

Package ‘mx.api’

April 21, 2026

Type Package

Title Minimal Matrix Client-Server API

Version 0.1.0

Date 2026-04-15

Description A minimal-dependency client for the 'Matrix' Client-Server HTTP API <<https://spec.matrix.org/>>, suitable for talking to a 'Synapse' homeserver <<https://element-hq.github.io/synapse/>>. Covers login, room management, message send and history, and media upload or download. End-to-end encryption is out of scope; use unencrypted rooms or a separate crypto package.

License MIT + file LICENSE

URL <https://github.com/cornball-ai/mx.api>

BugReports <https://github.com/cornball-ai/mx.api/issues>

Imports curl, jsonlite

Suggests tinytest

Encoding UTF-8

NeedsCompilation no

Author Troy Hernandez [aut, cre] (ORCID: <<https://orcid.org/0009-0005-4248-604X>>), cornball.ai [cph]

Maintainer Troy Hernandez <troy@cornball.ai>

Repository CRAN

Date/Publication 2026-04-21 19:42:09 UTC

Contents

mx.api-package	2
mx_download	2
mx_login	3
mx_logout	4

mx_messages	4
mx_react	5
mx_read_receipt	6
mx_register	6
mx_rooms	7
mx_room_create	8
mx_room_join	8
mx_room_leave	9
mx_room_members	10
mx_room_name	10
mx_room_topic	11
mx_send	11
mx_session	12
mx_sync	13
mx_upload	14
mx_whoami	14
Index	16

mx.api-package	<i>mx.api: Minimal Matrix Client-Server API</i>
----------------	---

Description

Base-R bindings for the Matrix Client-Server HTTP API, suitable for talking to a Synapse home-server. Two dependencies: curl and jsonlite. End-to-end encryption is out of scope; use unencrypted rooms or a separate crypto package.

Author(s)

Maintainer: Troy Hernandez <troy@cornball.ai>

mx_download	<i>Download a media file by mxc URI</i>
-------------	---

Description

Download a media file by mxc URI

Usage

```
mx_download(session, mxc_url, dest)
```

Arguments

session	An "mx_session" object.
mxc_url	Character. An "mxc://server/id" URI.
dest	Character. Destination file path.

Value

The destination path, invisibly.

Examples

```
## Not run:
mx_download(s, "mxc://matrix.example/abc123", tempfile())

## End(Not run)
```

mx_login	<i>Log in to a Matrix homeserver</i>
----------	--------------------------------------

Description

Authenticates with a Matrix homeserver using password login and returns a session object carrying the access token and device id.

Usage

```
mx_login(server, user, password, device_id = NULL)
```

Arguments

server	Character. Homeserver base URL (e.g. "https://matrix.example").
user	Character. User localpart or full Matrix ID.
password	Character. Account password.
device_id	Character or NULL. Reuse an existing device id.

Value

An object of class "mx_session".

Examples

```
## Not run:
s <- mx_login("https://matrix.example", "alice", "hunter2")

## End(Not run)
```

mx_logout	<i>Log out of a Matrix session</i>
-----------	------------------------------------

Description

Invalidates the access token on the homeserver.

Usage

```
mx_logout(session)
```

Arguments

session	An "mx_session" object.
---------	-------------------------

Value

Invisible NULL.

Examples

```
## Not run:
mx_logout(s)

## End(Not run)
```

mx_messages	<i>Fetch historical messages from a room</i>
-------------	--

Description

Thin wrapper over the /rooms/{id}/messages endpoint.

Usage

```
mx_messages(session, room_id, from = NULL, dir = "b", limit = 50L)
```

Arguments

session	An "mx_session" object.
room_id	Character. The room ID.
from	Character or NULL. Pagination token; NULL starts at the most recent message.
dir	Character. "b" (backwards, default) or "f" (forwards).
limit	Integer. Maximum events to return.

Value

A list with fields chunk, start, end.

Examples

```
## Not run:
mx_messages(s, "!abc:matrix.example", limit = 20L)

## End(Not run)
```

 mx_react

Send a reaction (annotation) to a room event

Description

Posts an m.reaction event tying key (usually a thumbs-up or other emoji) to event_id. Matrix reactions are plain events under the hood; they relate to the target via m.annotation.

Usage

```
mx_react(session, room_id, event_id, key)
```

Arguments

session	An "mx_session" object.
room_id	Character. The room ID.
event_id	Character. The event being reacted to.
key	Character. The reaction key (usually an emoji).

Value

The event ID of the sent reaction.

Examples

```
## Not run:
mx_react(s, "!abc:matrix.example", "$eventid", "thumbs-up")

## End(Not run)
```

mx_read_receipt *Send a read receipt for a room event*

Description

Public receipt (`m.read`) advances the "seen" marker in other clients; private receipt (`m.read.private`) only advances the bot's own view. Defaults to public so user clients show "seen by @bot".

Usage

```
mx_read_receipt(session, room_id, event_id,
               receipt_type = c("m.read", "m.read.private"))
```

Arguments

<code>session</code>	An "mx_session" object.
<code>room_id</code>	Character. The room ID.
<code>event_id</code>	Character. The event to mark as read.
<code>receipt_type</code>	Character. "m.read" (default) or "m.read.private".

Value

Invisible NULL.

Examples

```
## Not run:
mx_read_receipt(s, "!abc:matrix.example", "$eventid")

## End(Not run)
```

mx_register *Register a new account on a Matrix homeserver*

Description

Creates a new user via POST `/_matrix/client/v3/register` using the `m.login.dummy` auth flow. Most homeservers only accept this when open registration is enabled (or a registration token is supplied). On success returns a ready-to-use `mx_session` — registration also logs the new user in.

Usage

```
mx_register(server, username, password, device_id = NULL,
            initial_device_display_name = NULL, inhibit_login = FALSE)
```

Arguments

server	Character. Homeserver base URL.
username	Character. Desired localpart (e.g. "alice").
password	Character. Account password.
device_id	Character or NULL. Device id to assign; a server-generated one is used if NULL.
initial_device_display_name	Character or NULL. Human-readable label for the device.
inhibit_login	Logical. When TRUE, the server creates the account but does not return a session; the call returns a list with the new user_id instead of an mx_session.

Value

An mx_session object on login, or a list with user_id when inhibit_login = TRUE.

Examples

```
## Not run:
s <- mx_register("https://matrix.example", "alice", "hunter2")

## End(Not run)
```

mx_rooms	<i>List rooms the user has joined</i>
----------	---------------------------------------

Description

List rooms the user has joined

Usage

```
mx_rooms(session)
```

Arguments

session	An "mx_session" object.
---------	-------------------------

Value

Character vector of room IDs.

Examples

```
## Not run:
mx_rooms(s)

## End(Not run)
```

mx_room_create	<i>Create a room</i>
----------------	----------------------

Description

Create a room

Usage

```
mx_room_create(session, name = NULL, topic = NULL, visibility = "private",
               preset = NULL, invite = character())
```

Arguments

session	An "mx_session" object.
name	Character or NULL. Human-readable room name.
topic	Character or NULL. Room topic.
visibility	Character. "private" (default) or "public".
preset	Character or NULL. A Matrix room preset ("private_chat", "trusted_private_chat", "public_chat").
invite	Character vector. Matrix IDs to invite.

Value

The new room ID as a character string.

Examples

```
## Not run:
room_id <- mx_room_create(s, name = "test", topic = "hello")

## End(Not run)
```

mx_room_join	<i>Join a room by ID or alias</i>
--------------	-----------------------------------

Description

Join a room by ID or alias

Usage

```
mx_room_join(session, room)
```

Arguments

session	An "mx_session" object.
room	Character. Room ID (!abc:server) or alias (#name:server).

Value

The joined room ID.

Examples

```
## Not run:
mx_room_join(s, "#general:matrix.example")

## End(Not run)
```

mx_room_leave	<i>Leave a room</i>
---------------	---------------------

Description

Leave a room

Usage

```
mx_room_leave(session, room_id)
```

Arguments

session	An "mx_session" object.
room_id	Character. The room ID.

Value

Invisible NULL.

Examples

```
## Not run:
mx_room_leave(s, "!abc:matrix.example")

## End(Not run)
```

mx_room_members	<i>List the members of a room</i>
-----------------	-----------------------------------

Description

List the members of a room

Usage

```
mx_room_members(session, room_id)
```

Arguments

session	An "mx_session" object.
room_id	Character. The room ID.

Value

Character vector of Matrix user IDs currently joined.

Examples

```
## Not run:
mx_room_members(s, "!abc:matrix.example")

## End(Not run)
```

mx_room_name	<i>Get a room's human-readable name</i>
--------------	---

Description

Reads the m.room.name state event. Returns NULL if the room has no name set or the state event is inaccessible.

Usage

```
mx_room_name(session, room_id)
```

Arguments

session	An "mx_session" object.
room_id	Character. The room ID.

Value

Character scalar or NULL.

Examples

```
## Not run:
mx_room_name(s, "!abc:matrix.example")

## End(Not run)
```

mx_room_topic	<i>Get a room's topic</i>
---------------	---------------------------

Description

Reads the m.room.topic state event. Returns NULL if the room has no topic set or the state event is inaccessible.

Usage

```
mx_room_topic(session, room_id)
```

Arguments

session	An "mx_session" object.
room_id	Character. The room ID.

Value

Character scalar or NULL.

Examples

```
## Not run:
mx_room_topic(s, "!abc:matrix.example")

## End(Not run)
```

mx_send	<i>Send a message to a room</i>
---------	---------------------------------

Description

Send a message to a room

Usage

```
mx_send(session, room_id, body, msgtype = "m.text", extra = NULL)
```

Arguments

session	An "mx_session" object.
room_id	Character. The room ID.
body	Character. The message body.
msgtype	Character. Matrix msgtype, default "m.text".
extra	List or NULL. Extra fields merged into the event content (e.g. formatted body, reply relation).

Value

The event ID of the sent message.

Examples

```
## Not run:
mx_send(s, "!abc:matrix.example", "hello world")

## End(Not run)
```

 mx_session

Reconstruct a session from saved credentials

Description

Reconstruct a session from saved credentials

Usage

```
mx_session(server, token, user_id, device_id)
```

Arguments

server	Character. Homeserver base URL.
token	Character. Access token from a prior login.
user_id	Character. Full Matrix ID (e.g. "@troy:example.org").
device_id	Character. Device id from the prior login.

Value

An object of class "mx_session".

Examples

```
s <- mx_session(  
  server = "https://matrix.example",  
  token = "syt_...",  
  user_id = "@alice:matrix.example",  
  device_id = "ABC123"  
)
```

mx_sync

One-shot sync against the homeserver

Description

Calls /sync once and returns immediately. For streaming behaviour, the caller writes its own loop, passing the previous batch's next_batch token as since.

Usage

```
mx_sync(session, since = NULL, timeout = 0L, filter = NULL)
```

Arguments

session	An "mx_session" object.
since	Character or NULL. Sync token from a prior sync.
timeout	Integer. Long-poll timeout in milliseconds (0 returns immediately).
filter	Character or NULL. Filter ID or inline JSON filter.

Value

The parsed sync response, including next_batch.

Examples

```
## Not run:  
batch <- mx_sync(s)  
next_batch <- batch$next_batch  
  
## End(Not run)
```

mx_upload	<i>Upload a file to the homeserver media repository</i>
-----------	---

Description

Upload a file to the homeserver media repository

Usage

```
mx_upload(session, path, content_type = NULL, filename = NULL)
```

Arguments

session	An "mx_session" object.
path	Character. Local file path.
content_type	Character or NULL. MIME type; guessed from the file extension if NULL.
filename	Character or NULL. Filename advertised to the server.

Value

An "mxc://" URI as a character string.

Examples

```
## Not run:  
uri <- mx_upload(s, "photo.png")  
  
## End(Not run)
```

mx_whoami	<i>Return the identity of the current session</i>
-----------	---

Description

Return the identity of the current session

Usage

```
mx_whoami(session)
```

Arguments

session	An "mx_session" object.
---------	-------------------------

Value

A list with `user_id` and `device_id`.

Examples

```
## Not run:  
mx_whoami(s)  
  
## End(Not run)
```

Index

`mx.api` (`mx.api-package`), 2
`mx.api-package`, 2
`mx_download`, 2
`mx_login`, 3
`mx_logout`, 4
`mx_messages`, 4
`mx_react`, 5
`mx_read_receipt`, 6
`mx_register`, 6
`mx_room_create`, 8
`mx_room_join`, 8
`mx_room_leave`, 9
`mx_room_members`, 10
`mx_room_name`, 10
`mx_room_topic`, 11
`mx_rooms`, 7
`mx_send`, 11
`mx_session`, 12
`mx_sync`, 13
`mx_upload`, 14
`mx_whoami`, 14