
MatrixCtl

Release 0.12.0-beta.2

Michael Sasser

Mar 23, 2023

CONTENTS:

1	Installation	3
1.1	Prerequisites	3
1.2	Installation with pip	3
2	Getting Started	5
2.1	Config File	5
2.2	Synapse Playbook	8
2.3	Access Token	8
2.4	SSH Public Key	11
3	Contributor Documentation	17
3.1	Contributor Covenant Code of Conduct	17
4	Changelog	77
4.1	0.12.0-beta.2 (2023-03-24)	77
4.2	0.12.0-beta.1 (2021-12-02)	77
4.3	0.11.5 (2021-12-01)	78
4.4	0.11.4 (2021-12-01)	78
4.5	0.11.3 (2021-11-16)	78
4.6	0.11.2 (2021-09-26)	79
4.7	0.11.1 (2021-09-25)	79
4.8	0.11.0 (2021-09-21)	79
4.9	0.10.3 (2021-06-26)	80
4.10	0.10.2 (2021-06-24)	81
4.11	0.10.1 (2021-06-17)	81
4.12	0.10.0 (2021-06-17)	81
4.13	0.9.0 (2021-04-23)	82
4.14	0.8.6 (2021-04-17)	82
4.15	0.8.5 (2021-02-24)	83
4.16	0.8.4 (2021-02-24)	83
4.17	0.8.3 (2021-02-24)	83
4.18	0.8.2 (2021-02-24)	83
4.19	0.8.1 (2020-12-02)	83
4.20	0.8.0 (2020-12-02)	84
4.21	0.7.0 (2020-09-25)	84
4.22	0.6.3 (2020-09-17)	84
4.23	0.6.2 (2020-09-16)	84
4.24	0.6.1 (2020-06-02)	85
4.25	0.6.0 (2020-05-12)	85
4.26	0.5.0 (2020-04-30)	85

4.27	0.4.0 (2020-04-22)	86
4.28	0.3.2 (2020-04-21)	86
4.29	0.3.1 (2020-04-21)	86
4.30	0.3.0 (2020-04-20)	87
4.31	0.2.2 (2020-04-13)	87
4.32	0.2.1 (2020-04-13)	87
4.33	0.2.0 (2020-04-12)	87
4.34	0.1.4 (2020-04-10)	88
4.35	0.1.3 (2020-04-10)	88
4.36	0.1.2 (2020-04-07)	88
4.37	0.1.1 (2020-04-07)	89
4.38	0.1.0 (2020-04-07)	89
5	Branching Model	91
6	Command line tool	93
7	Configuration File	95
8	Semantic Versioning	97
9	Indices and tables	99
10	License	101
	Python Module Index	103
	Index	105

MatrixCtl is a python program to control, manage, provision and deploy our matrix homeserver. I had a bunch of shell scripts doing that. Two weeks after using them I couldn't remember the order in which I have to use the arguments or which arguments where needed. It was a pain. So I decided I hack something together fast.

It's not the most elegant piece of software I wrote, but it should do the trick. I will continue to port the rest of the scripts and add new features.

INSTALLATION

1.1 Prerequisites

To be able to use **all** features of MatrixCtl you need to have:

- Python 3.8 or higher on your machine.
- deployed the instance of `synapse` with the `spantaleev/matrix-docker-ansible-deploy` ansible playbook.
- the access token of your administrator account.
- SSH access to the matrix server from your machine with a public key.

Note: If you don't need all features, you are good to start with python 3.8. This is the only mandatory prerequisite of this list.

See also:

We have a guide, how you accomplish the rest of the list in the *Getting Started* guide.

1.2 Installation with pip

To install MatrixCtl run `pip install matrixctl` with a `Python>=3.8`. If you already have a version of MatrixCtl installed, you can upgrade it with `pip install --upgrade matrixctl`.

GETTING STARTED

To use all features of MatrixCtl you need can do a few steps to make them compatible.

2.1 Config File

To use this program you need to have this config file in `/etc/matrixctl/config.yaml` or in `~/.config/matrixctl/config.yaml`.

This config file contains four sections:

- `ansible`
- `synapse`
- `api`
- `ssh`
- `database`

In `ansible` fill in the absolute path to your fully configured Playbook. Make sure `ansible` is configured correctly on your system. To get started, follow the *Synapse Playbook* guide. You need this section, if you want to use one of the following commands:

- `matrixctl adduser --ansible`
- `matrixctl deploy`
- `matrixctl start`
- `matrixctl restart`
- `matrixctl maintenance`
- `matrixctl check`

Note: If you want to run more than one playbook you can create a file which contains `import_playbook` lines like:
- `import_playbook: /PathTo/matrix-docker-ansible-deploy/setup.yml` and configure it as `playbook` in the `matrixctl` config file.

`synapse` is used to update (`git pull`) the `synapse` playbook You need this section, if you want to:

- `matrixctl update`

`api` is used to communicate with the `synapse` API directly. This is faster and has more additional functionality then the *Synapse* playbook. To get started, follow the *Access Token* guide. It is used for:

- `matrixctl adduser`
- `matrixctl deluser`
- `matrixctl users`
- `matrixctl user`
- `matrixctl users`
- `matrixctl upload`
- `matrixctl rooms`
- `matrixctl delroom`
- `matrixctl server-notice`
- `matrixctl purge-history`
- `matrixctl version`
- `matrixctl delete-local-media`
- `matrixctl get-event-context`
- `matrixctl is-admin`
- `matrixctl joinroom`
- `matrixctl make-room-admin`
- `matrixctl purge-remote-media`
- `matrixctl report`
- `matrixctl reports`
- `matrixctl set-admin`

ssh you can use additional functionality. It is used for:

- `matrixctl adduser-jisi`
- `matrixctl deluser-jisi`

Note: If you are not sure, what to fill in that config file, read the rest of the “Getting Started” section of this documentation.

Warning: Make sure, that other accounts of your local machine are not able to read or edit your config file. It contains sensitive data.

```
1 # Define your homeservers in "servers" here.
2 servers:
3     # Your default server. You can specify multiple servers here with arbitrary
4     # Names
5     default:
6
7     ansible:
8         # The absolute path to your playbook
9         playbook: /path/to/ansible/playbook
```

(continues on next page)

(continued from previous page)

```
10
11 synapse:
12   # The absolute path to the synapse playbook.
13   # This is only used for updating the playbook.
14   playbook: /path/to/synapse/playbook
15
16   # If your matrix server is deployed, you may want to fill out the API section.
17   # It enables matrixctl to run more and faster commands. You can deploy and
18   # provision your Server without this section. You also can create a user with
19   # "matrixctl adduser --ansible YourUsername" and add your privileges after
20   # that.
21   api:
22     # Your domain should be something like "michaelsasser.org" without the
23     # "matrix." in the front. MatrixCtl will add that, if needed. An IP-Address
24     # is not enough.
25     domain: example.com
26
27     # The username your admin user
28     username: johndoe
29
30     # To use the API you need to have an administrator account. Enter your Token
31     # here. If you use the element client you will find it your user settings
32     # (click on your username on the upper left corner on your browser) in the
33     # "Help & About" tab. If you scroll down click next to "Access-Token:" on
34     # "<click to reveal>". It will be marked for you. Copy it in here.
35     token: "MyMatrixToken"
36
37     # In some cases, MatrixCtl does need to make many requests. To speed those
38     # requests a notch, you can set a concurrent_limit which is greater than
39     # one. This sets a limit to how many asynchronous workers can be spawned
40     # by MatrixCtl. If you set the number to high, MatrixCtl needs more time
41     # to spawn the workers, then a synchronous request would take.
42     concurrent_limit: 10
43
44     # Here you can add your SSH configuration.
45     ssh:
46       address: matrix.example.com
47
48       # The default port is 22
49       port: 22
50
51       # The default username is your current login name.
52       user: john
53
54     # Define your maintenance tasks
55     maintenance:
56       tasks:
57         - compress-state # Compress synapses state table
58         - vacuum         # VACUUM the synapse database (garbage-collection)
59
60     # Add connection parameters to the Database
61     # Synapse does only read (SELECT) information from the database.
```

(continues on next page)

(continued from previous page)

```

62 # The user needs to be able to login to the synapse database
63 # and SELECT from the events and event_json tables.
64 database:
65     synapse_database: synapse # this is the playbooks default table name
66     synapse_user: matrixctl # the username (role) for the database
67     synapse_password: "RolePassword"
68     tunnel: true # true if an ssh tunnel should be used to connect
69
70     # The port that was used in the playbook (e.g.
71     # matrix_postgres_container_postgres_bind_port: 5432)
72     # or for your external database. For security reasons the port
73     # should be blocked by your firewall. If you enable the tunnel
74     # by setting tunnel: true, MatrixCtl activates a SSH tunnel.
75     port: 5432 # the remote port
76
77 # Another server.
78 foo:
79     # ...

```

If you configure database, you can use the following commands:

- `matrixctl get-event`
- `matrixctl get-events`

Note: You need to create a new PostgreSQL role. The must have the permission to login and SELECT permissions for the `json_events` and `events` table.

2.2 Synapse Playbook

If you want to update your Synapse instance with MatrixCtl you need to deploy them with a specific Ansible playbook. The `spantaleev/matrix-docker-ansible-deploy` project made it quite simple. To start, follow the [link here](#). This ansible playbook is well documented and helps beginners, to get a fast and powerful synapse instance with a bunch of optional “plugins” running in no time.

2.3 Access Token

To use the API of Synapse you need a access token of an administrator. To get an access token, you need an user account. MatrixCtl helps you to register one. Just run:

```

$ matrixctl adduser --ansible --admin myusername
Password:
Password (again):
Username: myusername
Password: **HIDDEN**
Admin:    yes
Is everything ok? [y/n]y

```

(continues on next page)

(continued from previous page)

```

PLAY [Set up a Matrix server]
↳*****

[...]

PLAY RECAP
↳*****
matrix.michaelsasser.org : ok=24   changed=0   unreachable=0   failed=0
↳skipped=34   rescued=0   ignored=0

```

Note: If you don't enter a password and press [ENTER] twice, a password will be generated for you. If you are satisfied with it, enter [y].

Now you have created your user with the user name "myusername". The argument `--admin` makes sure, that you create an administrator account instead of an user account. The `--ansible` argument is needed, because you currently have no access to the admin API. After all that steps you don't need the `--ansible` anymore.

Note: You can use this user account as your personal main user account.

Note: If you have already created an admin user account ignore this step and continue below.

Now Open `https://element.yourdomain.tld` to login.

Click on "Sign In" and enter your credentials. In this example, we used "myusername" for the user and the entered password to login.

After you are logged in, click on your user name in the top right corner. A small window will pop up. Click on Settings.

A bigger window with your user settings will pop up. Click on Help & About on the left side of that window. If you scroll down on the right hand side of this window, you will find the Advanced section. In the Advanced section you find Access Token: `<click to reveal>`.

Now click on `<click to reveal>`.

This is your access token. It is already highlighted for you. Just copy it into the config file into the `api` section.

2.3.1 Copy The Token

Now you can copy the token into the `api` section of your config file. If you don't have a config file head over to the *Config File* chapter.

```

...
api.
  domain: yourdomain.tld
  token:
↳MDAxZmxvY2F0aW9uIG1pY2hhZWxzYXNzZXIub3JnCjAwMTNpZGVudGhmaWVyIGtleQowMDEwY2lkIGdlbiA9IDEKMDAzMGNpZCB1c
...

```

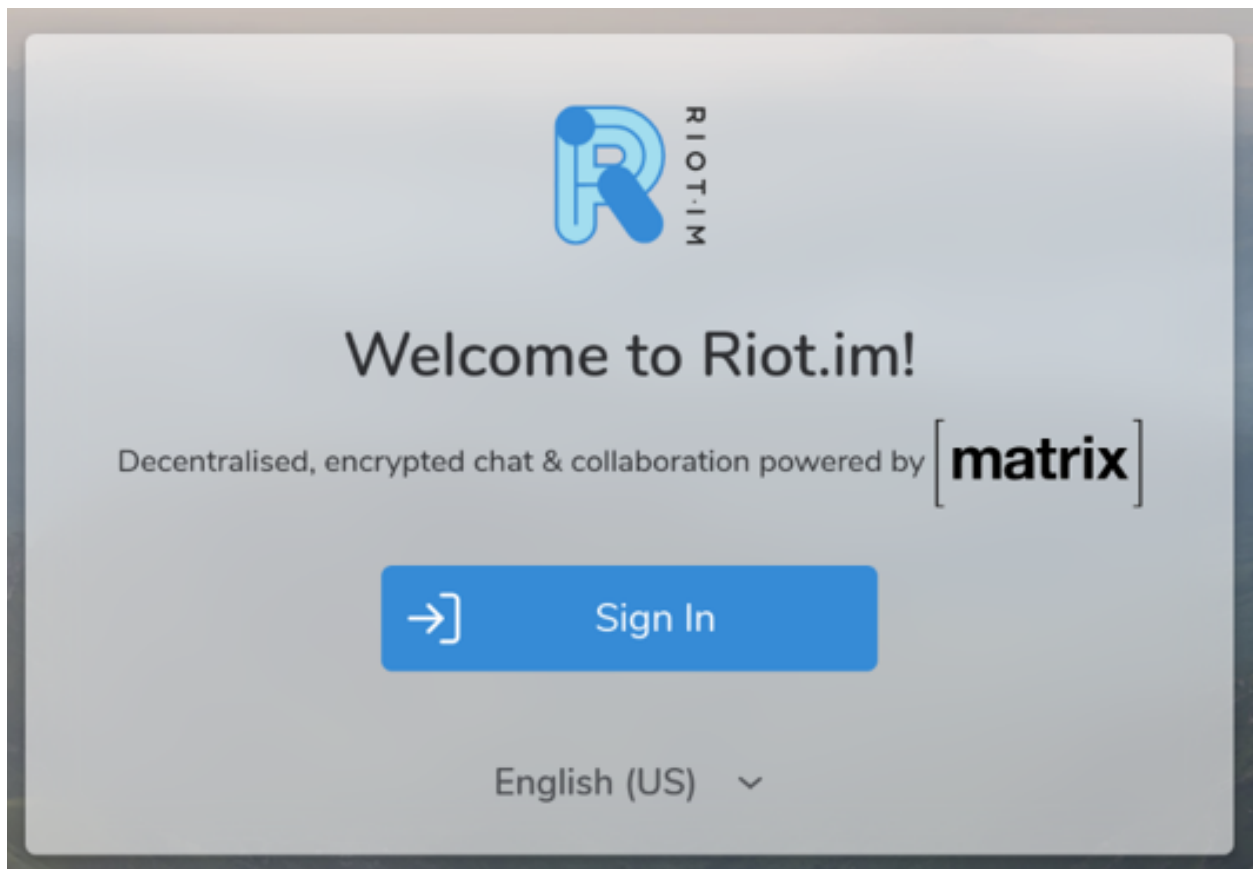


Fig. 2.1: The Element loginscreen

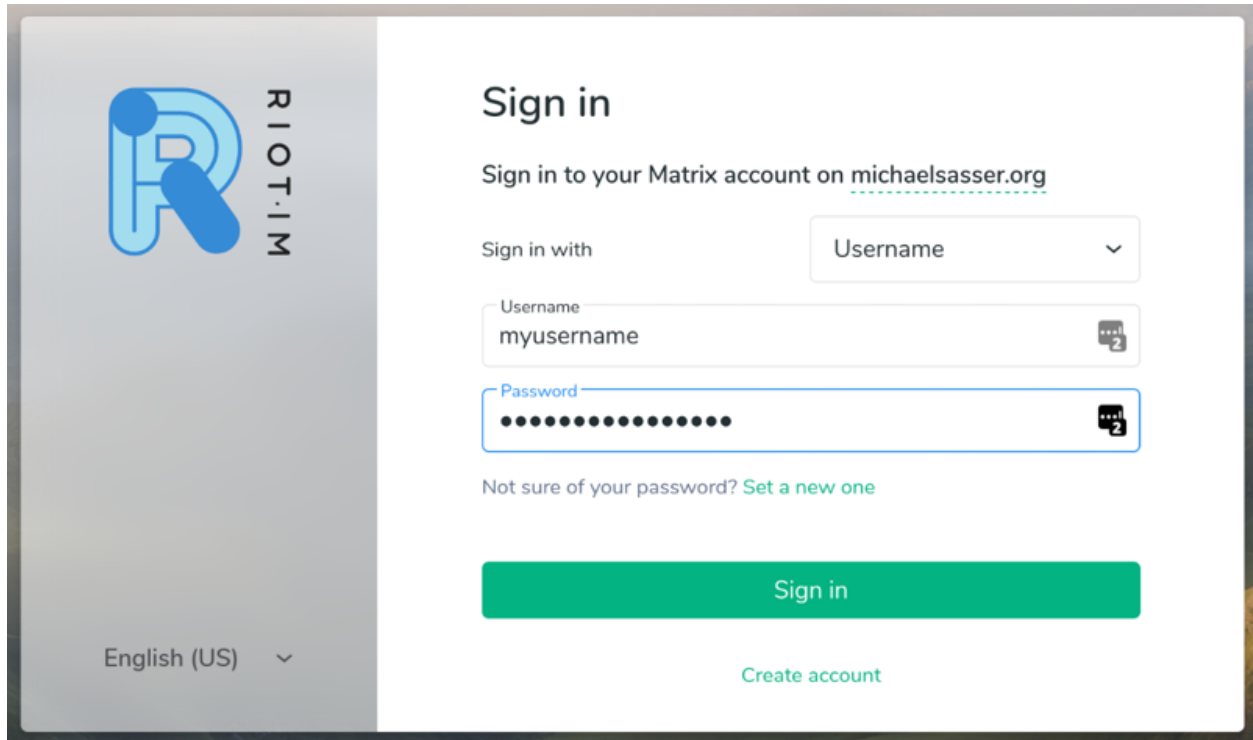


Fig. 2.2: Entering credentials

Note: To be able to use the admin API, you need to have `matrix_nginx_proxy_proxy_matrix_client_redirect_root_uri_to_d` "" and `matrix_nginx_proxy_proxy_matrix_client_api_forwarded_location_synapse_admin_api_enabled: true` in your `vars.yml` file. This will stop the playbook from setting up a redirect ``matrix.yourdomain.tld` to `element.yourdomain.tld`

Warning: Never ever, ever give this token to anyone else. If you have other administrators on that server, they should use their own token. With this token you can login and do anything on that matrix instance in **your name**.

2.4 SSH Public Key

To get easy access to the other matrix plugins (e.g. bridges) and other additional functionality, to communicate with the OCI containers, you need to have a ssh public key installed on your matrix host server. We use ssh access for the following:

- `matrixctl adduser-jitsi`
- `matrixctl deluser-jitsi`

Note: If you where already able to ran the [spantaleev/matrix-docker-ansible-deploy](#) playbook, you have installed the public key before. You are good to go and you can skip this chapter.

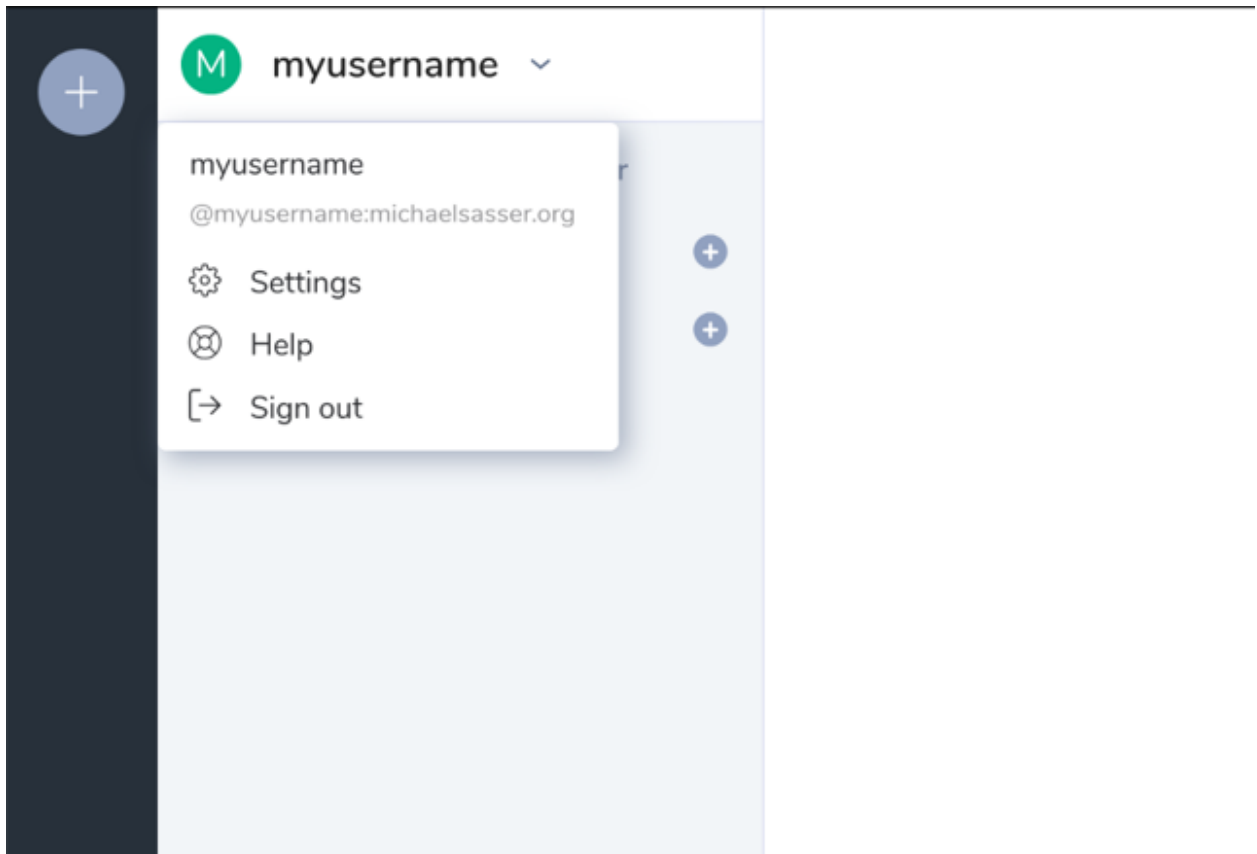


Fig. 2.3: Click on “Settings”

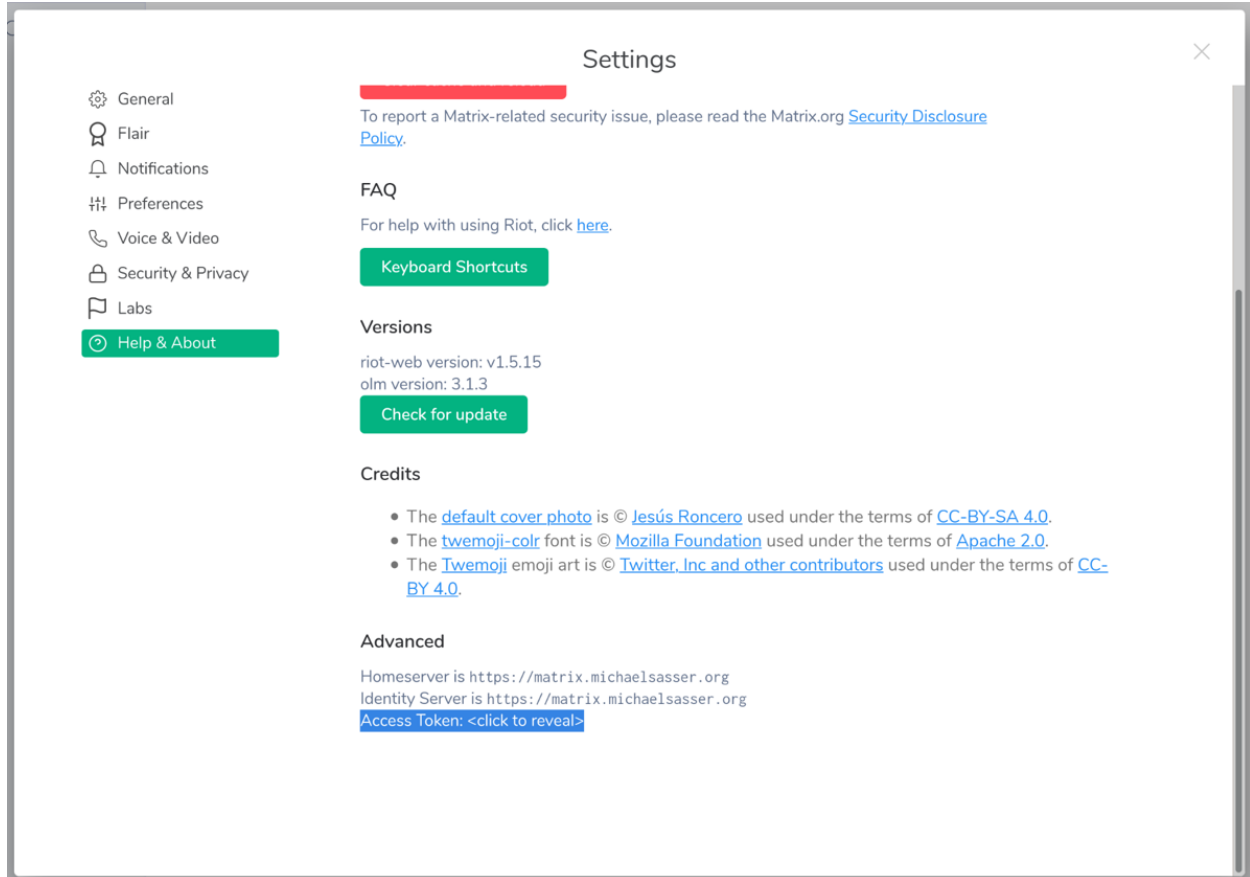


Fig. 2.4: Click on “<click to reveal>”

Advanced

Homeserver is `https://matrix.michaelsasser.org`
 Identity Server is `https://matrix.michaelsasser.org`
 Access Token:

```
MDAxZmxvY2F0aW9uIG1pY2hhZWxzYXNzZXlub3JnCjAwMTNpZGVudGlmaWVylGtle
QowMDEwY2lkIGdlbiA9IDEKMDAzMGNpZCB1c2VvYXZlID0gQG15dXNlcm5hbWU6b
WljaGFibHNhc3Nlci5vcmcKMDAxNmNpZCB0eXBID0gYWNjZXNzCjAwMjFjaWQgbm
9uY2UgPSA3WkBlKkdY3JTRG1CI3Z0CjAwMmZzaWduYXR1cmUgJblnYOAEQJVeHa
MgwnMsAagpZBc8CIC6Dwwy027tfJAK
```

Fig. 2.5: The revealed access token

Note: To get your public key installed you can use your own playbook like described in *Config File* chapter under the [SERVER] section. If you don't want to write your own playbook, follow this guide.

2.4.1 Check your key pair

Check, if you already have a key pair.

```
$ ls -la ~/.ssh/id_*.pub
-rw-r--r-- 1 michael users 767 30. Sep 2014 /home/michael/.ssh/id_rsa.pub
```

If the output looks like the above, you have generated a key pair in the past and you can continue in the next section *Copy Public Key*.

If it looks something like below or prints something like you can continue in the section: *Generate key pair*.

```
$ ls -la ~/.ssh/id_*.pub
zsh: no matches found: /home/michael/.ssh/id_*.pub
# or
$ ls -la ~/.ssh/id_*.pub
ls: cannot access '/home/michael/.ssh/id_*.pub': No such file or directory
```

2.4.2 Generate key pair

To generate your key pair run:

```
$ mkdir ~/.ssh
$ ssh-keygen -t rsa -b 4096 -C "your_email@domain.tld"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:UjqL4jzmuk2YjVqzVHNIay2TShDss5wMHyq3V7ZlI1M your_email@domain.tld
The key's randomart image is:
+---[RSA 4096]-----+
|o                    |
| o                   |
|o . . .             |
|.+. . = oE          |
|+o=.X *.S          |
|o@o+ *=++          |
|=..O..o.* .        |
|.B++ . .           |
|+=*o                |
+----[SHA256]-----+
```

If prints something like below you need to install openssh, sshd or openssh-client (depends on your distribution).

```
$ mkdir ~/.ssh
$ ssh-keygen -t rsa -b 4096 -C "your_email@domain.tld"
bash: ssh-keygen: command not found
```

On Arch linux the installation of openssh would look like:

```
$ pacman -Sy openssh
:: Synchronizing package databases...
core is up to date
extra is up to date
community is up to date
resolving dependencies...
looking for conflicting packages...

Packages (4) dnssec-anchors-20190629-2  ldns-1.7.1-2  libedit-20191231_3.1-1  openssh-8.
↳2p1-3

Total Download Size:  1.40 MiB
Total Installed Size: 7.31 MiB

:: Proceed with installation? [Y/n] y
:: Retrieving packages...
libedit-20191231_3.1-1-x86_64      106.9 KiB   656 KiB/s 00:00 [#####]
↳#####] 100%
dnssec-anchors-20190629-2-any      3.1 KiB   0.00 B/s 00:00 [#####]
↳#####] 100%
ldns-1.7.1-2-x86_64              435.9 KiB   895 KiB/s 00:00 [#####]
↳#####] 100%
openssh-8.2p1-3-x86_64           884.7 KiB  1355 KiB/s 00:01 [#####]
↳#####] 100%
(4/4) checking keys in keyring [#####]
↳#####] 100%
(4/4) checking package integrity [#####]
↳#####] 100%
(4/4) loading package files [#####]
↳#####] 100%
(4/4) checking for file conflicts [#####]
↳#####] 100%
(4/4) checking available disk space [#####]
↳#####] 100%
:: Processing package changes...
(1/4) installing libedit [#####]
↳#####] 100%
(2/4) installing dnssec-anchors [#####]
↳#####] 100%
(3/4) installing ldns [#####]
↳#####] 100%
Optional dependencies for ldns
libpcap: ldns-dpa tool [installed]
(4/4) installing openssh [#####]
↳#####] 100%
Optional dependencies for openssh
xorg-xauth: X11 forwarding
```

(continues on next page)

(continued from previous page)

```
x11-ssh-askpass: input passphrase in X
libfido2: FIDO/U2F support
:: Running post-transaction hooks...
(1/4) Reloading system manager configuration...
Skipped: Current root is not booted.
(2/4) Creating temporary files...
[/usr/lib/tmpfiles.d/journal-nocow.conf:26] Failed to resolve specifier: uninitialized /
→etc detected, skipping
All rules containing unresolvable specifiers will be skipped.
(3/4) Arming ConditionNeedsUpdate...
(4/4) Cleaning up package cache...
```

2.4.3 Copy Public Key

Now copy your public key to your Server:

```
$ ssh-copy-id -i ~/.ssh/id_rsa.pub user@matrix.domain.tld
```

CONTRIBUTOR DOCUMENTATION

First off, thank you for considering contributing to MatrixCtl. Please make sure to read our Code of Conduct before you start Contributing to MatrixCtl.

3.1 Contributor Covenant Code of Conduct

3.1.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, caste, color, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

3.1.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind
- Trolling, insulting or derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

3.1.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

3.1.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

3.1.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement at Abuse@MichaelSasser.org. All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

3.1.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

3.1.7 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 2.0, available at https://www.contributor-covenant.org/version/2/0/code_of_conduct.html.

Community Impact Guidelines were inspired by Mozilla's code of conduct enforcement ladder.

For answers to common questions about this code of conduct, see the FAQ at <https://www.contributor-covenant.org/faq>. Translations are available at <https://www.contributor-covenant.org/translations>.

[homepage]: <https://www.contributor-covenant.org>

[v2.0]: https://www.contributor-covenant.org/version/2/0/code_of_conduct.html

[Mozilla CoC]: <https://github.com/mozilla/diversity>

[FAQ]: <https://www.contributor-covenant.org/faq>

[translations]: <https://www.contributor-covenant.org/translations>

3.1.8 I found a bug / I want to give feedback

If you found a bug or you want to give feedback, please create an [issue](#) using one of the templates.

3.1.9 I have a question

Please check the [discussions](#) first. When you don't find the right thread, feel free to create a new one.

3.1.10 Add a feature

Note: Before you start make sure you hand in an [issue](#). Describe, what you like to change/add, so others are informed, what you are about to change and why you want to change anything.

1. Make sure you have at least Python 3.9, [poetry](#), and [pre-commit](#) installed.
2. Create a fork of MatrixCtl.

3. Clone the fork (`origin`) to your local machine.
4. Add the original repository as a remote named `upstream`.
5. Create a new branch from the `develop` branch. Make sure you use the `git-flow` branching model scheme. (You don't necessarily need `git-flow`). Example: Let's say your issue was issue #42 and you want to create a feature. Your branch name would be `feature/#42` or `feature/#42-my-cool-feature`.
6. Install the required tools with `poetry install -E docs`
7. Implement your feature or fix the bug you described in your issue.
8. Create a Pull Request as soon as possible as draft, so other contributors are able to help you and follow your progress.
9. Make sure to add/alter the documentation.
10. Add/alter tests, to test your code.
11. Describe your changes in one sentence in a newsfragment in `./news/`. You find the categories in the `pyproject.toml` under the `[tool.towncrier] -> directory`. Example: Let's say your issue was issue #42 and you added a bugfix. Give the newsfragment the name `42.bugfix`. A feature would be called `42.feature`.
12. Run `tox`. If everything is green with no errors, you are good to go.
13. Publish your branch to your fork (`origin`).
14. Create a pull request from the Branch, which contains your changes to MatrixCtl's `develop` branch.
15. Once the pull request is reviewed and merged you can pull the changes from `upstream` (the original repository) to your local repository and start over again from 5.. Don't forget to create an issue first.

Note: Do not add any additional requirement without an approval first. Make sure to use the provided `Handlers`, `Helpers`, `Errors` (exceptions) and `Type Hints`.

Note: If you have any questions feel free to ask in the issues, pull requests and discussions.

Note: You often can use one of the `Addons` as template for a new addon.

Handlers

Handlers in MatrixCtl are used to handle the communication between the server and the `Addons` or to load config files.

YAML

Read and parse the configuration file with this module.

```
class matrixctl.handlers.yaml.JinjaUndefined(hint=None, obj=missing, name=None, exc=<class
                                             'jinja2.exceptions.UndefinedError'>)
```

Bases: `Undefined`

Use this class as undefined argument in a Jinja2 Template.

The class replaces every undefined template with an empty string.


```
class matrixctl.handlers.yaml.YAML(paths=None, server=None)
```

Bases: `object`

Use the YAML class to read and parse the configuration file(s).

```
DEFAULT_PATHS: list[Path] = [PosixPath('/etc/matrixctl/config'),
PosixPath('/home/docs/.config/matrixctl/config')]
```

```
JINJA_PREDEFINED: dict[str, str | int] = {'default_api_concurrent_limit': 4,
'default_ssh_port': 22, 'home': '/home/docs', 'user': 'docs'}
```

```
static apply_defaults(server)
```

Apply defaults to the configuration.

Return type

`ConfigServer`

Parameters

server

`[matrixctl.structures.ConfigServer]` The configuration of a (home)server.

Returns

server

`[matrixctl.structures.ConfigServer]` The configuration of a (home)server with applied defaults.

```
get(*keys)
```

Get a value from a config entry safely.

Usage

Pass strings, describing the path in the `self.__yaml` dictionary. Let's say, you are looking for the synapse path:

Return type

`Any`

Parameters

***keys**

`[str]` A tuple of strings describing the values you are looking for.

Returns

answer

`[any]` The value of the entry you described.

Examples

```
from matrixctl.handlers.yaml import YAML

yaml: YAML = YAML()
port: int = yaml.get("server", "ssh", "port")
print(port)
# Output: 22
```

static get_paths_to_config()

Generate a tuple of path which may contain a configuration file. :rtype: `tuple[Path, ...]`

Note: This function preserves the order. The priority of the user configuration in `XDG_CONFIG_HOME` is higher than the global configuration in `/etc/matrixctl/`. The priority of the file extension `yaml` is greater than the priority of the file extension `yaml`.

Warning: The paths returned by this function might not exist.

Returns

config_paths

[`tuple of pathlib.Path`] A tuple of paths, which might contain a config file.

get_server_config(paths, server)

Read and concentrate the config in one dict.

The `server: ...` will be removed from the dict. A new entry `server` will be created, which represents the selected server.

Return type

`Config`

Parameters

paths

[`Iterable of pathlib.Path`] The paths to the configfiles.

server

[`str`] The selected server. (Default: "default")

Returns

server_config

[`matrixctl.typehints.Config`] The config for the selected server.

Notes

When all files were empty or don't exist, an empty dict will be returned.

static read_from_file(yaml, path)

Read the config from a YAML file and render the Jinja2 templates. :rtype: `Config`

Note:

- The `Renderer` does one pass. This means, you can only render templated strings but not the templated string of another templated string.
 - If the file was empty or does not exist, an empty dict will be returned.
-

Parameters

yaml

[`ruamel.yaml.Yaml`] The yaml object.

path

[Path] The path where the config file is located.

Returns**full_config**

[matrixctl.typehints.Config] The full (with server name) config file as dict.

server: `str`

`matrixctl.handlers.yaml.secrets_filter(tree, key)`

Redact secrets when printing the configuration file.

Return type

`Any`

Parameters**tree**

[dict [str, str]] A partial of tree from tree_printer. (Can only be this type) afterwards.

key

[str] A dict key. (Can only be this type)

Returns

`None`

`matrixctl.handlers.yaml.tree_printer(tree, depth=0)`

Print the configuration file recursively.

Return type

`None`

Parameters**tree**

[any] Initial a matrixctl.typehints.Config and partials of it afterwards.

depth

[int] The depth of the table

Returns

`None`

API

Get access to the API of your homeserver.

```
class matrixctl.handlers.api.RequestBuilder(token, domain, path, scheme='https', subdomain='matrix',
api_path='_synapse/admin', api_version='v2', data=None,
json=None, content=None, method='GET', params={},
headers={}, concurrent_limit=4, timeout=5.0,
success_codes=(200, 201, 202, 203, 204, 205, 206, 207,
226))
```

Bases: `object`

Build the URL for an API request.

`api_path: str`
`api_version: str`
`concurrent_limit: int`
`content: Union[str, bytes, Iterable[bytes], None]`
`data: Optional[dict[str, Any]]`
`domain: str`
`headers: dict[str, str]`
`property headers_with_auth: dict[str, str]`
 Get the headers with bearer token.
 Parameters
 None
 Returns
 headers
 [dict [str, str]] Headers with auth. token.
`json: Optional[dict[str, Any]]`
`method: str`
`params: dict[str, Union[str, int]]`
`path: str`
`scheme: str`
`subdomain: str`
`success_codes: tuple[int, ...]`
`timeout: float`
`token: str`

`class matrixctl.handlers.api.RequestStrategy`(*limit: int, step_size: int, concurrent_limit: int, offset: int, iterations: int*)

Bases: `NamedTuple`

Use this `NamedTuple` as request strategy data.

This `NamedTuple` is only used in this module.

`concurrent_limit: int`
 Alias for field number 2

`iterations: int`
 Alias for field number 4

`limit: int`
 Alias for field number 0

offset: `int`

Alias for field number 3

step_size: `int`

Alias for field number 1

async `matrixctl.handlers.api.async_worker(input_queue, output_queue)`

Use this coro as worker to make (a)synchronous request.

Return type

`None`

Returns

`None`

See also:

RequestBuilder

`matrixctl.handlers.api.RequestBuilder`

Attributes

input_queue

`[asyncio.Queue]` The input queue, which provides the `RequestBuilder`.

output_queue

`[asyncio.Queue]` The output queue, which gets the responses of there requests.

async `matrixctl.handlers.api.exec_async_request(request_config)`

Use this coro to generate and run workers and group the responses.

Return type

`Response | list[Response]`

Returns

responses

`[list of httpx.Response or httpx.Response]` Depending on `concurrent_limit` an `request_config`.

See also:

RequestBuilder

`matrixctl.handlers.api.RequestBuilder`

Attributes

request_config

`[RequestBuilder or Generator [RequestBuilder, None, None]]` An instance of an `RequestBuilder` or a list of `RequestBuilder`. If the function gets a `RequestBuilder`, the request will be synchronous. If it gets a `Generator`, the request will be asynchronous.

concurrent_limit

`[int]` The maximum of concurrent workers. (This information must be pulled from the `config`.)

`matrixctl.handlers.api.generate_worker_configs(request_config, next_token, limit)`

Create workers for async requests (minus the already done sync request).

Return type

`Generator[RequestBuilder, None, None]`

Yields

request_config

`[matrixctl.handlers.api.RequestBuilder]` Yields a fully configured RequestsBuilder for every request that has to be done to get all entries.

Notes

Warning Call-By-Reference like behavior! The param `limit` and the `concurrent_limit` in `request_config` will get changed in this function. Make sure to only use them after using this function!

Attributes

request_config

`[matrixctl.handlers.api.RequestBuilder]` An instance of an RequestBuilder from which was used for an initial synchronous request to get the first part of the data and the other two arguments from the response.

next_token

`[int]` The value, which defines from where to start in the next request. You get this value from the response of an initial synchronous request.

total

`[int]` The value which defines how many entries there are. You get this value from the response of an initial synchronous request.

`async matrixctl.handlers.api.group_async_results(input_size, output_queue)`

Use this coro to group the requests afterwards in a single list.

Return type

`list[Exception | Response]`

Returns

responses

`[list of httpx.Response or httpx.Response]` Depending on `concurrent`, it is a `httpx.Response` if `concurrent` is true, otherwise it is a `list` of `httpx.Response`.

Attributes

input_size

`[int]` The number of items in the queue.

output_queue

`[asyncio.Queue]` The output queue, which holds the responses of there requests.

concurrent

`[bool]` When True, make requests concurrently. When False, make requests synchronously.

`matrixctl.handlers.api.preplan_request_strategy(limit, concurrent_limit, max_step_size=100)`

Use this function as helper for optimizing asynchronous requests.

Return type

`RequestStrategy`

Returns

RequestStrategy

`[matrixctl.handlers.api.RequestStrategy]` A Named tuple with the RequestStrategy values.

Attributes**limit**

`[int]` A user entered limit or total.

concurrent_limit: int

The concurrent limit from the config file.

max_step_size

`[int, default=100]` The maximal step size, which is a soft limit. It is usually 100, but that value might be different. Check out the API documentation. We usually take the default one.

`matrixctl.handlers.api.request(request_config)`

Make a (a)synchronous request to the synapse API and receive a response.

Return type

`list[Response] | Response`

Returns**response**

`[httpx.Response]` Returns the response

See also:***RequestBuilder***

`matrixctl.handlers.api.RequestBuilder`

Attributes**request_config**

`[RequestBuilder or Generator [RequestBuilder, None, None]]` An instance of an RequestBuilder or a list of RequestBuilder. If the function gets a RequestBuilder, the request will be synchronous. If it gets a Generator, the request will be asynchronous.

concurrent_limit

`[int]` The maximum of concurrent workers. (This information must be pulled from the config.)

Ansible

Run a ansible playbook with this module.

`matrixctl.handlers.ansible.ansible_run(playbook, tags=None, extra_vars=None)`

Run an ansible playbook.

Return type

`None`

Parameters**playbook**

`[pathlib.Path]` The path to the ansible Playbook

tags

`[str, optional]` The tags to use

extra_vars
[dict [str, str], optional] The extra_vars to use.

Returns

None

Git (VCS)

Update and manage the synapse playbook repository with this module.

class matrixctl.handlers.vcs.VCS(*path*)

Bases: `object`

Update and manage a repository.

property `datetime_last_pulled_commit`: `datetime`

Get the datetime the commit was pulled last from git.

This is used to determine which messages will be produced in the table.

Parameters

None

Returns

`datetime`
[datetime.datetime] The datetime object.

log(*since=None*)

Print a table of date, user and commit message since the last pull.

Return type

None

Parameters

since
[datetime.datetime, optional, default=None] The datetime the last commit was pulled.

Returns

None

pull()

Git pull the latest commits from GH.

Return type

None

Parameters

None

Returns

None

SSH

Run and evaluate commands on the host machine of your synapse server.

```
class matrixctl.handlers.ssh.SSH(address, user=None, port=22)
```

Bases: `object`

Run and evaluate commands on the host machine of your synapse server.

address: `str`

port: `int`

run_cmd(cmd)

Run a command on the host machine and receive a response.

Return type

`SSHResponse`

Parameters

cmd

`[str]` The command to run.

tty

`[bool]` Request a pseudo-terminal from the server (default: False)

Returns

response

`[matrixctl.handlers.ssh.SSHResponse]` Receive stdin, stdout and stderr as response.

user: `str`

```
class matrixctl.handlers.ssh.SSHResponse(stdin: str | None, stdout: str | None, stderr: str | None)
```

Bases: `NamedTuple`

Store the response of a SSH command as response.

stderr: `Optional[str]`

Alias for field number 2

stdin: `Optional[str]`

Alias for field number 0

stdout: `Optional[str]`

Alias for field number 1

Table

Use this handler to generate and print tables.

```
matrixctl.handlers.table.cells_to_str(part, none)
```

Convert all cells to strings and format None values.

Return type

`list[list[str]]`

Parameters

part

[collections.abc.Sequence of collections.abc.Sequence of str] Data or header, in which every cell will be to casted to to strings.

none

[str] A string, which is used to replace None with the specific string.

Returns

part

[list of list of str]

The part, where every cell is of type str.

matrixctl.handlers.table.find_newlines(*data*)

Find newlines and return a dict with positions (key) and occurrences.

Return type

dict[int, int]

Parameters

data

[list of str] Data or headers of the table.

Returns

pos

[dict [int, int]] A dictionary {r: n}, where n are newlines in row r.

Notes

The function only adds an entry to the dict, if there is at least one newline in a row.

matrixctl.handlers.table.format_table_row(*line*, *max_column_len*)

Format a table row into a str.

Return type

str

Parameters

line

[list of str] A data or headers row, which will be formatted to a string.

max_column_len

[tuple of int] A n-tuple which describes the longest string per column. (n is the number of columns)

Returns

row_string

[str] A formatted string, which represents a table row.

matrixctl.handlers.table.get_colum_length(*data*, *headers*)

Transpose rows and find longest line.

Return type

tuple[int, ...]

Parameters

data

[list of list of str] The data part of the table.

headers

[None or list of list of str] The headers part of the table.

Returns**column_length_tuple**

[tuple of int] A n-tuple which describes the longest string per column. (n is the number of columns)

`matrixctl.handlers.table.handle_newlines(part, newlines)`

Update and insert new lines.

Return type

tuple[list[list[str]], set[int]]

Parameters**part**

[list of list of str] Data or headers of the table.

newlines

[dict [int, int]] A dictionary {r: n}, where n are newlines in row r.

Returns**part**

[list of list of str] The part contains the supplemented and updated rows.

inhibit_sep

[set of int] The `inhibit_sep` set contains the line numbers where a separator should be inhibited because the lines handled by this function are belonging together.

`matrixctl.handlers.table.newlines_in_row(row)`

Get the highest number of newlines per row.

The highest number of newlines for a row is used to determine in how many rows the row gets expanded, to get one row per newline - 1.

Return type

int

Parameters**row**

[list of str] Data or headers of the table.

Returns**max_newlines**

[int] The highest number of newlines per row.

`matrixctl.handlers.table.table(table_data, table_headers=None, sep=True, none='-')`

Create a table from data and a optional headers.

Return type

Generator[str, None, None]

Parameters**table_data**

[collections.abc.Sequence of collections.abc.Sequence of str] Data.

table_headers

[collections.abc.Sequence of str, Optional] Headers.

sep

[bool, default = True] True, when there should be a separator between every row of data.

none

[str, default = "-"] A string, which is used to replace None with the specific string.

Yields

table

[str] The table (row for row).

`matrixctl.handlers.table.transpose_newlines_to_rows(split, occurrences)`

Transpose newlines in new rows.

Return type

Generator[list[str], None, None]

Parameters

split

[list of list of str] A list of substring-lists, split from one row, which contains newline characters. The substring-lists are containing strings, which have been split into substrings.

occurrences

[int] The maximal number of newlines across the row.

Yields

row

[list of str] A row for each occurrence.

Database

Talk to the the database.

`class matrixctl.handlers.db.DBConnectionBuilder(host: str, database: str, username: str, password: str, port: int = 5432, timeout: int = 10, scheme: str = 'postgres')`

Bases: `NamedTuple`

Build the URL for an API request.

database: str

Alias for field number 1

host: str

Alias for field number 0

password: str

Alias for field number 3

port: int

Alias for field number 4

scheme: str

Alias for field number 6

timeout: `int`

Alias for field number 5

username: `str`

Alias for field number 2

`matrixctl.handlers.db.db_connect(yaml)`

Connect to a PostgreSQL database.

Return type

`Iterator[Connection]`

Parameters

yaml

`[matrixctl.handlers.yaml.YAML]` The configuration file handler.

Yields

conn

`[psycopg.Connection]` A new Connection instance.

`matrixctl.handlers.db.ssh_tunnel(host, username, remote_port, enabled=True, port=22)`

Create an SSH tunnel.

Return type

`Iterator[Optional[int]]`

Parameters

host

`[str]` The remote host e.g. `127.0.0.1` or `host.domain.tld`.

username

`[str]` The username of the user.

remote_port

`[int]` The port of the application, which should be tunneled.

enabled

`[bool, default: True]` True if the tunnel should be enabled or False if not.

port

`[int, default: 22]` The ssh port

private_key

`[Path or str, optioal]` The path to the private key (Currently Disabled)

Yields

tun

`[int]` The remote port

None

`[None]` Yields none, when the tunnel is disabled (`enabled = False`).

Notes

The tunnel will only be created, when it is enabled. If the tunnel is disabled (`enabled = False`), the function will yield `None` instead of the local bind port.

Examples

```
with ssh_tunnel("127.0.0.1", myuser, 5432) as remote_port:
    print(f"The local bind port is: {local_bind_port}")
# The local bind port is: 8765
```

Helpers

Helpers or helper function are common functions used throughout the project.

Addon Manager

Use this module as addon manager.

```
matrixctl.addon_manager.import_addons_from(addon_directory, addon_module, parser_name)
```

Import addons in (global) addons.

Return type

`None`

Parameters

addon_directory

[`str`] The absolute path as string to the addon directory.

addon_module

[`str`] The import path (with dots `.` not slashes `/`) to the addons from project root e.g. “matrixctl.addons”.

parser_name

[`str`] The name of the module the subparser is in.

..Note:

The nothing will be imported, when the subparser is not in (global) addons. To add the subparse to addons you need to decorate the subparsers with `matrixctl.addon_manager.subparser`

Returns

none

[`None`] The function always returns `None`.

```
matrixctl.addon_manager.setup(func)
```

Add subparsers to the (main) parser.

Return type

`ArgumentParser`

Parameters

func

[`matrixctl.addon_manager.ParserSetupType`] A callback to the main parser.

Returns**parser**

[`argparse.ArgumentParser`] The parser which includes all subparsers.

`matrixctl.addon_manager.subparser(func)`

Decorate subparsers with, to add them to (global) addons on import.

Return type

`Callable[[_SubParsersAction], None]`

Parameters**func**

[`matrixctl.addon_manager.SubParserType`] A subparser.

..Note:

The nothing will be imported, when the subparser is not in (global) addons. To add the subparse to addons you need to decorate the subparsers with `matrixctl.addon_manager.subparser`

Returns**decorated_func**

[`matrixctl.addon_manager.SubParserType`] The same subparser which was used as argument. (Without any changes)

Package Version

Get the packages version.

The package's version is determined by first checking if a `pyproject.toml` exists. If this is given, the version variable is searched line by line in the file using a regular expression. When a match occurs, the version is returned. If the ``pyproject.toml does not exist, e.g. because the package was installed, it uses the version stored in the package's metadata. In any case, if the version could not be determined, it will return None.`

`matrixctl.package_version.get_version(name, file)`

Get the version of a Python package.

Return type

`Optional[str]`

Parameters**name**

[`str`] The packages `__name__`.

file

[`str`] The `__name__` of `__init__.py`

Returns**version**

[`str` or `None`] The package version, if the package is installed and the version of it is stored in the packages metadata.

Examples

```
# file: __init__.py

from .package_version import get_version

__version__: str | None = get_version(__name__, __file__)

# or
__version__: str = get_version(__name__, __file__) or "Unknown"

# Optional:
if __version__ is None:
    raise ValueError("Could not find the version of the package.")
```

```
# file: conf.py (sphinx)

import sys

sys.path.insert(0, os.path.abspath("../"))
sys.path.insert(0, os.path.abspath("../.."))

from matrixctl.package_version import get_version

__version__: str = (
    get_version("matrixctl", Path(__file__).parent) or "Unknown"
)
```

Sanitizers

Use the functions of this module as printing helpers.

class `matrixctl.sanitizers.MessageType`(*value*)

Bases: `Enum`

Use this enum for describing message types.

Supported events:

message_type	Usage
m.room.message	This event is used when sending messages in a room
m.room.name	This event sets the name of an room
m.room.topic	This events sets the room topic
m.room.avatar	This event sets the room avatar
m.room.pinned_events	This event pins events
m.room.member	Adjusts the membership state for a user in a room
m.room.join_rules	This event sets the join rules
m.room.create	This event creates a room
m.room.power_levels	This event sets a rooms power levels
m.room.redaction	This event redacts other events


```

M_ROOM_AVATAR = 'm.room.avatar'
M_ROOM_CREATE = 'm.room.create'
M_ROOM_JOIN_RULES = 'm.room.join_rules'
M_ROOM_MEMBER = 'm.room.member'
M_ROOM_MESSAGE = 'm.room.message'
M_ROOM_NAME = 'm.room.name'
M_ROOM_PINNED_EVENTS = 'm.room.pinned_events'
M_ROOM_POWER_LEVELS = 'm.room.power_levels'
M_ROOM_REDACTION = 'm.room.redaction'
M_ROOM_TOPIC = 'm.room.topic'

```

`matrixctl.sanitizers.sanitize(pattern, identifier, error_message)`

Create a new sanitizer based on compiled RegEx expressions.

A helper function for simplifying the latter sanitize identifier specific functions.

Return type

`Union[str, Literal[False], None]`

Parameters

pattern

`[typing.Pattern]` The RegEx pattern used for the specific sanitizing

identifier

`[typing.Any, optional]` The identifier to sanitize based on the pattern

error_message

`[str]` The error string used for logging errors

Returns

result

`[typing.Literal[False] or str, optional]` The function returns None if `identifier` is None, the sanitized string, when it is valid, otherwise False

`matrixctl.sanitizers.sanitize_event_identifier(event_identifier)`

Sanitize an event identifier.

Return type

`Union[str, Literal[False], None]`

Parameters

event_identifier

`[typing.Any]` The event identifier to sanitize

Returns

result

`[typing.Literal[False] or str, optional]` The function returns None if `event_identifier` is None, the sanitized string, when it is valid, otherwise False

Examples

```
>>> sanitize_event_identifler(
...     "$event-abcdefghijklmH4omLrEumu7Pd01Qp-LySpK_Y"
... )
'$event-abcdefghijklmH4omLrEumu7Pd01Qp-LySpK_Y'
```

```
>>> sanitize_event_identifler(
...     " $event-abcdefghijklmH4omLrEumu7Pd01Qp-LySpK_Y "
... )
'$event-abcdefghijklmH4omLrEumu7Pd01Qp-LySpK_Y'
```

```
>>> sanitize_event_identifler("something invalid")
False
```

```
>>> sanitize_event_identifler(None)
```

`matrixctl.sanitizers.sanitize_message_type(message_type)`

Sanitize an message type.

Return type

`Union[MessageType, Literal[False], None]`

Parameters

`message_type`

`[typing.Any]` The event identifier to sanitize

Returns

`message_type_sanitized`

`[typing.Literal[False] or MessageType, optional]` The function returns `None` if `message_type` is `None`, `MessageType`, if it is valid, otherwise `False`

Examples

```
>>> sanitize_message_type("m.room.message")
<MessageType.M_ROOM_MESSAGE: 'm.room.message'>
```

```
>>> sanitize_message_type("M.RooM.MeSsAgE")
<MessageType.M_ROOM_MESSAGE: 'm.room.message'>
```

```
>>> sanitize_message_type(" m.room.message ")
<MessageType.M_ROOM_MESSAGE: 'm.room.message'>
```

```
>>> sanitize_message_type(MessageType.M_ROOM_MESSAGE)
<MessageType.M_ROOM_MESSAGE: 'm.room.message'>
```

```
>>> sanitize_message_type("something invalid")
False
```

```
>>> sanitize_message_type(None)
```

`matrixctl.sanitizers.sanitize_room_identifier(room_identifier)`

Sanitize an room identifier.

Return type

`Union[str, Literal[False], None]`

Parameters

room_identifier

`[typing.Any]` The room identifier to sanitize

Returns

room_identifier_sanitized

`[typing.Literal[False] or str, optional]` The function returns `None` if `room_identifier` is `None`, the sanitized string, when it is valid, otherwise `False`

Examples

```
>>> sanitize_room_identifier(
...     "!room:domain.tld"
... )
'!room:domain.tld'
```

```
>>> sanitize_room_identifier(
...     " !room:domain.tld "
... )
'!room:domain.tld'
```

```
>>> sanitize_room_identifier("something invalid")
False
```

```
>>> sanitize_room_identifier(None)
```

`matrixctl.sanitizers.sanitize_user_identifier(user_identifier)`

Sanitize an user identifier.

Return type

`Union[str, Literal[False], None]`

Parameters

user_identifier

`[typing.Any]` The user identifier to sanitize

Returns

event_identifier_sanitized

`[typing.Literal[False] or str, optional]` The function returns `None` if `user_identifier` is `None`, the sanitized string, when it is valid, otherwise `False`

Examples

```
>>> sanitize_user_identifier(  
...     "@user:domain.tld"  
... )  
'@user:domain.tld'
```

```
>>> sanitize_user_identifier(  
...     " @user:domain.tld "  
... )  
'@user:domain.tld'
```

```
>>> sanitize_user_identifier("something invalid")  
False
```

```
>>> sanitize_user_identifier(None)
```

Print

Use the functions of this module as printing helpers.

`matrixctl.print_helpers.human_readable_bool(b)`

Use this helper function to get a “yes” or “no” string from a “bool”.

Return type

`str`

Parameters

b

[`any`] The value to “convert”.

Returns

answer

[`str`] "Yes" if expression is True, or "False" if expression is False.

`matrixctl.print_helpers.timestamp_to_dt(ts, sep='')`

Convert a timestamp (in ms) to a datetime string.

Return type

`str`

Parameters

ts

[`str`] The value to “convert”.

sep

[`str`] The separator between the date and the time.

Returns

dt

[`str`] A datetime string (e.g. 2021-08-21 04:55:55)

Password

Use the functions of this module as helpers for passwords.

`matrixctl.password_helpers.ask_password()`

Ask the user to create a password.

The user will be asked twice for a password. After that the function compares the two entered passwords. If they are the same, and not empty, the function will return the password.

Return type

`Union[str, NoReturn]`

Parameters

None

Returns

password

`[str]` The user entered password.

`matrixctl.password_helpers.ask_question(question='Is everything correct?')`

Asks the user a simple yes/no a question.

Return type

`bool`

Parameters

question

`[str]` The yes/no question the user should be asked

Returns

answer

`[bool]` True if the answer was y / j, or False if the answer was n

Notes

- The user entered value is case-insensitive.
- If the user answered with an invalid answer (not y / j / n) the function asks again.

`matrixctl.password_helpers.create_user(user, admin=None)`

Ask the user to create a password.

The user will be asked twice for a password. After that the function compares the two entered passwords. If they are the same, and not empty, the function will ask the user if the data is correct without disclosing the password.

Return type

`Union[str, NoReturn]`

Parameters

user

`[str]` The username.

admin

`[bool or none, default is None]` True, if the user will be an admin, False, if the user will not have elevated permissions. None, if the admin permissions are not an criteria. The field will be omitted in the data.

Returns

password
[str] The user entered password.

Type Hints, Structures and Errors

MatrixCtl is strictly typed to avoid some bugs and help contributors in the future to easily identify what they are dealing with. They can be used by third party tools such as type checkers, IDEs, linters, etc.

In addition we make use of TypedDict to create typed structures (add type hints to e.g. the configuration).

MatrixCtl specifies some additional errors. Those errors are informing the user that, getting a traceback is a bug in this application. They are giving the person instructions, how to hand in a bug report.

Type Hints

Use this module for custom type definitions.

Structures

Use this module for structures.

class matrixctl.structures.**Config**

Bases: TypedDict

Cast the YAML config to a typed dict.

server: *ConfigServer*

servers: dict[str, *ConfigServer*]

class matrixctl.structures.**ConfigServer**

Bases: TypedDict

Add a *server* to the YAML config structure.

ansible: *ConfigServerAnsible*

api: *ConfigServerAPI*

maintenance: *ConfigServerMaintenance*

ssh: *ConfigServerSSH*

synapse: *ConfigServerSynapse*

class matrixctl.structures.**ConfigServerAPI**

Bases: TypedDict

Add *api* to *server* in the YAML config structure.

concurrent_limit: int

domain: str

token: str

username: `str`

class `matrixctl.structures.ConfigServerAnsible`

Bases: `TypedDict`

Add *ansible* to *server* in the YAML config structure.

playbook: `str`

class `matrixctl.structures.ConfigServerMaintenance`

Bases: `TypedDict`

Add *maintenance* to *server* in the YAML config structure.

tasks: `list[str]`

class `matrixctl.structures.ConfigServerSSH`

Bases: `TypedDict`

Add *ssh* to *server* in the YAML config structure.

address: `str`

port: `int`

user: `str`

class `matrixctl.structures.ConfigServerSynapse`

Bases: `TypedDict`

Add *synapse* to *server* in the YAML config structure.

playbook: `str`

Errors

Use the exceptions of this module for the application.

exception `matrixctl.errors.ConfigFileError`(*message=None, payload=None*)

Bases: `Error`

Use this exception class for everything related to the config file.

exception `matrixctl.errors.Error`(*message=None, payload=None*)

Bases: `Exception`

Use this exception class as base error for the project.

BUGMSG: `str` = 'If you discover this message, please try updating MatrixCtl. If you see this message again, we would be glad, if you would run the same command again in debug-mode (`matrixctl -d [...]`) and hand in a "Bug report" at <https://github.com/MichaelSasser/matrixctl/issues> with the complete output.\n\nPython version: 3.10.8 final\nMatrixCtl version: 0.12.0b2 \n'

exception `matrixctl.errors.ExitQWorker`

Bases: `Exception`

Use this exception when you want to exit an Queue worker.

exception `matrixctl.errors.InternalResponseError`(*message=None, payload=None*)

Bases: `Error`

Use this exception class for everything else.

Addons

Addons are the commands listed below, e.g. `adduser` to create a new user on the homeserver or `deploy` to deploy the ansible playbook.

adduser

Use this module to add the `adduser` subcommand to `matrixctl`.

`matrixctl.addons.adduser.parser.subparser_adduser`(*subparsers*)

Create a subparser for the `matrixctl adduser` command.

Parameters

`subparsers`

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to add the `adduser` subcommand to `matrixctl`.

`matrixctl.addons.adduser.addon.addon`(*arg, yaml*)

Add a User to the synapse instance.

It runs `ask_password()` first. If `ask_password()` returns `None` it generates a password with `gen_password()`. Then it gives the user a overview of the username, password and if the new user should be generated as admin (if you added the `--admin` argument). Next, it asks a question, if the entered values are correct with the `ask_question` function.

If the `ask_question` function returns `True`, it continues. If not, it starts from the beginning.

Depending on the `--ansible` switch it runs the `adduser` command via ansible or the API

Return type

`int`

Parameters

`arg`

[`argparse.Namespace`] The Namespace object of `argparse`'s `parse_args()`.

`yaml`

[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns

`err_code`

[`int`] Non-zero value indicates error code, or zero on success.

deluser

Use this module to add the deluser subcommand to matrixctl.

```
matrixctl.addons.deluser.parser.subparser_deluser(subparsers)
```

Create a subparser for the matrixctl deluser command.

Parameters

subparsers

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to add the deluser subcommand to matrixctl.

```
matrixctl.addons.deluser.addon.addon(arg, yaml)
```

Delete a user from the the matrix instance.

Return type

`int`

Parameters

arg

[`argparse.Namespace`] The Namespace object of `argparse`'s `parse_args()`

yaml

[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns

err_code

[`int`] Non-zero value indicates error code, or zero on success.

adduser-jitsi

Use this module to add the adduser-jitsi subcommand to matrixctl.

```
matrixctl.addons.adduser_jitsi.parser.subparser_adduser_jitsi(subparsers)
```

Create a subparser for the matrixctl adduser-jitsi command.

Parameters

subparsers

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to add the adduser-jitsi subcommand to matrixctl.

```
matrixctl.addons.adduser_jitsi.addon.addon(arg, yaml)
```

Add a User to the jitsi instance.

Return type

`int`

Parameters

arg
[`argparse.Namespace`] The Namespace object of `argparse.parse_args()`.

yaml
[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns

err_code
[`int`] Non-zero value indicates error code, or zero on success.

deluser-jitsi

Use this module to add a `deluser-jitsi` subcommand to `matrixctl`.

`matrixctl.addons.deluser_jitsi.parser.subparser_deluser_jitsi(subparsers)`

Create a subparser for the `matrixctl deluser-jitsi` command.

Parameters

subparsers
[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to add a `deluser-jitsi` subcommand to `matrixctl`.

`matrixctl.addons.deluser_jitsi.addon.addon(arg, yaml)`

Delete a user from the `jitsi` instance.

It uses the `Ssh` class from the `ssh_handler`.

Return type

`int`

Parameters

arg
[`argparse.Namespace`] The Namespace object of `argparse.parse_args()`

yaml
[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns

err_code
[`int`] Non-zero value indicates error code, or zero on success.

check

Use this module to add the check subcommand to matrixctl.

```
matrixctl.addons.check.parser.subparser_check(subparsers)
```

Create a subparser for the matrixctl check command.

Parameters

subparsers

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to add the check subcommand to matrixctl.

```
matrixctl.addons.check.addon.addon(_, yaml)
```

Check the deployment with audible.

Return type

`int`

Parameters

arg

[`argparse.Namespace`] The Namespace object of `argparse`'s `parse_args()`

yaml

[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns

err_code

[`int`] Non-zero value indicates error code, or zero on success.

deploy

Use this module to add the deploy subcommand to matrixctl.

```
matrixctl.addons.deploy.parser.subparser_deploy(subparsers)
```

Create a subparser for the matrixctl deploy command.

Parameters

subparsers

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to add the deploy subcommand to matrixctl.

```
matrixctl.addons.deploy.addon.addon(arg, yaml)
```

Deploy the ansible playbook.

Return type

`int`

Parameters

arg
[`argparse.Namespace`] The Namespace object of `argparse.parse_args()`

yaml
[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns

err_code
[`int`] Non-zero value indicates error code, or zero on success.

maintenance

Use this module to add the `maintenance` subcommand to `matrixctl`.

`matrixctl.addons.maintenance.parser.subparser_maintenance(subparsers)`

Create a subparser for the `matrixctl maintenance` command.

Parameters

subparsers
[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to add the `maintenance` subcommand to `matrixctl`.

`class matrixctl.addons.maintenance.addon.Task(value)`

Bases: `Enum`

Use this enum for describing the maintenance task.

Supported tasks:

tasks	Description
<code>vacuum</code>	Reclaims storage occupied by dead tuples.
<code>compress_state</code>	Compress Synapse State Tables.

`COMPRESS_STATE = 'rust-synapse-compress-state'`

`VACUUM = 'run-postgres-vacuum'`

`matrixctl.addons.maintenance.addon.addon(arg, yaml)`

Run the maintenance procedure of the ansible playbook.

Return type

`int`

Parameters

arg
[`argparse.Namespace`] The Namespace object of `argparse.parse_args()`.

yaml
[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns**err_code**

[int] Non-zero value indicates error code, or zero on success.

`matrixctl.addons.maintenance.addon.print_tasks()`

Print a list of all available tasks.

Return type

None

start

Use this module to add the (re)start subcommand to matrixctl.

`matrixctl.addons.start.parser.subparser_restart(subparsers)`

Create a subparser for the matrixctl restart command.

Parameters**subparsers**

[argparse._SubParsersAction or typing.Any] The object which is returned by `parser.add_subparsers()`.

Returns

None

See also:**matrixctl.start.subparser_start**

Subparser for matrixctl start.

Notes

This is a alias for matrixctl start

`matrixctl.addons.start.parser.subparser_start(subparsers)`

Create a subparser for the matrixctl start command.

Parameters**subparsers**

[argparse._SubParsersAction of typing.Any] The object which is returned by `parser.add_subparsers()`.

Returns

None

Use this module to add the (re)start subcommand to matrixctl.

`matrixctl.addons.start.addon.addon(_, yaml)`

Start/Restart the OCI containers.

Return type

int

Parameters

arg
[argparse.Namespace] The Namespace object of argparse's parse_args().

yaml
[matrixctl.handlers.yaml.YAML] The configuration file handler.

Returns

err_code
[int] Non-zero value indicates error code, or zero on success.

stop

Use this module to add the stop subcommand to matrixctl.

matrixctl.addons.stop.parser.**subparser_stop**(subparsers)

Create a subparser for the matrixctl stop command.

Parameters

subparsers
[argparse._SubParsersAction of typing.Any] The object which is returned by parser.add_subparsers().

Returns

None

Use this module to add the stop subcommand to matrixctl.

matrixctl.addons.stop.addon.**addon**(_, yaml)

Stop the OCI containers.

Return type

int

Parameters

arg
[argparse.Namespace] The Namespace object of argparse's parse_args().

yaml
[matrixctl.handlers.yaml.YAML] The configuration file handler.

Returns

err_code
[int] Non-zero value indicates error code, or zero on success.

update

Use this module to add the update subcommand to matrixctl.

matrixctl.addons.update.parser.**subparser_update**(subparsers)

Create a subparser for the matrixctl update command.

Parameters

subparsers
[argparse._SubParsersAction of typing.Any] The object which is returned by parser.add_subparsers().

Returns**None**

Use this module to add the update subcommand to `matrixctl`.

```
matrixctl.addons.update.addon.addon(, yaml)
```

Update the synapse playbook with git.

Return type**int****Parameters****arg**

[`argparse.Namespace`] The Namespace object of `argparse`'s `parse_args()`.

yaml

[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns**err_code**

[`int`] Non-zero value indicates error code, or zero on success.

user

Use this module to add the user subcommand to `matrixctl`.

```
matrixctl.addons.user.parser.subparser_user(subparsers)
```

Create a subparser for the `matrixctl user` command.

Parameters**subparsers**

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns**None**

Use this module to add the user subcommand to `matrixctl`.

```
matrixctl.addons.user.addon.addon(arg, yaml)
```

List information about an registered user.

It uses the admin API to get a python dictionary with the information. The `generate_user_tables` function makes the information human readable.

Return type**int****Parameters****arg**

[`argparse.Namespace`] The Namespace object of `argparse`'s `parse_args()`.

yaml

[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns

err_code

[int] Non-zero value indicates error code, or zero on success.

Examples

```

$ matrixctl user dwight
User:
+-----+-----+
| Name           | dwight |
| Password Hash  | $2b$12$9DUNderm1ffL1NincPap3RC |
|                | ompaNY1725.s1OUghAvEnu5cranT0n |
| Guest         | False  |
| Admin          | True   |
| Consent Version |        |
| Consent Server Notice Sent |        |
| Appservice Id  |        |
| Creation Ts    | 2020-04-14 13:04:21 |
| User Type      |        |
| Deactivated    | False  |
| Displayname    | Dwight Schrute |
| Avatar Url     | mxc://dunder-mifflin.com/sCr4 |
|                | nt0nsr4ng13rW45Cr33d |
+-----+-----+

Threepid:
+-----+-----+
| Medium         | email |
| Address        | dwight_schrute@dunder-mifflin.com |
| Validated At   | 2020-04-14 15:30:21.123000 |
| Added At      | 2020-04-14 15:29:19.100000 |
+-----+-----+

```

If the user does not exist, the return looks like:

Use this module to add the `rooms` subcommand to `matrixctl`.

`matrixctl.addons.user.to_table.generate_user_tables(user_dict, len_domain)`

Generate a main user table and threepid user tables.

The function generates first a main user table and then for every threepid an additional table from a `user_dict`. It renames and makes the output human readable.

Return type

`list[list[tuple[str, str]]]`

Parameters**user_dict**

[dict [str, Any]] The line as dict, a JSON string which was converted to a Python dictionary. (This is not a `Collections.UserDict`)

len_domain

[int] The length in characters of the domain.

Returns

err_code

[int] A list in the format: [[main], threepids_0, ... ,threepids_n]

Notes

This function is a recursive function.

`matrixctl.addons.user.to_table.make_human_readable(k, user_dict, len_domain)`

Make a key/value pair of a user (line) human readable, by modifying.

Return type

tuple[str, str]

Parameters**k**

[str] The key

user_dict

[dict [str, Any]] The line as dict, a JSON string which was converted to a Python dictionary. (This is not a `Collections.UserDict`)

len_domain

[int] The length in characters of the domain.

Returns**err_code**

[int] Non-zero value indicates error code, or zero on success.

Notes

This function is used as helper by `matrixctl.user.generate_user_tables`.

`matrixctl.addons.user.to_table.to_table(user_dict, len_domain)`

Use this function as helper to print the room table.

Return type

Generator[str, None, None]

Parameters**user_dict**

[matrixctl.typehints.JsonDict] The user data from the API

len_domain

[int] The length of the homeservers domain.

Yields**table_lines**

[str] The table lines.

users

Use this module to add the users subcommand to `matrixctl`.

`matrixctl.addons.users.parser.subparser_users(subparsers)`

Create a subparser for the `matrixctl users` command.

Parameters

subparsers

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to add the users subcommand to `matrixctl`.

`matrixctl.addons.users.addon.addon(arg, yaml)`

Print a table/json of the matrix users.

This function generates and prints a table of users or uses json as output format.

The table can be modified. `:rtype: int`

- If you want guests in the table use the `--with-guests` switch.
- If you want deactivated user in the table use the `--with-deactivated` switch.

Parameters

arg

[`argparse.Namespace`] The Namespace object of `argparse`'s `parse_args()`.

yaml

[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns

err_code

[`int`] Non-zero value indicates error code, or zero on success.

Notes

- Needs API version 2 (`synapse 1.28` or greater) to work.
- API version 1 is deprecated. If you encounter problems please upgrade to the latest `synapse` release.

Use this module to add the rooms subcommand to `matrixctl`.

`matrixctl.addons.users.to_table.to_table(users_list, len_domain)`

Use this function as helper to pint the users table.

Return type

`Generator[str, None, None]`

Parameters

users_list

[`list` of `matrixctl.typehints.JsonDict`] A list of rooms from the API.

len_domain

[[int](#)] The length of the homeservers domain.

Yields**table_lines**

[[str](#)] The table lines.

Examples

```
$ matrixctl users
+-----+-----+-----+-----+-----+-----+
| Name   | Deactivated | Shadow-Banned | Admin | Guest | Display Name |
+-----+-----+-----+-----+-----+-----+
| dwight | No          | No             | Yes   | No    | Dwight       |
| pam    | No          | No             | No    | No    | Pam          |
| jim    | No          | No             | No    | No    | Jim          |
| creed  | No          | Yes            | No    | No    | Creed        |
| stanley | No          | No             | No    | No    | Stanley      |
| kevin  | No          | No             | No    | No    | Cookie       |
| angela | No          | No             | No    | No    | Angela       |
| phyllis | No          | No             | No    | No    | Phyllis      |
| tobi   | No          | No             | No    | No    | TobiHR       |
| michael | No          | No             | Yes   | No    | Best Boss    |
| andy   | No          | No             | No    | No    | Andy         |
+-----+-----+-----+-----+-----+-----+

```

report

Use this module to add the `report` subcommand to `matrixctl`.

`matrixctl.addons.report.parser.subparser_report(subparsers)`

Create a subparser for the `matrixctl report` command.

Parameters**subparsers**

[[argparse._SubParsersAction](#) of [typing.Any](#)] The object which is returned by `parser.add_subparsers()`.

Returns

[None](#)

Use this module to add the `users` subcommand to `matrixctl`.

`matrixctl.addons.report.addon.addon(arg, yaml)`

Print a table of the reported events.

Return type

[int](#)

Parameters**arg**

[[argparse.Namespace](#)] The `Namespace` object of `argparse.parse_args()`.

yaml

[*matrixctl.handlers.yaml.YAML*] The configuration file handler.

Returns

err_code

[*int*] Non-zero value indicates error code, or zero on success.

reports

Use this module to add the `reports` subcommand to `matrixctl`.

`matrixctl.addons.reports.parser.subparser_reports(subparsers)`

Create a subparser for the `matrixctl reports` command.

Parameters

subparsers

[*argparse._SubParsersAction* of *typing.Any*] The object which is returned by `parser.add_subparsers()`.

Returns

None

Use this module to add the `reports` subcommand to `matrixctl`.

`matrixctl.addons.reports.addon.addon(arg, yaml)`

Print a table/json of the reported events.

Return type

int

Parameters

arg

[*argparse.Namespace*] The `Namespace` object of `argparse.parse_args()`.

yaml

[*matrixctl.handlers.yaml.YAML*] The configuration file handler.

Returns

err_code

[*int*] Non-zero value indicates error code, or zero on success.

Use this module to add the `rooms` subcommand to `matrixctl`.

`matrixctl.addons.reports.to_table.to_table(events_raw)`

Use this function as helper to print the events as table.

Return type

Generator[*str*, *None*, *None*]

Parameters

events_raw

[*list* of *matrixctl.typehints.JsonDict*] A list of events from the API.

Yields

table_lines

[*str*] The table lines.

Examples

```

$ matrixctl reports
+-----+
| ID          | 2          |
| Date        | 2021-05-08 |
| Time        | 21:04:55  |
| Score       | -100       |
| Canonical Alias | -         |
| Room Name   | SomeRoom   |
| Room ID     | !AbCdEfGhIjKlMnOpQr:domain.tld
| Event ID    | $Q_sksd348jaidj93jff9ojwef9h329ofijewhf932h9f
| Defendant   | @mallory:matrix.org
| Plaintiff   | @alice:myhomeverver.tld
| Reason      | Likes JavaScript
+-----+
| ID          | 1          |
| Date        | 2020-08-15 |
| Time        | 09:09:57  |
| Score       | -100       |
| Canonical Alias | -         |
| Room Name   | -          |
| Room ID     | !AbCdEfGhIjKlMnOpQr:matrix.org
| Event ID    | $123456789012345678901:matrix.org
| Defendant   | @eve:matrix.org
| Plaintiff   | @bob:myhomeserver.tld
| Reason      | Hates The Office (US)
+-----+

```

rooms

Use this module to add the rooms subcommand to `matrixctl`.

```
matrixctl.addons.rooms.parser.subparser_rooms(subparsers)
```

Create a subparser for the `matrixctl rooms` command.

Parameters

subparsers

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to add the rooms subcommand to `matrixctl`.

```
matrixctl.addons.rooms.addon.addon(arg, yaml)
```

Generate a table of the matrix rooms.

Return type

`int`

Parameters

arg
[`argparse.Namespace`] The Namespace object of `argparse.parse_args()`.

yaml
[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns

err_code
[`int`] Non-zero value indicates error code, or zero on success.

`matrixctl.addons.rooms.addon.filter_empty_rooms(rooms, local_users=True)`

Filter for empty rooms.

Return type

[`list[dict[str, Any]]`]

Parameters

rooms
[`list of matrixctl.typehints.JsonDict`] A rooms list.

local_users
[`bool`] `true`: Filter, if no local user is in the room. `false`: Filter, if no user is in the room.

Returns

rooms
[`list of matrixctl.typehints.JsonDict`] The filtered list.

`matrixctl.addons.rooms.addon.generate_output(rooms, to_json)`

Use this helper to generate the output.

Return type

[`None`]

Parameters

rooms
[`list of matrixctl.typehints.JsonDict`] A list of rooms from the API.

to_json
[`bool`] `True`, when the output should be in the JSON format. `False`, when the output should be a table.

Returns

[`None`]

Use this module to add the rooms subcommand to `matrixctl`.

`matrixctl.addons.rooms.to_table.to_table(rooms_list)`

Use this function as helper to pint the room table.

Return type

[`Generator[str, None, None]`]

Parameters

rooms_list
[`list of matrixctl.typehints.JsonDict`] A list of rooms from the API.

Yields

table_lines
 [str] The table lines.

purge-history

The purge-history command allows to purge historic events from the database.

Use this module to add the `purge-history` subcommand to `matrixctl`.

`matrixctl.addons.purge_history.parser.subparser_purge_history(subparsers)`

Create a subparser for the `matrixctl purge-history` command.

Parameters

subparsers
 [argparse._SubParsersAction of typing.Any] The object which is returned by `parser.add_subparsers()`.

Returns

None

The purge-history command allows to purge historic events from the database.

Use this module to add the `purge-history` subcommand to `matrixctl`.

`matrixctl.addons.purge_history.addon.addon(arg, yaml)`

Purge historic message events from the Database.

Return type

int

Parameters

arg
 [argparse.Namespace] The Namespace object of `argparse.parse_args()`.

yaml
 [`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns

err_code

[int] Non-zero value indicates error code, or zero on success.

The purge-history command allows to purge historic events from the database.

Use this module to add the `purge-history` subcommand to `matrixctl`.

`matrixctl.addons.purge_history.dialog.dialog_input(arg)`

Ask questions and sanitize them.

Return type

Union[dict[str, str | int], NoReturn]

Parameters

arg
 [argparse.Namespace] The Namespace object of `argparse.parse_args()`.

Returns

request_body

[`typing.Dict [str, str]` or `NoReturn`] Non-zero value indicates error code, or zero on success.

The purge-history command allows to purge historic events from the database.

Use this module to add the `purge-history` subcommand to `matrixctl`.

`matrixctl.addons.purge_history.handler.handle_purge_status(yaml, purge_id)`

Check the status of the purge history request.

Return type

`int`

Parameters

yaml

[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

purge_id: str

The purge id from a purge history request.

Returns

response: `matrixctl.typehints.JsonDict`, optional

The response as dict, containing the status.

The purge-history command allows to purge historic events from the database.

Use this module to add the `purge-history` subcommand to `matrixctl`.

`matrixctl.addons.purge_history.timing.check_point_in_time(event_or_timestamp)`

Check the the type of the point in time and set the correct body.

Return type

`Optional[dict[str, str | int]]`

Parameters

event_or_timestamp

[`str`] The event_id or timestamp (UNIX epoch, in milliseconds).

Returns

request_body: `Dict [str, str or int]`

A dict, which can be merged with the `request_body` dict.

delroom

Use this module to add the `delroom` subcommand to `matrixctl`.

`matrixctl.addons.delroom.parser.subparser_delroom(subparsers)`

Create a subparser for the `matrixctl delroom` command.

Parameters

subparsers

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to add the delroom subcommand to matrixctl.

`matrixctl.addons.delroom.addon.addon(arg, yaml)`

Delete an empty room from the database.

Return type

`int`

Parameters

arg

`[argparse.Namespace]` The Namespace object of argparse's `parse_args()`

yaml

`[matrixctl.handlers.yaml.YAML]` The configuration file handler.

Returns

err_code

`[int]` Non-zero value indicates error code, or zero on success.

`matrixctl.addons.delroom.addon.handle_arguments(arg)`

Build the parameters used for the delroom request.

Return type

`dict[str, Any]`

Parameters

arg

`[argparse.Namespace]` The Namespace object of argparse's `parse_args()`

Returns

body

`[matrixctl.typehints.JsonDict]` The params.

`matrixctl.addons.delroom.addon.handle_status(yaml, delete_id)`

Handle the status of a delete room request.

Return type

`dict[str, Any]`

Parameters

yaml

`[matrixctl.handlers.yaml.YAML]` The configuration file handler.

delete_id: str

The delete id of a delete room request.

Returns

response: matrixctl.typehints.JsonDict, optional

The response as dict, containing the status.

upload

Use this module to add the upload subcommand to matrixctl.

`matrixctl.addons.upload.parser.subparser_upload(subparsers)`

Create a subparser for the matrixctl upload command.

Parameters

subparsers

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to add the upload subcommand to matrixctl.

`matrixctl.addons.upload.addon.addon(arg, yaml)`

Upload a file or image to the matrix instance.

Return type

`int`

Parameters

arg

[`argparse.Namespace`] The Namespace object of argparse's `parse_args()`.

yaml

[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns

err_code

[`int`] Non-zero value indicates error code, or zero on success.

server-notice

Use this module to add the serve-notice subcommand to matrixctl.

`matrixctl.addons.server_notice.parser.subparser_server_notice(subparsers)`

Create a subparser for the matrixctl server-notice command.

Parameters

subparsers

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to add the serve-notice subcommand to matrixctl.

`matrixctl.addons.server_notice.addon.addon(arg, yaml)`

Send a server notice to a matrix instance.

Return type

`int`

Parameters

arg
 [argparse.Namespace] The Namespace object of argparse's parse_args().

yaml
 [matrixctl.handlers.yaml.YAML] The configuration file handler.

Returns

err_code
 [int] Non-zero value indicates error code, or zero on success.

Notes

- It uses the synapse admin API.
- Note that “server notices” must be enabled in homeserver.yaml before this API can be used.

get-event

Use this module to get an event from the Database.

matrixctl.addons.get_event.parser.subparser_get_event(subparsers)

Create a subparser for the matrixctl get-event command.

Parameters

subparsers
 [argparse._SubParsersAction of typing.Any] The object which is returned by parser.add_subparsers().

Returns

None

Use this module to get an event from the Database.

matrixctl.addons.get_event.addon.addon(arg, yaml)

Get an Event from the Server.

It connects via paramiko to the server and runs the psql command provided by the synapse playbook to run a query on the Database.

Return type

int

Parameters

arg
 [argparse.Namespace] The Namespace object of argparse's parse_args()

yaml
 [matrixctl.handlers.yaml.YAML] The configuration file handler.

Returns

err_code
 [int] Non-zero value indicates error code, or zero on success.

get-events

Use this module to get an events of an user from the Database.

```
matrixctl.addons.get_events.parser.subparser_get_events(subparsers)
```

Create a subparser for the matrixctl get-event command.

Parameters

subparsers

[argparse._SubParsersAction of typing.Any] The object which is returned by parser.add_subparsers().

Returns

None

Use this module to get an event from the Database.

```
matrixctl.addons.get_events.addon.addon(arg, yaml)
```

Get Events from the Server.

It connects via paramiko to the server and runs the psql command provided by the synapse playbook to run a query on the Database.

Return type

int

Parameters

arg

[argparse.Namespace] The Namespace object of argparse's parse_args()

yaml

[matrixctl.handlers.yaml.YAML] The configuration file handler.

Returns

err_code

[int] Non-zero value indicates error code, or zero on success.

joinroom

Use this module to add the joinroom subcommand to matrixctl.

```
matrixctl.addons.joinroom.parser.subparser_deluser(subparsers)
```

Create a subparser for the matrixctl joinroom command.

Parameters

subparsers

[argparse._SubParsersAction of typing.Any] The object which is returned by parser.add_subparsers().

Returns

None

Use this module to add the joinroom subcommand to matrixctl.

`matrixctl.addons.joinroom.addon.addon(arg, yaml)`

Join a user to an room.

Return type

`int`

Parameters

arg

`[argparse.Namespace]` The Namespace object of argparse's `parse_args()`

yaml

`[matrixctl.handlers.yaml.YAML]` The configuration file handler.

Returns

err_code

`[int]` Non-zero value indicates error code, or zero on success.

Notes

- You can only modify the membership of local users.
- The the token of server administrator used to authenticate against the homeserver must be in the room and must have permission to invite users.

is-admin

Use this module to add the `is-admin` subcommand to `matrixctl`.

`matrixctl.addons.is_admin.parser.subparser_is_admin(subparsers)`

Create a subparser for the `matrixctl is-admin` command.

Parameters

subparsers

`[argparse._SubParsersAction of typing.Any]` The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to add the `is-admin` subcommand to `matrixctl`.

`matrixctl.addons.is_admin.addon.addon(arg, yaml)`

Delete a user is an admin.

Return type

`int`

Parameters

arg

`[argparse.Namespace]` The Namespace object of argparse's `parse_args()`

yaml

`[matrixctl.handlers.yaml.YAML]` The configuration file handler.

Returns

err_code

[int] Non-zero value indicates error code, or zero on success.

Notes

If a user does not exist it still will return "admin": false or No.

set-admin

Use this module to add the set-admin subcommand to matrixctl.

matrixctl.addons.set_admin.parser.**subparser_set_admin**(subparsers)

Create a subparser for the matrixctl set-admin command.

Parameters

subparsers

[argparse._SubParsersAction of typing.Any] The object which is returned by parser.add_subparsers().

Returns

None

Use this module to get the number of user in a Matrix room.

matrixctl.addons.set_admin.addon.**addon**(arg, yaml)

Change whether a user is an admin or not.

Return type

int

Parameters

arg

[argparse.Namespace] The Namespace object of argparse's parse_args()

yaml

[matrixctl.handlers.yaml.YAML] The configuration file handler.

Returns

err_code

[int] Non-zero value indicates error code, or zero on success.

make-room-admin

Add the make-room-admin subcommand to matrixctl.

matrixctl.addons.make_room_admin.parser.**subparser_make_room_admin**(subparsers)

Create a subparser for the matrixctl make-room-admin command.

Parameters

subparsers

[argparse._SubParsersAction of typing.Any] The object which is returned by parser.add_subparsers().

Returns

None

Use this module to grant a user room admin status.

Grant another user the highest power available to a local user who is in the room Matrix room.

```
matrixctl.addons.make_room_admin.addon.addon(arg, yaml)
```

Grant a user room admin status.

By default the server admin (the caller) is granted power, but another user can optionally be specified.

Return type

`int`

Parameters**arg**

[`argparse.Namespace`] The Namespace object of `argparse.parse_args()`

yaml

[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns**err_code**

[`int`] Non-zero value indicates error code, or zero on success.

get-event-context

Add the `get-event-context` subcommand to `matrixctl`.

```
matrixctl.addons.get_event_context.parser.subparser_get_event_context(subparsers)
```

Create a subparser for the `matrixctl get-event-context` command.

Parameters**subparsers**

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

None

Use this module to find the context of an event.

```
matrixctl.addons.get_event_context.addon.addon(arg, yaml)
```

Find the context of an event.

Return type

`int`

Parameters**arg**

[`argparse.Namespace`] The Namespace object of `argparse.parse_args()`

yaml

[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns**err_code**

[`int`] Non-zero value indicates error code, or zero on success.

purge-remote-media

Add the `purge-remote-media` subcommand to `matrixctl`.

```
matrixctl.addons.purge_remote_media.parser.subparser_purge_remote_media(subparsers)
```

Create a subparser for the `matrixctl purge-remote-media` command.

Parameters

subparsers

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to delete remote media.

```
matrixctl.addons.purge_remote_media.addon.addon(arg, yaml)
```

Remove remote media.

Return type

`int`

Parameters

arg

[`argparse.Namespace`] The `Namespace` object of `argparse`'s `parse_args()`

yaml

[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns

err_code

[`int`] Non-zero value indicates error code, or zero on success.

```
matrixctl.addons.purge_remote_media.addon.handle_timestamp(timestamp, force)
```

Ask or generate timestamp.

Return type

`int`

Parameters

timestamp

[`int`, optional] The timestamp

force

[`bool`] Don't ask any questions. All questions are answered with `True`.

Returns

timestamp

[`int`] The same timestamp but sanitized, or the timestamp of this exact time.

delete-local-media

Add the `delete-local-media` subcommand to `matrixctl`.

```
matrixctl.addons.delete_local_media.parser.subparser_delete_local_media(subparsers)
```

Create a subparser for the `matrixctl delete-local-media` command.

Parameters

subparsers

[`argparse._SubParsersAction` of `typing.Any`] The object which is returned by `parser.add_subparsers()`.

Returns

`None`

Use this module to delete local media.

```
matrixctl.addons.delete_local_media.addon.addon(arg, yaml)
```

Delete local media.

Return type

`int`

Parameters

arg

[`argparse.Namespace`] The `Namespace` object of `argparse`'s `parse_args()`

yaml

[`matrixctl.handlers.yaml.YAML`] The configuration file handler.

Returns

err_code

[`int`] Non-zero value indicates error code, or zero on success.

```
matrixctl.addons.delete_local_media.addon.handle_timestamp(timestamp, force)
```

Ask or generate timestamp.

Return type

`int`

Parameters

timestamp

[`int`, optional] The timestamp

force

[`bool`] Don't ask any questions. All questions are answered with `True`.

Returns

timestamp

[`int`] The same timestamp but sanitized, or the timestamp of this exact time.

Application

Application

Use MatrixCtl to control, manage, provision and deploy your homeserver.

Tests

Handlers

Handlers in MatrixCtl are used to handle the communication between the server and the *Addons* or to load config files.

Fixtures

YAML

Test the yaml handler.

```
tests.matrixctl.handlers.test_yaml.test_get_ansible_playbook(yaml)
```

Test ansible -> playbook.

Return type

None

```
tests.matrixctl.handlers.test_yaml.test_get_api_domain(yaml)
```

Test api -> domain.

Return type

None

```
tests.matrixctl.handlers.test_yaml.test_get_api_token(yaml)
```

Test api -> token.

Return type

None

```
tests.matrixctl.handlers.test_yaml.test_get_api_username(yaml)
```

Test api -> username.

Return type

None

```
tests.matrixctl.handlers.test_yaml.test_get_ssh_address(yaml)
```

Test ssh -> address.

Return type

None

```
tests.matrixctl.handlers.test_yaml.test_get_ssh_port(yaml)
```

Test ssh -> port.

Return type

None

tests.matrixctl.handlers.test_yaml.test_get_ssh_user(*yaml*)

Test ssh -> user.

Return type

None

tests.matrixctl.handlers.test_yaml.test_get_synapse_playbook(*yaml*)

Test synapse -> playbook.

Return type

None

tests.matrixctl.handlers.test_yaml.test_repr(*yaml*)

Test __repr__().

Return type

None

tests.matrixctl.handlers.test_yaml.test_str(*yaml*)

Test __str__().

Return type

None

Sanitizers

Test the sanitizers.

tests.matrixctl.test_sanitizers.test_sanitize_event_identifier_1()

Test valid identifier.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_event_identifier_2()

Test valid identifier with spaces around.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_event_identifier_3()

Test invalid identifier (without \$).

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_event_identifier_4()

Test invalid identifier (empty str).

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_event_identifier_5()

Test missing identifier (None).

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_event_identifier_6()

Test wrong type.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_invalid()

Test invalid message type as string.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_lower_case()

Test valid message type as lowercase string.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_avatar_1()

Test valid message type as MessageType.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_avatar_2()

Test valid message type as string.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_create_1()

Test valid message type as MessageType.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_create_2()

Test valid message type as string.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_join_rules_1()

Test valid message type as MessageType.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_join_rules_2()

Test valid message type as string.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_member_1()

Test valid message type as MessageType.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_member_2()

Test valid message type as string.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_message_1()

Test valid message type as MessageType.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_message_2()

Test valid message type as string.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_name_1()

Test valid message type as MessageType.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_name_2()

Test valid message type as string.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_pinned_events_1()

Test valid message type as MessageType.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_pinned_events_2()

Test valid message type as string.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_power_levels_1()

Test valid message type as MessageType.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_power_levels_2()

Test valid message type as string.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_redaction_1()

Test valid message type as MessageType.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_redaction_2()

Test valid message type as string.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_topic_1()

Test valid message type as MessageType.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_m_room_topic_2()

Test valid message type as string.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_none()

Test none case.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_message_type_with_spaces()

Test valid message type as string with spaces around.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_room_identifier_1()

Test valid identifier.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_room_identifier_2()

Test valid identifier with spaces around.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_room_identifier_3()

Test invalid identifier (without \$).

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_room_identifier_4()

Test invalid identifier (empty str).

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_room_identifier_5()

Test missing identifier (None).

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_room_identifier_6()

Test wrong type.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_user_identifier_1()

Test valid identifier.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_user_identifier_2()

Test valid identifier with spaces around.

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_user_identifier_3()

Test invalid identifier (without \$).

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_user_identifier_4()

Test invalid identifier (empty str).

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_user_identifier_5()

Test missing identifier (None).

Return type

None

tests.matrixctl.test_sanitizers.test_sanitize_user_identifier_6()

Test wrong type.

Return type

None

CHANGELOG

This is the changelog of MatrixCtl. You can find the issue tracker on [GitHub](#).

4.1 0.12.0-beta.2 (2023-03-24)

4.1.1 Bugfixes

- Fix M_UNKNOWN bug when using `purge-history`. (#165)
- Ensure MatrixCtl does not log the database password for synapse in debug mode (#460)

4.1.2 Removals & Deprecations

- Add a deprecation warning to the `adduser-jitsi` and `deluser-jitsi` commands. They are planned for removal in MatrixCtl v0.13.0. (#453)

4.1.3 Miscellaneous

- Add tests to the sanitizers (#315)
- Update pre-commit plugin flake8 to new repo url (#448)
- Fix rtd: Invalid configuration option: `python.version` (#450)

4.2 0.12.0-beta.1 (2021-12-02)

4.2.1 Behavior & Breaking Changes

- This release changes how MatrixCtl connects to the database. Therefore the configuration file must be changed. Please check the [documentation](#) for more information. (#313)

4.2.2 Features & Improvements

- `get-event` and `get-events` are not using `psycopg` instead of a `docker` command (#313)

4.2.3 Bugfixes

- Empty data for the table handler does no longer raise an error. (#309)
- The message Deleted Rooms: 0 in `purge-remote-media` has been corrected to Deleted Media Files: 0 (#311)

4.3 0.11.5 (2021-12-01)

No significant changes.

4.4 0.11.4 (2021-12-01)

4.4.1 Features & Improvements

- Update to *Delete Room API v2*. (#305)

4.4.2 Bugfixes

- Fix a bug introduced in `be411cf0c1a9413bf25ca1b72004150c032555c2`, after the last release because the `httpx` typehints are incorrect. (#307)

4.4.3 Miscellaneous

- Fix incorrect typehints in the API handler (#287)

4.5 0.11.3 (2021-11-16)

4.5.1 Features & Improvements

- Add `is-admin` addon to determine if a user is a server admin. (#252)
- Add `set-admin` addon to promote/demote users to/from homeserver admin (#254)
- Add `make-room-admin` addon (#265)
- Add `get-event-context` addon. (#267)
- Add `-f|--force` switch to `purge-history` to answer all questions with `yes`. (#271)
- Add `-e|--empty` switch argument to `rooms`, to only show empty rooms. (#273)
- Add `purge-remote-media` addon. (#275)
- `delroom` now uses the “Delete Room API” instead of the old “Purge Room API”, which is deprecated. (#277)

- Add delete-local-media addon. (#278)
- Debloat matrixctl --help (#281)

4.5.2 Miscellaneous

- Remove dependency single_source (#245)
- Generate the release body with a script while running the release action. (#284)

4.6 0.11.2 (2021-09-26)

4.6.1 Features & Improvements

- Add the joinroom (join a user to a room) addon to MatrixCtl. (#89)

4.6.2 Miscellaneous

- The API handler was refactored, which results roughly in a 10% speed increase for asynchronous requests. (#235)

4.7 0.11.1 (2021-09-25)

4.7.1 Features & Improvements

- paramiko now creates a known_hosts entry, if it does not exist. (#231)

4.7.2 Bugfixes

- Fix: adduser, deluser, delroom, server-notice, purge-history. (#233)

4.8 0.11.0 (2021-09-21)

4.8.1 Behavior & Breaking Changes

- The config file now is using the YAML format instead of the TOML format. (#174)
- Drop support for python 3.8. (#181)
- The password generation of MatrixCtl has been removed (#193)
- All servers in the config (config.yaml) file now need too be grouped below servers:. (#213)
- Remove --number and -n in the rooms addon and replace it with [limit]. (#217)

4.8.2 Features & Improvements

- Add `rust-synapse-compress-state` to the maintenance command. (#163)
- Multiple servers can be specified in the config file. (#174)
- Per-server maintenance task configuration. (#184)
- Optimized startup time by lazy importing addons by a factor of 10. Added a `addon_manager` which now manages imports of the addon (sub)parsers. (#187)
- Add `get_events` addon, which gets user-events from the DB. (#198)
- Add `reports` addon. (#200)
- Add `report` addon. (#202)
- Replace `tabulate` with the new `table` handler. (#206)
- With the `-j` or `--to-json` argument, the output of `reports`, `rooms`, `users` and `user` can be set to the JSON format. (#211)
- All API requests which need multiple requests to collect all data are now asynchronous. Add a optional `[limit]` argument to the `users` and `reports` addon. (#217)
- Add (one-pass) Jinja2 support for the configuration file. (#229)

4.8.3 Miscellaneous

- Add tests for the `yaml` handler. (#174)
- Commands or subcommands are now located in `matrixctl.addons` as packages and considered addons. Addons are splitted in `parser.py` and `addon.py`. It is now allowed to use multiple modules for one addon. (#187)
- More flexible `yaml` handler. (#213)

4.9 0.10.3 (2021-06-26)

4.9.1 Features & Improvements

- The docks have moved back to (<https://matrixctl.readthedocs.io/>). (#69)

4.9.2 Bugfixes

- Make MatrixCtl compatible with Python 3.8. (#146)

4.9.3 Improved Documentation

- Add Contribution Guidelines (#149)

4.9.4 Miscellaneous

- The event_id of the command get-event now gets sanitized. (#143)

4.10 0.10.2 (2021-06-24)

4.10.1 Features & Improvements

- Add start/restart switch to the deploy subcommand to start/restart the server right after the deployment. (#132)
- Added the new command get-event, which gets an event by event_id from the Database and prints it as JSON. (#139)

4.10.2 Miscellaneous

- Rewritten API handler. (#136)
- Fixed: Wrong version while developing in virtual environment. (#141)

4.11 0.10.1 (2021-06-17)

4.11.1 Features & Improvements

- Update type hinting according to PEP 585. (#123)

4.12 0.10.0 (2021-06-17)

4.12.1 Behavior & Breaking Changes

- Drop support for Python 3.8 for tests and typing. (#121)

4.12.2 Features & Improvements

- add purge-history to purge historic events from the DB (#86)
- Modules are using logger instead of logging. (#117)
- Use secure, temporary directory for ansible_runner's private data. (#119)

4.12.3 Miscellaneous

- Moved `mypy.ini` into `pyproject.toml`. (#113)
- Fix of false-positive CWE-798: Use of Hard-coded Credentials. (#115)
- Update pre-commit and dependencies. (#121)

4.13 0.9.0 (2021-04-23)

4.13.1 Behavior & Breaking Changes

- add shadow-banned (needs synapse v1.28 or greater) and displayname to the table output of ``matrixctl users`. (#30)

4.13.2 Features & Improvements

- Add the stop command to `matrixctl`, which stops all OCI containers. (#74)

4.13.3 Improved Documentation

- Fixed the commandline tool example in the docs. (#68)
- Removed the program name from every title of the changelog. We now only use the version number and the date. (#79)

4.14 0.8.6 (2021-04-17)

4.14.1 Features & Improvements

- The application now uses `__main__.py` instead of `application.py`. Developers are now able to use `python matrixctl` from the project root to start the application. (#60)
- Add tox as simple way to check the changelog, testbuild the docs, run pre-commit and run tests (#64)

4.14.2 Bugfixes

- Fix `TypeError` when enabling debug mode and using the API. (#45)

4.14.3 Miscellaneous

- Add `CHANGELOG.rst` to project root generated by `towncrier`. This is the first release using the new changelog generation procedure. If you want to see the previous changelog please check our [releases on GitHub](#). (#61)

4.15 0.8.5 (2021-02-24)

4.15.1 Bugfixes

- Add the new `serve-notice` feature.

4.16 0.8.4 (2021-02-24)

Note: This version of MatrixCtl has not been released.

4.17 0.8.3 (2021-02-24)

Note: This version of MatrixCtl has not been released.

4.18 0.8.2 (2021-02-24)

Note: This version of MatrixCtl has not been released.

4.18.1 Features & Improvements

- feature `upload` which makes it possible to upload files and images. It returns the `mxr://` uri.
- feature `server-notice`.

4.18.2 Miscellaneous

- Changed docs to classic python theme.

4.19 0.8.1 (2020-12-02)

4.19.1 Behavior & Breaking Changes

- The `update` command now uses config: `[SYNAPSE] -> Playbook` instead of `[SYNAPSE] -> Path`

4.19.2 Features & Improvements

- Add missing [SYNAPSE] (config file) documentation.

4.20 0.8.0 (2020-12-02)

4.20.1 Behavior & Breaking Changes

- The option to run multiple playbooks with matrixctl. The user should use `- import_playbook: /PathTo/matrix-docker-ansible-deploy/setup.yml` in an own playbook. (#20)(#21)

4.20.2 Features & Improvements

- The ansible handler now uses `ansible-runner` instead of `subprocess` (#20)(#21)
- The api handler now gives the user a hint, when the admin api is disabled.

4.21 0.7.0 (2020-09-25)

4.21.1 Behavior & Breaking Changes

- Removed the `--with-bots`, “bots” are now shown by default (#15)

4.21.2 Bugfixes

- Fixed the deploy control logic (#18)

4.22 0.6.3 (2020-09-17)

4.22.1 Features & Improvements

- With the help of two args it is possible to deploy the two playbooks independently: `--s/--synapse`: Only deploy the synapse playbook, `--a/--ansible`: Only deploy your own playbook.

4.23 0.6.2 (2020-09-16)

4.23.1 Bugfixes

- It is now possible to deploy, when only one of [ANSIBLE] or [SYNAPSE] are configured.

4.24 0.6.1 (2020-06-02)

4.24.1 Features & Improvements

- If the access-token has changed or is wrong, MatrixCtl now throws a specific error, which tells the user, what went wrong. (#12)
- Replace the assertions from the API handler with proper `TypeError`.

4.25 0.6.0 (2020-05-12)

4.25.1 Behavior & Breaking Changes

- Changed `users --no-bots` or `users -b` to `users --with-bots` or `users -b`
- Changed `users --guests` or `users -g` to `users --with-guests` or `users -g`

4.25.2 Features & Improvements

- `users --with-deactivated` or `users -d` (#2)

4.25.3 Bugfixes

- SSH handler logs an error if unable to connect (#7)

4.26 0.5.0 (2020-04-30)

Warning: Since the `synapse-janitor` is not safe to use anymore, please **do not** use the `maintenance` command for any MatrixCtl version below 0.5.0!

4.26.1 Behavior & Breaking Changes

- Fixed typo in the `maintenance` command.

4.26.2 Removals & Deprecations

- Removed `run-postgres-synapse-janitor` from `maintenance` because it may destroy the DB (#8)(#465 (spantaleev/matrix-docker-ansible-deploy))

4.27 0.4.0 (2020-04-22)

Warning: Since the `synapse-janitor` is not safe to use anymore, please **do not** use the `maintenance` command for any MatrixCtl version below 0.5.0!

4.27.1 Behavior & Breaking Changes

- `rooms` submodule: Changed argument `--order_by_size` to `--order-by-size`.

4.27.2 Features & Improvements

- Add the `version` command.
- Add the `delroom` command.
- Add more debug output to the API handler (`params`, `data`, `method` and `censored headers`)

4.28 0.3.2 (2020-04-21)

Warning: Since the `synapse-janitor` is not safe to use anymore, please **do not** use the `maintenance` command for any MatrixCtl version below 0.5.0!

4.28.1 Features & Improvements

- Add the `rooms` command.

4.29 0.3.1 (2020-04-21)

Warning: Since the `synapse-janitor` is not safe to use anymore, please **do not** use the `maintenance` command for any MatrixCtl version below 0.5.0!

Note: This version of MatrixCtl has not been released.

4.30 0.3.0 (2020-04-20)

Warning: Since the `synapse-janitor` is not safe to use anymore, please **do not** use the `maintenance` command for any MatrixCtl version below 0.5.0!

Note: No significant changes to the Project.

Project restructured.

4.31 0.2.2 (2020-04-13)

Warning: Since the `synapse-janitor` is not safe to use anymore, please **do not** use the `maintenance` command for any MatrixCtl version below 0.5.0!

4.31.1 Features & Improvements

- Added docs to the Project (`gh-pages` branch).

4.31.2 Bugfixes

- `matixctl adduser --ansible`. MatrixCtl was not able to create a user with the `--ansible` argument.

4.32 0.2.1 (2020-04-13)

Warning: Since the `synapse-janitor` is not safe to use anymore, please **do not** use the `maintenance` command for any MatrixCtl version below 0.5.0!

Note: This version of MatrixCtl has not been released.

4.33 0.2.0 (2020-04-12)

Warning: Since the `synapse-janitor` is not safe to use anymore, please **do not** use the `maintenance` command for any MatrixCtl version below 0.5.0!

4.33.1 Behavior & Breaking Changes

- The command `list-user` has been renamed to `users`.

4.33.2 Features & Improvements

- Add the command `user`.

4.34 0.1.4 (2020-04-10)

Warning: Since the `synapse-janitor` is not safe to use anymore, please **do not** use the `maintenance` command for any MatrixCtl version below 0.5.0!

4.34.1 Features & Improvements

- Add the command `start`.
- Add the command `restart` (alias for `start`).
- Add the command `check`.

4.35 0.1.3 (2020-04-10)

Warning: Since the `synapse-janitor` is not safe to use anymore, please **do not** use the `maintenance` command for any MatrixCtl version below 0.5.0!

4.35.1 Features & Improvements

- Add the command `adduser-jitsi`.
- Add the command `deluser-jitsi`.

4.36 0.1.2 (2020-04-07)

Warning: Since the `synapse-janitor` is not safe to use anymore, please **do not** use the `maintenance` command for any MatrixCtl version below 0.5.0!

First official release.

4.36.1 Features & Improvements

- Add the command `list-users`.

4.37 0.1.1 (2020-04-07)

Warning: Since the `synapse-janitor` is not safe to use anymore, please **do not** use the `maintenance` command for any MatrixCtl version below 0.5.0!

Note: No significant changes to the Project.

4.37.1 Trivial Changes

- Fixed GitHub Workflow.

4.38 0.1.0 (2020-04-07)

Warning: Since the `synapse-janitor` is not safe to use anymore, please **do not** use the `maintenance` command for any MatrixCtl version below 0.5.0!

Note: No significant changes to the Project.

Internal Release

BRANCHING MODEL

This repository uses the [git-flow](#) branching model by [Vincent Driessen](#). It has two branches with infinite lifetime:

- [master](#)
- [develop](#)

The master branch gets updated on every release. The develop branch is the merging branch.

COMMAND LINE TOOL

MatrixCtl as a pure commandline tool. You can use it as package, if you like, but breaking changes may introduced, even in a minor change.

```
usage: matrixctl [-h] [--version] [-d] [-s SERVER] [-c CONFIG] Command ...
```

MatrixCtl is a simple, but feature-rich tool to remotely control, manage, provision and [↪deploy](#) Matrix homeservers.

options:

```
-h, --help          show this help message and exit
--version          show program's version number and exit
-d, --debug        Enables debugging mode.
-s SERVER, --server SERVER
                   Select the server. (default: "default")
-c CONFIG, --config CONFIG
                   A path to an alternative config file.
```

Commands:

The following are commands, you can use to accomplish various tasks.

Command

adduser	Add users to the homeserver
adduser-jitsi	[DEPRECATED] Add users to a jitsi server
check	Checks the deployment with Ansible
delete-local-media	Delete cached (local) media that was last accessed before a specific point in time
delroom	Shutdown a room
deluser	Deactivate users
deluser-jitsi	[DEPRECATED] Delete jitsi users
deploy	Provision and deploy the Ansible playbook
get-event	Get an event from the database
get-event-context	Get the context of an event
get-events	Get events from the database
is-admin	Check, if a user is a homeserver administrator
joinroom	Join a user to a room
maintenance	Run maintenance tasks
make-room-admin	Grant a user the highest power level available to a local user in this room
purge-history	Purge historic events from the database
purge-remote-media	Purge cached, remote media

(continues on next page)

(continued from previous page)

report	Get a report event by report identifier
reports	Lists reported events
rooms	List rooms
server-notice	Send a server notice to a user
set-admin	Change whether a user is a homeserver admin or not
start	Starts all OCI containers
restart	Restarts all OCI containers (alias for start)
stop	Stop and disable all OCI containers
update	Updates the ansible playbook repository
upload	Upload a media file.
user	Get information about a specific user
users	Lists all users of the homeserver
version	Get the version information of the Synapse instance

Thank you for using MatrixCtl!

Check out the docs: <https://matrixctl.rtfid.io>

Report bugs to: <https://github.com/MichaelSasser/matrixctl/issues/new/choose>

CONFIGURATION FILE

To use this program you need to have this config file in `/etc/matrixctl/config` or in `~/.config/matrixctl/config`.

```
1 # Define your homeservers in "servers" here.
2 servers:
3     # Your default server. You can specify multiple servers here with arbitrary
4     # Names
5     default:
6
7     ansible:
8         # The absolute path to your playbook
9         playbook: /path/to/ansible/playbook
10
11     synapse:
12         # The absolute path to the synapse playbook.
13         # This is only used for updating the playbook.
14         playbook: /path/to/synapse/playbook
15
16     # If your matrix server is deployed, you may want to fill out the API section.
17     # It enables matrixctl to run more and faster commands. You can deploy and
18     # provision your Server without this section. You also can create a user with
19     # "matrixctl adduser --ansible YourUsername" and add your privileges after
20     # that.
21     api:
22         # Your domain should be something like "michaelsasser.org" without the
23         # "matrix." in the front. MatrixCtl will add that, if needed. An IP-Address
24         # is not enough.
25         domain: example.com
26
27         # The username your admin user
28         username: johndoe
29
30         # To use the API you need to have an administrator account. Enter your Token
31         # here. If you use the element client you will find it your user settings
32         # (click on your username on the upper left corner on your browser) in the
33         # "Help & About" tab. If you scroll down click next to "Access-Token:" on
34         # "<click to reveal>". It will be marked for you. Copy it in here.
35         token: "MyMatrixToken"
36
37     # In some cases, MatrixCtl does need to make many requests. To speed those
```

(continues on next page)

```
38 # requests a notch, you can set a concurrent_limit which is greater than
39 # one. This sets a limit to how many asynchronous workers can be spawned
40 # by MatrixCtl. If you set the number to high, MatrixCtl needs more time
41 # to spawn the workers, then a synchronous request would take.
42 concurrent_limit: 10
43
44 # Here you can add your SSH configuration.
45 ssh:
46   address: matrix.example.com
47
48   # The default port is 22
49   port: 22
50
51   # The default username is your current login name.
52   user: john
53
54 # Define your maintenance tasks
55 maintenance:
56   tasks:
57     - compress-state # Compress synapses state table
58     - vacuum        # VACUUM the synapse database (garbage-collection)
59
60 # Add connection parameters to the Database
61 # Synapse does only read (SELECT) information from the database.
62 # The user needs to be able to login to the synapse database
63 # and SELECT from the events and event_json tables.
64 database:
65   synapse_database: synapse # this is the playbooks default table name
66   synapse_user: matrixctl  # the username (role) for the database
67   synapse_password: "RolePassword"
68   tunnel: true           # true if an ssh tunnel should be used to connect
69
70 # The port that was used in the playbook (e.g.
71 # matrix_postgres_container_postgres_bind_port: 5432)
72 # or for your external database. For security reasons the port
73 # should be blocked by your firewall. If you enable the tunnel
74 # by setting tunnel: true, MatrixCtl activates a SSH tunnel.
75   port: 5432             # the remote port
76
77 # Another server.
78 foo:
79   # ...
```

SEMANTIC VERSIONING

After release “1.0.0” this repository will use [SemVer](#) for its release cycle.

Note: Before release “1.0.0” it uses “0.y.z” as recommended by SemVer. This means that breaking changes result in a version change at “y” position (e.g. “0.1.0” -> “0.2.0”). Non breaking changes result in a “z” change (e.g. “0.1.1” -> “0.1.2”).

INDICES AND TABLES

- genindex
- modindex
- search

**CHAPTER
TEN**

LICENSE

Copyright © 2020 Michael Sasser <Info@MichaelSasser.org>. Released under the GPLv3 license.

PYTHON MODULE INDEX

m

matrixctl, 70
matrixctl.addon_manager, 34
matrixctl.addons.adduser.addon, 44
matrixctl.addons.adduser.parser, 44
matrixctl.addons.adduser_jitsi.addon, 45
matrixctl.addons.adduser_jitsi.parser, 45
matrixctl.addons.check.addon, 47
matrixctl.addons.check.parser, 47
matrixctl.addons.delete_local_media.addon, 69
matrixctl.addons.delete_local_media.parser, 69
matrixctl.addons.delroom.addon, 61
matrixctl.addons.delroom.parser, 60
matrixctl.addons.deluser.addon, 45
matrixctl.addons.deluser.parser, 45
matrixctl.addons.deluser_jitsi.addon, 46
matrixctl.addons.deluser_jitsi.parser, 46
matrixctl.addons.deploy.addon, 47
matrixctl.addons.deploy.parser, 47
matrixctl.addons.get_event.addon, 63
matrixctl.addons.get_event.parser, 63
matrixctl.addons.get_event_context.addon, 67
matrixctl.addons.get_event_context.parser, 67
matrixctl.addons.get_events.addon, 64
matrixctl.addons.get_events.parser, 64
matrixctl.addons.is_admin.addon, 65
matrixctl.addons.is_admin.parser, 65
matrixctl.addons.joinroom.addon, 64
matrixctl.addons.joinroom.parser, 64
matrixctl.addons.maintenance.addon, 48
matrixctl.addons.maintenance.parser, 48
matrixctl.addons.make_room_admin.addon, 67
matrixctl.addons.make_room_admin.parser, 66
matrixctl.addons.purge_history.addon, 59
matrixctl.addons.purge_history.dialog, 59
matrixctl.addons.purge_history.handler, 60
matrixctl.addons.purge_history.parser, 59
matrixctl.addons.purge_history.timing, 60
matrixctl.addons.purge_remote_media.addon, 68
matrixctl.addons.purge_remote_media.parser, 68
matrixctl.addons.report.addon, 55
matrixctl.addons.report.parser, 55
matrixctl.addons.reports.addon, 56
matrixctl.addons.reports.parser, 56
matrixctl.addons.reports.to_table, 56
matrixctl.addons.rooms.addon, 57
matrixctl.addons.rooms.parser, 57
matrixctl.addons.rooms.to_table, 58
matrixctl.addons.server_notice.addon, 62
matrixctl.addons.server_notice.parser, 62
matrixctl.addons.set_admin.addon, 66
matrixctl.addons.set_admin.parser, 66
matrixctl.addons.start.addon, 49
matrixctl.addons.start.parser, 49
matrixctl.addons.stop.addon, 50
matrixctl.addons.stop.parser, 50
matrixctl.addons.update.addon, 51
matrixctl.addons.update.parser, 50
matrixctl.addons.upload.addon, 62
matrixctl.addons.upload.parser, 62
matrixctl.addons.user.addon, 51
matrixctl.addons.user.parser, 51
matrixctl.addons.user.to_table, 52
matrixctl.addons.users.addon, 54
matrixctl.addons.users.parser, 54
matrixctl.addons.users.to_table, 54
matrixctl.errors, 43
matrixctl.handlers.ansible, 27
matrixctl.handlers.api, 23
matrixctl.handlers.db, 32
matrixctl.handlers.ssh, 29
matrixctl.handlers.table, 29
matrixctl.handlers.vcs, 28
matrixctl.handlers.yaml, 20
matrixctl.package_version, 35
matrixctl.password_helpers, 41
matrixctl.print_helpers, 40
matrixctl.sanitizers, 36
matrixctl.structures, 42
matrixctl.typehints, 42

t

tests.matrixctl.handlers.test_yaml, 70

tests.matrixctl.test_sanitizers, 71

A

addon() (in module *matrixctl.addons.adduser.addon*), 44
 addon() (in module *matrixctl.addons.adduser_jitsi.addon*), 45
 addon() (in module *matrixctl.addons.check.addon*), 47
 addon() (in module *matrixctl.addons.delete_local_media.addon*), 69
 addon() (in module *matrixctl.addons.delroom.addon*), 61
 addon() (in module *matrixctl.addons.deluser.addon*), 45
 addon() (in module *matrixctl.addons.deluser_jitsi.addon*), 46
 addon() (in module *matrixctl.addons.deploy.addon*), 47
 addon() (in module *matrixctl.addons.get_event.addon*), 63
 addon() (in module *matrixctl.addons.get_event_context.addon*), 67
 addon() (in module *matrixctl.addons.get_events.addon*), 64
 addon() (in module *matrixctl.addons.is_admin.addon*), 65
 addon() (in module *matrixctl.addons.joinroom.addon*), 64
 addon() (in module *matrixctl.addons.maintenance.addon*), 48
 addon() (in module *matrixctl.addons.make_room_admin.addon*), 67
 addon() (in module *matrixctl.addons.purge_history.addon*), 59
 addon() (in module *matrixctl.addons.purge_remote_media.addon*), 68
 addon() (in module *matrixctl.addons.report.addon*), 55
 addon() (in module *matrixctl.addons.reports.addon*), 56
 addon() (in module *matrixctl.addons.rooms.addon*), 57
 addon() (in module *matrixctl.addons.server_notice.addon*), 62
 addon() (in module *matrixctl.addons.set_admin.addon*), 66
 addon() (in module *matrixctl.addons.start.addon*), 49
 addon() (in module *matrixctl.addons.stop.addon*), 50
 addon() (in module *matrixctl.addons.update.addon*), 51

addon() (in module *matrixctl.addons.upload.addon*), 62
 addon() (in module *matrixctl.addons.user.addon*), 51
 addon() (in module *matrixctl.addons.users.addon*), 54
 address (*matrixctl.handlers.ssh.SSH* attribute), 29
 address (*matrixctl.structures.ConfigServerSSH* attribute), 43
 ansible (*matrixctl.structures.ConfigServer* attribute), 42
 ansible_run() (in module *matrixctl.handlers.ansible*), 27
 api (*matrixctl.structures.ConfigServer* attribute), 42
 api_path (*matrixctl.handlers.api.RequestBuilder* attribute), 23
 api_version (*matrixctl.handlers.api.RequestBuilder* attribute), 24
 apply_defaults() (*matrixctl.handlers.yaml.YAML* static method), 21
 ask_password() (in module *matrixctl.password_helpers*), 41
 ask_question() (in module *matrixctl.password_helpers*), 41
 async_worker() (in module *matrixctl.handlers.api*), 25

B

BUGMSG (*matrixctl.errors.Error* attribute), 43

C

cells_to_str() (in module *matrixctl.handlers.table*), 29
 check_point_in_time() (in module *matrixctl.addons.purge_history.timing*), 60
 COMPRESS_STATE (*matrixctl.addons.maintenance.addon.Task* attribute), 48
 concurrent_limit (*matrixctl.handlers.api.RequestBuilder* attribute), 24
 concurrent_limit (*matrixctl.handlers.api.RequestStrategy* attribute), 24
 concurrent_limit (*matrixctl.structures.ConfigServerAPI* attribute), 42

- Config (class in *matrixctl.structures*), 42
 ConfigFileError, 43
 ConfigServer (class in *matrixctl.structures*), 42
 ConfigServerAnsible (class in *matrixctl.structures*), 43
 ConfigServerAPI (class in *matrixctl.structures*), 42
 ConfigServerMaintenance (class in *matrixctl.structures*), 43
 ConfigServerSSH (class in *matrixctl.structures*), 43
 ConfigServerSynapse (class in *matrixctl.structures*), 43
 content (*matrixctl.handlers.api.RequestBuilder* attribute), 24
 create_user() (in module *matrixctl.password_helpers*), 41
- ## D
- data (*matrixctl.handlers.api.RequestBuilder* attribute), 24
 database (*matrixctl.handlers.db.DBConnectionBuilder* attribute), 32
 datetime_last_pulled_commit (*matrixctl.handlers.vcs.VCS* property), 28
 db_connect() (in module *matrixctl.handlers.db*), 33
 DBConnectionBuilder (class in *matrixctl.handlers.db*), 32
 DEFAULT_PATHS (*matrixctl.handlers.yaml.YAML* attribute), 21
 dialog_input() (in module *matrixctl.addons.purge_history.dialog*), 59
 domain (*matrixctl.handlers.api.RequestBuilder* attribute), 24
 domain (*matrixctl.structures.ConfigServerAPI* attribute), 42
- ## E
- Error, 43
 exec_async_request() (in module *matrixctl.handlers.api*), 25
 ExitQWorker, 43
- ## F
- filter_empty_rooms() (in module *matrixctl.addons.rooms.addon*), 58
 find_newlines() (in module *matrixctl.handlers.table*), 30
 format_table_row() (in module *matrixctl.handlers.table*), 30
- ## G
- generate_output() (in module *matrixctl.addons.rooms.addon*), 58
 generate_user_tables() (in module *matrixctl.addons.user.to_table*), 52
 generate_worker_configs() (in module *matrixctl.handlers.api*), 25
 get() (*matrixctl.handlers.yaml.YAML* method), 21
 get_column_length() (in module *matrixctl.handlers.table*), 30
 get_paths_to_config() (*matrixctl.handlers.yaml.YAML* static method), 21
 get_server_config() (*matrixctl.handlers.yaml.YAML* method), 22
 get_version() (in module *matrixctl.package_version*), 35
 group_async_results() (in module *matrixctl.handlers.api*), 26
- ## H
- handle_arguments() (in module *matrixctl.addons.delroom.addon*), 61
 handle_newlines() (in module *matrixctl.handlers.table*), 31
 handle_purge_status() (in module *matrixctl.addons.purge_history.handler*), 60
 handle_status() (in module *matrixctl.addons.delroom.addon*), 61
 handle_timestamp() (in module *matrixctl.addons.delete_local_media.addon*), 69
 handle_timestamp() (in module *matrixctl.addons.purge_remote_media.addon*), 68
 headers (*matrixctl.handlers.api.RequestBuilder* attribute), 24
 headers_with_auth (*matrixctl.handlers.api.RequestBuilder* property), 24
 host (*matrixctl.handlers.db.DBConnectionBuilder* attribute), 32
 human_readable_bool() (in module *matrixctl.print_helpers*), 40
- ## I
- import_addons_from() (in module *matrixctl.addon_manager*), 34
 InternalResponseError, 43
 iterations (*matrixctl.handlers.api.RequestStrategy* attribute), 24
- ## J
- JINJA_PREDEFINED (*matrixctl.handlers.yaml.YAML* attribute), 21
 JinjaUndefined (class in *matrixctl.handlers.yaml*), 20
 json (*matrixctl.handlers.api.RequestBuilder* attribute), 24

L

limit (*matrixctl.handlers.api.RequestStrategy* attribute), 24

log() (*matrixctl.handlers.vcs.VCS* method), 28

M

M_ROOM_AVATAR (*matrixctl.sanitizers.MessageType* attribute), 36

M_ROOM_CREATE (*matrixctl.sanitizers.MessageType* attribute), 37

M_ROOM_JOIN_RULES (*matrixctl.sanitizers.MessageType* attribute), 37

M_ROOM_MEMBER (*matrixctl.sanitizers.MessageType* attribute), 37

M_ROOM_MESSAGE (*matrixctl.sanitizers.MessageType* attribute), 37

M_ROOM_NAME (*matrixctl.sanitizers.MessageType* attribute), 37

M_ROOM_PINNED_EVENTS (*matrixctl.sanitizers.MessageType* attribute), 37

M_ROOM_POWER_LEVELS (*matrixctl.sanitizers.MessageType* attribute), 37

M_ROOM_REDACTION (*matrixctl.sanitizers.MessageType* attribute), 37

M_ROOM_TOPIC (*matrixctl.sanitizers.MessageType* attribute), 37

maintenance (*matrixctl.structures.ConfigServer* attribute), 42

make_human_readable() (in module *matrixctl.addons.user.to_table*), 53

matrixctl module, 70

matrixctl.addon_manager module, 34

matrixctl.addons.adduser.addon module, 44

matrixctl.addons.adduser.parser module, 44

matrixctl.addons.adduser_jitsi.addon module, 45

matrixctl.addons.adduser_jitsi.parser module, 45

matrixctl.addons.check.addon module, 47

matrixctl.addons.check.parser module, 47

matrixctl.addons.delete_local_media.addon module, 69

matrixctl.addons.delete_local_media.parser module, 69

matrixctl.addons.delroom.addon module, 61

matrixctl.addons.delroom.parser module, 60

matrixctl.addons.deluser.addon module, 45

matrixctl.addons.deluser.parser module, 45

matrixctl.addons.deluser_jitsi.addon module, 46

matrixctl.addons.deluser_jitsi.parser module, 46

matrixctl.addons.deploy.addon module, 47

matrixctl.addons.deploy.parser module, 47

matrixctl.addons.get_event.addon module, 63

matrixctl.addons.get_event.parser module, 63

matrixctl.addons.get_event_context.addon module, 67

matrixctl.addons.get_event_context.parser module, 67

matrixctl.addons.get_events.addon module, 64

matrixctl.addons.get_events.parser module, 64

matrixctl.addons.is_admin.addon module, 65

matrixctl.addons.is_admin.parser module, 65

matrixctl.addons.joinroom.addon module, 64

matrixctl.addons.joinroom.parser module, 64

matrixctl.addons.maintenance.addon module, 48

matrixctl.addons.maintenance.parser module, 48

matrixctl.addons.make_room_admin.addon module, 67

matrixctl.addons.make_room_admin.parser module, 66

matrixctl.addons.purge_history.addon module, 59

matrixctl.addons.purge_history.dialog module, 59

matrixctl.addons.purge_history.handler module, 60

matrixctl.addons.purge_history.parser module, 59

matrixctl.addons.purge_history.timing module, 60

matrixctl.addons.purge_remote_media.addon module, 68

matrixctl.addons.purge_remote_media.parser module, 68

matrixctl.addons.report.addon
 module, 55

matrixctl.addons.report.parser
 module, 55

matrixctl.addons.reports.addon
 module, 56

matrixctl.addons.reports.parser
 module, 56

matrixctl.addons.reports.to_table
 module, 56

matrixctl.addons.rooms.addon
 module, 57

matrixctl.addons.rooms.parser
 module, 57

matrixctl.addons.rooms.to_table
 module, 58

matrixctl.addons.server_notice.addon
 module, 62

matrixctl.addons.server_notice.parser
 module, 62

matrixctl.addons.set_admin.addon
 module, 66

matrixctl.addons.set_admin.parser
 module, 66

matrixctl.addons.start.addon
 module, 49

matrixctl.addons.start.parser
 module, 49

matrixctl.addons.stop.addon
 module, 50

matrixctl.addons.stop.parser
 module, 50

matrixctl.addons.update.addon
 module, 51

matrixctl.addons.update.parser
 module, 50

matrixctl.addons.upload.addon
 module, 62

matrixctl.addons.upload.parser
 module, 62

matrixctl.addons.user.addon
 module, 51

matrixctl.addons.user.parser
 module, 51

matrixctl.addons.user.to_table
 module, 52

matrixctl.addons.users.addon
 module, 54

matrixctl.addons.users.parser
 module, 54

matrixctl.addons.users.to_table
 module, 54

matrixctl.errors
 module, 43

matrixctl.handlers.ansible
 module, 27

matrixctl.handlers.api
 module, 23

matrixctl.handlers.db
 module, 32

matrixctl.handlers.ssh
 module, 29

matrixctl.handlers.table
 module, 29

matrixctl.handlers.vcs
 module, 28

matrixctl.handlers.yaml
 module, 20

matrixctl.package_version
 module, 35

matrixctl.password_helpers
 module, 41

matrixctl.print_helpers
 module, 40

matrixctl.sanitizers
 module, 36

matrixctl.structures
 module, 42

matrixctl.typehints
 module, 42

MessageType (*class in matrixctl.sanitizers*), 36

method (*matrixctl.handlers.api.RequestBuilder* attribute), 24

module

- matrixctl, 70
- matrixctl.addon_manager, 34
- matrixctl.addons.adduser.addon, 44
- matrixctl.addons.adduser.parser, 44
- matrixctl.addons.adduser_jitsi.addon, 45
- matrixctl.addons.adduser_jitsi.parser, 45
- matrixctl.addons.check.addon, 47
- matrixctl.addons.check.parser, 47
- matrixctl.addons.delete_local_media.addon, 69
- matrixctl.addons.delete_local_media.parser, 69
- matrixctl.addons.delroom.addon, 61
- matrixctl.addons.delroom.parser, 60
- matrixctl.addons.deluser.addon, 45
- matrixctl.addons.deluser.parser, 45
- matrixctl.addons.deluser_jitsi.addon, 46
- matrixctl.addons.deluser_jitsi.parser, 46
- matrixctl.addons.deploy.addon, 47
- matrixctl.addons.deploy.parser, 47
- matrixctl.addons.get_event.addon, 63
- matrixctl.addons.get_event.parser, 63
- matrixctl.addons.get_event_context.addon, 67

- matrixctl.addons.get_event_context.parser, 67
 - matrixctl.addons.get_events.addon, 64
 - matrixctl.addons.get_events.parser, 64
 - matrixctl.addons.is_admin.addon, 65
 - matrixctl.addons.is_admin.parser, 65
 - matrixctl.addons.joinroom.addon, 64
 - matrixctl.addons.joinroom.parser, 64
 - matrixctl.addons.maintenance.addon, 48
 - matrixctl.addons.maintenance.parser, 48
 - matrixctl.addons.make_room_admin.addon, 67
 - matrixctl.addons.make_room_admin.parser, 66
 - matrixctl.addons.purge_history.addon, 59
 - matrixctl.addons.purge_history.dialog, 59
 - matrixctl.addons.purge_history.handler, 60
 - matrixctl.addons.purge_history.parser, 59
 - matrixctl.addons.purge_history.timing, 60
 - matrixctl.addons.purge_remote_media.addon, 68
 - matrixctl.addons.purge_remote_media.parser, 68
 - matrixctl.addons.report.addon, 55
 - matrixctl.addons.report.parser, 55
 - matrixctl.addons.reports.addon, 56
 - matrixctl.addons.reports.parser, 56
 - matrixctl.addons.reports.to_table, 56
 - matrixctl.addons.rooms.addon, 57
 - matrixctl.addons.rooms.parser, 57
 - matrixctl.addons.rooms.to_table, 58
 - matrixctl.addons.server_notice.addon, 62
 - matrixctl.addons.server_notice.parser, 62
 - matrixctl.addons.set_admin.addon, 66
 - matrixctl.addons.set_admin.parser, 66
 - matrixctl.addons.start.addon, 49
 - matrixctl.addons.start.parser, 49
 - matrixctl.addons.stop.addon, 50
 - matrixctl.addons.stop.parser, 50
 - matrixctl.addons.update.addon, 51
 - matrixctl.addons.update.parser, 50
 - matrixctl.addons.upload.addon, 62
 - matrixctl.addons.upload.parser, 62
 - matrixctl.addons.user.addon, 51
 - matrixctl.addons.user.parser, 51
 - matrixctl.addons.user.to_table, 52
 - matrixctl.addons.users.addon, 54
 - matrixctl.addons.users.parser, 54
 - matrixctl.addons.users.to_table, 54
 - matrixctl.errors, 43
 - matrixctl.handlers.ansible, 27
 - matrixctl.handlers.api, 23
 - matrixctl.handlers.db, 32
 - matrixctl.handlers.ssh, 29
 - matrixctl.handlers.table, 29
 - matrixctl.handlers.vcs, 28
 - matrixctl.handlers.yaml, 20
 - matrixctl.package_version, 35
 - matrixctl.password_helpers, 41
 - matrixctl.print_helpers, 40
 - matrixctl.sanitizers, 36
 - matrixctl.structures, 42
 - matrixctl.typehints, 42
 - tests.matrixctl.handlers.test_yaml, 70
 - tests.matrixctl.test_sanitizers, 71
- ## N
- newlines_in_row() (in module *matrixctl.handlers.table*), 31
- ## O
- offset (*matrixctl.handlers.api.RequestStrategy* attribute), 24
- ## P
- params (*matrixctl.handlers.api.RequestBuilder* attribute), 24
 - password (*matrixctl.handlers.db.DBConnectionBuilder* attribute), 32
 - path (*matrixctl.handlers.api.RequestBuilder* attribute), 24
 - playbook (*matrixctl.structures.ConfigServerAnsible* attribute), 43
 - playbook (*matrixctl.structures.ConfigServerSynapse* attribute), 43
 - port (*matrixctl.handlers.db.DBConnectionBuilder* attribute), 32
 - port (*matrixctl.handlers.ssh.SSH* attribute), 29
 - port (*matrixctl.structures.ConfigServerSSH* attribute), 43
 - preplan_request_strategy() (in module *matrixctl.handlers.api*), 26
 - print_tasks() (in module *matrixctl.addons.maintenance.addon*), 49
 - pull() (*matrixctl.handlers.vcs.VCS* method), 28
- ## R
- read_from_file() (*matrixctl.handlers.yaml.YAML* static method), 22
 - request() (in module *matrixctl.handlers.api*), 27
 - RequestBuilder (class in *matrixctl.handlers.api*), 23
 - RequestStrategy (class in *matrixctl.handlers.api*), 24
 - run_cmd() (*matrixctl.handlers.ssh.SSH* method), 29
- ## S
- sanitize() (in module *matrixctl.sanitizers*), 37

- sanitize_event_identifiler() (in module *matrixctl.sanitizers*), 37
 sanitize_message_type() (in module *matrixctl.sanitizers*), 38
 sanitize_room_identifiler() (in module *matrixctl.sanitizers*), 39
 sanitize_user_identifiler() (in module *matrixctl.sanitizers*), 39
 scheme (matrixctl.handlers.api.RequestBuilder attribute), 24
 scheme (matrixctl.handlers.db.DBConnectionBuilder attribute), 32
 secrets_filter() (in module *matrixctl.handlers.yaml*), 23
 server (matrixctl.handlers.yaml.YAML attribute), 23
 server (matrixctl.structures.Config attribute), 42
 servers (matrixctl.structures.Config attribute), 42
 setup() (in module *matrixctl.addon_manager*), 34
 SSH (class in *matrixctl.handlers.ssh*), 29
 ssh (matrixctl.structures.ConfigServer attribute), 42
 ssh_tunnel() (in module *matrixctl.handlers.db*), 33
 SSHResponse (class in *matrixctl.handlers.ssh*), 29
 stderr (matrixctl.handlers.ssh.SSHResponse attribute), 29
 stdin (matrixctl.handlers.ssh.SSHResponse attribute), 29
 stdout (matrixctl.handlers.ssh.SSHResponse attribute), 29
 step_size (matrixctl.handlers.api.RequestStrategy attribute), 25
 subdomain (matrixctl.handlers.api.RequestBuilder attribute), 24
 subparser() (in module *matrixctl.addon_manager*), 35
 subparser_adduser() (in module *matrixctl.addons.adduser.parser*), 44
 subparser_adduser_jitsi() (in module *matrixctl.addons.adduser_jitsi.parser*), 45
 subparser_check() (in module *matrixctl.addons.check.parser*), 47
 subparser_delete_local_media() (in module *matrixctl.addons.delete_local_media.parser*), 69
 subparser_delroom() (in module *matrixctl.addons.delroom.parser*), 60
 subparser_deluser() (in module *matrixctl.addons.deluser.parser*), 45
 subparser_deluser() (in module *matrixctl.addons.joinroom.parser*), 64
 subparser_deluser_jitsi() (in module *matrixctl.addons.deluser_jitsi.parser*), 46
 subparser_deploy() (in module *matrixctl.addons.deploy.parser*), 47
 subparser_get_event() (in module *matrixctl.addons.get_event.parser*), 63
 subparser_get_event_context() (in module *matrixctl.addons.get_event_context.parser*), 67
 subparser_get_events() (in module *matrixctl.addons.get_events.parser*), 64
 subparser_is_admin() (in module *matrixctl.addons.is_admin.parser*), 65
 subparser_maintenance() (in module *matrixctl.addons.maintenance.parser*), 48
 subparser_make_room_admin() (in module *matrixctl.addons.make_room_admin.parser*), 66
 subparser_purge_history() (in module *matrixctl.addons.purge_history.parser*), 59
 subparser_purge_remote_media() (in module *matrixctl.addons.purge_remote_media.parser*), 68
 subparser_report() (in module *matrixctl.addons.report.parser*), 55
 subparser_reports() (in module *matrixctl.addons.reports.parser*), 56
 subparser_restart() (in module *matrixctl.addons.start.parser*), 49
 subparser_rooms() (in module *matrixctl.addons.rooms.parser*), 57
 subparser_server_notice() (in module *matrixctl.addons.server_notice.parser*), 62
 subparser_set_admin() (in module *matrixctl.addons.set_admin.parser*), 66
 subparser_start() (in module *matrixctl.addons.start.parser*), 49
 subparser_stop() (in module *matrixctl.addons.stop.parser*), 50
 subparser_update() (in module *matrixctl.addons.update.parser*), 50
 subparser_upload() (in module *matrixctl.addons.upload.parser*), 62
 subparser_user() (in module *matrixctl.addons.user.parser*), 51
 subparser_users() (in module *matrixctl.addons.users.parser*), 54
 success_codes (matrixctl.handlers.api.RequestBuilder attribute), 24
 synapse (matrixctl.structures.ConfigServer attribute), 42
- ## T
- table() (in module *matrixctl.handlers.table*), 31
 Task (class in *matrixctl.addons.maintenance.addon*), 48
 tasks (matrixctl.structures.ConfigServerMaintenance attribute), 43
 test_get_ansible_playbook() (in module *tests.matrixctl.handlers.test_yaml*), 70
 test_get_api_domain() (in module *tests.matrixctl.handlers.test_yaml*), 70
 test_get_api_token() (in module *tests.matrixctl.handlers.test_yaml*), 70
 test_get_api_username() (in module *tests.matrixctl.handlers.test_yaml*), 70

test_get_ssh_address()	(in module tests.matrixctl.handlers.test_yaml), 70	test_sanitize_message_type_m_room_power_levels_1()	(in module tests.matrixctl.test_sanitizers), 73
test_get_ssh_port()	(in module tests.matrixctl.handlers.test_yaml), 70	test_sanitize_message_type_m_room_power_levels_2()	(in module tests.matrixctl.test_sanitizers), 73
test_get_ssh_user()	(in module tests.matrixctl.handlers.test_yaml), 70	test_sanitize_message_type_m_room_redaction_1()	(in module tests.matrixctl.test_sanitizers), 73
test_get_synapse_playbook()	(in module tests.matrixctl.handlers.test_yaml), 71	test_sanitize_message_type_m_room_redaction_2()	(in module tests.matrixctl.test_sanitizers), 73
test_repr()	(in module tests.matrixctl.handlers.test_yaml), 71	test_sanitize_message_type_m_room_topic_1()	(in module tests.matrixctl.test_sanitizers), 74
test_sanitize_event_identifiser_1()	(in module tests.matrixctl.test_sanitizers), 71	test_sanitize_message_type_m_room_topic_2()	(in module tests.matrixctl.test_sanitizers), 74
test_sanitize_event_identifiser_2()	(in module tests.matrixctl.test_sanitizers), 71	test_sanitize_message_type_none()	(in module tests.matrixctl.test_sanitizers), 74
test_sanitize_event_identifiser_3()	(in module tests.matrixctl.test_sanitizers), 71	test_sanitize_message_type_with_spaces()	(in module tests.matrixctl.test_sanitizers), 74
test_sanitize_event_identifiser_4()	(in module tests.matrixctl.test_sanitizers), 71	test_sanitize_room_identifiser_1()	(in module tests.matrixctl.test_sanitizers), 74
test_sanitize_event_identifiser_5()	(in module tests.matrixctl.test_sanitizers), 71	test_sanitize_room_identifiser_2()	(in module tests.matrixctl.test_sanitizers), 74
test_sanitize_event_identifiser_6()	(in module tests.matrixctl.test_sanitizers), 71	test_sanitize_room_identifiser_3()	(in module tests.matrixctl.test_sanitizers), 74
test_sanitize_message_type_invalid()	(in module tests.matrixctl.test_sanitizers), 72	test_sanitize_room_identifiser_4()	(in module tests.matrixctl.test_sanitizers), 74
test_sanitize_message_type_lower_case()	(in module tests.matrixctl.test_sanitizers), 72	test_sanitize_room_identifiser_5()	(in module tests.matrixctl.test_sanitizers), 74
test_sanitize_message_type_m_room_avatar_1()	(in module tests.matrixctl.test_sanitizers), 72	test_sanitize_room_identifiser_6()	(in module tests.matrixctl.test_sanitizers), 74
test_sanitize_message_type_m_room_avatar_2()	(in module tests.matrixctl.test_sanitizers), 72	test_sanitize_user_identifiser_1()	(in module tests.matrixctl.test_sanitizers), 75
test_sanitize_message_type_m_room_create_1()	(in module tests.matrixctl.test_sanitizers), 72	test_sanitize_user_identifiser_2()	(in module tests.matrixctl.test_sanitizers), 75
test_sanitize_message_type_m_room_create_2()	(in module tests.matrixctl.test_sanitizers), 72	test_sanitize_user_identifiser_3()	(in module tests.matrixctl.test_sanitizers), 75
test_sanitize_message_type_m_room_join_rules_1()	(in module tests.matrixctl.test_sanitizers), 72	test_sanitize_user_identifiser_4()	(in module tests.matrixctl.test_sanitizers), 75
test_sanitize_message_type_m_room_join_rules_2()	(in module tests.matrixctl.test_sanitizers), 72	test_sanitize_user_identifiser_5()	(in module tests.matrixctl.test_sanitizers), 75
test_sanitize_message_type_m_room_member_1()	(in module tests.matrixctl.test_sanitizers), 72	test_sanitize_user_identifiser_6()	(in module tests.matrixctl.test_sanitizers), 75
test_sanitize_message_type_m_room_member_2()	(in module tests.matrixctl.test_sanitizers), 72	test_str()	(in module tests.matrixctl.handlers.test_yaml), 71
test_sanitize_message_type_m_room_message_1()	(in module tests.matrixctl.test_sanitizers), 73	tests.matrixctl.handlers.test_yaml	module, 70
test_sanitize_message_type_m_room_message_2()	(in module tests.matrixctl.test_sanitizers), 73	tests.matrixctl.test_sanitizers	module, 71
test_sanitize_message_type_m_room_name_1()	(in module tests.matrixctl.test_sanitizers), 73	timeout	(matrixctl.handlers.api.RequestBuilder attribute), 24
test_sanitize_message_type_m_room_name_2()	(in module tests.matrixctl.test_sanitizers), 73	timeout	(matrixctl.handlers.db.DBConnectionBuilder attribute), 32
test_sanitize_message_type_m_room_pinned_events_1()	(in module tests.matrixctl.test_sanitizers), 73	time_stamp_to_dt()	(in module matrixctl.print_helpers), 40
test_sanitize_message_type_m_room_pinned_events_2()	(in module tests.matrixctl.test_sanitizers), 73	to_table()	(in module matrixctl.addons.reports.to_table), 56

`to_table()` (in module `matrixctl.addons.rooms.to_table`), 58

`to_table()` (in module `matrixctl.addons.user.to_table`), 53

`to_table()` (in module `matrixctl.addons.users.to_table`), 54

`token` (`matrixctl.handlers.api.RequestBuilder` attribute), 24

`token` (`matrixctl.structures.ConfigServerAPI` attribute), 42

`transpose_newlines_to_rows()` (in module `matrixctl.handlers.table`), 32

`tree_printer()` (in module `matrixctl.handlers.yaml`), 23

U

`user` (`matrixctl.handlers.ssh.SSH` attribute), 29

`user` (`matrixctl.structures.ConfigServerSSH` attribute), 43

`username` (`matrixctl.handlers.db.DBConnectionBuilder` attribute), 33

`username` (`matrixctl.structures.ConfigServerAPI` attribute), 42

V

`VACUUM` (`matrixctl.addons.maintenance.addon.Task` attribute), 48

`VCS` (class in `matrixctl.handlers.vcs`), 28

Y

`YAML` (class in `matrixctl.handlers.yaml`), 20