asynctmdb Documentation

Release 0.0.3-dev

Azat Ibrakov

Sep 16, 2017

Contents:

1 Subpackages **1**

1.1 methods package 1

1.1.1 authentication methods module 1

1.1.2 find methods module 2

1.1.3 genres methods module 3

1.1.4 movies methods module 3

1.2 IMDb package 7

1.2.1 find module 7

1.2.2 title ID module 7

2 Submodules **9**

2.1 common module 9

2.2 requests module 9

3 Indices and tables **11**

Python Module Index **13**

CHAPTER 1

Subpackages

methods package

authentication methods module

asynctmdb.methods.authentication.**create_request_token**(*, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *session: aiohttp.client.ClientSession*, *date_time_format: str = '%Y-%m-%d %H:%M:%S %Z'*) → typing.Dict[str, typing.Union[int, str, datetime.datetime]]

Create a temporary request token that can be used to validate a TMDb user login.

More details about how this works can be found here.

More info at TMDb docs.

asynctmdb.methods.authentication.**create_session**(*, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *request_token: str*, *session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Union[int, str]]

Create a fully valid session.

This method can be used once a user has validated the request token.

More details about how this works can be found here.

More info at TMDb docs.

`asynctmdb.methods.authentication.`**`validate_request_token`**(*, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *username: str*, *password: str*, *request_token: str*, *session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Union[int, str]]

Validate a request token with username and password.

Caution

Please note, using this method is strongly discouraged.

The preferred method of validating a request token is to have a user authenticate the request via the TMDb website.

More details about how this works can be found here.

More info at TMDb docs.

`asynctmdb.methods.authentication.`**`create_guest_session`**(*, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *session: aiohttp.client.ClientSession*, *date_time_format: str = '%Y-%m-%d %H:%M:%S %Z'*) → typing.Dict[str, typing.Union[int, str, datetime.datetime]]

Create a new guest session.

Guest sessions are a type of session that will let a user rate movies and TV shows but not require them to have a TMDb user account.

More information about user authentication can be found here.

Please note, there should be generated only a single guest session per user (or device) as you will be able to attach the ratings to a TMDb user account in the future.

There is also IP limits in place so you should always make sure it's the end user doing the guest session actions.

If a guest session is not used for the first time within 24 hours, it will be automatically deleted.

More info at TMDb docs.

find methods module

`asynctmdb.methods.find.`**`by_id`**(*external_id: str*, *, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *language: str = None*, *external_source: str*, *session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Search for objects by an external id (for instance, an IMDb ID).

This method will search all objects (movies, TV shows and people) and return the results in a single response.

The supported external sources for each object are as follows.

	Movies	TV Shows	TV Seasons	TV Episodes	People
IMDb ID	✓	✓		✓	✓
Freebase MID		✓	✓	✓	✓
Freebase ID		✓	✓	✓	✓
TVDB ID		✓	✓	✓	
TVRage ID		✓	✓	✓	✓

More info at TMDb docs.

genres methods module

`asynctmdb.methods.genres.`**`movie`**(**, api_base_url: str = 'https://api.themoviedb.org/3', api_key: str, language: str = None, session: aiohttp.client.ClientSession*) → typing.List[typing.Dict[str, typing.Union[int, str]]]

Get the list of official genres for movies.

More info at TMDb docs.

`asynctmdb.methods.genres.`**`tv`**(**, api_base_url: str = 'https://api.themoviedb.org/3', api_key: str, language: str = None, session: aiohttp.client.ClientSession*) → typing.List[typing.Dict[str, typing.Union[int, str]]]

Get the list of official genres for TV shows.

More info at TMDb docs.

movies methods module

`asynctmdb.methods.movies.`**`details`**(*movie_id: int, *, api_base_url: str = 'https://api.themoviedb.org/3', api_key: str, language: str = None, append_to_response: str = None, session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Get the primary information about a movie.

Supports `append_to_response`. Read more about this here.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`account_states`**(*movie_id: int, *, api_base_url: str = 'https://api.themoviedb.org/3', api_key: str, session_id: str, session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Grab the following account states for a session:

- Movie rating
- If it belongs to your watchlist
- If it belongs to your favourite list

More info at TMDb docs.

`asynctmdb.methods.movies.`**`alternative_titles`**(*movie_id: int, *, api_base_url: str = 'https://api.themoviedb.org/3', api_key: str, country: str = None, session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Get all of the alternative titles for a movie.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`changes`**(*movie_id: int*, ***, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *start_date: datetime.datetime = None*, *end_date: datetime.datetime = None*, *page: int = None*, *session: aiohttp.client.ClientSession*, *format_string: str = '%Y-%m-%d %H:%M:%S %Z'*) → typing.Dict[str, typing.Any]

Get the changes for a movie.

By default only the last 24 hours are returned.

You can query up to 14 days in a single query by using the `start_date` and `end_date` query parameters.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`credits`**(*movie_id: int*, ***, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Get the cast and crew for a movie.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`images`**(*movie_id: int*, ***, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *language: str = None*, *include_image_language: str = None*, *session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Get the images that belong to a movie.

Querying images with a `language` parameter will filter the results.

If you want to include a fallback language (especially useful for backdrops) you can use the `include_image_language` parameter. This should be a comma seperated value like so:

```
include_image_language="en,null"
```

More info at TMDb docs.

`asynctmdb.methods.movies.`**`keywords`**(*movie_id: int*, ***, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Get the keywords that have been added to a movie.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`release_dates`**(*movie_id: int*, ***, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *session: aiohttp.client.ClientSession*, *format_string: str = '%Y-%m-%dT%H:%M:%S.%fZ'*) → typing.Dict[str, typing.Any]

Get the release date along with the certification for a movie.

Release dates support different types:

1. Premiere
2. Theatrical (limited)
3. Theatrical
4. Digital
5. Physical
6. TV

More info at TMDb docs.

`asynctmdb.methods.movies.`**`videos`**(*movie_id: int*, ***, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *language: str = None*, *session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Get the videos that have been added to a movie.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`translations`**(*movie_id: int*, ***, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Get a list of translations that have been created for a movie.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`recommendations`**(*movie_id: int*, ***, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *language: str = None*, *page: int = None*, *session: aiohttp.client.ClientSession*, *format_string: str = '%Y-%m-%d'*) → typing.Dict[str, typing.Any]

Get a list of recommended movies for a movie.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`similar`**(*movie_id: int*, ***, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *language: str = None*, *page: int = None*, *session: aiohttp.client.ClientSession*, *format_string: str = '%Y-%m-%d'*) → typing.Dict[str, typing.Any]

Get a list of similar movies.

This is **not** the same as the "Recommendation" system you see on the website.

These items are assembled by looking at keywords and genres.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`reviews`**(*movie_id: int*, ***, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *language: str = None*, *page: int = None*, *session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Get the user reviews for a movie.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`lists`**(*movie_id: int*, ***, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *language: str = None*, *page: int = None*, *session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Get a list of lists that this movie belongs to.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`rate`**(*movie_id: int*, ***, *rating: float*, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *guest_session_id: str = None*, *session_id: str = None*, *session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Rate a movie.

A valid session or guest session ID is required. You can read more about how this works here.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`delete_rating`**(*movie_id: int, *, api_base_url: str = 'https://api.themoviedb.org/3', api_key: str, guest_session_id: str = None, session_id: str = None, session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Remove your rating for a movie.

A valid session or guest session ID is required. You can read more about how this works here.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`latest`**(*, *api_base_url: str = 'https://api.themoviedb.org/3', api_key: str, language: str = None, session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Get the most newly created movie.

This is a live response and will continuously change.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`now_playing`**(*, *api_base_url: str = 'https://api.themoviedb.org/3', api_key: str, language: str = None, page: int = None, region: str = None, session: aiohttp.client.ClientSession, format_string: str = '%Y-%m-%d', dates_keys: typing.Iterable[str] = ['minimum', 'maximum']*) → typing.Dict[str, typing.Any]

Get a list of movies in theatres.

This is a release type query that looks for all movies that have a release type of 2 or 3 within the specified date range.

You can optionally specify a `region` parameter which will narrow the search to only look for theatrical release dates within the specified country.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`popular`**(*, *api_base_url: str = 'https://api.themoviedb.org/3', api_key: str, language: str = None, page: int = None, region: str = None, session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Get a list of the current popular movies on TMDb.

This list updates daily.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`top_rated`**(*, *api_base_url: str = 'https://api.themoviedb.org/3', api_key: str, language: str = None, page: int = None, region: str = None, session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Get the top rated movies on TMDb.

More info at TMDb docs.

`asynctmdb.methods.movies.`**`upcoming`**(*, *api_base_url: str = 'https://api.themoviedb.org/3', api_key: str, language: str = None, page: int = None, region: str = None, session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Get a list of upcoming movies in theatres.

This is a release type query that looks for all movies that have a release type of 2 or 3 within the specified date range.

You can optionally specify a region parameter which will narrow the search to only look for theatrical release dates within the specified country.

More info at TMDb docs.

IMDb package

find module

`asynctmdb.imdb.find.`**`movie`**(*imdb_id: str*, ***, *api_base_url: str = 'https://api.themoviedb.org/3'*, *api_key: str*, *language: str = None*, *session: aiohttp.client.ClientSession*) → typing.Dict[str, typing.Any]

Search TMDb movie details by IMDb ID.

Raise `ValueError` if IMDb ID is invalid or there is no details found.

title ID module

`asynctmdb.imdb.title_id.`**`from_int`**(*int_id: int*, ***, *length: int = 7*) → str

Convert integer IMDb id to string representation.

CHAPTER 2

Submodules

common module

class `asynctmdb.common.`**`StatusCode`**

Bases: `enum.IntEnum`

An enumeration.

`AUTHENTICATION_FAILED` = **3**

`INTERNAL_ERROR` = **11**

`INVALID_API_KEY` = **7**

`INVALID_PAGE` = **22**

`RESOURCE_NOT_FOUND` = **34**

`SESSION_DENIED` = **17**

`SUCCESS` = **1**

`SUCCESSFULLY_DELETED` = **13**

requests module

`asynctmdb.requests.`**`send`**(**, method: typing.Callable[[aiohttp.client.ClientSession, str, typing.Any], asynctmdb.types.AsyncContextManager], method_url: str, session: aiohttp.client.ClientSession, json_body: typing.Dict[str, typing.Any] = None, **params: typing.Dict[str, str]*) → typing.Dict[str, typing.Any]

Perform HTTP request with JSON-serializable response.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

asynctmdb.common, 9
asynctmdb.imdb.find, 7
asynctmdb.imdb.title_id, 7
asynctmdb.methods.authentication, 1
asynctmdb.methods.find, 2
asynctmdb.methods.genres, 3
asynctmdb.methods.movies, 3
asynctmdb.requests, 9

Index

A

account_states() (in module asynctmdb.methods.movies), 3
alternative_titles() (in module asynctmdb.methods.movies), 3
asynctmdb.common (module), 9
asynctmdb.imdb.find (module), 7
asynctmdb.imdb.title_id (module), 7
asynctmdb.methods.authentication (module), 1
asynctmdb.methods.find (module), 2
asynctmdb.methods.genres (module), 3
asynctmdb.methods.movies (module), 3
asynctmdb.requests (module), 9
AUTHENTICATION_FAILED (asynctmdb.common.StatusCode attribute), 9

B

by_id() (in module asynctmdb.methods.find), 2

C

changes() (in module asynctmdb.methods.movies), 3
create_guest_session() (in module asynctmdb.methods.authentication), 2
create_request_token() (in module asynctmdb.methods.authentication), 1
create_session() (in module asynctmdb.methods.authentication), 1
credits() (in module asynctmdb.methods.movies), 4

D

delete_rating() (in module asynctmdb.methods.movies), 5
details() (in module asynctmdb.methods.movies), 3

F

from_int() (in module asynctmdb.imdb.title_id), 7

I

images() (in module asynctmdb.methods.movies), 4
INTERNAL_ERROR (asynctmdb.common.StatusCode attribute), 9
INVALID_API_KEY (asynctmdb.common.StatusCode attribute), 9
INVALID_PAGE (asynctmdb.common.StatusCode attribute), 9

K

keywords() (in module asynctmdb.methods.movies), 4

L

latest() (in module asynctmdb.methods.movies), 6
lists() (in module asynctmdb.methods.movies), 5

M

movie() (in module asynctmdb.imdb.find), 7
movie() (in module asynctmdb.methods.genres), 3

N

now_playing() (in module asynctmdb.methods.movies), 6

P

popular() (in module asynctmdb.methods.movies), 6

R

rate() (in module asynctmdb.methods.movies), 5
recommendations() (in module asynctmdb.methods.movies), 5
release_dates() (in module asynctmdb.methods.movies), 4
RESOURCE_NOT_FOUND (asynctmdb.common.StatusCode attribute), 9
reviews() (in module asynctmdb.methods.movies), 5

S

send() (in module asynctmdb.requests), 9
SESSION_DENIED (asynctmdb.common.StatusCode attribute), 9
similar() (in module asynctmdb.methods.movies), 5

StatusCode (class in asynctmdb.common), 9
SUCCESS (asynctmdb.common.StatusCode attribute), 9
SUCCESSFULLY_DELETED (asynctmdb.common.StatusCode attribute), 9

T

top_rated() (in module asynctmdb.methods.movies), 6
translations() (in module asynctmdb.methods.movies), 5
tv() (in module asynctmdb.methods.genres), 3

U

upcoming() (in module asynctmdb.methods.movies), 6

V

validate_request_token() (in module asynctmdb.methods.authentication), 1
videos() (in module asynctmdb.methods.movies), 5