
synadm
Release 0.31

Johannes Tiefenbacher

Oct 12, 2021

CONTENTS:

- 1 Command Line Reference** **1**
- 1.1 Main Command 1
- 1.2 Config 2
- 1.3 User 3
- 1.4 Room 8
- 1.5 Media 12
- 1.6 History 14
- 1.7 Group 15
- 1.8 Matrix 16
- 1.9 Regtok 18

- 2 Python Package Documentation** **21**
- 2.1 Synadm CLI Package 21
- 2.2 Synadm Package 22

- 3 Indices and tables** **29**

- Python Module Index** **31**

- Index** **33**

COMMAND LINE REFERENCE

1.1 Main Command

1.1.1 synadm

the Matrix-Synapse admin CLI

```
synadm [OPTIONS] COMMAND [ARGS]...
```

Options

- v, --verbose**
enable INFO (-v) or DEBUG (-vv) logging on console
- batch, --no-batch**
enable batch behavior (no interactive prompts)
- o, --output <output>**
override default output format.
Options yaml | json | human | pprint | y | j | h | p |
- c, --config-file <config_file>**
configuration file path
Default ~/.config/synadm.yaml

Commands

- config**
Modify synadm's configuration.
- group**
Manage groups (communities)
- history**
purge historic events from Synapse database
- matrix**
Execute Matrix API calls.
- media**
Manage local and remote media

regtok

Manage registration tokens

room

Manipulate rooms and room membership

user

list, add, modify, deactivate/erase users,...

version

Get the synapse server version

1.2 Config

1.2.1 synadm config

Modify synadm's configuration. Configuration details are generally always asked interactively. Command line options override the suggested defaults in the prompts.

```
synadm config [OPTIONS]
```

Options

-u, --user <user>

admin user for accessing the Synapse admin API's

-t, --token <token>

admin user's access token for the Synapse admin API's

-b, --base-url <base_url>

the base URL Synapse is running on. Typically this is <https://localhost:8008> or <https://localhost:8448>. If Synapse is configured to expose its admin API's to the outside world it might as well be something like this: <https://example.org:8448>

-p, --admin-path <admin_path>

the path Synapse provides its admin API's, usually the default is alright for most installations.

-m, --matrix-path <matrix_path>

the path Synapse provides the Matrix client API's, usually the default is alright for most installations.

-w, --timeout <timeout>

the timeout for http queries to admin API's or Matrix Client API's. The default is 7 seconds.

-o, --output <output>

how should synadm display data by default? 'human' gives a tabular or list view depending on the fetched data. This mode needs your terminal to be quite wide! 'json' displays exactly as the API responded. 'pprint' shows nicely formatted json. 'yaml' is the currently recommended output format. It doesn't need as much terminal width as 'human' does. Note that the default output format can always be overridden by using global switch -o (eg 'synadm -o pprint user list').

Options yaml | json | human | pprint

1.3 User

1.3.1 synadm user

list, add, modify, deactivate/erase users, reset passwords

```
synadm user [OPTIONS] COMMAND [ARGS]...
```

deactivate

deactivate or gdpr-erase a user. Provide matrix user ID (@user:server) as argument. It removes active access tokens, resets the password, and deletes third-party IDs (to prevent the user requesting a password reset).

```
synadm user deactivate [OPTIONS] USER_ID
```

Options

-e, --gdpr-erase

marks the user as GDPR-erased. This means messages sent by the user will still be visible by anyone that was in the room when these messages were sent, but hidden from users joining the room afterwards.

Default False

Arguments

USER_ID

Required argument

details

view details of a user account.

```
synadm user details [OPTIONS] USER_ID
```

Arguments

USER_ID

Required argument

list

list and search for users

```
synadm user list [OPTIONS]
```

Options

- f, --from** <from_>
offset user listing by given number. This option is also used for pagination.
Default 0
- l, --limit** <limit>
limit user listing to given number
Default 100
- g, --guests, -G, --no-guests**
show guest users.
- d, --deactivated**
also show deactivated/erased users
Default False
- n, --name** <name>
search users by name - filters to only return users with user ID localparts or displaynames that contain this value (localpart is the left part before the colon of the matrix ID (@user:server))
- i, --user-id** <user_id>
search users by ID - filters to only return users with Matrix IDs (@user:server) that contain this value

login

Get an access token for a given user.

Useful for when admins wish to do actions on behalf of a user.

If no `-expire*` option is given, a default token expiry time of exactly 1 day (24h) is used. If it's desired that the token never expires, use `-expire-never`

This API does not generate a new device for the user, and so will not appear in their `/devices` list, and in general the target user should not be able to tell they have been logged in as.

To expire the token before the expiry date/time is reached, call the standard `/logout` API with the token. Note: The token will expire if the admin user calls `/logout/all` from any of their devices, but the token will not expire if the target user does the same.

```
synadm user login [OPTIONS] USER_ID
```

Options

- d, --expire-days** <expire_days>
expire token after this number of days.
- expire** <expire>
expire token after this point in time. Eg. '2021-01-01', see above for available date/time formats.
- expire-ts** <expire_ts>
expire token after this point in time giving a unix timestamp in ms.
- expire-never**
never expire token.
Default False

Arguments

USER_ID

Required argument

media

list all local media uploaded by a user. Provide matrix user ID (@user:server) as argument.

Gets a list of all local media that a specific user_id has created. By default, the response is ordered by descending creation date and ascending media ID. The newest media is on top. You can change the order with options `-order-by` and `-reverse`.

Caution. The database only has indexes on the columns `media_id`, `user_id` and `created_ts`. This means that if a different sort order is used (`upload_name`, `last_access_ts`, `media_length`, `media_type`, `quarantined_by` or `safe_from_quarantine`), this can cause a large load on the database, especially for large environments

```
synadm user media [OPTIONS] USER_ID
```

Options

-f, --from <from>

offset media listing by given number. This option is also used for pagination.

Default 0

-l, --limit <limit>

limit media listing to given number

Default 100

-s, --sort <sort>

The method by which to sort the returned list of media. If the ordered field has duplicates, the second order is always by ascending `media_id`, which guarantees a stable ordering.

Options `media_id` | `upload_name` | `created_ts` | `last_access_ts` | `media_length` | `media_type` | `quarantined_by` | `safe_from_quarantine`

-r, --reverse

Direction of media order. If set it will reverse the sort order of `-order-by` method.

Arguments

USER_ID

Required argument

membership

list all rooms a user is member of. Provide matrix user ID (@user:server) as argument.

```
synadm user membership [OPTIONS] USER_ID
```

Options

--aliases, --ids

Display rooms as canonical aliases or room ID's. [default: aliases]

Arguments

USER_ID

Required argument

modify

create or modify a local user. Provide matrix user ID (@user:server) as argument.

```
synadm user modify [OPTIONS] USER_ID
```

Options

-p, --password-prompt

set password interactively.

-P, --password <password>

set password on command line.

-n, --display-name <display_name>

set display name. defaults to the value of user_id

-t, --threepid <threepid>

add a third-party identifier. This can be an email address or a phone number. Threepids are used for several things: For use when logging in, as an alternative to the user id. In the case of email, as an alternative contact to help with account recovery, as well as to receive notifications of missed messages. Format: medium value (eg. `-threepid email <user@example.org>`). This option can also be stated multiple times, i.e. a user can have multiple threepids configured.

-v, --avatar-url <avatar_url>

set avatar URL. Must be a MXC URI (https://matrix.org/docs/spec/client_server/r0.6.0#matrix-content-mxc-uris)

-a, --admin, -u, --no-admin

grant user admin permission. Eg user is allowed to use the admin API

--activate

re-activate user.

--deactivate

deactivate user. Use with caution! Deactivating a user removes their active access tokens, resets their password, kicks them out of all rooms and deletes third-party identifiers (to prevent the user requesting a password reset). See also "user deactivate" command.

Arguments

USER_ID

Required argument

password

change a user's password. To prevent the user from being logged out of all sessions use option -n

```
synadm user password [OPTIONS] USER_ID
```

Options

-n, --no-logout

don't log user out of all sessions on all devices.

-p, --password <password>

new password

Arguments

USER_ID

Required argument

search

a simplified shortcut to 'synadm user list -d -g -n <search-term>' (Searches for users by name/matrix-ID, including deactivated users as well as guest users). Also it executes a case-insensitive search compared to the original command.

```
synadm user search [OPTIONS] SEARCH_TERM
```

Options

-f, --from <from_>

offset user listing by given number. This option is also used for pagination.

Default 0

-l, --limit <limit>

maximum amount of users to return.

Default 100

Arguments

SEARCH_TERM

Required argument

whois

Return information about the active sessions for a specific user

```
synadm user whois [OPTIONS] USER_ID
```

Arguments

USER_ID

Required argument

1.4 Room

1.4.1 synadm room

Manipulate rooms and room membership

```
synadm room [OPTIONS] COMMAND [ARGS]...
```

delete

delete and possibly purge a room

```
synadm room delete [OPTIONS] ROOM_ID
```

Options

-u, --new-room-user-id <new_room_user_id>

If set, a new room will be created with this user ID as the creator and admin, and all users in the old room will be moved into that room. If not set, no new room will be created and the users will just be removed from the old room. The user ID must be on the local server, but does not necessarily have to belong to a registered user.

-n, --room-name <room_name>

A string representing the name of the room that new users will be invited to. Defaults to “Content Violation Notification”

-m, --message <message>

A string containing the first message that will be sent as new_room_user_id in the new room. Ideally this will clearly convey why the original room was shut down. Defaults to “Sharing illegal content on this server is not permitted and rooms in violation will be blocked.”

-b, --block

If set, this room will be added to a blocking list, preventing future attempts to join the room

Default False

--no-purge

Prevent removing of all traces of the room from your database.

Default False

Arguments**ROOM_ID**

Required argument

details

get room details

```
synadm room details [OPTIONS] ROOM_ID
```

Arguments**ROOM_ID**

Required argument

join

join a room

```
synadm room join [OPTIONS] ROOM_ID_OR_ALIAS USER_ID
```

Arguments**ROOM_ID_OR_ALIAS**

Required argument

USER_ID

Required argument

list

list and search for rooms

```
synadm room list [OPTIONS]
```

Options

- f, --from** <from_>
offset room listing by given number. This option is also used for pagination.
Default 0
- l, --limit** <limit>
Maximum amount of rooms to return.
Default 100
- n, --name** <name>
Filter rooms by their room name. Search term can be contained in any part of the room name)
- s, --sort** <sort>
The method in which to sort the returned list of rooms.
Options name | canonical_alias | joined_members | joined_local_members | version | creator | encryption | federatable | public | join_rules | guest_access | history_visibility | state_events
- r, --reverse**
Direction of room order. If set it will reverse the sort order of `-order-by` method.

make-admin

grant a user room admin permission. If the user is not in the room, and it is not publicly joinable, then invite the user.

```
synadm room make-admin [OPTIONS] ROOM_ID
```

Options

- u, --user-id** <user_id>
By default the server admin (the caller) is granted power, but another user can optionally be specified.

Arguments

ROOM_ID
Required argument

members

list current room members

```
synadm room members [OPTIONS] ROOM_ID
```

Arguments

ROOM_ID

Required argument

resolve

lookup room ID from alias or vice versa

```
synadm room resolve [OPTIONS] ROOM_ID_OR_ALIAS
```

Options

-r, --reverse

Fetch all room aliases corresponding to a given room ID, instead of the other way round.

Default False

Arguments

ROOM_ID_OR_ALIAS

Required argument

search

a simplified shortcut to 'synadm room list -n <search-term>'. Also it executes a case-insensitive search compared to the original command.

```
synadm room search [OPTIONS] SEARCH_TERM
```

Options

-f, --from <from_>

offset room listing by given number. This option is also used for pagination.

Default 0

-l, --limit <limit>

Maximum amount of rooms to return.

Default 100

Arguments

SEARCH_TERM

Required argument

1.5 Media

1.5.1 synadm media

Manage local and remote media

```
synadm media [OPTIONS] COMMAND [ARGS]...
```

delete

delete media by ID, size or age

```
synadm media delete [OPTIONS]
```

Options

- i, --media-id** <media_id>
the media with this specific media ID will be deleted.
- d, --before-days** <before_days>
delete all media that was last accessed before this number of days ago.
- b, --before** <before>
delete all media that was last accessed before this date/time. Eg. '2021-01-01', see above for available date/time formats.
- t, --before-ts** <before_ts>
delete all media that was last accessed before this unix timestamp in ms.
- size, --kib** <size>
delete all media that is larger than this size in KiB (1 KiB = 1024 bytes).
- delete-profiles, --all**
also delete files that are still used in image data (e.g user profile, room avatar). If set, these files will be deleted too. Not valid when a specific media is being deleted (`-media-id`)

list

list local media by room or user

```
synadm media list [OPTIONS]
```


Options

- r, --room-id** <room_id>
list all media in room with this room ID ('!abcdefg').
- u, --user-id** <user_id>
list all media uploaded by user with this matrix ID (@user:server).
- f, --from** <from_>
offset media listing by given number. This option is also used for pagination but only supported together with `-user-id`.
- Default** 0
- l, --limit** <limit>
limit media listing to given number. This option is only supported together with `-user-id`.
- Default** 100
- s, --sort** <sort>
The method by which to sort the returned list of media. If the ordered field has duplicates, the second order is always by ascending `media_id`, which guarantees a stable ordering. This option is only supported together with `-user-id`.
- Options** `media_id` | `upload_name` | `created_ts` | `last_access_ts` | `media_length` | `media_type` | `quarantined_by` | `safe_from_quarantine`
- R, --reverse**
Direction of media order. If set it will reverse the sort order of `-order-by` method. This option is only supported together with `-user-id`.

protect

protect specific media from being quarantined

```
synadm media protect [OPTIONS] MEDIA_ID
```

Arguments

MEDIA_ID
Required argument

purge

Purge old cached remote media

```
synadm media purge [OPTIONS]
```

Options

- d, --before-days** <before_days>
Purge all media that was last accessed before this number of days ago.
- b, --before** <before>
Purge all media that was last accessed before this date/time. Eg. '2021-01-01', see above for available date/time formats.
- t, --before-ts** <before_ts>
Purge all media that was last accessed before this unix timestamp in ms.

quarantine

quarantine media in rooms, by users or by media ID

```
synadm media quarantine [OPTIONS]
```

Options

- i, --media-id** <media_id>
the media with this specific media ID will be quarantined.
- r, --room-id** <room_id>
all media in room with this room ID (!abcdefg) will be quarantined.
- u, --user-id** <user_id>
all media uploaded by user with this matrix ID (@user:server) will be quarantined.
- s, --server-name** <server_name>
the server name of the media, mandatory when `-media-id` is used and `_remote_` media should be processed. For locally stored media this option can be omitted.

1.6 History

1.6.1 synadm history

purge historic events from Synapse database

```
synadm history [OPTIONS] COMMAND [ARGS]...
```

purge

purge room events before a point in time or before an event ID.

The purge history API allows server admins to purge historic events from their database, reclaiming disk space.

Depending on the amount of history being purged a call to the API may take several minutes or longer. During this period users will not be able to paginate further back in the room from the point being purged from.

Note that Synapse requires at least one message in each room, so it will never delete the last message in a room.

By default, events sent by local users are not deleted, as they may represent the only copies of this content in existence. (Events sent by remote users are deleted.)

Room state data (such as joins, leaves, topic) is always preserved.

The API starts the purge running, and returns immediately with a JSON body with a purge id. Use 'synadm history purge-status <purge id>' to poll for updates on the running purge.

```
synadm history purge [OPTIONS] ROOM_ID
```

Options

- i, --before-event-id** <before_event_id>
purge all history before this event ID. Event ID's contain \$, which is interpreted by the shell. Make sure it is surrounded by single quotes ('id').
- d, --before-days** <before_days>
purge all history before this number of days ago.
- before** <before>
purge all history before this point in time. Eg. '2021-01-01', see above for available date/time formats.
- before-ts** <before_ts>
purge all history before this point in time giving a unix timestamp in ms.
- delete-local**
this option overrides the default behavior and forces removal of events sent by local users.

Arguments

ROOM_ID
Required argument

purge-status

view status of a recent history purge. Provide purge ID as argument.

The status will be one of active, complete, or failed.

```
synadm history purge-status [OPTIONS] PURGE_ID
```

Arguments

PURGE_ID
Required argument

1.7 Group

1.7.1 synadm group

Manage groups (communities)

```
synadm group [OPTIONS] COMMAND [ARGS]...
```

delete

delete a local group (community)

```
synadm group delete [OPTIONS] GROUP_ID
```

Arguments

GROUP_ID

Required argument

1.8 Matrix

1.8.1 synadm matrix

Execute Matrix API calls.

```
synadm matrix [OPTIONS] COMMAND [ARGS]...
```

login

Login to Matrix via username/password and receive an access token.

The response also contains a newly generated device ID and further information about user and homeserver.

Each successful login will show up in the user's devices list marked with a display name of 'synadm matrix login command'.

```
synadm matrix login [OPTIONS] USER_ID
```

Options

-p, --password <password>

The Matrix user's password. If missing, an interactive password prompt is shown.

Arguments

USER_ID

Required argument

raw

Execute a raw request to the Matrix API.

The endpoint argument is the part of the URL *after* the configured base URL and Matrix path (see *synadm config*). A simple get request would e.g like this: *synadm matrix raw client/versions*

Use either `-token` or `-prompt` to provide a user's token and execute Matrix commands on their behalf. Respect the privacy of others! Be responsible!

The precedence rules for token reading are: 1. Interactive input using `-prompt`; 2. Set on CLI via `-token` string; 3. Read from environment variable `$MTOKEN`; 4. Preconfigured admin token set in *synadm*'s config file.

```
synadm matrix raw [OPTIONS] ENDPOINT
```

Options

-m, --method <method>

The HTTP method used for the request.

Default get

Options get | post | put | delete

-d, --data <data>

The JSON string sent in the body of post, put and delete requests - provided as a string. Make sure to escape it from shell interpretation by using single quotes. E.g `'{"key1": "value1", "key2": 123}'`

Default {}

-f, --data-file <data_file>

Read JSON data from file. To read from stdin use `-` as the filename argument.

-t, --token <token>

Token used for Matrix authentication instead of the configured admin user's token. If `-token` (and `-prompt`) option is missing, the token is read from environment variable `$MTOKEN` instead. To make sure a user's token does not show up in system logs, don't provide it on the shell directly but set `$MTOKEN` with shell command *read MTOKEN*.

-p, --prompt

Prompt for the token used for Matrix authentication. This option always overrides `$MTOKEN`.

Default False

Arguments

ENDPOINT

Required argument

Environment variables

MTOKEN

Provide a default for `--token`

1.9 Regtok

1.9.1 synadm regtok

Manage registration tokens

```
synadm regtok [OPTIONS] COMMAND [ARGS]...
```

delete

Delete a registration token

```
synadm regtok delete [OPTIONS] TOKEN
```

Arguments

TOKEN

Required argument

details

View details of the given token

```
synadm regtok details [OPTIONS] TOKEN
```

Options

-d, --datetime, -t, --timestamp

Display expiry time in a human readable format, or as a unix timestamp in milliseconds. [default: datetime].

Arguments

TOKEN

Required argument

list

List registration tokens

```
synadm regtok list [OPTIONS]
```

Options

-v, --valid, -V, --invalid

List only valid/invalid tokens.

-d, --datetime, -t, --timestamp

Display expiry time in a human readable format, or as a unix timestamp in milliseconds. [default: datetime].

new

Create a new registration token

```
synadm regtok new [OPTIONS]
```

Options

-n, --token <token>

Set the registration token. The default is a random value generated by the server.

-l, --length <length>

The length of the randomly generated token if the token is not specified.

Default 16

-u, --uses-allowed <uses_allowed>

The number of times the token can be used to complete a registration before it becomes invalid. [default: unlimited uses]

-t, --expiry-ts <expiry_ts>

The latest time the registration token is valid. Given as the number of milliseconds since 1970-01-01 00:00:00 UTC.

[default: no expiry]

-e, --expire-at <expire_at>

The latest time the registration token is valid. See above for available date/time formats. [default: no expiry]

update

Update a registration token

```
synadm regtok update [OPTIONS] TOKEN
```

Options

- u, --uses-allowed** <uses_allowed>
The number of times the token can be used to complete a registration before it becomes invalid. Use -1 for an unlimited number of uses. [default: unchanged]
- t, --expiry-ts** <expiry_ts>
The latest time the registration token is valid. Given as the number of milliseconds since 1970-01-01 00:00:00 UTC. Use -1 for no expiration. [default: unchanged]
- e, --expire-at** <expire_at>
The latest time the registration token is valid. See above for available date/time formats. [default: unchanged]

Arguments

- TOKEN**
Required argument

PYTHON PACKAGE DOCUMENTATION

2.1 Synadm CLI Package

The `synadm.cli` package contains all the command line frontend code (using the Python Click module), see the sections for each subcommand for help on usage and have a look into the files inside the `synadm/cli` directory to see how the CLI frontend is constructed. The main command is defined in `__init__.py` and each subcommand lives in its own module (e.g `user.py`, `room.py`, ...)

Click command definition code is not very well suited for autogenerated Python package documentations (in this case using Sphinx autodoc) but one thing in this package that is worth mentioning and generating documentation for, is the `synadm.cli.APIHelper` class which is the connection between the CLI code and the backend code in `synadm.api`:

CLI base functions and settings

```
class synadm.cli.APIHelper (config_path, verbose, batch, output_format_cli)
```

Bases: object

API client enriched with CLI-level functions, used as a proxy to the client object

```
FORMATTERS = {'human': <function humanize>, 'json': <function dumps>, 'pprint': <fu
```

```
CONFIG = {'admin_path': '/_synapse/admin', 'base_url': 'http://localhost:8008', 'mat
```

```
init_logger (verbose)
```

Log both to console (defaults to WARNING) and file (DEBUG)

```
_set_formatter (_output_format)
```

```
load ()
```

Load the configuration and initialize the client

```
write_config (config)
```

Write a new version of the configuration to file

```
output (data)
```

Output data object using the configured formatter

2.2 Synadm Package

The `synadm` package contains the `synadm.api` module which is responsible for accessing the Synapse admin API endpoints and the regular Matrix API endpoints. Also some helper methods, e.g to calculate time-related things are to be found in this module:

Synapse admin API and regular Matrix API clients

Most API calls defined in this module respect the API's defaults and only pass what's necessary in the request body.

A fully qualified Matrix user ID looks like this: `@user:server`, where `server` often is a domain name only, e.g `@user@example.org`

See https://github.com/matrix-org/synapse/tree/master/docs/admin_api for documentation of the Synapse admin APIs and the Matrix spec at <https://matrix.org/docs/spec/#matrix-apis>.

class `synadm.api.ApiRequest` (*log, user, token, base_url, path, timeout, debug*)

Bases: `object`

Basic API request handling and helper utilities

This is subclassed by `SynapseAdmin` and `Matrix`

query (*method, urlpart, params=None, data=None, token=None*)

Generic wrapper around requests methods

Handles requests methods, logging and exceptions

Parameters

- **urlpart** (*string*) – the path to the API endpoint, excluding `self.base_url` and `self.path` (the part after `proto://fqdn:port/path`).
- **params** (*dict, optional*) – URL parameters (`?param1&paarm2`). Defaults to `None`.
- **data** (*dict, optional*) – request body used in POST, PUT, DELETE requests. Defaults to `None`.

Returns

usually a JSON string containing the response of the API; responses that are not 200(OK) (usually error messages returned by the API) will also be returned as JSON strings; on exceptions the error type and description are logged and `None` is returned.

Return type `string` or `None`

`_timestamp_from_days_ago` (*days*)

Get a unix timestamp in ms from days ago

Parameters **days** (*int*) – number of days

Returns a unix timestamp in milliseconds (ms)

Return type `int`

`_timestamp_from_days_ahead` (*days*)

Get a unix timestamp in ms for the given number of days ahead

Parameters **days** (*int*) – number of days

Returns a unix timestamp in milliseconds (ms)

Return type `int`

`_timestamp_from_datetime` (*datetime*)

Get a unix timestamp in ms from a datetime object

Parameters `_datetime` (*datetime object*) – an object built by `datetime.datetime`

Returns a unix timestamp in milliseconds (ms)

Return type int

`_datetime_from_timestamp` (*timestamp*)

Get a datetime object from a unix timestamp in ms

Parameters `timestamp` (*int*) – a unix timestamp in milliseconds (ms)

Returns an object built by `datetime.datetime`

Return type datetime object

class `synadm.api.Matrix` (*log, user, token, base_url, matrix_path, timeout, debug*)

Bases: `synadm.api.ApiRequest`

Matrix API client

Inheritance:

ApiRequest (object): parent class containing general properties and methods for requesting REST API's

`user_login` (*user_id, password*)

Login as a Matrix user and retrieve an access token

Parameters

- `user_id` (*string*) – a fully qualified Matrix user ID
- `password` (*string*) – the Matrix user's password

Returns

JSON string containing a token suitable to access the Matrix API on the user's behalf, a `device_id` and some more details on Matrix server and user.

Return type string

`room_get_id` (*room_alias*)

Get the room ID for a given room alias

Parameters `room_alias` (*string*) – A Matrix room alias (`#name:example.org`)

Returns

A dict containing the room ID for the alias. If `room_id` is missing in the response we return the whole response as it might contain Synapse's error message.

Return type string, dict or None

`room_get_aliases` (*room_id*)

Get a list of room aliases for a given room ID

Parameters `room_id` (*string*) – A Matrix room ID (`!abc123:example.org`)

Returns

A dict containing a list of room aliases, Synapse's error message or None on exceptions.

Return type dict or None

`raw_request` (*endpoint, method, data, token=None*)

server_name ()

Fetch the local server name

using the API for retrieving public server keys: https://matrix.org/docs/spec/server_server/r0.1.4#retrieving-server-keys

Returns

the local Matrix server name or None if the query method could not fetch it for any reason.

Return type string

class `synadm.api.SynapseAdmin` (*log, user, token, base_url, admin_path, timeout, debug*)

Bases: `synadm.api.ApiRequest`

Synapse admin API client

Inheritance:

ApiRequest (object): parent class containing general properties and methods for requesting REST API's

user_list (*_from, _limit, _guests, _deactivated, _name, _user_id*)

List and search users

Parameters

- **_from** (*int*) – offsets user list by this number, used for pagination
- **_limit** (*int*) – maximum number of users returned, used for pagination
- **_guests** (*bool*) – enable/disable fetching of guest users
- **_deactivated** (*bool*) – enable/disable fetching of deactivated users
- **_name** (*string*) – user name localpart to search for, see Synapse admin API docs for details
- **_user_id** (*string*) – fully qualified Matrix user ID to search for

Returns JSON string containing the found users

Return type string

user_membership (*user_id, return_aliases, matrix_api*)

Get a list of rooms the given user is member of

Parameters

- **user_id** (*string*) – Fully qualified Matrix user ID
- **room_aliases** (*bool*) – Return human readable room aliases instead of room ID's if applicable.
- **matrix_api** (*object*) – An initialized Matrix object needs to be passes as we need some Matrix API functionality here.

Returns

JSON string containing the admin API's response or None if an exception occurred. See Synapse admin API docs for details.

Return type string

user_deactivate (*user_id, gdpr_erase*)

Delete a given user

Parameters

- **user_id** (*string*) – fully qualified Matrix user ID
- **gdpr_erase** (*bool*) – enable/disable gdpr-erasing the user, see Synapse admin API docs for details.

Returns

JSON string containing the admin API's response or None if an exception occurred. See Synapse admin API docs for details.

Return type string

user_password (*user_id, password, no_logout*)

Set the user password, and log the user out if requested

Parameters

- **user_id** (*string*) – fully qualified Matrix user ID
- **password** (*string*) – new password that should be set
- **no_logout** (*bool*) – the API defaults to logging out the user after password reset via the admin API, this option can be used to disable this behaviour.

Returns JSON string containing the admin API's response or None if an exception occurred. See Synapse admin API docs for details.

Return type string

user_details (*user_id*)

Get information about a given user

Note that the admin API docs describe this function as “Query User Account”.

Parameters **user_id** (*string*) – fully qualified Matrix user ID

Returns

JSON string containing the admin API's response or None if an exception occurred. See Synapse admin API docs for details.

Return type string

user_login (*user_id, expire_days, expire, _expire_ts*)

Get an access token that can be used to authenticate as that user.

If one of the args `expire_days`, `expire` or `_expire_ts` is set, the `valid_until_ms` field will be sent to the API endpoint. If this is not the case the default of the API would be used. At the time of writing, this would be that tokens never expire.

Note: If this method is called by the CLI frontend code (`synadm.cli.user.user_login_cmd`), a default expiry date of 1 day (24h) is passed.

Parameters

- **user_id** (*string*) – fully qualified Matrix user ID
- **expire_days** (*int*) – token should expire after this number of days
- **expire** (*datetime*) – token should expire after this date/time - a datetime object (e.g. as generated by `Click.DateTime()`)
- **_expire_ts** (*int*) – token should expire after this date/time - a unix timestamp in ms.

Returns JSON string containing the admin API's response or None if an exception occurred. See Synapse admin API docs for details.

Return type string

user_modify (*user_id, password, display_name, threepid, avatar_url, admin, deactivation*)

Create or update information about a given user

Threepid should be passed as a tuple in a tuple

user_whois (*user_id*)

Return information about the active sessions for a specific user

room_join (*room_id_or_alias, user_id*)

Allow an administrator to join an user account with a given *user_id* to a room with a given *room_id_or_alias*

room_list (*_from, limit, name, order_by, reverse*)

List and search rooms

room_details (*room_id*)

Get details about a room

room_members (*room_id*)

Get a list of room members

room_delete (*room_id, new_room_user_id, room_name, message, block, no_purge*)

Delete a room and purge it if requested

room_make_admin (*room_id, user_id*)

Grant a user room admin permission. If the user is not in the room, and it is not publicly joinable, then invite the user.

room_media_list (*room_id*)

Get a list of known media in an (unencrypted) room.

media_quarantine (*server_name, media_id*)

Quarantine a single piece of local or remote media

room_media_quarantine (*room_id*)

Quarantine all local and remote media in a room

user_media_quarantine (*user_id*)

Quarantine all local and remote media of a user

user_media (*user_id, _from, limit, order_by, reverse*)

Get a user's uploaded media

media_delete (*server_name, media_id*)

Delete a specific (local) *media_id*

media_delete_by_date_or_size (*server_name, before_days, before, _before_ts, _size_gt, delete_profiles*)

Delete local media by date and/or size FIXME and/or?

media_protect (*media_id*)

Protect a single piece of local or remote media

from being quarantined

purge_media_cache (*before_days, before, _before_ts*)

Purge old cached remote media

version ()

Get the server version

group_delete (*group_id*)

Delete a local group (community)

purge_history (*room_id, before_event_id, before_days, before, _before_ts, delete_local*)

Purge room history

purge_history_status (*purge_id*)

Get status of a recent history purge

The status will be one of active, complete, or failed.

regtok_list (*valid, readable_expiry*)

List registration tokens

Parameters

- **valid** (*bool*) – List only valid (if True) or invalid (if False) tokens. Default is to list all tokens regardless of validity.
- **readable_expiry** (*bool*) – If True, replace the `expiry_time` field with a human readable datetime. If False, `expiry_time` will be a unix timestamp.

Returns

JSON string containing the admin API's response or None if an exception occurred. See Synapse admin API docs for details.

Return type string

regtok_details (*token, readable_expiry*)

Get details about the given registration token

Parameters

- **token** (*string*) – The registration token in question
- **readable_expiry** (*bool*) – If True, replace the `expiry_time` field with a human readable datetime. If False, `expiry_time` will be a unix timestamp.

Returns

JSON string containing the admin API's response or None if an exception occurred. See Synapse admin API docs for details.

Return type string

regtok_new (*token, length, uses_allowed, expiry_ts, expire_at*)

Create a new registration token

Parameters

- **token** (*string*) – Registration token to create. Default is randomly generated by the server.
- **length** (*int*) – The length of the token to generate if the token is not provided.
- **uses_allowed** (*int*) – The number of times the token can be used to complete a registration before it becomes invalid.
- **expiry_ts** (*int*) – The latest time the registration token is valid. Given as the number of milliseconds since 1970-01-01 00:00:00 UTC.
- **expire_at** (*click.DateTime*) – The latest time the registration token is valid.

Returns

JSON string containing the admin API's response or None if an exception occurred. See Synapse admin API docs for details.

Return type string

regtok_update (*token*, *uses_allowed*, *expiry_ts*, *expire_at*)

Update a registration token

Parameters

- **token** (*string*) – Registration token to update.
- **uses_allowed** (*int*) – The number of times the token can be used to complete a registration before it becomes invalid.
- **expiry_ts** (*int*) – The latest time the registration token is valid. Given as the number of milliseconds since 1970-01-01 00:00:00 UTC. -1 indicates no expiry.
- **expire_at** (*click.DateTime*) – The latest time the registration token is valid.

Returns

JSON string containing the admin API's response or None if an exception occurred. See Synapse admin API docs for details.

Return type string

regtok_delete (*token*)

Delete a registration token

Parameters **token** (*string*) – The registration token to delete

Returns

JSON string containing the admin API's response or None if an exception occurred. See Synapse admin API docs for details.

Return type string

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

`synadm.api`, 22

`synadm.cli`, 21

Symbols

`_datetime_from_timestamp()`
(synadm.api.ApiRequest method), 23
`_set_formatter()` *(synadm.cli.APIHelper method)*,
 21
`_timestamp_from_datetime()`
(synadm.api.ApiRequest method), 22
`_timestamp_from_days_ago()`
(synadm.api.ApiRequest method), 22
`_timestamp_from_days_ahead()`
(synadm.api.ApiRequest method), 22
 -G
 synadm-user-list command line
 option, 4
 -P
 synadm-user-modify command line
 option, 6
 -R
 synadm-media-list command line
 option, 13
 -V
 synadm-regtok-list command line
 option, 19
 --activate
 synadm-user-modify command line
 option, 6
 --admin
 synadm-user-modify command line
 option, 6
 --admin-path <admin_path>
 synadm-config command line option, 2
 --aliases
 synadm-user-membership command
 line option, 6
 --all
 synadm-media-delete command line
 option, 12
 --avatar-url <avatar_url>
 synadm-user-modify command line
 option, 6
 --base-url <base_url>
 synadm-config command line option, 2
 --batch
 synadm command line option, 1
 --before <before>
 synadm-history-purge command line
 option, 15
 synadm-media-delete command line
 option, 12
 synadm-media-purge command line
 option, 14
 --before-days <before_days>
 synadm-history-purge command line
 option, 15
 synadm-media-delete command line
 option, 12
 synadm-media-purge command line
 option, 14
 --before-event-id <before_event_id>
 synadm-history-purge command line
 option, 15
 --before-ts <before_ts>
 synadm-history-purge command line
 option, 15
 synadm-media-delete command line
 option, 12
 synadm-media-purge command line
 option, 14
 --block
 synadm-room-delete command line
 option, 8
 --config-file <config_file>
 synadm command line option, 1
 --data <data>
 synadm-matrix-raw command line
 option, 17
 --data-file <data_file>
 synadm-matrix-raw command line
 option, 17
 --datetime
 synadm-regtok-details command line
 option, 18
 synadm-regtok-list command line
 option, 19

```
--deactivate
    synadm-user-modify command line
    option,6
--deactivated
    synadm-user-list command line
    option,4
--delete-local
    synadm-history-purge command line
    option,15
--delete-profiles
    synadm-media-delete command line
    option,12
--display-name <display_name>
    synadm-user-modify command line
    option,6
--expire <expire>
    synadm-user-login command line
    option,4
--expire-at <expire_at>
    synadm-regtok-new command line
    option,19
    synadm-regtok-update command line
    option,20
--expire-days <expire_days>
    synadm-user-login command line
    option,4
--expire-never
    synadm-user-login command line
    option,4
--expire-ts <expire_ts>
    synadm-user-login command line
    option,4
--expiry-ts <expiry_ts>
    synadm-regtok-new command line
    option,19
    synadm-regtok-update command line
    option,20
--from <from_>
    synadm-media-list command line
    option,13
    synadm-room-list command line
    option,10
    synadm-room-search command line
    option,11
    synadm-user-list command line
    option,4
    synadm-user-media command line
    option,5
    synadm-user-search command line
    option,7
--gdpr-erase
    synadm-user-deactivate command
    line option,3
--guests
    synadm-user-list command line
    option,4
--ids
    synadm-user-membership command
    line option,6
--invalid
    synadm-regtok-list command line
    option,19
--kib <size>
    synadm-media-delete command line
    option,12
--length <length>
    synadm-regtok-new command line
    option,19
--limit <limit>
    synadm-media-list command line
    option,13
    synadm-room-list command line
    option,10
    synadm-room-search command line
    option,11
    synadm-user-list command line
    option,4
    synadm-user-media command line
    option,5
    synadm-user-search command line
    option,7
--matrix-path <matrix_path>
    synadm-config command line option,2
--media-id <media_id>
    synadm-media-delete command line
    option,12
    synadm-media-quarantine command
    line option,14
--message <message>
    synadm-room-delete command line
    option,8
--method <method>
    synadm-matrix-raw command line
    option,17
--name <name>
    synadm-room-list command line
    option,10
    synadm-user-list command line
    option,4
--new-room-user-id <new_room_user_id>
    synadm-room-delete command line
    option,8
--no-admin
    synadm-user-modify command line
    option,6
--no-batch
    synadm command line option,1
--no-guests
```

```

    synadm-user-list command line
        option,4
--no-logout
    synadm-user-password command line
        option,7
--no-purge
    synadm-room-delete command line
        option,8
--output <output>
    synadm command line option,1
    synadm-config command line option,2
--password <password>
    synadm-matrix-login command line
        option,16
    synadm-user-modify command line
        option,6
    synadm-user-password command line
        option,7
--password-prompt
    synadm-user-modify command line
        option,6
--prompt
    synadm-matrix-raw command line
        option,17
--reverse
    synadm-media-list command line
        option,13
    synadm-room-list command line
        option,10
    synadm-room-resolve command line
        option,11
    synadm-user-media command line
        option,5
--room-id <room_id>
    synadm-media-list command line
        option,13
    synadm-media-quarantine command
        line option,14
--room-name <room_name>
    synadm-room-delete command line
        option,8
--server-name <server_name>
    synadm-media-quarantine command
        line option,14
--size
    synadm-media-delete command line
        option,12
--sort <sort>
    synadm-media-list command line
        option,13
    synadm-room-list command line
        option,10
    synadm-user-media command line
        option,5
    --threepid <threepid>
        synadm-user-modify command line
            option,6
    --timeout <timeout>
        synadm-config command line option,2
    --timestamp
        synadm-regtok-details command line
            option,18
        synadm-regtok-list command line
            option,19
    --token <token>
        synadm-config command line option,2
        synadm-matrix-raw command line
            option,17
        synadm-regtok-new command line
            option,19
    --user <user>
        synadm-config command line option,2
    --user-id <user_id>
        synadm-media-list command line
            option,13
        synadm-media-quarantine command
            line option,14
        synadm-room-make-admin command
            line option,10
        synadm-user-list command line
            option,4
    --uses-allowed <uses_allowed>
        synadm-regtok-new command line
            option,19
        synadm-regtok-update command line
            option,20
    --valid
        synadm-regtok-list command line
            option,19
    --verbose
        synadm command line option,1
-a
    synadm-user-modify command line
        option,6
-b
    synadm-config command line option,2
    synadm-media-delete command line
        option,12
    synadm-media-purge command line
        option,14
    synadm-room-delete command line
        option,8
-c
    synadm command line option,1
-d
    synadm-history-purge command line
        option,15
    synadm-matrix-raw command line

```

```

    option,17
synadm-media-delete command line
    option,12
synadm-media-purge command line
    option,14
synadm-regtok-details command line
    option,18
synadm-regtok-list command line
    option,19
synadm-user-list command line
    option,4
synadm-user-login command line
    option,4
-e
synadm-regtok-new command line
    option,19
synadm-regtok-update command line
    option,20
synadm-user-deactivate command
line option,3
-f
synadm-matrix-row command line
    option,17
synadm-media-list command line
    option,13
synadm-room-list command line
    option,10
synadm-room-search command line
    option,11
synadm-user-list command line
    option,4
synadm-user-media command line
    option,5
synadm-user-search command line
    option,7
-g
synadm-user-list command line
    option,4
-i
synadm-history-purge command line
    option,15
synadm-media-delete command line
    option,12
synadm-media-quarantine command
line option,14
synadm-user-list command line
    option,4
-l
synadm-media-list command line
    option,13
synadm-regtok-new command line
    option,19
synadm-room-list command line
    option,10
synadm-room-search command line
    option,11
synadm-user-list command line
    option,4
synadm-user-media command line
    option,5
synadm-user-search command line
    option,7
-m
synadm-config command line option,2
synadm-matrix-row command line
    option,17
synadm-room-delete command line
    option,8
-n
synadm-regtok-new command line
    option,19
synadm-room-delete command line
    option,8
synadm-room-list command line
    option,10
synadm-user-list command line
    option,4
synadm-user-modify command line
    option,6
synadm-user-password command line
    option,7
-o
synadm command line option,1
synadm-config command line option,2
-p
synadm-config command line option,2
synadm-matrix-login command line
    option,16
synadm-matrix-row command line
    option,17
synadm-user-modify command line
    option,6
synadm-user-password command line
    option,7
-r
synadm-media-list command line
    option,13
synadm-media-quarantine command
line option,14
synadm-room-list command line
    option,10
synadm-room-resolve command line
    option,11
synadm-user-media command line
    option,5
-s
synadm-media-list command line
    option,13

```


synadm-media-quarantine command line option, 14
 synadm-room-list command line option, 10
 synadm-user-media command line option, 5
 -t
 synadm-config command line option, 2
 synadm-matrix-raw command line option, 17
 synadm-media-delete command line option, 12
 synadm-media-purge command line option, 14
 synadm-regtok-details command line option, 18
 synadm-regtok-list command line option, 19
 synadm-regtok-new command line option, 19
 synadm-regtok-update command line option, 20
 synadm-user-modify command line option, 6
 -u
 synadm-config command line option, 2
 synadm-media-list command line option, 13
 synadm-media-quarantine command line option, 14
 synadm-regtok-new command line option, 19
 synadm-regtok-update command line option, 20
 synadm-room-delete command line option, 8
 synadm-room-make-admin command line option, 10
 synadm-user-modify command line option, 6
 -v
 synadm command line option, 1
 synadm-regtok-list command line option, 19
 synadm-user-modify command line option, 6
 -w
 synadm-config command line option, 2

A

APIHelper (*class in synadm.cli*), 21
 ApiRequest (*class in synadm.api*), 22

C

CONFIG (*synadm.cli.APIHelper attribute*), 21

E

ENDPOINT
 synadm-matrix-raw command line option, 17

F

FORMATTERS (*synadm.cli.APIHelper attribute*), 21

G

group_delete() (*synadm.api.SynapseAdmin method*), 26

GROUP_ID
 synadm-group-delete command line option, 16

I

init_logger() (*synadm.cli.APIHelper method*), 21

L

load() (*synadm.cli.APIHelper method*), 21

M

Matrix (*class in synadm.api*), 23

media_delete() (*synadm.api.SynapseAdmin method*), 26

media_delete_by_date_or_size() (*synadm.api.SynapseAdmin method*), 26

MEDIA_ID
 synadm-media-protect command line option, 13

media_protect() (*synadm.api.SynapseAdmin method*), 26

media_quarantine() (*synadm.api.SynapseAdmin method*), 26

module
 synadm.api, 22
 synadm.cli, 21

O

output() (*synadm.cli.APIHelper method*), 21

P

purge_history() (*synadm.api.SynapseAdmin method*), 27

purge_history_status() (*synadm.api.SynapseAdmin method*), 27

PURGE_ID
 synadm-history-purge-status command line option, 15

`purge_media_cache()` (*synadm.api.SynapseAdmin method*), 26

Q

`query()` (*synadm.api.ApiRequest method*), 22

R

`raw_request()` (*synadm.api.Matrix method*), 23

`regtok_delete()` (*synadm.api.SynapseAdmin method*), 28

`regtok_details()` (*synadm.api.SynapseAdmin method*), 27

`regtok_list()` (*synadm.api.SynapseAdmin method*), 27

`regtok_new()` (*synadm.api.SynapseAdmin method*), 27

`regtok_update()` (*synadm.api.SynapseAdmin method*), 28

`room_delete()` (*synadm.api.SynapseAdmin method*), 26

`room_details()` (*synadm.api.SynapseAdmin method*), 26

`room_get_aliases()` (*synadm.api.Matrix method*), 23

`room_get_id()` (*synadm.api.Matrix method*), 23

ROOM_ID

`synadm-history-purge` command line option, 15

`synadm-room-delete` command line option, 9

`synadm-room-details` command line option, 9

`synadm-room-make-admin` command line option, 10

`synadm-room-members` command line option, 11

ROOM_ID_OR_ALIASES

`synadm-room-join` command line option, 9

`synadm-room-resolve` command line option, 11

`room_join()` (*synadm.api.SynapseAdmin method*), 26

`room_list()` (*synadm.api.SynapseAdmin method*), 26

`room_make_admin()` (*synadm.api.SynapseAdmin method*), 26

`room_media_list()` (*synadm.api.SynapseAdmin method*), 26

`room_media_quarantine()` (*synadm.api.SynapseAdmin method*), 26

`room_members()` (*synadm.api.SynapseAdmin method*), 26

S

SEARCH_TERM

`synadm-room-search` command line option, 12

`synadm-user-search` command line option, 8

`server_name()` (*synadm.api.Matrix method*), 23

`synadm` command line option

`--batch`, 1

`--config-file <config_file>`, 1

`--no-batch`, 1

`--output <output>`, 1

`--verbose`, 1

`-c`, 1

`-o`, 1

`-v`, 1

`synadm.api` module, 22

`synadm.cli` module, 21

`synadm-config` command line option

`--admin-path <admin_path>`, 2

`--base-url <base_url>`, 2

`--matrix-path <matrix_path>`, 2

`--output <output>`, 2

`--timeout <timeout>`, 2

`--token <token>`, 2

`--user <user>`, 2

`-b`, 2

`-m`, 2

`-o`, 2

`-p`, 2

`-t`, 2

`-u`, 2

`-w`, 2

`synadm-group-delete` command line option

GROUP_ID, 16

`synadm-history-purge` command line option

`--before <before>`, 15

`--before-days <before_days>`, 15

`--before-event-id <before_event_id>`, 15

`--before-ts <before_ts>`, 15

`--delete-local`, 15

`-d`, 15

`-i`, 15

ROOM_ID, 15

`synadm-history-purge-status` command line option

PURGE_ID, 15

`synadm-matrix-login` command line option

`--password <password>`, 16

`-p`, 16

```

    USER_ID, 16
synadm-matrix-raw command line option
  --data <data>, 17
  --data-file <data_file>, 17
  --method <method>, 17
  --prompt, 17
  --token <token>, 17
  -d, 17
  -f, 17
  -m, 17
  -p, 17
  -t, 17
  ENDPOINT, 17
synadm-media-delete command line
  option
  --all, 12
  --before <before>, 12
  --before-days <before_days>, 12
  --before-ts <before_ts>, 12
  --delete-profiles, 12
  --kib <size>, 12
  --media-id <media_id>, 12
  --size, 12
  -b, 12
  -d, 12
  -i, 12
  -t, 12
synadm-media-list command line option
  -R, 13
  --from <from_>, 13
  --limit <limit>, 13
  --reverse, 13
  --room-id <room_id>, 13
  --sort <sort>, 13
  --user-id <user_id>, 13
  -f, 13
  -l, 13
  -r, 13
  -s, 13
  -u, 13
synadm-media-protect command line
  option
  MEDIA_ID, 13
synadm-media-purge command line option
  --before <before>, 14
  --before-days <before_days>, 14
  --before-ts <before_ts>, 14
  -b, 14
  -d, 14
  -t, 14
synadm-media-quarantine command line
  option
  --media-id <media_id>, 14
  --room-id <room_id>, 14
  --server-name <server_name>, 14
  --user-id <user_id>, 14
  -i, 14
  -r, 14
  -s, 14
  -u, 14
synadm-regtok-delete command line
  option
  TOKEN, 18
synadm-regtok-details command line
  option
  --datetime, 18
  --timestamp, 18
  -d, 18
  -t, 18
  TOKEN, 18
synadm-regtok-list command line option
  -V, 19
  --datetime, 19
  --invalid, 19
  --timestamp, 19
  --valid, 19
  -d, 19
  -t, 19
  -v, 19
synadm-regtok-new command line option
  --expire-at <expire_at>, 19
  --expiry-ts <expiry_ts>, 19
  --length <length>, 19
  --token <token>, 19
  --uses-allowed <uses_allowed>, 19
  -e, 19
  -l, 19
  -n, 19
  -t, 19
  -u, 19
synadm-regtok-update command line
  option
  --expire-at <expire_at>, 20
  --expiry-ts <expiry_ts>, 20
  --uses-allowed <uses_allowed>, 20
  -e, 20
  -t, 20
  -u, 20
  TOKEN, 20
synadm-room-delete command line option
  --block, 8
  --message <message>, 8
  --new-room-user-id
    <new_room_user_id>, 8
  --no-purge, 8
  --room-name <room_name>, 8
  -b, 8
  -m, 8

```

```

    -n, 8
    -u, 8
    ROOM_ID, 9
synadm-room-details command line
    option
    ROOM_ID, 9
synadm-room-join command line option
    ROOM_ID_OR_ALIAS, 9
    USER_ID, 9
synadm-room-list command line option
    --from <from_>, 10
    --limit <limit>, 10
    --name <name>, 10
    --reverse, 10
    --sort <sort>, 10
    -f, 10
    -l, 10
    -n, 10
    -r, 10
    -s, 10
synadm-room-make-admin command line
    option
    --user-id <user_id>, 10
    -u, 10
    ROOM_ID, 10
synadm-room-members command line
    option
    ROOM_ID, 11
synadm-room-resolve command line
    option
    --reverse, 11
    -r, 11
    ROOM_ID_OR_ALIAS, 11
synadm-room-search command line option
    --from <from_>, 11
    --limit <limit>, 11
    -f, 11
    -l, 11
    SEARCH_TERM, 12
synadm-user-deactivate command line
    option
    --gdpr-erase, 3
    -e, 3
    USER_ID, 3
synadm-user-details command line
    option
    USER_ID, 3
synadm-user-list command line option
    -G, 4
    --deactivated, 4
    --from <from_>, 4
    --guests, 4
    --limit <limit>, 4
    --name <name>, 4
    --no-guests, 4
    --user-id <user_id>, 4
    -d, 4
    -f, 4
    -g, 4
    -i, 4
    -l, 4
    -n, 4
synadm-user-login command line option
    --expire <expire>, 4
    --expire-days <expire_days>, 4
    --expire-never, 4
    --expire-ts <expire_ts>, 4
    -d, 4
    USER_ID, 5
synadm-user-media command line option
    --from <from_>, 5
    --limit <limit>, 5
    --reverse, 5
    --sort <sort>, 5
    -f, 5
    -l, 5
    -r, 5
    -s, 5
    USER_ID, 5
synadm-user-membership command line
    option
    --aliases, 6
    --ids, 6
    USER_ID, 6
synadm-user-modify command line option
    -P, 6
    --activate, 6
    --admin, 6
    --avatar-url <avatar_url>, 6
    --deactivate, 6
    --display-name <display_name>, 6
    --no-admin, 6
    --password <password>, 6
    --password-prompt, 6
    --threepid <threepid>, 6
    -a, 6
    -n, 6
    -p, 6
    -t, 6
    -u, 6
    -v, 6
    USER_ID, 7
synadm-user-password command line
    option
    --no-logout, 7
    --password <password>, 7
    -n, 7
    -p, 7

```

USER_ID, 7
 synadm-user-search command line option
 --from <from_>, 7
 --limit <limit>, 7
 -f, 7
 -l, 7
 SEARCH_TERM, 8
 synadm-user-whois command line option
 USER_ID, 8
 SynapseAdmin (*class in synadm.api*), 24

T

TOKEN

synadm-regtok-delete command line option, 18
 synadm-regtok-details command line option, 18
 synadm-regtok-update command line option, 20

U

user_deactivate() (*synadm.api.SynapseAdmin method*), 24
 user_details() (*synadm.api.SynapseAdmin method*), 25
 USER_ID
 synadm-matrix-login command line option, 16
 synadm-room-join command line option, 9
 synadm-user-deactivate command line option, 3
 synadm-user-details command line option, 3
 synadm-user-login command line option, 5
 synadm-user-media command line option, 5
 synadm-user-membership command line option, 6
 synadm-user-modify command line option, 7
 synadm-user-password command line option, 7
 synadm-user-whois command line option, 8
 user_list() (*synadm.api.SynapseAdmin method*), 24
 user_login() (*synadm.api.Matrix method*), 23
 user_login() (*synadm.api.SynapseAdmin method*), 25
 user_media() (*synadm.api.SynapseAdmin method*), 26
 user_media_quarantine() (*synadm.api.SynapseAdmin method*), 26

user_membership() (*synadm.api.SynapseAdmin method*), 24
 user_modify() (*synadm.api.SynapseAdmin method*), 26
 user_password() (*synadm.api.SynapseAdmin method*), 25
 user_whois() (*synadm.api.SynapseAdmin method*), 26

V

version() (*synadm.api.SynapseAdmin method*), 26

W

write_config() (*synadm.cli.APIHelper method*), 21