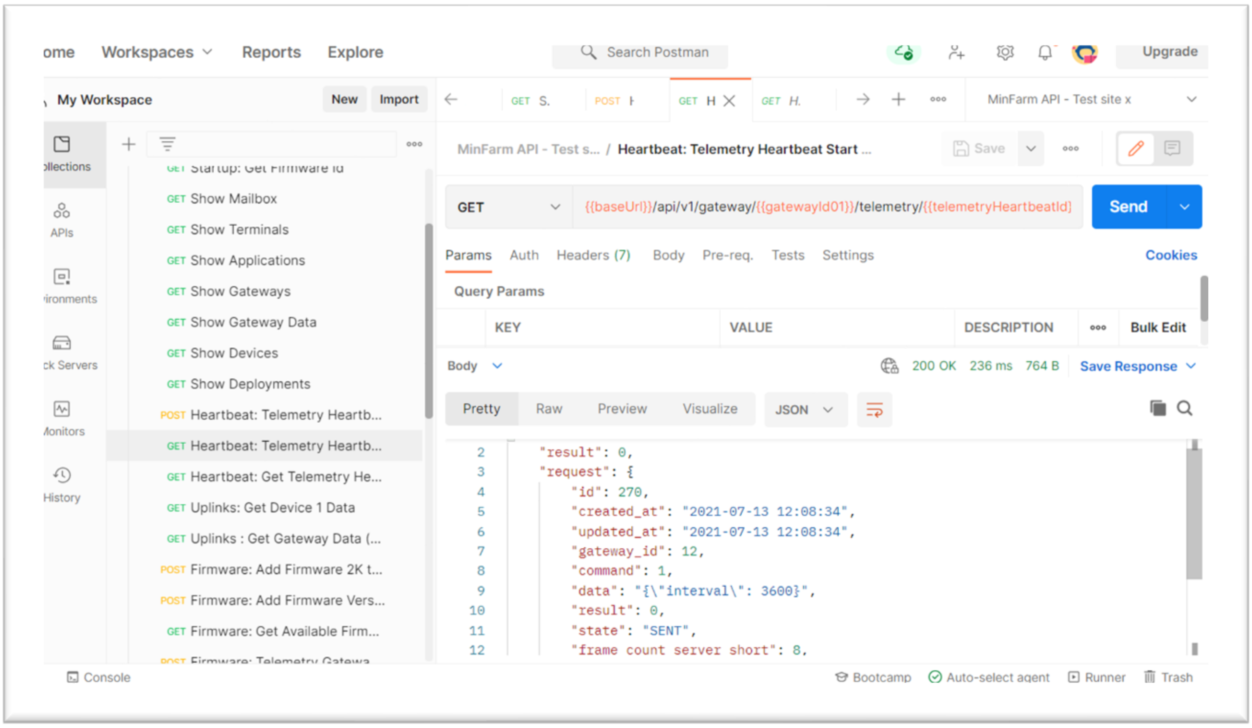
Description: Poll status of queued command.

Type of Command: Queued.

URL: {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry/{{telemetryHeartbeatId}}

Expected Response: 200 OK.

Example Immediate Response: The same id is returned. “result” of 0 to show it is pending.



GET Startup: Get Firmware Id
GET Show Mailbox
GET Show Terminals
GET Show Applications
GET Show Gateways
GET Show Gateway Data
GET Show Devices
GET Show Deployments
POST Heartbeat: Telemetry Heartb...
GET Heartbeat: Telemetry Heartb...
GET Heartbeat: Get Telemetry He...
GET Uplinks: Get Device 1 Data
GET Uplinks : Get Gateway Data (...
POST Firmware: Add Firmware 2K t...
POST Firmware: Add Firmware Vers...
GET Firmware: Get Available Firm...
POST Firmware: Telemetry Gatewa

MinFarm API - Test s... / Heartbeat: Telemetry Heartbeat Start ...

GET {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry/{{telemetryHeartbeatId}} Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION

Body 200 OK 236 ms 764 B Save Response

Pretty Raw Preview Visualize JSON

```
    "result": 0,
    "request": {
        "id": 270,
        "created_at": "2021-07-13 12:08:34",
        "updated_at": "2021-07-13 12:08:34",
        "gateway_id": 12,
        "command": 1,
        "data": "{\"interval\": 3600}",
        "result": 0,
        "state": "SENT",
        "frame count server short": 8,
```

Example Response after approximately 5 minutes: The same id is returned. “result” of 1 to show it is successful.

```
{
    "result": 1,
    "request": {
        "id": 270,
        "created_at": "2021-07-13 12:08:34",
        "updated_at": "2021-07-13 12:08:34",
        "gateway_id": 12,
        "command": 1,
        "data": "{\"interval\": 3600}",
        "result": 1,
        "state": "FINISHED",
        "frame_count_server_short": 9,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a5032652232389 37df18fa77ac236b
2b9dded6068740ec2ca539b28"
    },
    "response": {
        "id": 597,
        "created_at": "2021-07-13 12:10:03",
        "updated_at": "2021-07-13 12:10:03",
        "gateway_id": 12,
        "command": 1,
        "data": "{}",
        "result": 1,
        "state": "FINISHED",
        "req_frame_count_server_short": 9,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28",
        "telemetry_req_id": 270
    }
}
```

Request: **GET Heartbeat: Get Telemetry Heartbeat Pulses**

Description: Shows heartbeat pulses from Gateway - at specified interval.

Type of Command: Instant.

URL: {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry/heartbeat-pulses

Expected Response: 200 OK.

Example Response: 3600 seconds after Request is sent. With an interval of 3600 seconds expect one heartbeat message per hour. 3 heartbeats are shown in the example below.

```
[
    {
        "id": 561,
        "created_at": "2021-07-13 15:10:02",
        "updated_at": "2021-07-13 15:10:02",
        "gateway_id": 12,
        "command": 2,
        "data": "{}",
        "result": 1,
        "state": "FINISHED",
        "req_frame_count_server_short": 0,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28",
        "telemetry_req_id": null
    },
    {
        "id": 599,
        "created_at": "2021-07-13 14:10:02",
        "updated_at": "2021-07-13 14:10:02",
        "gateway_id": 12,
        "command": 2,
        "data": "{}",
        "result": 1,
        "state": "FINISHED",
        "req_frame_count_server_short": 0,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28",
        "telemetry_req_id": null
    },
    {
        "id": 598,
        "created_at": "2021-07-13 13:11:04",
        "updated_at": "2021-07-13 13:11:04",
        "gateway_id": 12,
        "command": 2,
        "data": "{}",
        "result": 1,
        "state": "FINISHED",
        "req_frame_count_server_short": 0,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28",
        "telemetry_req_id": null
    }
]
```

9.12 UPLINKS: GET DEVICE 1 UPLINKS

Request: **GET Uplinks: Get Device 1 Uplinks**

Description: Shows any uplink messages coming from the Device. Note different Devices have different reporting intervals.

Type of Command: Instant.

URL: {{baseUrl}}/api/v1/device/{{deviceId1}}/data

Expected Response: 200 OK.

Example Response:

```
[
    {
        "id": 7874,
        "created_at": "2021-07-13 15:45:03",
        "updated_at": "2021-07-13 15:45:03",
        "gateway_id": 12,
        "idp_uplink_id": 9317,
        "dev_eui": "008000000400B024",
        "dev_f_port": 1,
        "dev_f_cnt_up": 104,
        "dev_frm_payload": 
"00000000000000000000800A15",
        "state": "SENDING"
    }
]
```

9.13 UPLINKS: GET GATEWAY UPLINKS

Request: **GET Uplinks: Get Gateway Uplinks**

Description: Shows uplinks received from the Gateway for all Devices associated with this Gateway. The most recent 50 uplinks are returned.

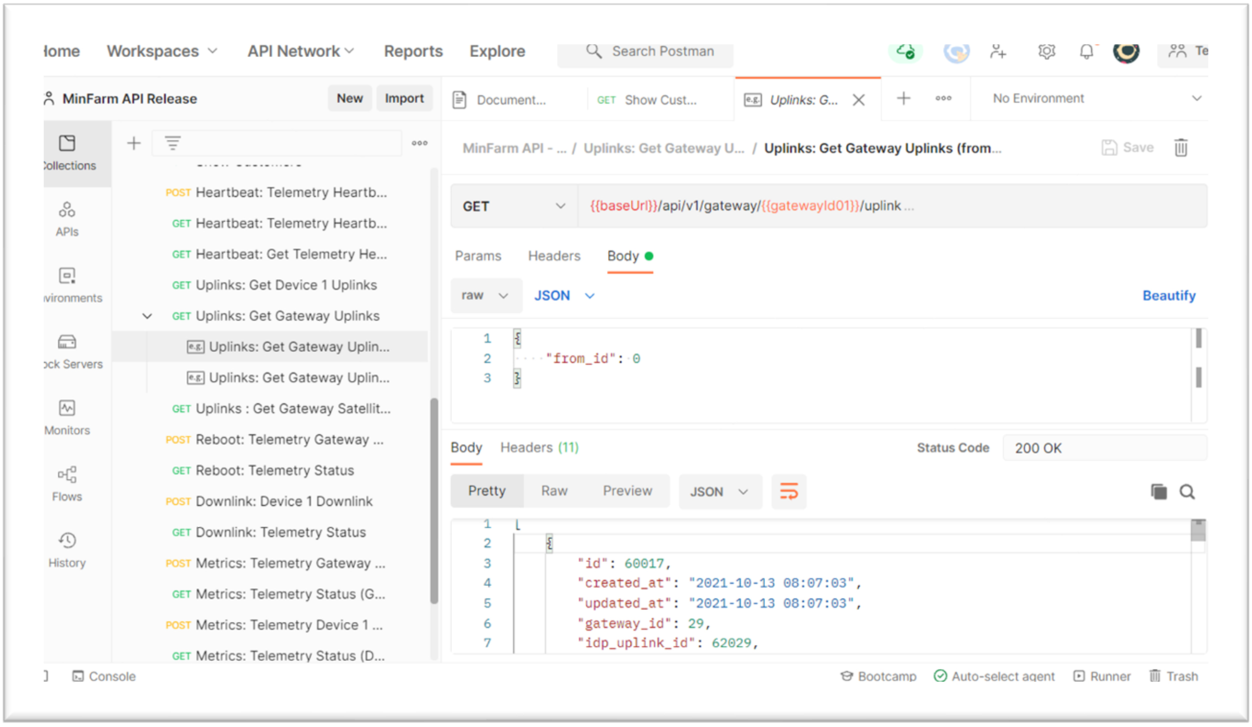
Type of Command: Instant.

URL: {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/uplink

Expected Response: 200 OK.

Two examples are given in your Collection. If from_id is set to 0, the oldest 50 uplinks are returned. If from_id is omitted, the most recent 50 uplinks are returned.

Example Response 1 where a body of from_id set to 0 is added:



POST Heartbeat: Telemetry Heartb...
GET Heartbeat: Telemetry Heartb...
GET Heartbeat: Get Telemetry He...
GET Uplinks: Get Device 1 Uplinks
GET Uplinks: Get Gateway Uplinks
Uplinks: Get Gateway Uplin...
Uplinks: Get Gateway Uplin...
GET Uplinks : Get Gateway Satellit...
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Downlink: Device 1 Downlink
GET Downlink: Telemetry Status
POST Metrics: Telemetry Gateway ...
GET Metrics: Telemetry Status (G...
POST Metrics: Telemetry Device 1 ...
GET Metrics: Telemetry Status (D...

MinFarm API - ... / Uplinks: Get Gateway U... / Uplinks: Get Gateway Uplinks (from...

GET {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/uplink ...

```
{
    "from_id": 0
}
```

Body Headers (11) Status Code 200 OK

```
[
    {
        "id": 60017,
        "created_at": "2021-10-13 08:07:03",
        "updated_at": "2021-10-13 08:07:03",
        "gateway_id": 29,
        "idp_uplink_id": 62029,
```

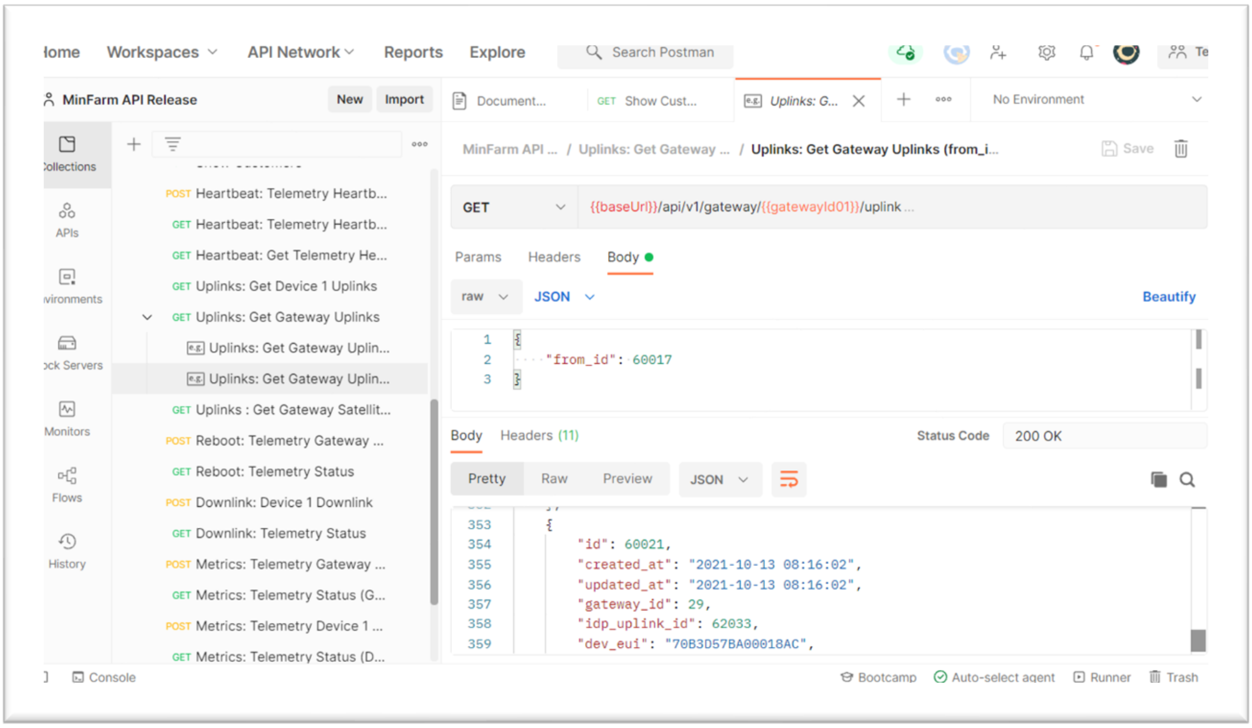
Example of two uplinks received:

```
[
    {
        "id": 60017,
        "created_at": "2021-10-13 08:07:03",
        "updated_at": "2021-10-13 08:07:03",
        "gateway_id": 29,
        "idp_uplink_id": 62029,
        "dev_eui": "98208E0000001590",
        "dev_f_port": 1,
        "dev_f_cnt_up": 138,
        "dev_frm_payload": "80C005D2F241A8BE77",
        "state": "FINISHED",
        "gateway_timestamp": "2021-10-13 08:05:54"
    },
    {
        "id": 60016,
        "created_at": "2021-10-13 08:06:03",
        "updated_at": "2021-10-13 08:06:03",
        "gateway_id": 29,
        "idp_uplink_id": 62028,
        "dev_eui": "98208E0000001590",
        "dev_f_port": 1,
        "dev_f_cnt_up": 137,
        "dev_frm_payload":
"2000026CF60000000000000000000000",
        "state": "FINISHED",
        "gateway_timestamp": "2021-10-13 08:05:13"
    }
]
```

A good way of using this endpoint is to initially call the endpoint with from_id set to 0 (to get the oldest uplinks) or to omit from_id (to get the latest uplinks). Store the id of the most recent uplink returned. When calling the endpoint again, set from_id to this stored id. The endpoint will only return new uplinks with an id greater than from_id.

Example Response 2 where a body of from_id set to 60017 is added:

All ids greater than 60017 are returned.


Home
Workspaces
API Network
Reports
Explore
Search Postman
MinFarm API Release
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
POST Heartbeat: Telemetry Heartb...
GET Heartbeat: Telemetry Heartb...
GET Heartbeat: Get Telemetry He...
GET Uplinks: Get Device 1 Uplinks
GET Uplinks: Get Gateway Uplinks
Uplinks: Get Gateway Uplin...
Uplinks: Get Gateway Uplin...
GET Uplinks : Get Gateway Satellit...
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Downlink: Device 1 Downlink
GET Downlink: Telemetry Status
POST Metrics: Telemetry Gateway ...
GET Metrics: Telemetry Status (G...
POST Metrics: Telemetry Device 1 ...
GET Metrics: Telemetry Status (D...
Document...
GET Show Cust...
Uplinks: G...
No Environment
Save
Params
Headers
Body
raw
JSON
Beautify
Body
Headers (11)
Pretty
Raw
Preview
JSON
Console
Bootcamp
Auto-select agent
Runner
Trash

MinFarm API ... / Uplinks: Get Gateway ... / Uplinks: Get Gateway Uplinks (from_i...

GET {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/uplink ...

```
{
    "from_id": 60017
}
```

Status Code 200 OK

```
    {
        "id": 60021,
        "created_at": "2021-10-13 08:16:02",
        "updated_at": "2021-10-13 08:16:02",
        "gateway_id": 29,
        "idp_uplink_id": 62033,
        "dev_eui": "70B3D57BA00018AC",
```

9.14 UPLINKS: GET GATEWAY SATELLITE MESSAGES FOR DEVICE 1

Request: **GET Uplinks : Get Gateway Satellite Messages (Filter for Device 1)**

Description: Shows satellite messages from the Gateway filtered for a specific Device. This gives extra information than the command in Section 9.12 above, in that it returns the entire message that is sent over satellite, including for example ota message size.

Type of Command: Instant.

URL: {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/data

Body:

```
{
        "search": "008000000400B024"
}
```

Enter the dev_eui of the Device you would like information on. This is the same dev_eui as used when setting up a Deployment in the MinFarm Dashboard.

Expected Response: 200 OK.

Example Response where “dev_eui” of Device is `008000000400B024`:

```
[
    {
        "id": 9317,
        "created_at": "2021-07-13 15:45:03",
        "updated_at": "2021-07-13 15:45:03",
        "message_id": "59462588",
        "message_utc": "2021-07-13 15:44:58",
        "receive_utc": "2021-07-13 15:44:58",
        "mobile_id": "01043241SKYAA0A",
        "region_name": "EMEARB7",
        "sin": "30",
        "payload": "\"\"",
        "ota_message_size": 22,
        "custom_data":
"{\"frame_count_client\":43,\"dev_f_cnt_up\":104,\"d
ev_frm_payload\":\"0000000000000000000800A15\",\"mi
c\":\"14413CDA\",\"dev_eui\":\"008000000400B024\",\"
dev_f_port\":1,\"encrypt_message\":1,\"command_type\
":30}"
    }
]
```

9.15 DOWNLINK: DEVICE 1

Request: **POST Downlink: Device 1 Downlink**

Description: Initiates remote downlink configuration of Device, *e.g.* you can select to change the reporting period of a Device.

Type of Command: Queued.

URL: {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry

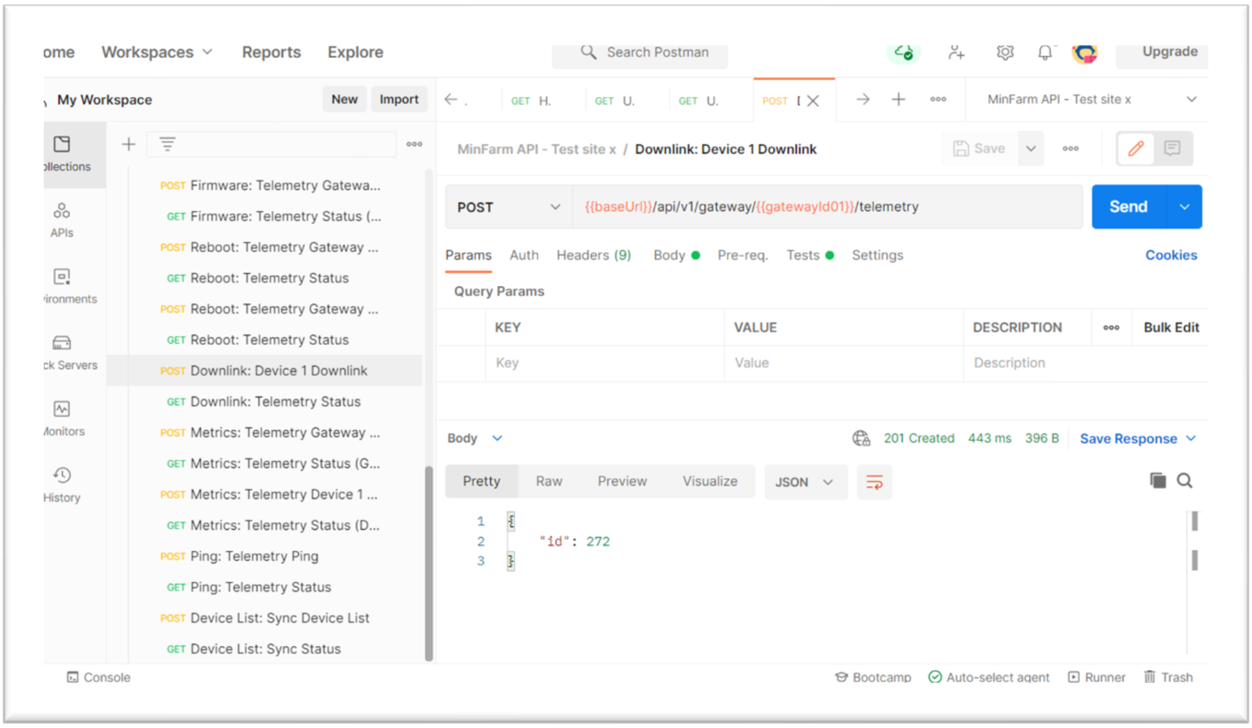
Body:

```
{
    "command": 7,
    "dev_eui": "008000000400B024",
    "dev_f_port": 100,
    "dev_frm_payload": "7265706F727420706572696F6420
31383030"
}
```

A telemetry "command" of 7 initiates remote configuration of your Device. Enter the "dev_eui" of the Device you would like to select. This is the same "dev_eui" as used when setting up a Deployment in the MinFarm Dashboard. Also, enter "dev_frm_payload". "dev_frm_payload" is the payload in hex. This is Device specific, and you will need to contact the manufacturer for this information.

Expected Response: 201 Created.

Example Response: Returns an id.



POST Firmware: Telemetry Gatewa...
GET Firmware: Telemetry Status (...
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Downlink: Device 1 Downlink
GET Downlink: Telemetry Status
POST Metrics: Telemetry Gateway ...
GET Metrics: Telemetry Status (G...
POST Metrics: Telemetry Device 1 ...
GET Metrics: Telemetry Status (D...
POST Ping: Telemetry Ping
GET Ping: Telemetry Status
POST Device List: Sync Device List
GET Device List: Sync Status

MinFarm API - Test site x / Downlink: Device 1 Downlink

POST {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry

Params Auth Headers (9) Body ● Pre-req. Tests ● Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body 201 Created 443 ms 396 B Save Response

Pretty Raw Preview Visualize JSON

```
{
    "id": 272
}
```

Request: **GET Downlink: Telemetry Status**

Description: Poll status of queued command.

Type of Command: Queued.

URL:
{{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry/{{telemetryDownlinkId}}

Expected Response: 200 OK.

Example Immediate Response: The same id is returned. “result” of 0 to show it is pending.

```
{
    "result": 0,
    "request": {
        "id": 272,
        "created_at": "2021-07-13 16:00:04",
        "updated_at": "2021-07-13 16:00:04",
        "gateway_id": 12,
        "command": 7,
        "data": "{\"dev_eui\": \"008000000400B024\",
\"dev_f_port\": 100, \"dev_frm_payload\":
\"7265706F727420706572696F642031383030\"}",
        "result": 0,
        "state": "QUEUED",
        "frame_count_server_short": null,
        "network_session_id": null
    },
    "response": []
}
```

Example Response after approximately 5 minutes: The same id is returned. "result" of 1 to show it is successfully scheduled for transmission to the Device by the Gateway.

```
{
    "result": 1,
    "request": {
        "id": 272,
        "created_at": "2021-07-13 16:00:04",
        "updated_at": "2021-07-13 16:00:04",
        "gateway_id": 12,
        "command": 7,
        "data": "{\"dev_eui\": \"008000000400B024\",
\"dev_f_port\": 100, \"dev_frm_payload\":
\"7265706F727420706572696F642031383030\"}",
        "result": 1,
        "state": "FINISHED",
        "frame_count_server_short": 10,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28"
    },
    "response": {
        "id": 598,
        "created_at": "2021-07-13 16:03:02",
        "updated_at": "2021-07-13 16:03:02",
        "gateway_id": 12,
        "command": 7,
        "data": "{}",
        "result": 1,
        "state": "FINISHED",
        "req_frame_count_server_short": 10,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28",
        "telemetry_req_id": 272
    }
}
```

9.16 GATEWAY METRICS

Request: **POST Metrics: Telemetry Gateway Metrics**

Description: Initiates Gateway metrics. The firmware version is returned, and last Gateway reboot time.

Type of Command: Queued.

URL: {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry

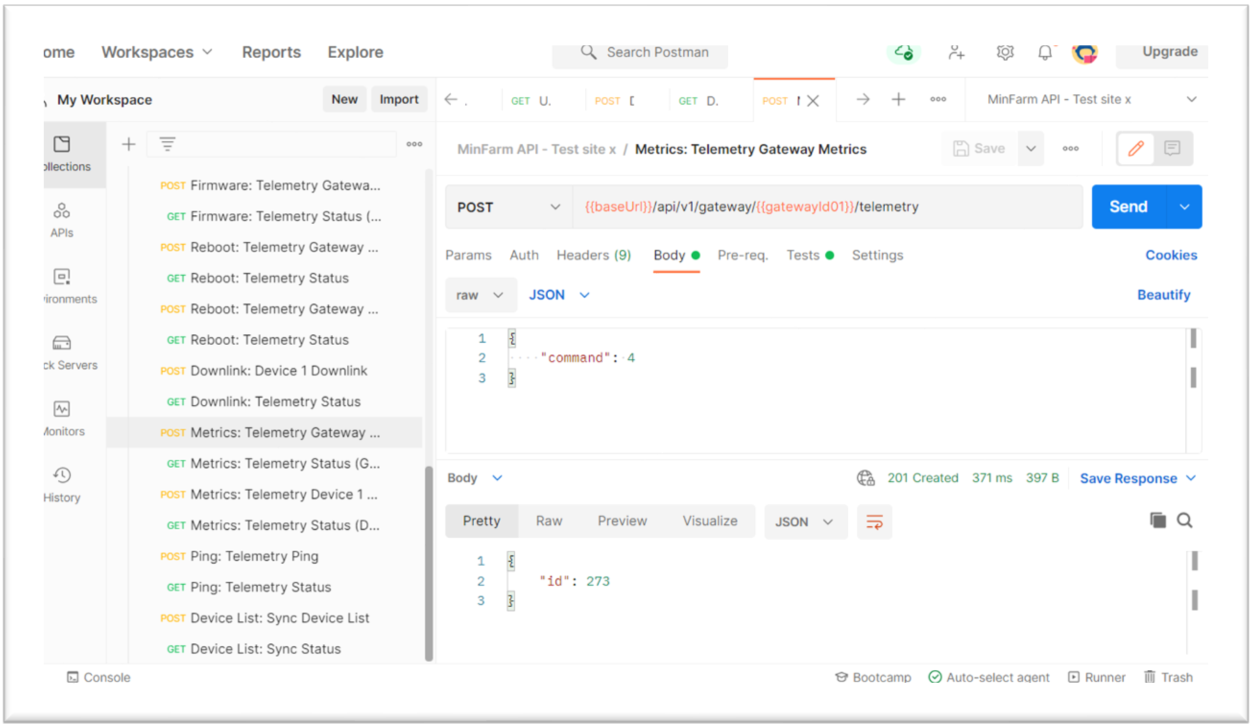
Body:

```
{
    "command": 4
}
```

A telemetry “command” of 4 initiates retrieval of Gateway metrics.

Expected Response: 201 Created.

Example Response: Returns an id.



POST Firmware: Telemetry Gatewa...
GET Firmware: Telemetry Status (...
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Downlink: Device 1 Downlink
GET Downlink: Telemetry Status
POST Metrics: Telemetry Gateway ...
GET Metrics: Telemetry Status (G...
POST Metrics: Telemetry Device 1 ...
GET Metrics: Telemetry Status (D...
POST Ping: Telemetry Ping
GET Ping: Telemetry Status
POST Device List: Sync Device List
GET Device List: Sync Status

MinFarm API - Test site x / Metrics: Telemetry Gateway Metrics

POST {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry

```
{
····"command": 4
}
```

Body 201 Created 371 ms 397 B

```
{
    "id": 273
}
```

Request: **GET Metrics: Telemetry Status (Gateway)**

Description: Poll status of queued command.

Type of Command: Queued.

URL: {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry/{{telemetryMetricsGatewayId}}

Expected Response: 200 OK.

Example Immediate Response: The same id is returned. “result” of 0 to show it is pending.

```
{
    "result": 0,
    "request": {
        "id": 273,
        "created_at": "2021-07-13 16:24:59",
        "updated_at": "2021-07-13 16:24:59",
        "gateway_id": 12,
        "command": 4,
        "data": "[]",
        "result": 0,
        "state": "SENT",
        "frame_count_server_short": 11,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28"
    },
    "response": []
}
```

Example Response after approximately 5 minutes: The same id is returned. “result” of 1 to show it is successful. “last_power_on”, and firmware version* are shown.

* firmware version seen as `"version_major\":2,\"version_minor\":1,\"version_patch\":0` *i.e.* 2.1.0 in this example.

```
{
    "result": 1,
    "request": {
        "id": 273,
        "created_at": "2021-07-13 16:24:59",
        "updated_at": "2021-07-13 16:24:59",
        "gateway_id": 12,
        "command": 4,
        "data": "[]",
        "result": 1,
        "state": "FINISHED",
        "frame_count_server_short": 11,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28"
    },
    "response": {
        "id": 600,
        "created_at": "2021-07-13 16:26:03",
        "updated_at": "2021-07-13 16:26:03",
        "gateway_id": 12,
        "command": 4,
        "data": "{\"last_power_on\": \"2021-07-12 
15:04:10\", \"version_major\": 2, \"version_minor\": 
1, \"version_patch\": 0, \"cpu_usage_since_boot\": 
21, \"cpu_usage_since_last_call\": 0}",
        "result": 1,
        "state": "FINISHED",
        "req_frame_count_server_short": 11,
        "network_session_id": 
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28",
        "telemetry_req_id": 273
    }
}
```

9.17 DEVICE METRICS

Request: **POST Metrics: Telemetry Device 1 Metrics**

Description: Initiates retrieval of Device metrics. Various Device parameters are returned, *e.g.* “per”, snr”, “rssi”, “uplink_sf”, “downlink_sf”, “last_seen_time”.

“per”: packet error rate

“snr”: signal to noise ratio

“rssi”: received signal strength indicator

“uplink_sf”: uplink spreading factor

“downlink_sf”: downlink spreading factor

“last_seen_time”: last message from Device

Type of Command: Queued.

URL: {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry

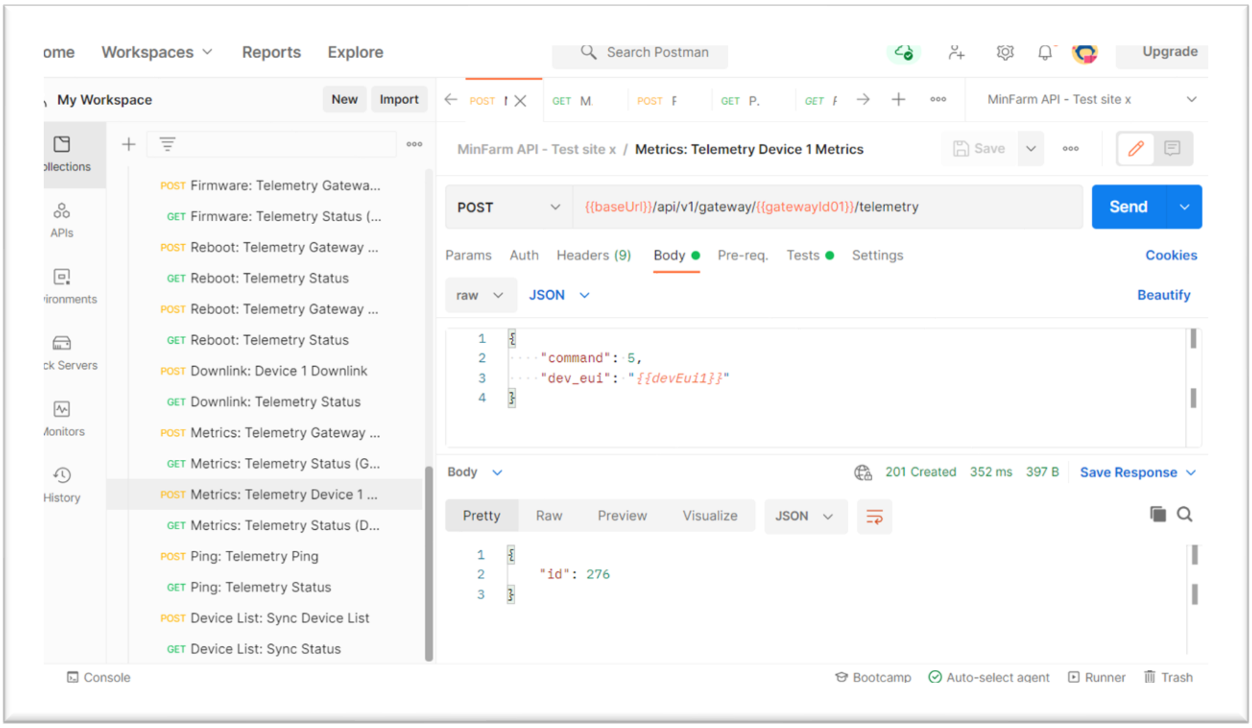
Body:

```
{
    "command": 5,
    "dev_eui": "008000000400B024"
}
```

A telemetry “command” of 5 initiates retrieval of Device metrics. Enter the dev_eui of the Device you would like metrics on. This is the same dev_eui as used when setting up a Deployment in the MinFarm Dashboard.

Expected Response: 201 Created.

Example Response: Returns an id.



MinFarm API - Test site x / Metrics: Telemetry Device 1 Metrics

POST {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry

```
{
    "command": 5,
    "dev_eui": "{{devEui1}}"
}
```

201 Created 352 ms 397 B

```
{
    "id": 276
}
```

ome
Workspaces
Reports
Explore
Search Postman
Upgrade
My Workspace
New
Import
POST Firmware: Telemetry Gatewa...
GET Firmware: Telemetry Status (...
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Downlink: Device 1 Downlink
GET Downlink: Telemetry Status
POST Metrics: Telemetry Gateway ...
GET Metrics: Telemetry Status (G...
POST Metrics: Telemetry Device 1 ...
GET Metrics: Telemetry Status (D...
POST Ping: Telemetry Ping
GET Ping: Telemetry Status
POST Device List: Sync Device List
GET Device List: Sync Status
Send
Params
Auth
Headers (9)
Body
Pre-req.
Tests
Settings
Cookies
raw
JSON
Beautify
Save Response
Pretty
Raw
Preview
Visualize
Console
Bootcamp
Auto-select agent
Runner
Trash

Request: **GET Metrics: Telemetry Status (Device)**

Description: Poll status of queued command.

Type of Command: Queued.

URL:
{{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry/{{telemetryMetricsDeviceId}}

Expected Response: 200 OK.

Example Immediate Response: The same id is returned. “result” of 0 to show it is pending.

```
{
    "result": 0,
    "request": {
        "id": 276,
        "created_at": "2021-07-13 16:54:41",
        "updated_at": "2021-07-13 16:54:41",
        "gateway_id": 12,
        "command": 5,
        "data": "{\"dev_eui\":
\"008000000400B024\"}",
        "result": 0,
        "state": "SENT",
        "frame_count_server_short": 14,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28"
    },
    "response": []
}
```

Example Response after approximately 5 minutes: The same id is returned. “result” of 1 to show it is successful.

```
{
    "result": 1,
    "request": {
        "id": 276,
        "created_at": "2021-07-13 16:54:41",
        "updated_at": "2021-07-13 16:54:41",
        "gateway_id": 12,
        "command": 5,
        "data": "{\"dev_eui\":
\"008000000400B024\"}",
        "result": 1,
        "state": "FINISHED",
        "frame_count_server_short": 14,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28"
    },
    "response": {
        "id": 603,
        "created_at": "2021-07-13 16:57:03",
        "updated_at": "2021-07-13 16:57:03",
        "gateway_id": 12,
        "command": 5,
        "data": "{\"per\": 0, \"snr\": 9, \"rssi\":
-54, \"join_time\": \"2021-01-15 12:11:15\",
\"uplink_sf\": \"SF12BW125\", \"downlink_sf\":
\"SF10BW125\", \"last_seen_time\": \"2021-07-13
16:43:02\"}",
        "result": 1,
        "state": "FINISHED",
        "req_frame_count_server_short": 14,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28",
        "telemetry_req_id": 276
    }
}
```

9.18 GATEWAY PING

Request: **POST Ping: Telemetry Ping**

Description: Initiates a ping to the Gateway. A quick way of seeing if your Gateway is active and receiving data.

Type of Command: Queued.

URL: {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry

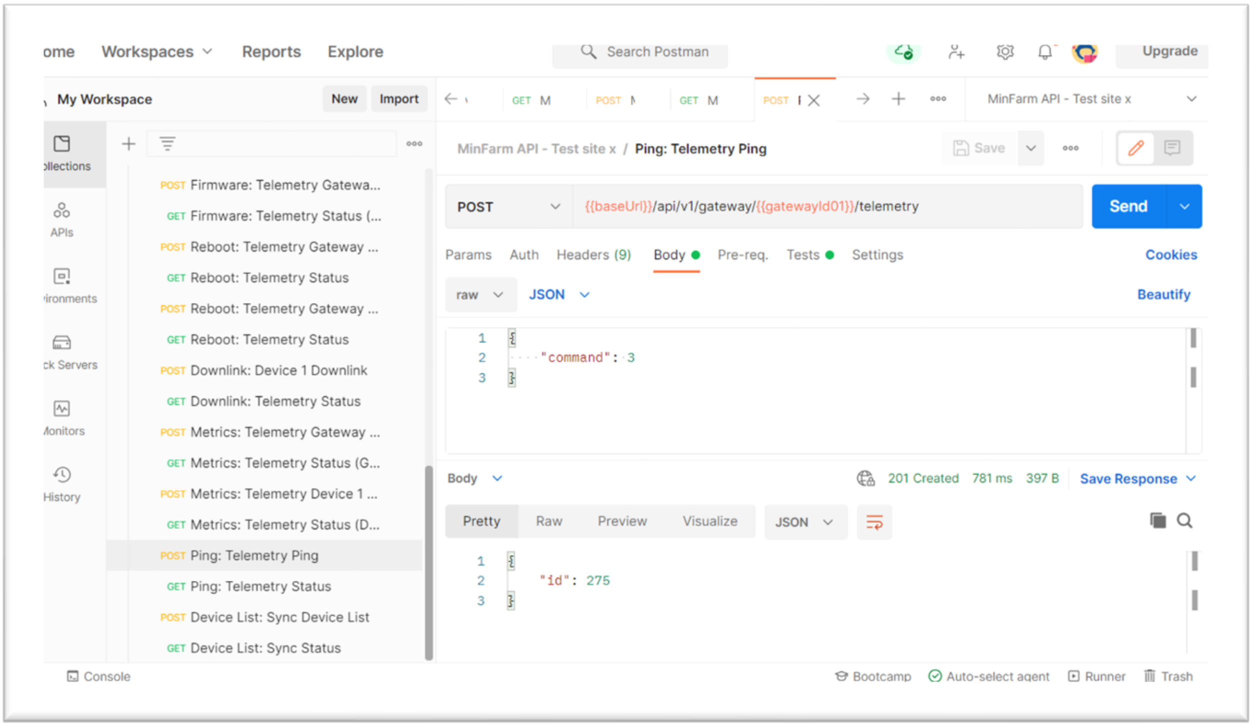
Body:

```
{
    "command": 3
}
```

A telemetry “command” of 3 initiates a Gateway ping.

Expected Response: 201 Created.

Example Response: Returns an id.



ome Workspaces Reports Explore Search Postman Upgrade

My Workspace New Import

MinFarm API - Test site x

POST Firmware: Telemetry Gatewa...
GET Firmware: Telemetry Status (...
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Downlink: Device 1 Downlink
GET Downlink: Telemetry Status
POST Metrics: Telemetry Gateway ...
GET Metrics: Telemetry Status (G...
POST Metrics: Telemetry Device 1 ...
GET Metrics: Telemetry Status (D...
POST Ping: Telemetry Ping
GET Ping: Telemetry Status
POST Device List: Sync Device List
GET Device List: Sync Status

MinFarm API - Test site x / Ping: Telemetry Ping

POST {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

raw JSON Beautify

```
{
    "command": 3
}
```

Body 201 Created 781 ms 397 B Save Response

Pretty Raw Preview Visualize JSON

```
{
    "id": 275
}
```

Console Bootcamp Auto-select agent Runner Trash

Request: **<u>GET Ping: Telemetry Status</u>**

Description: Poll status of queued command.

Type of Command: Queued.

URL:
{{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry/{{telemetryPingId}}

Expected Response: 200 OK.

Example Immediate Response: The same id is returned. “result” of 0 to show it is pending.

```
{
    "result": 0,
    "request": {
        "id": 275,
        "created_at": "2021-07-13 16:45:08",
        "updated_at": "2021-07-13 16:45:08",
        "gateway_id": 12,
        "command": 3,
        "data": "[]",
        "result": 0,
        "state": "QUEUED",
        "frame_count_server_short": null,
        "network_session_id": null
    },
    "response": []
}
```

Example Response after approximately 5 minutes: The same id is returned. “result” of 1 to show it is successful.

```
{
    "result": 1,
    "request": {
        "id": 275,
        "created_at": "2021-07-13 16:45:08",
        "updated_at": "2021-07-13 16:45:08",
        "gateway_id": 12,
        "command": 3,
        "data": "[]",
        "result": 1,
        "state": "FINISHED",
        "frame_count_server_short": 13,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28"
    },
    "response": {
        "id": 602,
        "created_at": "2021-07-13 16:47:02",
        "updated_at": "2021-07-13 16:47:02",
        "gateway_id": 12,
        "command": 3,
        "data": "{}",
        "result": 1,
        "state": "FINISHED",
        "req_frame_count_server_short": 13,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28",
        "telemetry_req_id": 275
    }
}
```

9.19 GATEWAY REBOOT

Request: **POST Reboot: Telemetry Gateway Reboot**

Description: Initiates reboot of Gateway.

Type of Command: Queued.

URL: {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry

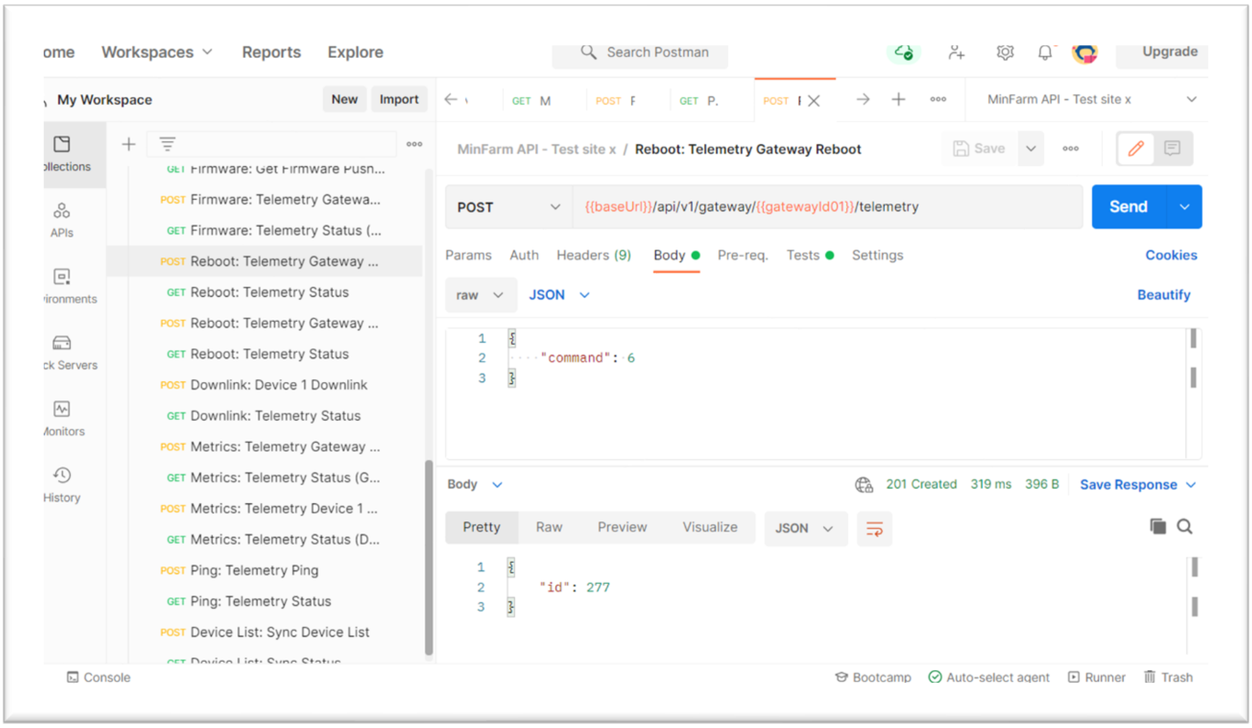
Body:

```
{
    "command": 6
}
```

A telemetry "command" of 6 initiates a Gateway restart.

Expected Response: 201 Created.

Example Response: An id is returned.


ome
Workspaces
Reports
Explore
Search Postman
Upgrade
My Workspace
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
GET Firmware: Get Firmware Push...
POST Firmware: Telemetry Gatewa...
GET Firmware: Telemetry Status (...
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Reboot: Telemetry Gateway ...
GET Reboot: Telemetry Status
POST Downlink: Device 1 Downlink
GET Downlink: Telemetry Status
POST Metrics: Telemetry Gateway ...
GET Metrics: Telemetry Status (G...
POST Metrics: Telemetry Device 1 ...
GET Metrics: Telemetry Status (D...
POST Ping: Telemetry Ping
GET Ping: Telemetry Status
POST Device List: Sync Device List
GET Device List: Sync Status
MinFarm API - Test site x
Save
Send
Params
Auth
Headers (9)
Body
Pre-req.
Tests
Settings
Cookies
Beautify
Pretty
Raw
Preview
Visualize
Save Response
Console
Bootcamp
Auto-select agent
Runner
Trash

MinFarm API - Test site x / Reboot: Telemetry Gateway Reboot

POST {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry

```
{
    "command": 6
}
```

Body — 201 Created 319 ms 396 B — JSON

```
{
    "id": 277
}
```

Request: **GET Reboot: Telemetry Status**

Description: Poll status of queued command.

Type of Command: Queued.

URL: {{baseUrl}}/api/v1/gateway/{{gatewayId01}}/telemetry/{{telemetryRebootId}}

Expected Response: 200 OK.

Example Immediate Response: The same id is returned. “result” of 0 to show it is pending.

```
{
    "result": 0,
    "request": {
        "id": 277,
        "created_at": "2021-07-13 17:04:50",
        "updated_at": "2021-07-13 17:04:50",
        "gateway_id": 12,
        "command": 6,
        "data": "[]",
        "result": 0,
        "state": "QUEUED",
        "frame_count_server_short": null,
        "network_session_id": null
    },
    "response": []
}
```

Example Response after approximately 5 minutes: The same id is returned. “result” of 1 to show it is successful.

```
{
    "result": 1,
    "request": {
        "id": 277,
        "created_at": "2021-07-13 17:04:50",
        "updated_at": "2021-07-13 17:04:50",
        "gateway_id": 12,
        "command": 6,
        "data": "[]",
        "result": 1,
        "state": "FINISHED",
        "frame_count_server_short": 15,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28"
    },
    "response": {
        "id": 604,
        "created_at": "2021-07-13 17:07:03",
        "updated_at": "2021-07-13 17:07:03",
        "gateway_id": 12,
        "command": 6,
        "data": "{}",
        "result": 1,
        "state": "FINISHED",
        "req_frame_count_server_short": 15,
        "network_session_id":
"6dbc4662ba4ea29c05f6e82d2d34a40b21ba9c4e72bfc03a191
5216f1ed5a75efd5fd9672a503265223238937df18fa77ac236b
2b9dded6068740ec2ca539b28",
          "telemetry_req_id": 277
     }
}
```

After approximately 20 minutes check the Gateway metrics as described in Section 9.16. The Gateway reboot time is displayed in the field “last_power_on”.

10.0 CONTACT DETAILS

MINFARMTECH

MinFarm Tech Ltd

Webpage: www.minfarmtech.com

Email: support@minfarm.se

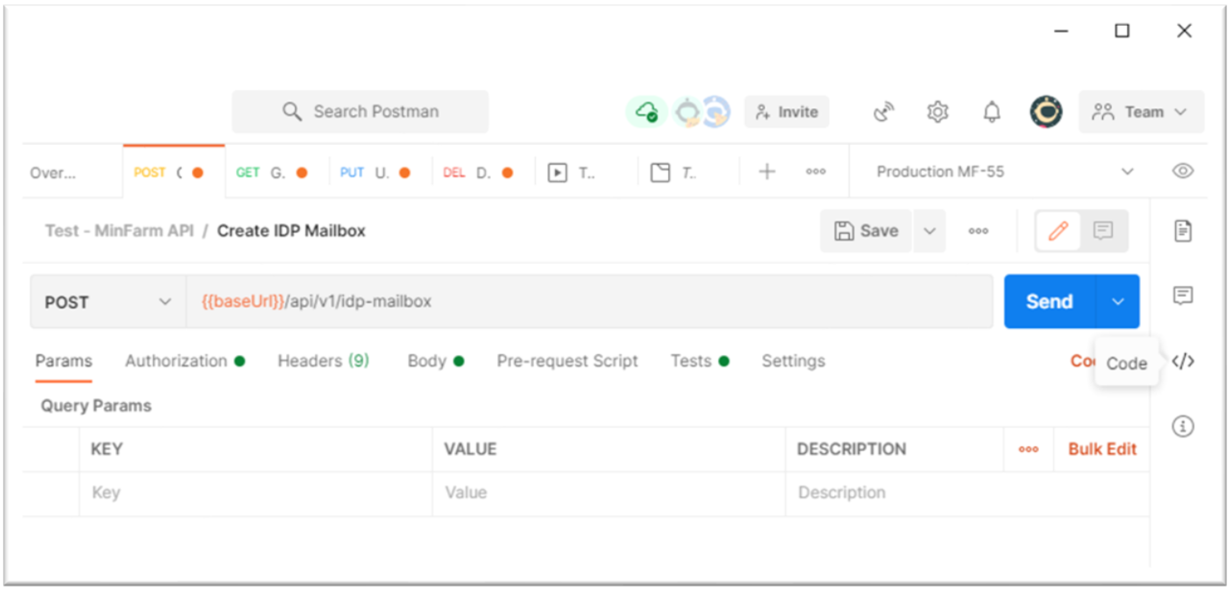
- To make a simple call to a remote URL - enter the request URL, select a method in the drop-down menu (*i.e.* GET, POST etc.), add a Personal Access Token in the Authorization tab of the Request Header, select a JSON file response, and select <Send>. If successful, a JSON file response is obtained with a successful status code.

- A Collection can be created. A Collection is a group of requests. These are listed to the left of the home screen. A Collection can be run automatically by selecting <Runner> on the home page. This is an easy way of automating a Collection.

- It is usually helpful to setup an Environment. An Environment contains variables that can be used in requests. Each variable is given a name and a value. An Environment allows variables to be stored and reused (*e.g.* Personal Access Tokens, URL *etc.*), so if a value needs to be updated, it only needs to be changed in one place, and not in every request. There are Environment variables and Collection variables.

- The Postman Console, to the bottom left of the screen, allows for deeper troubleshooting. All raw data can be viewed in the Postman Console.

- Postman has a great tool for advanced users. Select the Code symbol to the right of the home screen. Code snippets of the HTTP Request are shown for a number of coding languages.



Test - MinFarm API / Create IDP Mailbox

POST {{baseUrl}}/api/v1/idp-mailbox

Params | Authorization ● | Headers (9) | Body ● | Pre-request Script | Tests ● | Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description