FarmBot API Guide

Getting Started

Obtain FarmBot API Token

Use the following code to get your FarmBot API token. This will prompt for login details and download a JSON file with the token.

```
import json
from getpass import getpass
from farmbot import FarmbotToken

# inputs
SERVER = 'https://my.farm.bot'
EMAIL = input("Enter your email: ")
PASSWORD = getpass("Enter your password: ")

# get your FarmBot authorization token
token_string = FarmbotToken.download_token(EMAIL, PASSWORD, SERVER)
TOKEN = json.loads(token_string)
print(f'{TOKEN = }')

# save token to file
with open('farmbot_authorization_token.json', 'w') as f:
    f.write(json.dumps(TOKEN))
    print('token saved to file')
```

Using the Token

To use the token in your scripts, load it from the file:

```
# load token from file
with open('farmbot_authorization_token.json', 'r') as f:
    TOKEN = json.load(f)
```

Note: Some code from the Python library will only work if you still have the email and password defined. It is recommended to always have them defined.

Rest API

The FarmBot REST API allows for interaction with FarmBot systems via HTTP requests. Key functionalities include:

1. **Data Management:** Store, validate, and secure data, even offline.
2. **Email Notifications:** Manage password resets and critical errors.
3. **Image Handling:** Resize and store images from the onboard camera.

Common Endpoints:

- **/api/device:** Manage device settings.
- **/api/sequences:** Control sequence commands.
- **/api/sensor_readings:** Record sensor data.
- **/api/points:** Manage garden points (plants, weeds, tool slots).

This is an example request that will display the logs of the farmbot

```
import json

import requests



# load token from file
with open('farmbot_authorization_token.json', 'r') as f:
   TOKEN = json.load(f)

url = f'https:{TOKEN['token']['unencoded']['iss']}/api/logs'

headers = {'Authorization': 'Bearer ' + TOKEN['token']['encoded'],

          'content-type': 'application/json'}

response = requests.get(url, headers=headers)

print(json.dumps(response.json(), indent=2))
```

Farmbot library

The farmbot library gives a couple ways to interact with farmbot this are some key functions and their purposes:

1. **Connection Management:**
    - `connect()`, `disconnect()`
2. **Movement Commands:**
    - `move_absolute(x, y, z, speed)`
    - `move_relative(x, y, z, speed)`
3. **Device Control:**
    - `emergency_lock()`, `emergency_unlock()`
    - `power_off()`, `reboot()`
4. **Sensor Interaction:**
    - `read_pin(pin_number, pin_mode)`
    - `write_pin(pin_number, pin_value, pin_mode)`
5. **Miscellaneous:**
    - `take_photo()`
    - `send_message(msg, type)`

Although the python library is not complete since its missing some functionalities you can execute celery script with the `send_rpc` See how the celery script works below

Celery script

It's a JSON-based language used by FarmBot for task automation. It allows users to send commands and sequences to FarmBot devices in a structured format.

Node Structure Each Celery Script node has:

- `kind`: Specifies the node's purpose (e.g., `move_relative`).
- `args`: Key/value pairs defining the node's parameters.
- `comment` (optional): Helps with debugging.
- `body` (optional): Contains child nodes for complex sequences.

Common Nodes

- `move_relative`, `move_absolute`: Movement commands.
- `execute`: Runs predefined sequences.
- `rpc_request`, `rpc_ok`, `rpc_error`: Manage real-time device communication.

here is an example of how to reset

```
{
 "kind": "reboot",
 "args": { "package": "farmbot_os" },
 "comment": "Optional. Useful when debugging, but ignored by the system.",
 "body": []
}
```

Further explanation is on the farmbot GUI app

Every function is documented and includes additional helper functions.