pyaoscx

Release 2.4.0

Apr 18, 2023

Contents:

These modules are written for AOS-CX API v10.04 and later. These scripts are written for devices running AOS-CX firmware version 10.04 or greater.

See the Release Notes for more information.

Please note that pyaoscx v2 is **not** backwards compatible for pyaoscx v1 and earlier, so please specify the correct version when using pyaoscx in requirements.txt files

It is also important to note that the latest code commits on the Master branch in Git are usually ahead of the official releases and tags, so please be aware of this when cloning the repo versus doing a `pip install pyaoscx`

Structure

Detailed information about the structure and design can be found in the Design document.

- REST API call functions are found in the modules in /pyaoscx.

- REST API call functions are combined into other functions that emulate low-level processes. These low-level process functions are also placed in files in /pyaoscx.

- Functions from the /pyaoscx files (API functions and low-level functions) are combined to emulate larger network configuration processes (workflows). These workflow scripts stored in the /workflows folder.

How to contribute

Please see the accompanying CONTRIBUTING.md file for guidelines on how to contribute to this repository.

Git Workflow

This repo adheres to the 'shared repo' git workflow: 1. Clone the repo to a local machine:

```
```git clone <repo_URL>```
```

2. Checkout a local working branch:

   ```
   git checkout -b <local_working_branch_name>
   ```

3. Add and amend files in the local working branch:

   ```
   git add <file_name>
   ```

4. Commit regularly. Each commit should encompass a single logical change to the repo (e.g. adding a new function in /pyaoscx is one commit; writing docstrings for all functions in a module is another commit). Include an explanatory message with each commit:

   ```
   git commit -m "<Clear_explanation_of_commit_here>"
   ```

5. Push commits to github.hpe.com:

   ```
   git push origin <local_working_branch_name>
   ```

6. Merge changes using a Pull Request on github.hpe.com. Ensure the PR has a relevant title and additional comments if necessary. PRs should be raised regularly once code is tested and the user satisfied that it is ready for submission. Do not put off creaing a PR until a whole project is complete. The larger the PR, the difficult it is to successfully merge.

Setup

Before starting ensure the switch REST API is enabled. Instructions for checking and changing whether or not the REST API is enabled status are available in the *ArubaOS-CX Rest API Guide*. This includes making sure each device has an administrator account with a password, and each device has https-server rest access-mode read-write and enabled on the reachable vrf.

4.1 How to run this code

In order to run the workflow scripts, please complete the steps below: 1. install virtual env (refer https://docs.python.org/3/library/venv.html). Make sure python version 3 is installed in system.

```
```
$ python3 -m venv switchenv
```
```

2. Activate the virtual env

```
$ source switchenv/bin/activate
in Windows:
$ venv/Scripts/activate.bat
```

3. Install the pyaoscx package

```
(switchenv)$ pip install pyaoscx
```

4. Now you can run different workflows from pyaoscx/workflows (e.g. `print_system_info.py`)

5. Keep in mind that the workflows perform high-level configuration processes; they are highly dependent on the configuration already on the switch prior to running the workflows. For this reason, the comment at the top of each workflow script describes any necessary preconditions.

CHAPTER 5

Troubleshooting Issues

1. If you encounter module import errors, make sure that the package has been installed correctly.

Additionally, please read the RELEASE-NOTES.md file for the current release information and known issues.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search