Mist Python Library

Thomas MUNZER

Senior SE – South Europe

Mist Python3 Library explanations

©2019 Mist Systems. Proprietary & Confidential

Mist Python3 Library

What is it?

- A collection of Python3 files made to simplify the creation of python scripts using Mist APIs

What does it do?

- Automate some processes (authentication, org/site selection, …)
- Organize the API calls so there is no need to read the whole API documentation every time (e.g. *mlib.requests.sites.devices.get()* to retrieve the devices from a specific site)
- Accelerate the development process of Python scripts by providing reusable code and examples

Who can use it?

- Anyone who wants
- If someone wants to create new scripts based on this library, Python knowledge is required

Where can I get it?

- The whole library is available on github, and "may" be updated in the future with bug fixes and new example scripts
  Github: https://github.com/tmunzer/Mist_library

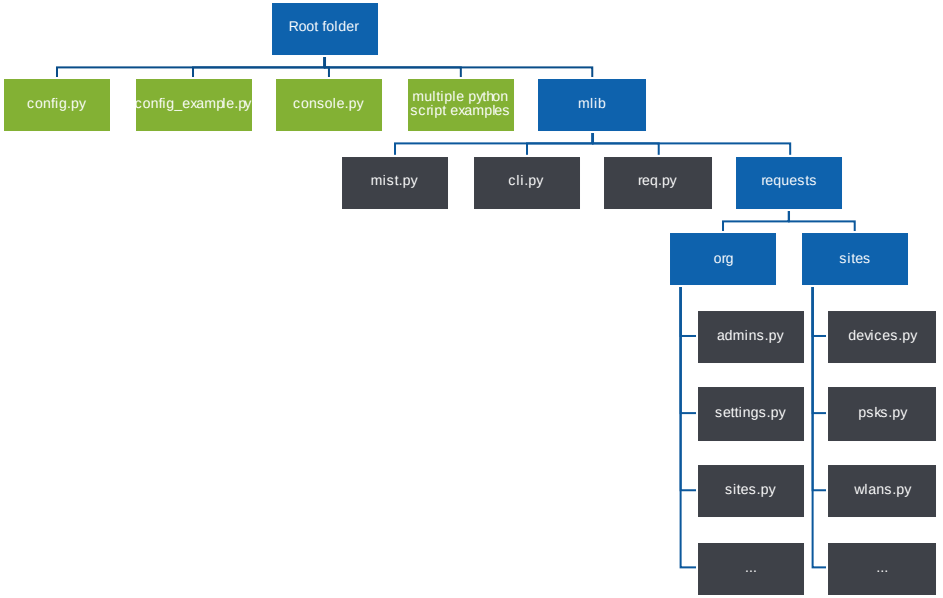
Limitations?

- All the API calls are not created yet, I'm creating them when needed. But new API calls can be easily added.
- 2FA is working fine with GET calls but seems to bring some issues with PUT and POST requests.

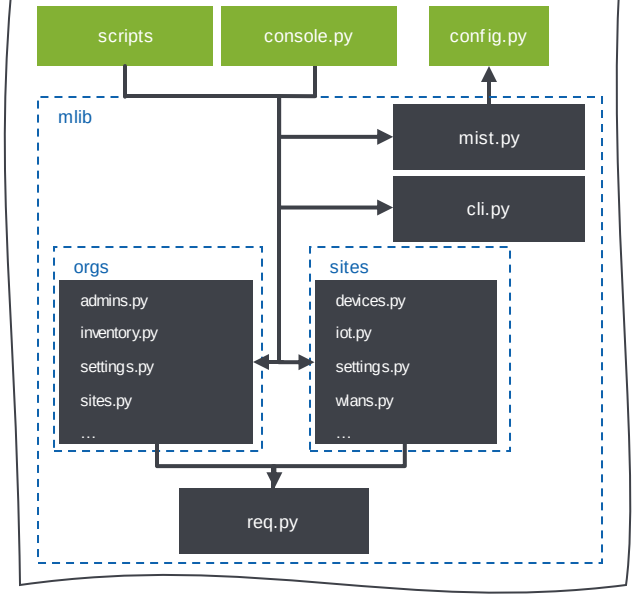
©2019 Mist Systems. Proprietary & Confidential

Script organization

Folders and files organization



Scripts dependencies



User files These files are the entry points for **end users**. They are configuration files or executable python scripts.

Library files These files are the library files, and will be used by **developer** to create new Python scripts.

©2019 Mist Systems. Proprietary & Confidential

Mist Python3 Library explanations

Configuration file

©2019 Mist Systems. Proprietary & Confidential

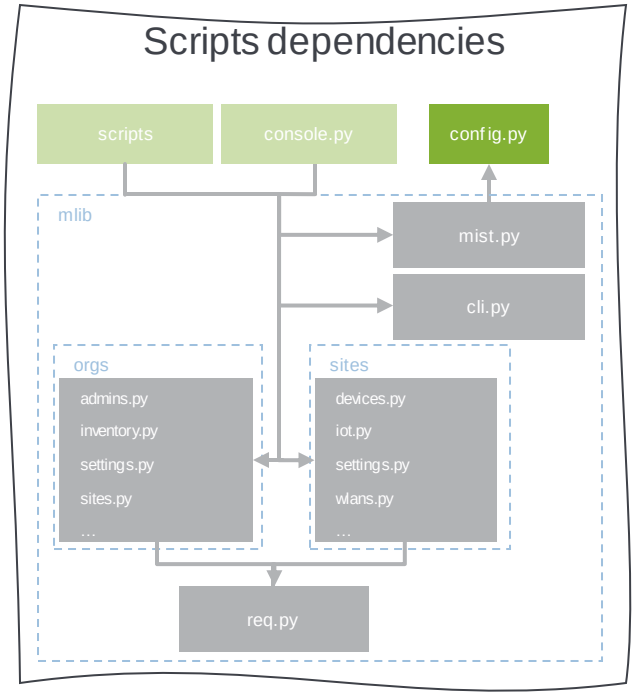
Config.py

Configuration file located inside the root folder.

Allows to store:

- the credentials
- the host to reach depending on the Cloud used (US, EU)
- the log level.

The format and explanation of the config.py file can be found in the config_example.py file.



©2019 Mist Systems. Proprietary & Confidential

Config.py

Note:

This is optional. If there is no “config.py” file, the script will contact the US cloud (api.mist.com) and ask for login/password every time.

The script configuration file allows you to store:

- Credentials:
  - Host: Optional. May be used to ask the script to use other Cloud than US Cloud. For example, use “api.eu.mist.com” for EU Cloud

AND

  - Login only: the password and OTP (if 2FA is enabled) will be asked by the script

OR

  - Login/Password: the OTP (if 2FA is enabled) will be asked by the script

OR

  - API Token: nothing will be asked by the script

- Log Level: this is optional. It allows you to chose the level of verbosity of the scripts

All the explanations are present in the “config_example.py” file.

Example:

```
credentials = {
"apitoken" : "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"host": "api.mist.com"
}
log_level = 6
```

©2019 Mist Systems. Proprietary & Confidential

Mist Python3 Library explanations

Python Console tool

©2019 Mist Systems. Proprietary & Confidential

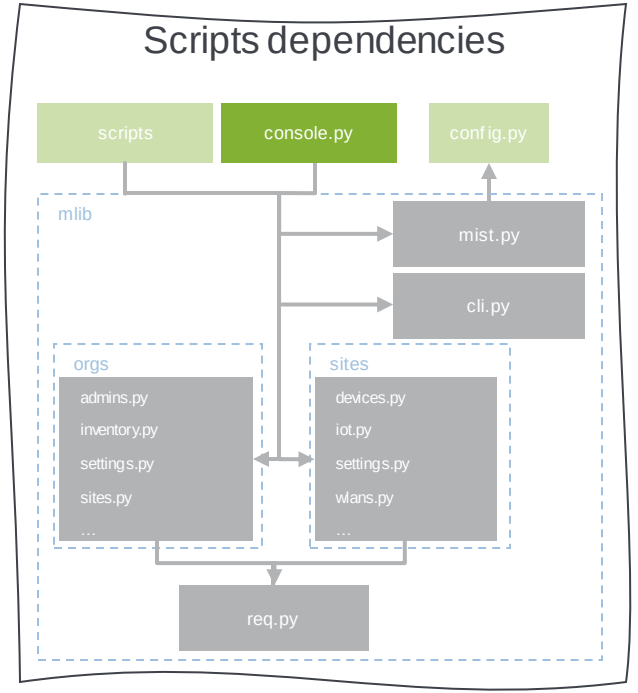
Console.py

Python3 module to easily use/test the Mist library from python CLI.

This module will:

- automatically start a session (i.e. authenticate the user based on the configuration stored in config.py or by asking credentials).
- Provide autocompletion to easily find the right syntax to use the Python3 Mist Library

(Work in Progress… there is plan to improve this part in the future)



©2019 Mist Systems. Proprietary & Confidential

Console.py

Start Python3 from the Mist Library folder	**tmunzer@tmunzer-mbp:~/data/4_dev/Mist/Mist_library$ python3** Python 3.7.4 (default, Sep 7 2019, 18:27:02) [Clang 10.0.1 (clang-1001.0.46.4)] on darwin Type "help", "copyright", "credits" or "license" for more information.
Load the module. When the module is loaded, it will automatically start a session.	**>>> import console** NOTICE: Login file found. NOTICE: API Token authentication used INFO: Request > GET https://api.mist.com/api/v1/self DEBUG: Response Status Code: 200
conole.get_org_id() and *console.get_site_id()* Available function to easily find an org_id or a site_id to be used in future requests	**>>> org_id = console.get_org_id()** Available organizations: 0) API test (id: ebad74f0-xxxx-xxxx-xxxx-850a0f98248a) 1) API test (id: ebad74f0-xxxx-xxxx-xxxx-850a0f98248a) Select an Org (0 to 1, or q to exit): 0 Selected org id: ebad74f0-xxxx-xxxx-xxxx-850a0f98248a
Autocompletion example. Use TAB to autocomplete and double TAB to see possible completions	**>>> console.mlib.org.** console.mlib.org.admins console.mlib.org.licenses console.mlib.org.sites console.mlib.org.channels console.mlib.org.settings console.mlib.org.stats console.mlib.org.inventory console.mlib.org.site_groups
See parameters to pass to the function Just send the function without parameters to see which parameters are required.	**>>> console.mlib.org.admins.get()** Traceback (most recent call last): File "<stdin>", line 1, in <module> TypeError: get() missing 2 required positional arguments: 'mist_session' and 'org_id'
Send a request and see the response Example of request. In this case, to get all the administrators from an org.	**>>> console.mlib.org.admins.get(console.session, org_id)** INFO: Request > GET https://api.mist.com/api/v1/orgs/ebad74f0-xxxx-xxxx-xxxx-850a0f98248a/admins DEBUG: Response Status Code: 200 {'result': [{'admin_id': '2c39a11b-xxxx-xxxx-xxxx-0b061c931809', 'privileges': [{'scope': 'org', 'role': 'admin', 'org_id': 'ebad74f0-xxxx-xxxx-xxxx-850a0f98248a'}], 'last_name': 'Skywalker', 'first_name': 'Anakin', 'email': 'askywalker@gmail.com'}, {'admin_id': '02a9d599-xxxx-xxxx-xxxx-59a111bf2333', 'privileges': [{'scope': 'org', 'role': 'admin', 'org_id': 'ebad74f0-xxxx-xxxx-xxxx-850a0f98248a'}], 'last_name': 'Skywalker', 'first_name': 'Luck', 'email': 'lskywalker@gmail.com'}'status_code': 200, 'error': '', 'uri': '/api/v1/orgs/ebad74f0-xxxx-xxxx-xxxx-850a0f98248a/admins'}

©2019 Mist Systems. Proprietary & Confidential

Mist Python3 Library explanations

Script examples

©2019 Mist Systems. Proprietary & Confidential

scripts

Set of Python3 scripts using the Mist Library.

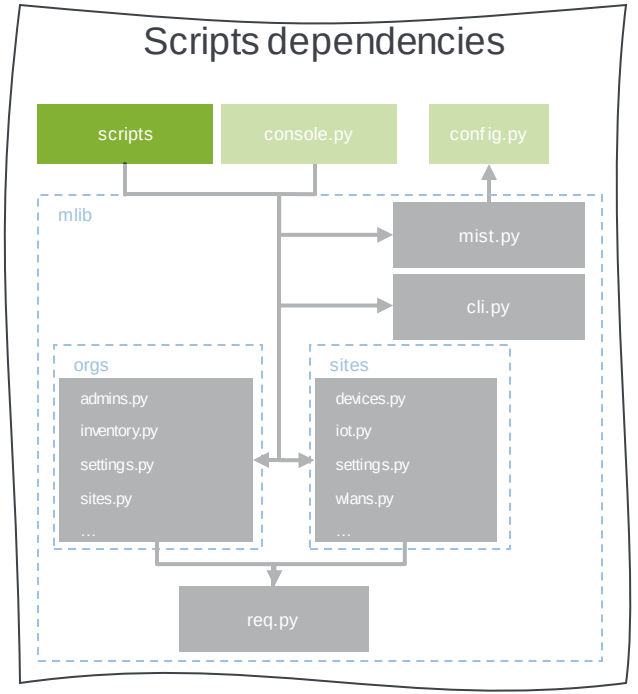
Example scripts are built to require a minimum input from users, but they can also be totally automated and/or run by an external software.

Each script has help and description at the beginning of the file.

If needed, new scripts can be added in the future to solve

Here is a lit of available scripts as of today (01/2020):

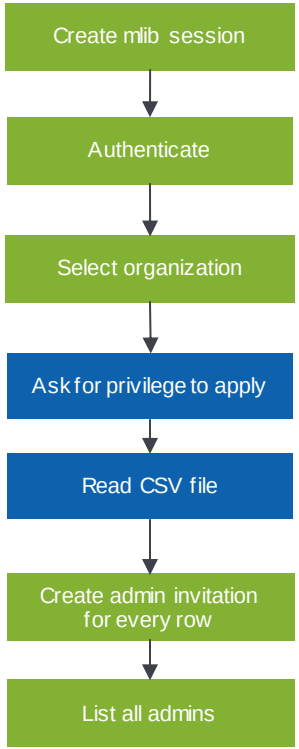
- org_admin_import.py
  read a CSV file to automatically generate admin invitations
- org_auto_site_assignment.py
  use APIs to enable autoprovisionning feature
- org_report_rogue.py
  generate a report (and save it as a CSV file) of all the rogue APs and clients from an organization or a site.
- site_conf_psk_import_csv.py
  read a CSV file to automatically create PSK
- site_conf_psk.py
  use APIs to create a new PSK
- site_conf_rogue.py
  use APIs to configure Rogue detection
- site_conf_webhook.py
  use APIs to configure webhooks (org or site level). Useful to create a webhook configuration for Splunk.
- site_conf_wlan.py
  use APIs to create/delete a WLAN



©2019 Mist Systems. Proprietary & Confidential

Example: org_admin_import.py

The entire org_admin_import.py script is less than 50 lines of code. Most of the code used is from the Mist Libraries.



Automated with the mist.py module (reusable library)
1 line of code:

```
mist = mist_lib.Mist_Session()
```

Automated with the cli.py module (reusable library)
1 line of code:

```
org_id = cli.select_org(mist)
```

Code specific to this script

Use pre-defined methods from the Mist Library (reusable library)
1 line of code to create an admin:

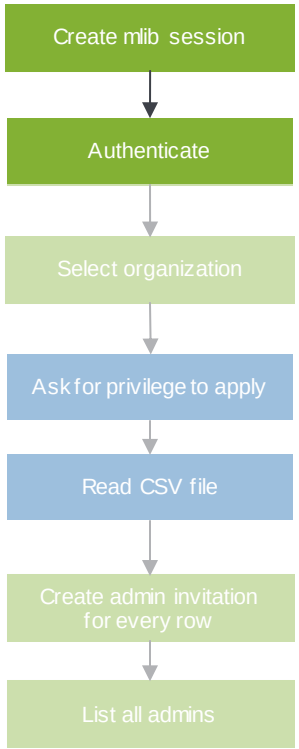
```
mist_lib.requests.org.admins.create_invite(mist, org_id, email, privileges, first_name, last_name)
```

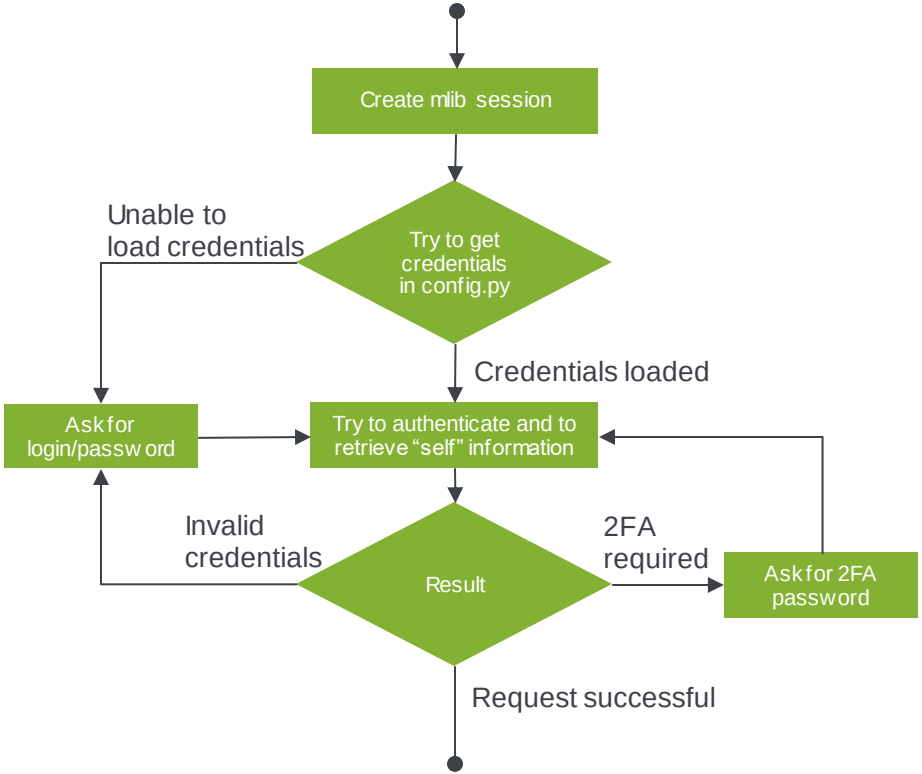
2 lines of code to get all the admins and display them nicely:

```
admins = mist_lib.requests.org.admins.get(mist, org_id)['result']
print(tabulate(admins))
```

©2019 Mist Systems. Proprietary & Confidential

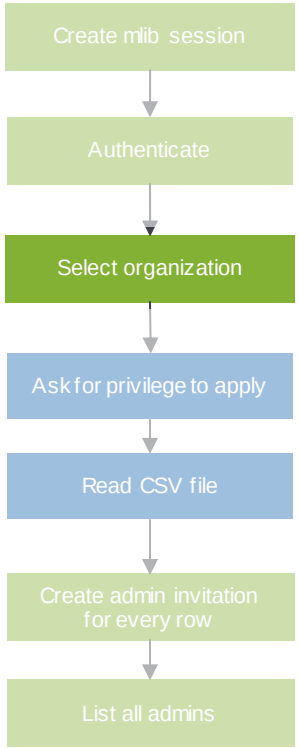
Example: org_admin_import.py

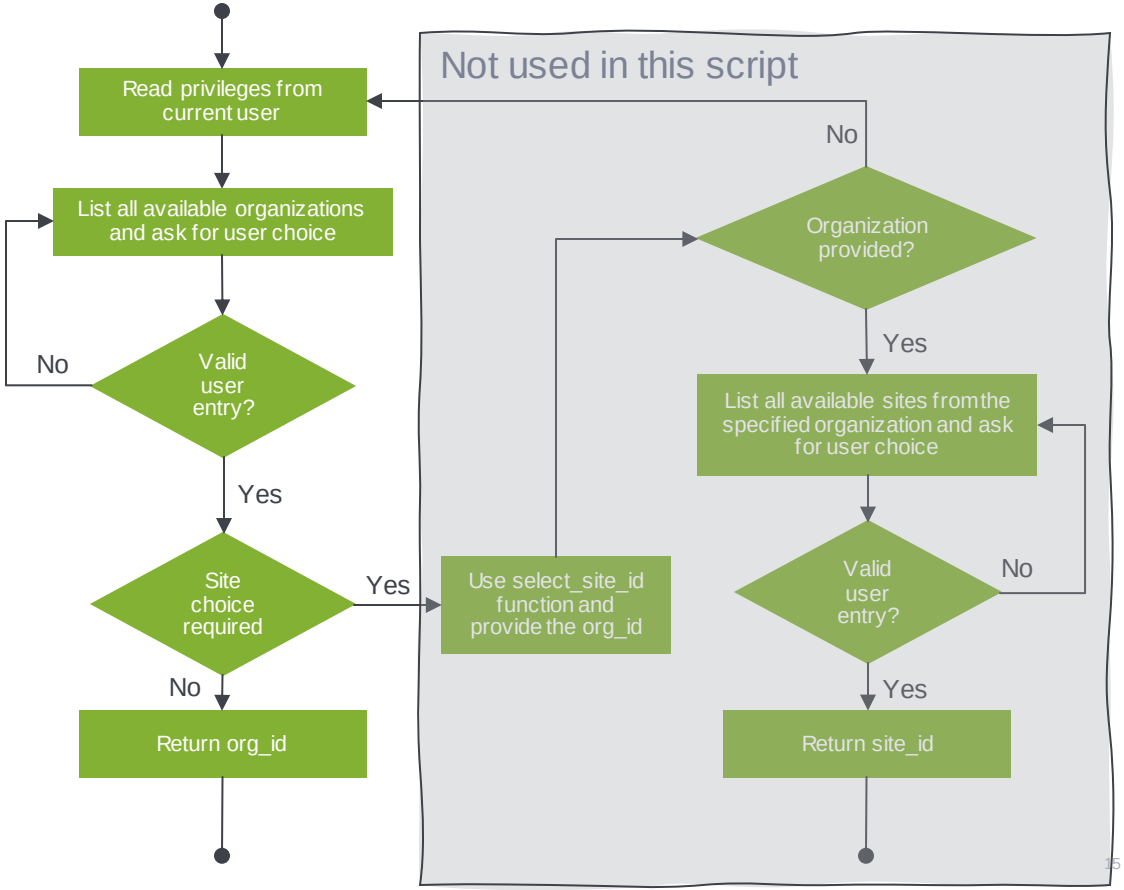




©2019 Mist Systems. Proprietary & Confidential

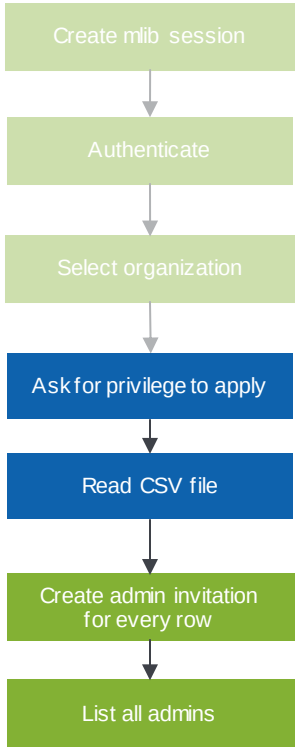
Example: org_admin_import.py

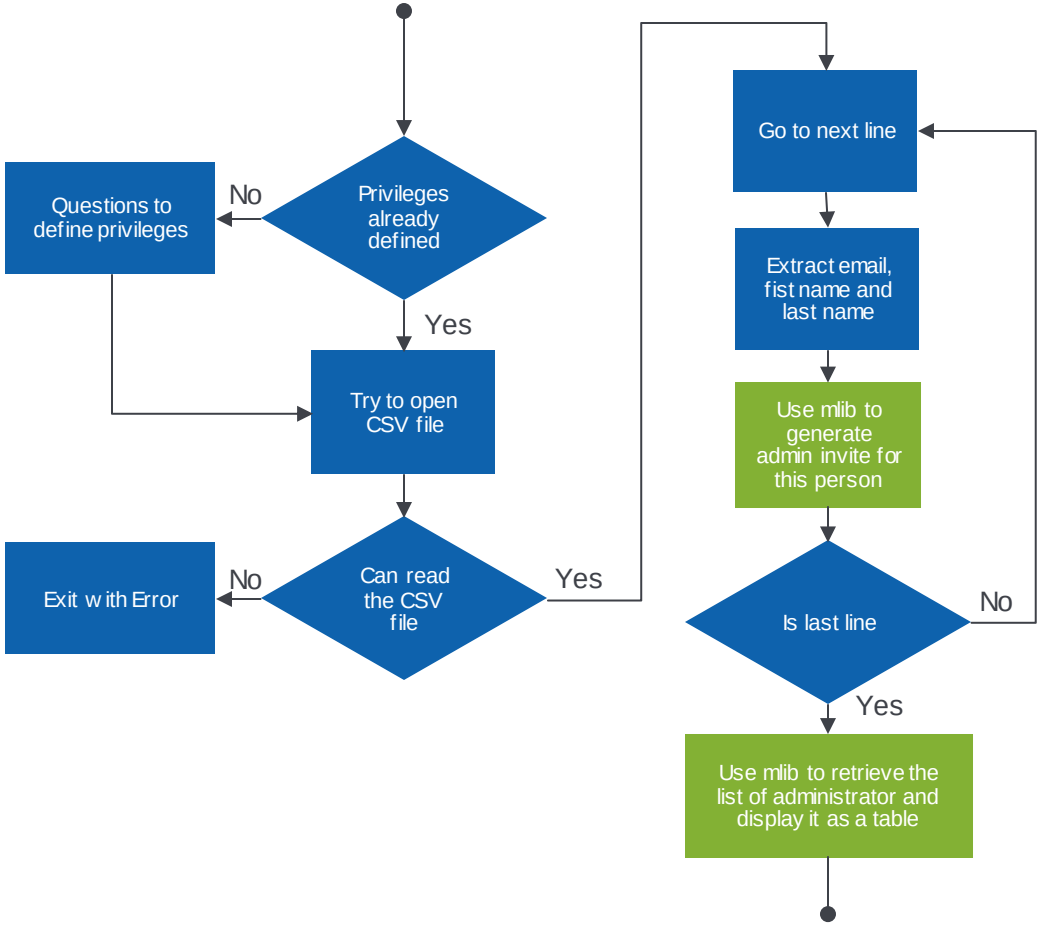




©2019 Mist Systems. Proprietary & Confidential

Example: org_admin_import.py





©2019 Mist Systems. Proprietary & Confidential

How to use it

©2019 Mist Systems. Proprietary & Confidential

Mist
A Juniper Company

Install Python | Download Mist Library | Create config.py | Test console.py | Test a script

Download and Install Python3 from https://www.python.org/downloads/

For Windows users, be sure to select **Customize Installation** and to enable **add Python to environment variables** during the installation wizard

Install library dependencies with pip:

```
pip install geopy tzwhere requests tabulate
OR (depending on your system)
pip3 install geopy tzwhere requests tabulate
```

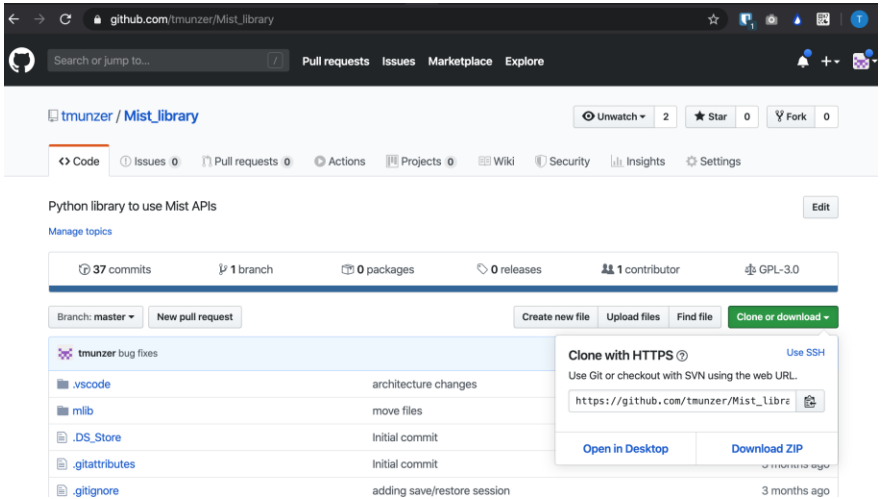
©2019 Mist Systems. Proprietary & Confidential

Mist
A Juniper Company

Install Python | Download Mist Library | Create config.py | Test console.py | Test a script

- Get the library from Github
- Go to https://github.com/tmunzer/Mist_library
- Click on **Clone or Download**
- then click on **Download ZIP**

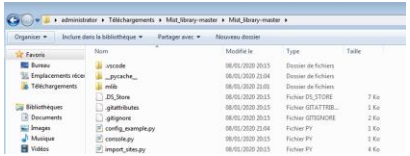
- Unzip the downloaded archive.



©2019 Mist Systems. Proprietary & Confidential

Install Python → Download Mist Library → Create config.py → Test console.py → Test a script

Go to the library folder. You should see files like **config_example.py.**



Create a file **config.py** and edit it.

I recommand to use text editors like atom (available on https://atom.io) or Visual Studio Code (available at https://code.visualstudio.com/)

Add the configuration to your **config.py** file.

Be sure to replace the login and password or the Access Token with your own credentials.

```
credentials = {
    "email": "user@domain",
    "password": "xxxxx",
    "host": 'api.mist.com'
}
```

OR

```
credentials = {
    "apitoken" : "xxxxxxxxxxxx",
    "host": "api.mist.com"
}
```

©2019 Mist Systems. Proprietary & Confidential

Mist
A Juniper Company

Install Python | Download Mist Library | Create config.py | Test console.py | Test a script

console.py is not build like as python script, but more like a python module. This means you must **start python**, and then **import the console module**:

Tips: You must run Python from the Mist Library folder

Tips: You don't have to add the ".py" extension w hen you want to load it.

```
tmunzer@tmunzer-mbp:~/4_dev/Mist/Mist_library$ python3
Python 3.7.4 (default, Sep  7 2019, 18:27:02)
[Clang 10.0.1 (clang-1001.0.46.4)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import console
```

Once the **console** module is loaded, it will automatically

- Display the help information
- Start a Mist session and store it in the **console.session** variable

```
>>> import console

    Info:
    This Python3 module is built to help you to use and test Mist Library

    Help:
    When this module is imported, it will automatically authenticate you, create a Mist session, and
    store is into the variable
            console.session

    You can use it afterward with Mist Library request. For example, you can do:
    The you will be able to use the script to generate and send request to the Mist Cloud. For example,
    you can do:
            console.mlib.org.channels.country_codes_get(console.session)

    Some API request will ask for an "org_id" or a "site_id". The following commands are here to help
    you to retrieve (and store) this information:
            console.get_org_id()
            console.get_site_id()

    Udage:
    console.help()            Show this message
    console.get_org_id()      Help you to retrieve an Org ID
    console.get_site_id()     Help you to retrieve a Site ID
    console.show(data)        Nicely display API call response. Where "data" is is the API response

NOTICE: Login file found.
NOTICE: API Token authentication used
INFO: Request > GET https://api.mist.com/api/v1/self
DEBUG: Response Status Code: 200
```

Warning

The current version is not totally functional when 2FA is enabled. Only GET requests may work.

©2019 Mist Systems. Proprietary & Confidential

Mist
A Juniper Company

Install Python | Download Mist Library | Create config.py | Test console.py | Test a script

- You can use **console.get_org_id()** and **console.get_site_id()** to easily retrieve and store your org and site IDs:

```
>>> mysite = console.get_site_id()

Available organizations:
0) API test (id: ebad74f0-[redacted]-850a0f98248a)
1) API test (id: ebad74f0-[redacted]-850a0f98248a)

Select an Org (0 to 1, or q to exit): 0

Available sites:
0) Secondary Site (id: 91e219cb-[redacted]-b-a7dcfdfebbd5)
1) Primary Site (id: 89d1af18-[redacted]-40e13bd6d3c0)

Select a Site (0 to 1, or q to exit): 1

Selected site id: 89d1af18-[redacted]-40e13bd6d3c0
>>>
```

- You can use the autocompletion feature to know which command is available.

```
>>> console.mlib.sites.devices.
console.mlib.sites.devices.create(      console.mlib.sites.devices.delete(      console.mlib.sites.devices.get(         console.mlib.sites
```

- You can send a command without parameters to know which parameters are required

```
>>> console.mlib.sites.devices.get()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: get() missing 2 required positional arguments: 'mist_session' and 'site_id'
```

- You can display the raw data received from the server by just typing the command

```
>>> console.mlib.sites.devices.get(console.session, mysite)
INFO: Request > GET https://api.mist.com/api/v1/sites/fa018c13-[redacted]-1eeb894a96c4/devices
DEBUG: Response Status Code: 200
{'result': [{'switch_config': {'wds': {}, 'eth1': {}, 'module': {}, 'eth0': {}}, 'orientation': 265, 'x_m': 7.380324693115277, 'site_id': 'fa018c13-[redacted]
ed_time': 157589123O, 'serial': [redacted], 'id': '00000000-0000-0000-10[redacted]', 'disable_module': False, 'ble_config': {'eddystone_uid_enabled':
'default', 'eddystone_url_enabled': False, 'beam_disabled': []}, 'hw_rev': '800', 'y_m': 9.637619688584998, 'disable_eth1': False, 'type': 'ap', 'map_id': '2b79
8, 'radio_config': {'ant_gain_5': None, 'ant_gain_24': None}, 'orientation_overwrite': True, 'name': 'lab-ms-ap-02', 'org_id': '203d3d02-[redacted]76896a33
: {'A1': {'enabled': True, 'name': 'my input 1'}, 'D0': {'enabled': True, 'value': 0}}, 'y': 675.11328125, 'x': 516.990234375, 'model': 'AP43'}, {'map_id': None
eight': 0, 'mac': [redacted], 'tag_id': 228170, 'tag_uuid': '203d3d02-[redacted]76896a3330f4', 'deviceprofile_id': None, 'created_time': 1578420314, 's
v': '800', 'org_id': '203d3d02-[redacted]76896a3330f4', 'y': 0, 'x': 0, 'model': 'AP21', 'type': 'ap', 'locating': False}], 'status_code': 200, 'error': ''
```

- Or you can use the command **console.show(data)** to display it nicely

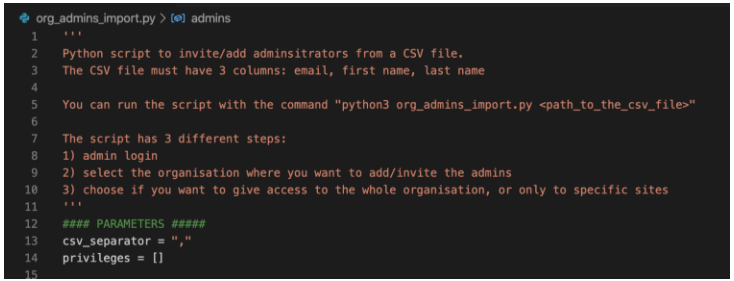
```
>>> console.show(console.mlib.sites.devices.get(console.session, mysite))
INFO: Request > GET https://api.mist.com/api/v1/sites/fa018c13-[redacted]-1eeb894a96c4/devices
DEBUG: Response Status Code: 200

switch_config                                orientation     x_m  site_id                          tag_uuid
config
        modified_time  mac            tag_id  radio_config                      orientation_overwrite   name          org_id
                                                                  y      x  model      height  notes    locating
------  -------------  -------------  ------  ------------------------------------  ------  ---------  ------  ---------  ------------------------------------
------  -------------  -------------  ------  ------------------------------------  ------  ---------  ------  ---------  ------------------------------------

{'wds': {}, 'eth1': {}, 'module': {}, 'eth0': {}}            265  7.38032  fa018c13-[redacted]-1eeb894a96c4  203d3d02-[redacted]-76896a3330f4
dystone_uid_enabled': False, 'ibeacon_enabled': False, 'beacon_enabled': True, 'power_mode': 'default', 'beacon_rate_mode': 'default', 'eddystone_url
d9275f          1577789874  [redacted]  240278  {'ant_gain_5': None, 'ant_gain_24': None}  True                    lab-ms-ap-02  203d3d02-[redacted]
': 'my input 1'}, 'D0': {'enabled': True, 'value': 0}}  675.113  516.99  AP43
                                                                  0      fa018c13-[redacted]-1eeb894a96c4  203d3d02-[redacted]-76896a3330f4
```

©2019 Mist Systems. Proprietary & Confidential

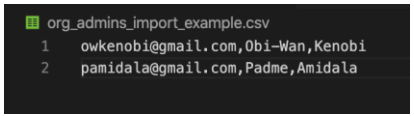
Mist
A Juniper Company

Install Python | Download Mist Library | Create config.py | Test console.py | Test a script

- Open the script file to see how to use it, and, in some case, to configure the script with your specifications



- If the script is used with an external file (e.g. a CSV file), prepare it.



- Open your terminal, go to the Mist Library folder, and start the script as explained in the help.

©2019 Mist Systems. Proprietary & Confidential

Mist
A Juniper Company

Install Python → Download Mist Library → Create config.py → Test console.py → Test a script

```
tmunzer@tmunzer-mbp:~/4_dev/Mist/Mist_library$ python3 ./org_admins_import.py ./org_admins_import_example.csv
INFO: Restoring session...
ERROR: Unable to load session...
NOTICE: Login file found.
NOTICE: API Token authentication used
INFO: Request > GET https://api.mist.com/api/v1/self
DEBUG: Response Status Code: 200

Available organizations:
0) API test (id: ebad74f0-xxxx-xxxx-xxxx-850a0f98248a)
1) API test (id: ebad74f0-xxxx-xxxx-xxxx-850a0f98248a)

Select an Org (0 to 1, or q to exit): 0
Which level of privilege at the org level ("s" for Super Admin, "n" for Network Admin,"o" for observer,"h" for helpdesk)? o
Do you want to select specific sites (y/N)? n
INFO: Request > GET https://api.mist.com/api/v1/orgs/ebad74f0-xxxx-xxxx-xxxx-850a0f98248a/sites
DEBUG: Response Status Code: 200
Opening CSV file ./org_admins_import_example.csv
owkenobi@gmail.com, Obi-Wan, Kenobi
INFO: Request > POST https://api.mist.com/api/v1/orgs/ebad74f0-xxxx-xxxx-xxxx-850a0f98248a/invites
pamidala@gmail.com, Padme, Amidala
INFO: Request > POST https://api.mist.com/api/v1/orgs/ebad74f0-xxxx-xxxx-xxxx-850a0f98248a/invites
DEBUG: Response Status Code: 200
INFO: Request > GET https://api.mist.com/api/v1/orgs/ebad74f0-xxxx-xxxx-xxxx-850a0f98248a/admins
DEBUG: Response Status Code: 200

admin_id                             privileges                                                
```

Questions?

Thank you!

Mist
A Juniper Company