
trakt.py

Release 4.4.0

Mar 19, 2021

Contents

1	API	1
1.1	Interfaces	1
1.2	Objects	15
1.3	Modules	35
2	Information	45
2.1	Changelog	45
2.2	License	57
3	Indices	59
	Python Module Index	61
	Index	63

1.1 Interfaces

1.1.1 Trakt['auth']

```
class trakt.interfaces.auth.AuthInterface(client)
```

```
    login(login, password, **kwargs)
```

```
    logout()
```

1.1.2 Trakt['calendars/all/*']

```
class trakt.interfaces.calendars.AllCalendarsInterface(client)
```

```
get(media, collection=None, start_date=None, days=None, **kwargs)
```

Retrieve calendar items.

The *all* calendar displays info for all shows airing during the specified period. The *my* calendar displays episodes for all shows that have been watched, collected, or watchlisted.

Parameters

- **source** (*str*) – Calendar source (*all* or *my*)
- **media** (*str*) – Media type (*dvd*, *movies* or *shows*)
- **collection** (*str* or *None*) – Collection type (*new*, *premieres*)
- **start_date** (*datetime* or *None*) – Start date (defaults to today)
- **days** (*int* or *None*) – Number of days to display (defaults to 7)
- **query** (*str* or *None*) – Search title or description.

- **years** (*int or str or tuple or None*) – Year or range of years (e.g. *2014*, or *2014-2016*)
- **genres** (*str or list of str or None*) – Genre slugs (e.g. *action*)
- **languages** (*str or list of str or None*) – Language codes (e.g. *en*)
- **countries** (*str or list of str or None*) – Country codes (e.g. *us*)
- **runtimes** (*str or tuple or None*) – Runtime range in minutes (e.g. *30-90*)
- **ratings** (*str or tuple or None*) – Rating range between *0* and *100* (e.g. *75-100*)
- **certifications** (*str or list of str or None*) – US Content Certification (e.g. *pg-13*, *tv-pg*)
- **networks** (*str or list of str or None*) – (TV) Network name (e.g. *HBO*)
- **status** (*str or list of str or None*) – (TV) Show status (e.g. *returning series*, *in production*, *ended*)

Returns Items

Return type list of `trakt.objects.video.Video`

new (*media*, ***kwargs*)

premieres (*media*, ***kwargs*)

1.1.3 Trakt['movies']

class `trakt.interfaces.movies.MoviesInterface` (*client*)

get (*id*, *extended=None*, ***kwargs*)

popular (*extended=None*, *page=None*, *per_page=None*, ***kwargs*)

recommended (*period=None*, *extended=None*, *page=None*, *per_page=None*, ***kwargs*)

trending (*extended=None*, *page=None*, *per_page=None*, ***kwargs*)

1.1.4 Trakt['oauth']

class `trakt.interfaces.oauth.OAuthInterface` (*client*)

authorize_url (*redirect_uri*, *response_type='code'*, *state=None*, *username=None*)

pin_url (**args*, ***kwargs*)

token (**args*, ***kwargs*)

token_exchange (*code=None*, *redirect_uri=None*, *grant_type='authorization_code'*, ***kwargs*)

token_refresh (*refresh_token=None*, *redirect_uri=None*, *grant_type='refresh_token'*, ***kwargs*)

1.1.5 Trakt['oauth/device']

class trakt.interfaces.oauth.device.**DeviceOAuthInterface** (*client*)

code (***kwargs*)

poll (*device_code, expires_in, interval, **kwargs*)
Construct the device authentication poller.

Parameters

- **device_code** (*str*) – Device authentication code
- **expires_in** – Device authentication code expiry (in seconds)
- **interval** (*int*) – Device authentication poll interval

Return type DeviceOAuthPoller

token (*device_code, **kwargs*)

1.1.6 Trakt['oauth/pin']

class trakt.interfaces.oauth.pin.**PinOAuthInterface** (*client*)

url ()

1.1.7 Trakt['scrobble']

class trakt.interfaces.scrobble.**ScrobbleInterface** (*client*)

action (***kwargs*)
Perform scrobble action.

Parameters

- **action** (*str*) – Action to perform (either start, pause or stop)
- **movie** (*dict*) – Movie definition (or *None*)

Example:

```
{
    'title': 'Guardians of the Galaxy',
    'year': 2014,
    'ids': {
        'tmdb': 118340
    }
}
```

- **show** (*dict*) – Show definition (or *None*)

Example:

```
{
    'title': 'Breaking Bad',
    'year': 2008,

    'ids': {
        'tvdb': 81189
    }
}
```

- **episode** (*dict*) – Episode definition (or *None*)

Example:

```
{
    "season": 3,
    "number": 11
}
```

- **progress** (*float*) – Current movie/episode progress percentage
- **kwargs** (*dict*) – Extra request options

Returns

Response (or *None*)

Example:

```
{
    'action': 'start',
    'progress': 1.25,

    'sharing': {
        'facebook': true,
        'twitter': true,
        'tumblr': false
    },

    'movie': {
        'title': 'Guardians of the Galaxy',
        'year': 2014,

        'ids': {
            'trakt': 28,
            'slug': 'guardians-of-the-galaxy-2014',
            'imdb': 'tt2015381',
            'tmdb': 118340
        }
    }
}
```

Return type *dict*

pause (***kwargs*)

Send the scrobble “pause” action.

Use this method when the video is paused. The playback progress will be saved and `Trakt['sync/playback'].get()` can be used to resume the video from this exact position. Un-pause a video by calling the `Trakt['scrobble'].start()` method again.

Parameters

- **movie** (*dict*) – Movie definition (or *None*)

Example:

```
{
    'title': 'Guardians of the Galaxy',
    'year': 2014,

    'ids': {
        'tmdb': 118340
    }
}
```

- **show** (*dict*) – Show definition (or *None*)

Example:

```
{
    'title': 'Breaking Bad',
    'year': 2008,

    'ids': {
        'tvdb': 81189
    }
}
```

- **episode** (*dict*) – Episode definition (or *None*)

Example:

```
{
    "season": 3,
    "number": 11
}
```

- **progress** (*float*) – Current movie/episode progress percentage
- **kwargs** (*dict*) – Extra request options

Returns

Response (or *None*)

Example:

```
{
    'action': 'pause',
    'progress': 75,

    'sharing': {
        'facebook': true,
        'twitter': true,
        'tumblr': false
    },

    'movie': {
        'title': 'Guardians of the Galaxy',
        'year': 2014,

        'ids': {
```

(continues on next page)

(continued from previous page)

```
        'trakt': 28,  
        'slug': 'guardians-of-the-galaxy-2014',  
        'imdb': 'tt2015381',  
        'tmdb': 118340  
    }  
}  
}
```

Return type `dict`

start (***kwargs*)

Send the scrobble “start” action.

Use this method when the video initially starts playing or is un-paused. This will remove any playback progress if it exists.

Note: A watching status will auto expire after the remaining runtime has elapsed. There is no need to re-send every 15 minutes.

Parameters

- **movie** (`dict`) – Movie definition (or *None*)

Example:

```
{  
    'title': 'Guardians of the Galaxy',  
    'year': 2014,  
    'ids': {  
        'tmdb': 118340  
    }  
}
```

- **show** (`dict`) – Show definition (or *None*)

Example:

```
{  
    'title': 'Breaking Bad',  
    'year': 2008,  
    'ids': {  
        'tvdb': 81189  
    }  
}
```

- **episode** (`dict`) – Episode definition (or *None*)

Example:

```
{  
    "season": 3,  
    "number": 11  
}
```

- **progress** (`float`) – Current movie/episode progress percentage
- **kwargs** (`dict`) – Extra request options

Returns

Response (or *None*)

Example:

```
{
    'action': 'start',
    'progress': 1.25,

    'sharing': {
        'facebook': true,
        'twitter': true,
        'tumblr': false
    },

    'movie': {
        'title': 'Guardians of the Galaxy',
        'year': 2014,

        'ids': {
            'trakt': 28,
            'slug': 'guardians-of-the-galaxy-2014',
            'imdb': 'tt2015381',
            'tmdb': 118340
        }
    }
}
```

Return type `dict`

`stop` (***kwargs*)

Send the scrobble “stop” action.

Use this method when the video is stopped or finishes playing on its own. If the progress is above 80%, the video will be scrobbled and the *action* will be set to **scrobble**.

If the progress is less than 80%, it will be treated as a *pause* and the *action* will be set to **pause**. The playback progress will be saved and `Trakt['sync/playback'].get()` can be used to resume the video from this exact position.

Note: If you prefer to use a threshold higher than 80%, you should use `Trakt['scrobble'].pause()` yourself so it doesn't create duplicate scrobbles.

Parameters

- **movie** (`dict`) – Movie definition (or *None*)

Example:

```
{
    'title': 'Guardians of the Galaxy',
    'year': 2014,

    'ids': {
        'tmdb': 118340
    }
}
```

- **show** (`dict`) – Show definition (or *None*)

Example:

```
{
    'title': 'Breaking Bad',
    'year': 2008,

    'ids': {
        'tvdb': 81189
    }
}
```

- **episode** (*dict*) – Episode definition (or *None*)

Example:

```
{
    "season": 3,
    "number": 11
}
```

- **progress** (*float*) – Current movie/episode progress percentage
- **kwargs** (*dict*) – Extra request options

Returns

Response (or *None*)

Example:

```
{
    'action': 'scrobble',
    'progress': 99.9,

    'sharing': {
        'facebook': true,
        'twitter': true,
        'tumblr': false
    },

    'movie': {
        'title': 'Guardians of the Galaxy',
        'year': 2014,

        'ids': {
            'trakt': 28,
            'slug': 'guardians-of-the-galaxy-2014',
            'imdb': 'tt2015381',
            'tmdb': 118340
        }
    }
}
```

Return type *dict*

1.1.8 Trakt['search']

class trakt.interfaces.search.**SearchInterface** (*client*)

lookup (*id*, *service=None*, *media=None*, *extended=None*, *page=None*, *per_page=None*, ***kwargs*)

Lookup items by their Trakt, IMDB, TMDB, TVDB, or TVRage ID.

Note: If you lookup an identifier without a *media* type specified it might return multiple items if the *service* is not globally unique.

Parameters

- **id** (*str* or *int*) – Identifier value to lookup
- **service** (*str*) – Identifier service

Possible values:

- trakt
- imdb
- tmdb
- tvdb
- tvrage

- **media** (*str* or list of *str*) – Desired media type (or *None* to return all matching items)

Possible values:

- movie
- show
- episode
- person
- list

- **extended** (*str*) – Level of information to include in response

Possible values:

- *None*: Minimal (e.g. title, year, ids) (**default**)
- full: Complete

- **kwargs** (*dict*) – Extra request options

Returns Results

Return type list of *trakt.objects.media.Media*

query (*query*, *media=None*, *year=None*, *fields=None*, *extended=None*, *page=None*, *per_page=None*, ***kwargs*)

Search by titles, descriptions, translated titles, aliases, and people.

Note: Results are ordered by the most relevant score.

Parameters

- **query** (*str*) – Search title or description
- **media** (*str* or list of *str*) – Desired media type (or *None* to return all matching items)

Possible values:

- movie

- show
 - episode
 - person
 - list
 - **year** (*str* or *int*) – Desired media year (or *None* to return all matching items)
 - **fields** (*str* or *list*) – Fields to search for query (or *None* to search all fields)
 - **extended** (*str*) – Level of information to include in response
- Possible values:**
- *None*: Minimal (e.g. title, year, ids) (**default**)
 - *full*: Complete
- **kwargs** (*dict*) – Extra request options

Returns Results

Return type list of *trakt.objects.media.Media*

1.1.9 Trakt['shows']

class *trakt.interfaces.shows.ShowsInterface* (*client*)

episode (*id*, *season*, *episode*, *extended=None*, ***kwargs*)

get (*id*, *extended=None*, ***kwargs*)

last_episode (*id*, *extended=None*, ***kwargs*)

next_episode (*id*, *extended=None*, ***kwargs*)

popular (*extended=None*, *page=None*, *per_page=None*, ***kwargs*)

progress (***kwargs*)

progress_collection (***kwargs*)

progress_watched (***kwargs*)

recommended (*period=None*, *extended=None*, *page=None*, *per_page=None*, ***kwargs*)

season (*id*, *season*, *extended=None*, ***kwargs*)

seasons (*id*, *extended=None*, ***kwargs*)

trending (*extended=None*, *page=None*, *per_page=None*, ***kwargs*)

1.1.10 Trakt['sync']

class *trakt.interfaces.sync.SyncInterface* (*client*)

last_activities (***kwargs*)

playback (**args*, ***kwargs*)

1.1.11 Trakt['sync/collection']

```
class trakt.interfaces.sync.collection.SyncCollectionInterface (client)
```

```
    add (**kwargs)
    flags = {'is_collected': True}
    get (media=None, store=None, params=None, extended=None, **kwargs)
    movies (**kwargs)
    remove (**kwargs)
    shows (**kwargs)
```

1.1.12 Trakt['sync/history']

```
class trakt.interfaces.sync.history.SyncHistoryInterface (client)
```

```
    add (**kwargs)
    episodes (**kwargs)
    flags = {'is_watched': True}
    get (media=None, id=None, start_at=None, end_at=None, store=None, extended=None, page=None,
        per_page=None, **kwargs)
    movies (**kwargs)
    remove (**kwargs)
    seasons (**kwargs)
    shows (**kwargs)
```

1.1.13 Trakt['sync/playback']

```
class trakt.interfaces.sync.playback.SyncPlaybackInterface (client)
```

```
    delete (**kwargs)
    episodes (**kwargs)
    flags = {}
    get (**kwargs)
    movies (**kwargs)
    shows (**kwargs)
```

1.1.14 Trakt['sync/ratings']

```
class trakt.interfaces.sync.ratings.SyncRatingsInterface (client)
```

```
    add (**kwargs)
```

```
all (**kwargs)
episodes (**kwargs)
flags = {}
get (**kwargs)
movies (**kwargs)
remove (**kwargs)
seasons (**kwargs)
shows (**kwargs)
```

1.1.15 Trakt['sync/watched']

```
class trakt.interfaces.sync.watched.SyncWatchedInterface (client)
```

```
    flags = {'is_watched': True}
    get (media=None, store=None, params=None, extended=None, **kwargs)
    movies (**kwargs)
    shows (**kwargs)
```

1.1.16 Trakt['sync/watchlist']

```
class trakt.interfaces.sync.watchlist.SyncWatchlistInterface (client)
```

```
    add (**kwargs)
    episodes (**kwargs)
    flags = {'in_watchlist': True}
    get (media=None, sort=None, store=None, extended=None, flat=False, page=None, per_page=None,
        **kwargs)
    movies (**kwargs)
    remove (**kwargs)
    seasons (**kwargs)
    shows (**kwargs)
```

1.1.17 Trakt['users']

```
class trakt.interfaces.users.UsersInterface (client)
```

```
    likes (**kwargs)
```

1.1.18 Trakt['users/*/following']

```
class trakt.interfaces.users.following.UsersFollowingInterface(client)
```

```
    get (username, extended=None, **kwargs)
```

1.1.19 Trakt['users/*/friends']

```
class trakt.interfaces.users.friends.UsersFriendsInterface(client)
```

```
    get (username, extended=None, **kwargs)
```

1.1.20 Trakt['users/*/history']

```
class trakt.interfaces.users.history.UsersHistoryInterface(client)
```

```
    episodes (**kwargs)
```

```
    flags = {'is_watched': True}
```

```
    get (username, media=None, id=None, start_at=None, end_at=None, store=None, extended=None,
        page=None, per_page=None, **kwargs)
```

```
    movies (**kwargs)
```

```
    seasons (**kwargs)
```

```
    shows (**kwargs)
```

1.1.21 Trakt['users/*/lists']

```
class trakt.interfaces.users.lists.UsersListsInterface(client)
```

```
    create (username, name, description=None, privacy='private', display_numbers=False, allow_comments=True, sort_by='rank', sort_how='asc', **kwargs)
        Create a new list.
```

Parameters

- **username** (str) – Username (or me)
- **name** (str) – Name
- **description** (bool) – Description
- **privacy** – Privacy (private, friends, or public)
- **display_numbers** – Flag indicating this list displays numbers
- **allow_comments** – Flag indicating this list allows comments
- **sort_by** (str) – Sort By (rank, added, title, released, runtime, popularity, percentage, votes, my_rating, random, watched, collected)
- **sort_how** (str) – Sort Direction (asc, or desc)

Returns List

Return type trakt.objects.CustomList

get (*username*, ***kwargs*)
Retrieve lists for user.

Parameters **username** (*str*) – Username (or me)

Returns List

Return type trakt.objects.CustomList

1.1.22 Trakt['users/*/lists/*']

```
class trakt.interfaces.users.lists.list_.UsersListInterface (client)
```

add (***kwargs*)

delete (***kwargs*)

get (*username*, *id*, ***kwargs*)

items (*username*, *id*, *media=None*, *extended=None*, *page=None*, *per_page=None*, ***kwargs*)

like (***kwargs*)

remove (***kwargs*)

unlike (***kwargs*)

update (***kwargs*)

1.1.23 Trakt['users/*/ratings']

```
class trakt.interfaces.users.ratings.UsersRatingsInterface (client)
```

all (***kwargs*)

episodes (***kwargs*)

get (*username*, *media=None*, *rating=None*, *store=None*, *extended=None*, *page=None*,
per_page=None, ***kwargs*)

movies (***kwargs*)

seasons (***kwargs*)

shows (***kwargs*)

1.1.24 Trakt['users/settings']

```
class trakt.interfaces.users.settings.UsersSettingsInterface (client)
```

get (***kwargs*)

1.1.25 Trakt['users/*/watchlist']

```
class trakt.interfaces.users.watchlist.UsersWatchlistInterface(client)

    episodes (**kwargs)

    flags = {'in_watchlist': True}

    get (username, media=None, sort=None, store=None, extended=None, page=None, per_page=None,
         **kwargs)

    movies (**kwargs)

    seasons (**kwargs)

    shows (**kwargs)
```

1.2 Objects

1.2.1 Comment

```
class trakt.objects.comment.Comment (client, keys)
    Bases: object
```

```
    comment = None
        Type str
        Comment body

    created_at = None
        Type datetime
        Timestamp of when this comment was created

    id
        Retrieve the comment identifier.
        Return type int

    keys = None
        Type list of tuple
        Keys (for trakt, imdb, tvdb, etc..), defined as:
        ..code-block:
```

```
[
    (<service>, <id>)
]
```

```
    liked_at = None
        Type datetime
        Timestamp of when this comment was liked

    likes = None
        Type int
```

Number of likes

parent_id = None

Type `int`

Parent comment id

pk

Retrieve the primary key (unique identifier for the comment).

Returns ("trakt", <id>) or None if no primary key is available

Return type tuple

replies = None

Type `int`

Number of replies

review = None

Type `bool`

Flag indicating this comment is a review

spoiler = None

Type `bool`

Flag indicating this comment has a spoiler

user = None

Type `dict`

Author details

user_rating = None

Type `float`

Author rating for the item

1.2.2 Episode

class trakt.objects.episode.**Episode** (*client, keys=None, index=None*)

Bases: `trakt.objects.video.Video`

available_translations = None

Type `list`

Available translations (for title, overview, etc..)

first_aired = None

Type `datetime`

First air date

get_key (*service*)

number_abs = None

Type `int`

Absolute episode number

pk

Retrieve the primary key (unique identifier for the item).

Provides the following identifiers (by media type):

- **movie:** imdb
- **show:** tvdb
- **season:** tvdb
- **episode:** tvdb
- **custom_list:** trakt
- **person:** tmdb

Returns (<service>, <value>) or None if no primary key is available

Return type tuple

runtime = None

Type int

Runtime

season = None

Type *trakt.objects.season.Season*

Season

show = None

Type *trakt.objects.show.Show*

Show

title = None

Type str

Title

to_dict ()

Dump episode to a dictionary.

Returns Episode dictionary

Return type dict

to_identifier ()

Retrieve the episode identifier.

Returns Episode identifier/definition

Return type dict

to_info (*args, **kwargs)

Dump episode to a dictionary.

Deprecated: use the `to_dict()` method instead.

updated_at = None

Type datetime

Updated date/time

1.2.3 List

```
class trakt.objects.list.base.List (client, keys, user)
    Bases: object

    allow_comments = None
        Type bool
        Flag indicating this list allows comments

    comment_count = None
        Type int
        Number of comments

    created_at = None
        Type datetime
        Timestamp of when this list was created

    description = None
        Type str
        Description

    display_numbers = None
        Type bool
        Flag indicating this list displays numbers

    id
        Retrieve the list identifier.
        Return type int

    item_count = None
        Type int
        Number of items

    items (**kwargs)
        Retrieve list items.
        Parameters kwargs (dict) – Extra request options
        Returns Current list items
        Return type list of trakt.objects.media.Media

    keys = None
        Type list of tuple
        Keys (for trakt, imdb, tvdb, etc..), defined as:
        ..code-block:
```

```
[
    (<service>, <id>)
]
```

like (**kwargs)

Like the list.

Parameters **kwargs** (`dict`) – Extra request options

Returns Boolean to indicate if the request was successful

Return type `bool`

like_count

Retrieve the number of likes.

Return type `int`

liked_at = None

Type `datetime`

Timestamp of when this list was liked

likes = None

Type `int`

Number of likes

name = None

Type `str`

Name

pk

Retrieve the primary key (unique identifier for the list).

Returns ("trakt", <id>) or None if no primary key is available

Return type `tuple`

privacy = None

Type `str`

Privacy

Possible values:

- private
- friends
- public

sort_by = None

Type `str`

Sort By

Possible values:

- rank
- added
- title
- released
- runtime

- popularity
- percentage
- votes
- my_rating
- random
- watched
- collected

sort_how = None

Type `str`

Sort Direction

Possible values:

- asc
- desc

unlike (***kwargs*)

Un-like the list.

Parameters **kwargs** (`dict`) – Extra request options

Returns Boolean to indicate if the request was successful

Return type `bool`

updated_at = None

Type `datetime`

Timestamp of when this list was last updated

user = None

Type `trakt.objects.User`

Author

username

Retrieve author username.

Return type `str`

1.2.4 CustomList

class `trakt.objects.list.custom.CustomList` (*client, keys, user*)

Bases: `trakt.objects.list.base.List`

add (*items, **kwargs*)

Add specified items to the list.

Parameters

- **items** (`list`) – Items that should be added to the list
- **kwargs** (`dict`) – Extra request options

Returns `Response`

Return type `dict`

delete (***kwargs*)
Delete the list.

Parameters **kwargs** (`dict`) – Extra request options

Returns Boolean to indicate if the request was successful

Return type `bool`

id
Retrieve the list identifier.

Return type `int`

items (***kwargs*)
Retrieve list items.

Parameters **kwargs** (`dict`) – Extra request options

Returns Current list items

Return type list of `trakt.objects.media.Media`

like (***kwargs*)
Like the list.

Parameters **kwargs** (`dict`) – Extra request options

Returns Boolean to indicate if the request was successful

Return type `bool`

like_count
Retrieve the number of likes.

Return type `int`

pk
Retrieve the primary key (unique identifier for the list).

Returns ("trakt", <id>) or None if no primary key is available

Return type `tuple`

remove (*items, **kwargs*)
Remove specified items from the list.

Parameters

- **items** (`list`) – Items that should be removed from the list
- **kwargs** (`dict`) – Extra request options

Returns Response

Return type `dict`

unlike (***kwargs*)
Un-like the list.

Parameters **kwargs** (`dict`) – Extra request options

Returns Boolean to indicate if the request was successful

Return type `bool`

update (***kwargs*)

Update the list with the current object attributes.

Parameters **kwargs** (*dict*) – Extra request options**Returns** Boolean to indicate if the request was successful**Return type** *bool***username**

Retrieve author username.

Return type *str*

1.2.5 Media

class `trakt.objects.media.Media` (*client, keys=None, index=None*)Bases: *object***get_key** (*service*)**images** = *None***Type** *dict*Images (or *None*), defined as:

```
{
  <type>: {
    <size>: <url>
  }
}
```

Type	Size	Dimensions
banner	full	1000x185 (movie/show), 758x140 (show)
clearart	full	1000x562
fanart	full	1920x1080 (typical), 1280x720
	medium	1280x720
	thumb	853x480
logo	full	800x310
poster	full	1000x1500
	medium	600x900
	thumb	300x450
thumb	full	1000x562 (movie), 500x281 (show)

in_watchlist = *None***Type** *bool*Flag indicating this item is in your watchlist (or *None*)**index** = *None***Type** *int*

Playlist item index

keys = *None***Type** *list of tuple*

Keys (for imdb, tvdb, etc..), defined as:

..code-block:

```
[
    (<service>, <id>)
]
```

last_updated_at = None

Type `datetime`

Timestamp of when this item was last updated (or *None*)

last_watched_at = None

Type `datetime`

Timestamp of when this item was last watched (or *None*)

listed_at = None

Type `datetime`

Timestamp of when this item was added to the list (or *None*)

overview = None

Type `str`

Overview (or *None*)

pk

Retrieve the primary key (unique identifier for the item).

Provides the following identifiers (by media type):

- **movie:** imdb
- **show:** tvdb
- **season:** tvdb
- **episode:** tvdb
- **custom_list:** trakt
- **person:** tmdb

Returns (<service>, <value>) or *None* if no primary key is available

Return type `tuple`

plays = None

Type `int`

Number of plays (or *None*)

rating = None

Type `int`

Community rating (0 - 10) (or *None*)

reset_at = None

Type `datetime`

Timestamp of when this item was reset (or *None*)

score = None

Type `float`

Search score (or *None*)

1.2.6 Movie

class `trakt.objects.movie.Movie` (*client, keys, index=None*)

Bases: `trakt.objects.video.Video`

available_translations = None

Type `list`

Available translations (for title, overview, etc..)

certification = None

Type `str`

Content certification (e.g PG-13)

genres = None

Type `list`

Genres

get_key (*service*)

homepage = None

Type `str`

Homepage URL

language = None

Type `str`

Language (for title, overview, etc..)

pk

Retrieve the primary key (unique identifier for the item).

Provides the following identifiers (by media type):

- **movie:** imdb
- **show:** tvdb
- **season:** tvdb
- **episode:** tvdb
- **custom_list:** trakt
- **person:** tmdb

Returns (<service>, <value>) or *None* if no primary key is available

Return type `tuple`

released = None

Type `date`

Release date

runtime = `None`

Type `int`

Duration (in minutes)

tagline = `None`

Type `str`

Tagline

title = `None`

Type `str`

Title

to_dict()

Dump movie to a dictionary.

Returns Movie dictionary

Return type `dict`

to_identifier()

Return the movie identifier which is compatible with requests that require movie definitions.

Returns Movie identifier/definition

Return type `dict`

to_info(*args, **kwargs)

Dump movie to a dictionary.

Deprecated: use the `to_dict()` method instead.

trailer = `None`

Type `str`

Trailer URL

updated_at = `None`

Type `datetime`

Updated date/time

user_count = `None`

Type `int`

Number of user recommendations (returned by the `Trakt['movies'].recommended()` and `Trakt['shows'].recommended()` methods)

watchers = `None`

Type `int`

Number of active watchers (returned by the `Trakt['movies'].trending()` and `Trakt['shows'].trending()` methods)

year = `None`

Type `int`

Year

1.2.7 Person

class trakt.objects.person.**Person** (*client, keys=None, index=None*)

Bases: `object`

index = `None`

Type `int`

Playlist item index

keys = `None`

Type list of tuple

Keys (for imdb, tvdb, etc..), defined as:

..code-block:

```
[
    (<service>, <id>)
]
```

listed_at = `None`

Type `datetime`

Timestamp of when this item was added to the list (or *None*)

name = `None`

Type `str`

Name

pk

Retrieve the primary key (unique identifier for the item).

Provides the following identifiers (by media type):

- **movie:** imdb
- **show:** tvdb
- **season:** tvdb
- **episode:** tvdb
- **custom_list:** trakt
- **person:** tmdb

Returns (<service>, <value>) or *None* if no primary key is available

Return type tuple

1.2.8 BaseProgress

class trakt.objects.progress.**BaseProgress** (*aired=None, completed=None*)

Bases: `object`

aired = `None`

```

        Type int
        Number of aired episodes
completed = None
        Type int
        Number of completed episodes
to_dict()
class trakt.objects.progress.CollectionProgress (client, aired=None, completed=None)
    Bases: trakt.objects.progress.Progress
    progress_type = 'collection'
to_dict()
    Dump progress to a dictionary.
    Returns Progress dictionary
    Return type dict
class trakt.objects.progress.EpisodeProgress (pk=None)
    Bases: object
completed = None
        Type bool
        Whether or not the episode has been watched or collected
pk = None
        Type int
        Episode Number
progress_timestamp = None
        Type datetime
        Date/time episode was collected or last watched
to_dict()
class trakt.objects.progress.Progress (client, aired=None, completed=None)
    Bases: trakt.objects.progress.BaseProgress
hidden_seasons = None
        Type dict
        Hidden Seasons, defined as {season_num: Season}
last_episode = None
        Type trakt.objects.episode.Episode
        Last Episode the user watched or collected
last_progress_change = None
        Type datetime
        Last watched or collected date/time
next_episode = None

```

Type `trakt.objects.episode.Episode`

Next Episode the user should watch or collect

progress_type = `None`

Type `str`

Progress Type (watched or collection)

reset_at = `None`

Type `datetime`

Reset date/time (not applicable for collected progress)

seasons = `None`

Type `dict`

Season Progress, defined as {season_num: SeasonProgress}

to_dict ()

Dump progress to a dictionary.

Returns Progress dictionary

Return type `dict`

class `trakt.objects.progress.SeasonProgress` (*pk=None, aired=None, completed=None*)

Bases: `trakt.objects.progress.BaseProgress`

episodes = `None`

Type `dict`

Episode Progress, defined as {episode_num: EpisodeProgress}

pk = `None`

Type `int`

Season Number

to_dict ()

class `trakt.objects.progress.WatchedProgress` (*client, aired=None, completed=None*)

Bases: `trakt.objects.progress.Progress`

progress_type = `'watched'`

to_dict ()

Dump progress to a dictionary.

Returns Progress dictionary

Return type `dict`

1.2.9 Rating

class `trakt.objects.rating.Rating` (*client, value=None, timestamp=None, votes=None*)

Bases: `object`

timestamp = `None`

Type `datetime`

Rating timestamp

value = None
 Type `int`
 Rating value (0 - 10)

votes = None
 Type `int`
 Number of votes

1.2.10 Season

class `trakt.objects.season.Season` (*client*, *keys=None*, *index=None*)

Bases: `trakt.objects.media.Media`

aired_episodes = None

Type `int`
 Aired episode count

episode_count = None

Type `int`
 Total episode count

episodes = None

Type `dict`
 Episodes, defined as {episode_num: Episode}
Note: this field might not be available with some methods

first_aired = None

Type `datetime`
 First air date

get_key (*service*)

pk
 Retrieve the primary key (unique identifier for the item).

Provides the following identifiers (by media type):

- **movie:** imdb
- **show:** tvdb
- **season:** tvdb
- **episode:** tvdb
- **custom_list:** trakt
- **person:** tmdb

Returns (<service>, <value>) or None if no primary key is available

Return type `tuple`

show = None

Type `trakt.objects.show.Show`

Show

`to_dict()`

Dump season to a dictionary.

Returns Season dictionary

Return type `dict`

`to_identifier()`

Return the season identifier which is compatible with requests that require season definitions.

Returns Season identifier/definition

Return type `dict`

`to_info(*args, **kwargs)`

Dump season to a dictionary.

Deprecated: use the `to_dict()` method instead.

1.2.11 Show

class `trakt.objects.show.Show(client, keys, index=None)`

Bases: `trakt.objects.media.Media`

aired_episodes = `None`

Type `int`

Aired episode count

airs = `None`

Type `dict`

Dictionary with day, time and timezone in which the show airs

available_translations = `None`

Type `list`

Available translations (for title, overview, etc..)

certification = `None`

Type `str`

Content certification (e.g TV-MA)

country = `None`

Type `str`

Country in which the show is aired

episodes()

Return a flat episode iterator.

Returns Iterator ((season_num, episode_num), Episode)

Return type iterator

first_aired = `None`

Type `datetime`

First air date

genres = None

Type list

Genres

get_key (*service*)

homepage = None

Type str

Homepage URL

language = None

Type str

Language (for title, overview, etc..)

network = None

Type str

Network in which the show is aired

pk

Retrieve the primary key (unique identifier for the item).

Provides the following identifiers (by media type):

- **movie:** imdb
- **show:** tvdb
- **season:** tvdb
- **episode:** tvdb
- **custom_list:** trakt
- **person:** tmdb

Returns (<service>, <value>) or None if no primary key is available

Return type tuple

runtime = None

Type int

Duration (in minutes)

seasons = None

Type dict

Seasons, defined as {season_num: Season}

Note: this field might not be available with some methods

status = None

Type str

Value of returning series (airing right now), in production (airing soon), planned (in development), canceled, or ended

title = None
 Type `str`
 Title

to_dict()
 Dump show to a dictionary.
 Returns Show dictionary
 Return type `dict`

to_identifier()
 Return the show identifier which is compatible with requests that require show definitions.
 Returns Show identifier/definition
 Return type `dict`

to_info(*args, **kwargs)
 Dump show to a dictionary.
 Deprecated: use the `to_dict()` method instead.

updated_at = None
 Type `datetime`
 Updated date/time

user_count = None
 Type `int`
 Number of user recommendations (returned by the `Trakt['movies'].recommended()` and `Trakt['shows'].recommended()` methods)

watchers = None
 Type `int`
 Number of active watchers (returned by the `Trakt['movies'].trending()` and `Trakt['shows'].trending()` methods)

year = None
 Type `int`
 Year

1.2.12 User

```
class trakt.objects.user.User(client, keys)  
    Bases: object  
    follow(**kwargs)  
    followed_at = None  
        Type datetime  
        Timestamp of when this user was followed  
    following(**kwargs)  
    friends(**kwargs)
```

friends_at = None

Type `datetime`

Timestamp of when this user was friended

history (***kwargs*)

id

Retrieve the user identifier.

Return type `int`

keys = None

Type `list of tuple`

Keys (for trakt, imdb, tvdb, etc..), defined as:

..code-block:

```
[
    (<service>, <id>)
]
```

name = None

Type `str`

Name

pk

Retrieve the primary key (unique identifier for the user).

Returns ("trakt", <id>) or None if no primary key is available

Return type `tuple`

private = None

Type `bool`

User profile is private

ratings (***kwargs*)

unfollow (***kwargs*)

username = None

Type `str`

Username

vip = None

Type `bool`

User has VIP

vip_ep = None

Type `bool`

User has VIP Executive Producer Credit

watchlist (***kwargs*)

1.2.13 Video

class `trakt.objects.video.Video` (*client*, *keys=None*, *index=None*)

Bases: `trakt.objects.media.Media`

action = `None`

Type `str`

Item action (e.g. history action: “checkin”, “scrobble” or “watch”)

collected_at = `None`

Type `datetime`

Timestamp of when this item was added to your collection (or *None*)

get_key (*service*)

id = `None`

Type `long`

Item id (e.g. history id)

is_collected = `None`

Type `bool`

Flag indicating this item has been collected (or *None*)

is_watched = `None`

Type `bool`

Flag indicating this item has been watched (or *None*)

paused_at = `None`

Type `datetime`

Timestamp of when this item was paused (or *None*)

pk

Retrieve the primary key (unique identifier for the item).

Provides the following identifiers (by media type):

- **movie:** imdb
- **show:** tvdb
- **season:** tvdb
- **episode:** tvdb
- **custom_list:** trakt
- **person:** tmdb

Returns (<service>, <value>) or *None* if no primary key is available

Return type `tuple`

progress = `None`

Type `float`

Playback progress for item (or *None*)

```
watched_at = None
```

Type `datetime`

Timestamp of when this item was watched (or *None*)

1.3 Modules

1.3.1 trakt

```
class trakt.Trakt
    Bases: object

    client = <trakt.client.TraktClient object>

    classmethod construct ()

exception trakt.RequestError (response)
    Bases: exceptions.Exception

exception trakt.ClientError (response)
    Bases: trakt.core.exceptions.RequestError

exception trakt.ServerError (response)
    Bases: trakt.core.exceptions.RequestError
```

1.3.2 trakt.client

```
class trakt.client.TraktClient (adapter_kwargs=None)
    Bases: trakt.core.emitter.Emitter

    base_url = 'https://api.trakt.tv'

    construct (media, item, keys=None, **kwargs)

    static get_ids (media, item, parent=None)

    site_url

    version = '4.4.0'
```

1.3.3 trakt.core

1.3.4 trakt.core.configuration

```
class trakt.core.configuration.Configuration (manager)
    Bases: object

    app (name=None, version=None, date=None, id=None)

    auth (login=None, token=None)

    client (id=None, secret=None)

    get (key, default=None)

    http (retry=False, max_retries=3, retry_sleep=5, timeout=(6.05, 24))
```

```
class trakt.core.configuration.ConfigurationManager
    Bases: object

    app (name=None, version=None, date=None, id=None)

    auth (login=None, token=None)

    client (id=None, secret=None)

    current

    get (key, default=None)

    http (retry=False, max_retries=3, retry_sleep=5, timeout=(6.05, 24))

class trakt.core.configuration.OAuthConfiguration (owner)
    Bases: object

    clear ()

    from_response (response=None, refresh=None, username=None)
```

1.3.5 trakt.core.context_collection

```
class trakt.core.context_collection.ContextCollection (base=None)
    Bases: object

    append (value)

    build (*args, **kwargs)

    clear (*args, **kwargs)

    current

    pop (index=None)

class trakt.core.context_collection.ListCollection (*lists)
    Bases: object

    append (*args, **kwargs)

    find_list (*args, **kwargs)

    lists (*args, **kwargs)

    pop (*args, **kwargs)
```

1.3.6 trakt.core.context_stack

```
class trakt.core.context_stack.Context (**kwargs)
    Bases: object

class trakt.core.context_stack.ContextStack
    Bases: object

    pop ()

    push (**kwargs)
```

1.3.7 trakt.core.emitter

```

class trakt.core.emitter.Emitter
    Bases: object

    emit (event, *args, **kwargs)

    emit_on (event, *args, **kwargs)

    off (event=None, func=None)

    on (events, func=None, on_bound=None)

    once (event, func=None)

    pipe (events, other)

    threading = False

    threading_workers = 2

class trakt.core.emitter.PipeHandler (event, callback)
    Bases: object

trakt.core.emitter.emit (emitter, event, *args, **kwargs)

trakt.core.emitter.off (emitter, event, func=None)

trakt.core.emitter.on (emitter, event, func=None)

trakt.core.emitter.once (emitter, event, func=None)

trakt.core.emitter.repr_trim (value, length=1000)

```

1.3.8 trakt.core.errors

```

trakt.core.errors.log_request_error (logger, response)

```

1.3.9 trakt.core.exceptions

```

exception trakt.core.exceptions.ClientError (response)
    Bases: trakt.core.exceptions.RequestError

exception trakt.core.exceptions.RequestError (response)
    Bases: exceptions.Exception

exception trakt.core.exceptions.RequestFailedError
    Bases: exceptions.Exception

exception trakt.core.exceptions.ServerError (response)
    Bases: trakt.core.exceptions.RequestError

```

1.3.10 trakt.core.helpers

```

trakt.core.helpers.clean_username (username)

trakt.core.helpers.deprecated (message)

trakt.core.helpers.dictfilter (d, **kwargs)

trakt.core.helpers.from_iso8601 (self, *args, **kwargs)

```

```
trakt.core.helpers.from_iso8601_date(value)
trakt.core.helpers.from_iso8601_datetime(value)
trakt.core.helpers.synchronized(f_lock, mode='full')
trakt.core.helpers.to_iso8601(self, *args, **kwargs)
trakt.core.helpers.to_iso8601_date(value)
trakt.core.helpers.to_iso8601_datetime(value)
trakt.core.helpers.try_convert(value, value_type, default=None)
```

1.3.11 trakt.core.http

```
class trakt.core.http.HTTPSAdapter(ssl_version=None, *args, **kwargs)
    Bases: requests.adapters.HTTPAdapter
```

```
    init_poolmanager(connections, maxsize, block=False, **pool_kwargs)
        Initializes a urllib3 PoolManager.
```

This method should not be called from user code, and is only exposed for use when subclassing the HTTPAdapter.

Parameters

- **connections** – The number of urllib3 connection pools to cache.
- **maxsize** – The maximum number of connections to save in the pool.
- **block** – Block when no free connections are available.
- **pool_kwargs** – Extra keyword arguments used to initialize the Pool Manager.

```
class trakt.core.http.HttpClient(client, adapter_kwargs=None, keep_alive=True)
    Bases: object
```

```
    configure(path=None)
```

```
    delete(path=None, params=None, data=None, **kwargs)
```

```
    get(path=None, params=None, data=None, **kwargs)
```

```
    post(path=None, params=None, data=None, **kwargs)
```

```
    proxies
```

```
    put(path=None, params=None, data=None, **kwargs)
```

```
    rebuild()
```

```
    request(method, path=None, params=None, data=None, query=None, authenticated=False, validate_token=True, exceptions=False, pagination=False, **kwargs)
```

```
    send(request)
```

```
    ssl_version
```

```
    validate()
```

1.3.12 trakt.core.pagination

```
class trakt.core.pagination.PaginationIterator (client, request, exceptions=False)
    Bases: object

    get (page)

    resolve ()

    with_mapper (mapper)
```

1.3.13 trakt.core.request

```
class trakt.core.request.TraktRequest (client, **kwargs)
    Bases: object

    construct_url ()
        Construct a full trakt request URI, with params and query.

    classmethod encode_query (parameters)

    classmethod encode_query_parameter (value)

    prepare ()

    transform_data ()

    transform_headers ()

    transform_method ()

    transform_parameters ()
```

1.3.14 trakt.helpers

```
trakt.helpers.build_url (*args, **kwargs)

trakt.helpers.has_attribute (obj, name)

trakt.helpers.setdefault (d, defaults, func=None)
```

1.3.15 trakt.interfaces.base

```
class trakt.interfaces.base.Interface (client)
    Bases: object

    get_data (response, exceptions=False, parse=True)

    http

    path = None

class trakt.interfaces.base.InterfaceProxy (interface, args)
    Bases: object

    trakt.interfaces.base.application (func)

    trakt.interfaces.base.authenticated (func)
```

1.3.16 `trakt.interfaces.sync.core`

1.3.17 `trakt.interfaces.sync.core.mixins`

```
class trakt.interfaces.sync.core.mixins.Add(client)
    Bases: trakt.interfaces.base.Interface

    add (**kwargs)

class trakt.interfaces.sync.core.mixins.Delete(client)
    Bases: trakt.interfaces.base.Interface

    delete (**kwargs)

class trakt.interfaces.sync.core.mixins.Get(client)
    Bases: trakt.interfaces.base.Interface

    flags = {}

    get (**kwargs)

    movies (**kwargs)

    shows (**kwargs)

class trakt.interfaces.sync.core.mixins.Remove(client)
    Bases: trakt.interfaces.base.Interface

    remove (**kwargs)
```

1.3.18 `trakt.mapper`

```
class trakt.mapper.CommentMapper
    Bases: trakt.mapper.core.base.Mapper

    classmethod comment (client, item, **kwargs)

class trakt.mapper.ListMapper
    Bases: trakt.mapper.core.base.Mapper

    classmethod custom_list (client, item, username=None, **kwargs)

    classmethod public_list (client, item, **kwargs)

    classmethod public_lists (client, items, **kwargs)

class trakt.mapper.ListItemMapper
    Bases: trakt.mapper.core.base.Mapper

    classmethod episode (client, item, **kwargs)

    classmethod episodes (client, items, **kwargs)

    classmethod list (client, item, **kwargs)

    classmethod movie (client, item, **kwargs)

    classmethod officiallist (client, item, **kwargs)

    classmethod person (client, item, **kwargs)

    classmethod process (client, item, media=None, **kwargs)

    classmethod process_many (client, items, **kwargs)
```

```

    classmethod season (client, item, **kwargs)

    classmethod seasons (client, items, **kwargs)

    classmethod show (client, item, **kwargs)

class trakt.mapper.ProgressMapper
    Bases: trakt.mapper.core.base.Mapper

    classmethod progress (client, progress_type, item, **kwargs)

class trakt.mapper.SearchMapper
    Bases: trakt.mapper.core.base.Mapper

    classmethod episode (client, item, **kwargs)

    classmethod episodes (client, items, **kwargs)

    classmethod list (client, item, **kwargs)

    classmethod movie (client, item, **kwargs)

    classmethod officiallist (client, item, **kwargs)

    classmethod person (client, item, **kwargs)

    classmethod process (client, item, media=None, **kwargs)

    classmethod process_many (client, items, **kwargs)

    classmethod show (client, item, **kwargs)

class trakt.mapper.SummaryMapper
    Bases: trakt.mapper.core.base.Mapper

    classmethod episode (client, item, parse_show=False, **kwargs)

    classmethod episodes (client, items, **kwargs)

    classmethod movie (client, item, **kwargs)

    classmethod movies (client, items, **kwargs)

    classmethod season (client, item, **kwargs)

    classmethod season_episode (client, season, episode_num, item=None, **kwargs)

    classmethod seasons (client, items, **kwargs)

    classmethod show (client, item, **kwargs)

    classmethod shows (client, items, **kwargs)

class trakt.mapper.SyncMapper
    Bases: trakt.mapper.core.base.Mapper

    classmethod episode (client, store, item, append=False, **kwargs)

    classmethod episodes (client, store, items, **kwargs)

    classmethod item (client, store, item, media=None, **kwargs)

    classmethod iterate_items (client, store, items, func, media=None, **kwargs)

    classmethod map_item (client, store, item, media, key=None, parent=None, append=False,
                          **kwargs)

    classmethod map_items (client, store, items, func, **kwargs)

    classmethod movie (client, store, item, **kwargs)

```

```
classmethod movies (client, store, items, **kwargs)
classmethod process (client, store, items, media=None, flat=False, **kwargs)
classmethod season (client, store, item, **kwargs)
classmethod seasons (client, store, items, **kwargs)
classmethod show (client, store, item, append=False, **kwargs)
classmethod show_episode (client, season, episode_num, item=None, **kwargs)
classmethod show_season (client, show, season_num, item=None, **kwargs)
classmethod shows (client, store, items, **kwargs)
class trakt.mapper.UserMapper
  Bases: trakt.mapper.core.base.Mapper
  classmethod user (client, item, **kwargs)
  classmethod users (client, items, **kwargs)
```

1.3.19 trakt.mapper.comment

```
class trakt.mapper.comment.CommentMapper
  Bases: trakt.mapper.core.base.Mapper
  classmethod comment (client, item, **kwargs)
```

1.3.20 trakt.mapper.core

1.3.21 trakt.mapper.core.base

```
class trakt.mapper.core.base.Mapper
  Bases: object
  classmethod construct (client, media, item, keys=None, **kwargs)
  static get_ids (media, item, parent=None)
```

1.3.22 trakt.mapper.list

```
class trakt.mapper.list.ListMapper
  Bases: trakt.mapper.core.base.Mapper
  classmethod custom_list (client, item, username=None, **kwargs)
  classmethod public_list (client, item, **kwargs)
  classmethod public_lists (client, items, **kwargs)
```

1.3.23 trakt.mapper.list_item

```
class trakt.mapper.list_item.ListItemMapper
  Bases: trakt.mapper.core.base.Mapper
  classmethod episode (client, item, **kwargs)
```

```

classmethod episodes (client, items, **kwargs)
classmethod list (client, item, **kwargs)
classmethod movie (client, item, **kwargs)
classmethod officiallist (client, item, **kwargs)
classmethod person (client, item, **kwargs)
classmethod process (client, item, media=None, **kwargs)
classmethod process_many (client, items, **kwargs)
classmethod season (client, item, **kwargs)
classmethod seasons (client, items, **kwargs)
classmethod show (client, item, **kwargs)

```

1.3.24 `trakt.mapper.progress`

```

class trakt.mapper.progress.ProgressMapper
    Bases: trakt.mapper.core.base.Mapper
    classmethod progress (client, progress_type, item, **kwargs)

```

1.3.25 `trakt.mapper.search`

```

class trakt.mapper.search.SearchMapper
    Bases: trakt.mapper.core.base.Mapper
    classmethod episode (client, item, **kwargs)
    classmethod episodes (client, items, **kwargs)
    classmethod list (client, item, **kwargs)
    classmethod movie (client, item, **kwargs)
    classmethod officiallist (client, item, **kwargs)
    classmethod person (client, item, **kwargs)
    classmethod process (client, item, media=None, **kwargs)
    classmethod process_many (client, items, **kwargs)
    classmethod show (client, item, **kwargs)

```

1.3.26 `trakt.mapper.summary`

```

class trakt.mapper.summary.SummaryMapper
    Bases: trakt.mapper.core.base.Mapper
    classmethod episode (client, item, parse_show=False, **kwargs)
    classmethod episodes (client, items, **kwargs)
    classmethod movie (client, item, **kwargs)
    classmethod movies (client, items, **kwargs)

```

```
classmethod season (client, item, **kwargs)  
classmethod season_episode (client, season, episode_num, item=None, **kwargs)  
classmethod seasons (client, items, **kwargs)  
classmethod show (client, item, **kwargs)  
classmethod shows (client, items, **kwargs)
```

1.3.27 `trakt.mapper.sync`

```
class trakt.mapper.sync.SyncMapper  
    Bases: trakt.mapper.core.base.Mapper  
  
    classmethod episode (client, store, item, append=False, **kwargs)  
    classmethod episodes (client, store, items, **kwargs)  
    classmethod item (client, store, item, media=None, **kwargs)  
    classmethod iterate_items (client, store, items, func, media=None, **kwargs)  
    classmethod map_item (client, store, item, media, key=None, parent=None, append=False, **kwargs)  
    classmethod map_items (client, store, items, func, **kwargs)  
    classmethod movie (client, store, item, **kwargs)  
    classmethod movies (client, store, items, **kwargs)  
    classmethod process (client, store, items, media=None, flat=False, **kwargs)  
    classmethod season (client, store, item, **kwargs)  
    classmethod seasons (client, store, items, **kwargs)  
    classmethod show (client, store, item, append=False, **kwargs)  
    classmethod show_episode (client, season, episode_num, item=None, **kwargs)  
    classmethod show_season (client, show, season_num, item=None, **kwargs)  
    classmethod shows (client, store, items, **kwargs)
```

1.3.28 `trakt.mapper.user`

```
class trakt.mapper.user.UserMapper  
    Bases: trakt.mapper.core.base.Mapper  
  
    classmethod user (client, item, **kwargs)  
    classmethod users (client, items, **kwargs)
```

1.3.29 `trakt.objects.core`

1.3.30 `trakt.objects.core.helpers`

```
trakt.objects.core.helpers.update_attributes (obj, dictionary, keys)
```

2.1 Changelog

2.1.1 4.4.0 (2021-03-19)

Added

- **Interfaces:**
 - `Trakt['movies'].recommended()`
 - `Trakt['shows'].recommended()`

2.1.2 4.3.0 (2021-03-19)

Added

- **Interfaces:**
 - `Trakt['movies'].popular()`
 - `Trakt['shows'].popular()`

Changed

- Updated “arrow” requirement to support v1.0.0 (#91)
- Updated “requests” requirement to exclude v2.16.0 (#88)

2.1.3 4.2.0 (2020-06-29)

Added

- **Interfaces:**
 - `Trakt['lists'].popular(), trending()`

- Trakt['users/*'] get(), follow(), unfollow()
- Trakt['users/*/watched'] get(), movies(), shows()

- **Objects:**

- PublicList

- **Properties:**

- Media
 - * plays
 - * last_updated_at
 - * last_watched_at
 - * reset_at

2.1.4 4.1.0 (2020-05-28)

Added

- Documentation is now generated for a number of modules that were previously missing

- **Interfaces:**

- Trakt['sync/watchlist']
 - * Now supports the sort parameter
 - Trakt['users/*/following']
 - Trakt['users/*/friends']
 - Trakt['users/*/history']
 - Trakt['users/*/ratings']
 - Trakt['users/*/watchlist']

- **Methods:**

- Trakt['sync/history'] seasons(), episodes()
 - Trakt['sync/ratings'] all()

- **Objects:**

- User

Changed

- **Methods:**

- Trakt['users/*/lists'] create() now cleans the provided username parameter.

2.1.5 4.0.0 (2020-05-15)

BREAKING

- **pagination=True** now always returns a **PaginationIterator** with:

- total_items - Total number of items
 - total_pages - Total number of pages

- `get (page)` - Fetch the specified page
- `__iter__` - Iterate over all items, automatically requesting the next page as required
- `Trakt['search'].lookup()` now always returns a list of matched items (*previously a list was only returned when more than one item was returned*)

Added

- Pages can now be requested without using `pagination=True` by using the `page` and `per_page` parameters on:

- `Trakt['movies'].trending()`
- `Trakt['search'].lookup(), query()`
- `Trakt['shows'].trending()`
- `Trakt['sync/history'].get(), movies(), shows()`
- `Trakt['sync/ratings'].get(), movies(), shows(), seasons(), episodes()`
- `Trakt['sync/watchlist'].get(), seasons(), episodes()`
- `Trakt['users'].likes()`
- `Trakt['users/*/lists/*'].items()`

- `Trakt['users/*/lists/*'].items()` now supports a `media` parameter (#76)

2.1.6 3.2.0 (2020-03-17)**Added**

- Exposed episode runtime (#69)
- Support extended parameter on `list items()` function (#71)

2.1.7 3.1.0 (2019-10-22)**Added**

- Support for absolute episode numbers (#63)
- Support for rating votes (#68)

Changed

- Dropped support for Python 3.4

2.1.8 3.0.0 (2018-10-26)**Added**

- Implemented the show progress interface (#60)
- Support for the `extended` parameter on sync methods (#57)

Changed

- Requests that raise exceptions are now retried (with `retry=True`)
- Dropped support for Python 2.6, 3.3

Fixed

- Issue disabling pagination on watchlist requests (#62)

2.1.9 2.14.1 (2017-03-07)

Added

- `__future__` imports to `trakt/hooks.py`

2.1.10 2.14.0 (2017-03-07)

Added

- Support for Python 3.6
- Package version is now automatically written into the `trakt.version` module on `python setup.py egg_info`
- `__future__` imports for more consistent Python 3 compatibility
- `SummaryMapper.episode` method now accepts the “`parse_show`” parameter
- Interfaces:
 - `Trakt['calendars']` (#56)

Changed

- Dropped support for Python 3.2
- Cleaned up documentation
- Switched to `pbr` setup configuration
- Ordering of import statements has been updated to be more consistent

Tests

- Replaced usage of `responses` in tests with `httmock`
- Updated `tox.ini` with additional test environments and `flake8` plugins
- Improved the `fixtures` directory structure

Travis CI

- Switched to using “`tox-travis`”, instead of running `pytest` directly
- Releases are now automatically uploaded to PyPI and GitHub Releases

2.1.11 2.13.0 (2017-02-16)

Added

- `Trakt['sync/playback'].delete(<id>)` method (#54)

2.1.12 2.12.0 (2017-01-11)

Added

- Improved token refreshing (*and added the “`oauth.refresh`” and “`oauth.refresh.rejected`” events*)
- `RequestFailedError` exception will now be raised if no response was returned (if `exceptions=True`)
- `Trakt.http.keep_alive` property (*defaults to `:code:'True'`*)
- `Trakt.http.ssl_version` property (*defaults to `:code:'None'` / `:code:'PROTOCOL_TLS'` / `:code:'PROTOCOL_SSLv23'`*)

Changed

- Switched default API endpoint to <https://api.trakt.tv>
- SSL protocol version is now automatically negotiated with the server (*instead of defaulting to TLS v1.0*)
- Warning will now be displayed if a deadlock is detected inside token refresh events
- Fixed some inconsistencies in the handling of error responses
- Updated bundled emitter module (`fuzeman/PyEmitter@3c558c7c2bc3ae07cb1e8e18b2c1c16be042c748`)
- Interfaces:
 - `Trakt['search']`
 - * Updated to use the new search endpoints
- Methods:
 - `Trakt['search'].lookup`
 - * Now supports the `extended` parameter
 - `Trakt['search'].query`
 - * Now supports the `fields` and `extended` parameters

Fixed

- `@authenticated` decorator wasn't applied to some methods, resulting in tokens not being refreshed
- Exception raised when `Trakt['scrobble']` methods are provided “`app_version`” or “`app_date`” parameters
- Error responses weren't being returned correctly with `parse=False`
- Issue handling `None` responses in the automatic token refresher
- Inconsistent handling of error responses in some methods

2.1.13 2.11.0 (2016-12-20)

Added

- Properties
 - `Movie`
 - * `tagline`
 - * `released`
 - * `runtime`
 - * `certification`

```
* updated_at
* homepage
* trailer
* language
* available_translations
* genres
- Show
  * first_aired
  * airs
  * runtime
  * certification
  * network
  * country
  * updated_at
  * status
  * homepage
  * language
  * available_translations
  * genres
  * aired_episodes
- Season
  * first_aired
  * episode_count
  * aired_episodes
- Episode
  * first_aired
  * updated_at
  * available_translations
```

Changed

- Methods on the `Trakt [ 'movies' ]` and `Trakt [ 'shows' ]` interfaces now support the extended parameter (#51)
- Minor improvements to property descriptions on the `Show` and `Episode` objects

2.1.14 2.10.1 (2016-12-15)**Changed**

- Removed some stray files from the source distribution

2.1.15 2.10.0 (2016-12-15)

Added

- Methods:
 - `Trakt['shows'].next_episode (#50)`
 - `Trakt['shows'].last_episode (#50)`

2.1.16 2.9.0 (2016-10-16)

Added

- Properties:
 - `Person.listed_at`
 - `Video.action`
- Methods:
 - `Trakt['sync/history'].get`
 - `Trakt['sync/history'].shows`
 - `Trakt['sync/history'].movies`

Changed

- Updated `SyncMapper` to support flat iterators
- Methods:
 - `Trakt['sync/ratings'].get`
 - * Flat iterator will now be returned if no `media` parameter is provided
 - `Trakt['sync/watchlist'].get`
 - * Pagination is now supported, can be enabled with `pagination=True`
 - * `media` parameter can now be specified as `None` to return all items (with no type filter)

Fixed

- Pagination wouldn't work correctly if a starting page was specified

2.1.17 2.8.0 (2016-09-17)

Added

- Lists containing people are now supported (instead of raising an exception)
- `SyncMapper` can now be used without the `store` parameter
- Objects:
 - `Person`
- Properties:
 - `Video.id` (history id)
 - `Video.watched_at` (history timestamp)

- Methods:
 - `Media.get_key(<service>)`

2.1.18 2.7.1 (2016-08-30)

Fixed

- Invalid classifier was defined in [setup.py]

2.1.19 2.7.0 (2016-08-30)

Added

- Support for multiple media options on the `Trakt['search'].query()` method
- Implemented the media parameter on the `Trakt['search'].lookup()` method

Changed

- “Request failed” warnings now display the request method and path to help with debugging
- Improved handling of requirements in [setup.py]

2.1.20 2.6.1 (2016-05-19)

Changed

- Updated request error messages

Fixed

- Authorization tokens generated with device authentication wouldn't refresh correctly

2.1.21 2.6.0 (2016-04-15)

Added

- `Trakt['oauth/device']` (see `examples/authentication/device.py` for usage details)
- `Trakt['shows'].seasons()` now supports the `extended="episodes"` parameter
- Pagination can now be enabled with `pagination=True`, **warnings** will be displayed if you ignore pagination responses

Changed

- Moved the `Trakt['oauth'].pin_url()` method to `Trakt['oauth/pin'].url()`, the old method still works but will display a deprecation **warning**

Fixed

- Issue retrieving lists by users with the `.` character in their usernames

2.1.22 2.5.2 (2016-02-19)

Added

- `in_watchlist` property to `Movie`, `Show`, `Season` and `Episode` objects (#45)
- `Trakt.site_url` setter to override automatic detection
- `HttpClient` now supports direct calls (#43)

Changed

- Tests are now included in builds, but are excluded from installations

2.1.23 2.5.1 (2015-09-25)

Fixed

- Issue installing `trakt.py` when “`six`” hasn’t been installed yet

2.1.24 2.5.0 (2015-09-24)

Added

- `Trakt['users'].likes()` method
- `CustomList.items()` method
- `Comment` object
- `Media.index` attribute (list item position/rank)
- Basic documentation generation (#29)
- `NullHandler` to the logger to avoid “No handler found” warnings (#33)

Changed

- ‘`movies`’ and ‘`shows`’ interface methods to support the `exceptions=True` parameter (#32)
- `Interface.get_data()` to only parse the response body if the request is successful (#32)

Fixed

- `TypeError` was raised in `SummaryMapper` if the request failed (#30, #31)
- Constructing “Special” episodes could raise an `AttributeError` (#38, #39)
- `Media._update()` “images” attribute
- Issue serializing `List` objects

2.1.25 2.4.1 (2015-09-12)

Fixed

- Issue where the “`_client`” attribute on objects was being serialized
- Issue installing `trakt.py` when “`arrow`” isn’t available yet

2.1.26 2.4.0 (2015-07-09)

Added

- `trending()` method to `Trakt['shows']` and `Trakt['movies']` interfaces (#23)
- `seasons()` and `episodes()` methods to the `Trakt['sync/watchlist']` interface (#26)
- Custom lists support (`Trakt['users/*/lists']`, `Trakt['users/*/lists/*']`) (#26)
- `__eq__()` method on the `Rating` class
- `proxies` attribute on `Trakt.http`

Changed

- `datetime` objects are now returned offset-aware (**make sure you use offset-aware ‘datetime’ objects when comparing timestamps now**)
- Force requests to use `ssl.PROTOCOL_TLSv1` connections for `https://` (#25)
- Return site url from `Trakt['oauth'].authorize_url()`
- Use season number from parent when one isn't defined in the episode

2.1.27 2.3.0 (2015-04-11)

Changes

- Added support for PIN authentication
- Added automatic OAuth token refreshing (*see “examples/pin.py” for an example*)
- Added `Trakt.configuration.oauth.from_response()` configuration method
- Added tests for the `Trakt['oauth']` interface
- Added tests to ensure authentication headers are being sent
- `Trakt['oauth']` methods now raise an exception if you are missing required configuration parameters
- `Trakt['oauth'].token()` method has been renamed to `Trakt['oauth'].token_exchange()` (*old method is still present for compatibility*)

Fixed

- `Trakt['oauth']` “_url” methods could raise an exception in some cases

2.1.28 2.2.0 (2015-04-02)

Changes

- Added unit tests (with `travis-ci.org` and `coveralls.io` integrations)
- Added `/movies`, `/shows`, `/search` and `/users/settings` interfaces
- Added parent properties (“show”, “season”)
- Added “images”, “overview” and “score” properties to the `Media` class
- Added “last_watched_at” property to movies and episodes
- Updated `/sync/playback` interface (to include type filtering)
- “progress” and “paused_at” properties are now included in `to_dict()`

Fixed

- “year” property could be returned as a string in some cases
- Catch an exception in `trakt.media_mapper`
- Catch a case where `Interface.get_data()` can raise a `KeyError: 'content-type' exception`

2.1.29 2.1.1 (2015-02-06)**Changes**

- Updated to use the new v2 API endpoint (`api-v2launch.trakt.tv`)
- Episode and Movie `to_dict()` method now always returns “plays” as an integer
- Added “http.retry_sleep” and “http.timeout” configuration parameters
- Setup travis/coveralls services

Fixed

- Python 3.x compatibility issues

2.1.30 2.1.0 (2015-02-05)**Changes**

- Added “exceptions” and “parse” parameter to `Interface.get_data()`
- Added additional error messages (502, 504, 520)
- Renamed media object `to_info()` method to `to_identifier()`
- Added new `to_dict()` method which returns a dictionary representation of the media object
- Request retrying (on 5xx errors) can now be enabled with `Trakt.configuration.http(retry=True)`
- `requests/urllib3` now retries requests on connection errors (default: 3 retries)

Fixed

- Thread synchronization issue with `trakt.core.configuration`
- `[/sync] last_activities()` used an incorrect path

2.1.31 2.0.8 (2015-01-06)

- Catch all response errors to avoid issues parsing the returned body

2.1.32 2.0.7 (2015-01-04)

- Handle a case where `[media_mapper]` processes an item with an empty “ids” dict

2.1.33 2.0.6 (2015-01-02)

- Switched to manual interface importing to avoid security restrictions

2.1.34 2.0.5 (2015-01-02)

- Convert all datetime properties to UTC

2.1.35 2.0.4 (2015-01-02)

- Allow for charset definitions in “Content-Type” response header

2.1.36 2.0.3 (2015-01-02)

- Display request failed messages in log (with error name/desc)

2.1.37 2.0.2 (2015-01-02)

- Fixed broken logging message

2.1.38 2.0.1 (2015-01-02)

- Properly handle responses where trakt.tv returns errors without a json body

2.1.39 2.0.0 (2014-12-31)

- Re-designed to support trakt 2.0 (note: this isn’t a drop-in update - interfaces, objects and methods have changed to match the new API)
- Support for OAuth and xAuth authentication methods
- Simple configuration system

2.1.40 0.7.0 (2014-10-24)

- “title” and “year” parameters are now optional on scrobble() and watching() methods
- [movie] Added unseen() method
- [show/episode] Added unseen() method

2.1.41 0.6.1 (2014-07-10)

- Return None if an action fails validation (instead of raising an exception)

2.1.42 0.6.0 (2014-06-23)

- Added Trakt.configure() method
- Rebuild session on socket.gaierror (workaround for urllib error)

2.1.43 0.5.3 (2014-05-10)

- Fixed bugs sending media actions
- Renamed `cancel_watching()` to `cancelwatching()`
- “title” and “year” parameters are now optional on media actions

2.1.44 0.5.2 (2014-04-20)

- [movie] Added `seen()`, `library()` and `unlibrary()` methods
- [movie] Implemented media mapping
- [rate] Added `shows()`, `episodes()` and `movies()` methods
- [show] Added `unlibrary()` method
- [show/episode] Added `library()` and `seen()` methods

2.1.45 0.5.1 (2014-04-19)

- Added `@authenticated` to `MediaInterface.send()`
- Fixed missing imports

2.1.46 0.5.0 (2014-04-18)

- Initial release

2.2 License

The MIT License (MIT)

Copyright (c) 2014 Dean Gardiner

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER 3

Indices

- `genindex`
- `modindex`

t

- trakt, 35
- trakt.client, 35
- trakt.core, 35
 - configuration, 35
 - context_collection, 36
 - context_stack, 36
 - emitter, 37
 - errors, 37
 - exceptions, 37
 - helpers, 37
 - http, 38
 - pagination, 39
 - request, 39
- trakt.helpers, 39
- trakt.interfaces.auth, 1
- trakt.interfaces.base, 39
- trakt.interfaces.calendars, 1
- trakt.interfaces.movies, 2
- trakt.interfaces.oauth, 2
 - device, 3
 - pin, 3
- trakt.interfaces.scrobble, 3
- trakt.interfaces.search, 8
- trakt.interfaces.shows, 10
- trakt.interfaces.sync, 10
 - collection, 11
 - core, 40
 - mixins, 40
 - history, 11
 - playback, 11
 - ratings, 11
 - watched, 12
 - watchlist, 12
- trakt.interfaces.users, 12
 - following, 13
 - friends, 13
 - history, 13
 - lists, 13
 - trakt.interfaces.users.lists.list_, 14
 - ratings, 14
 - settings, 14
 - watchlist, 15
- trakt.mapper, 40
 - comment, 42
 - core, 42
 - base, 42
 - list, 42
 - list_item, 42
 - progress, 43
 - search, 43
 - summary, 43
 - sync, 44
 - user, 44
- trakt.objects.comment, 15
- trakt.objects.core, 44
 - helpers, 44
- trakt.objects.episode, 16
- trakt.objects.list.base, 18
- trakt.objects.list.custom, 20
- trakt.objects.media, 22
- trakt.objects.movie, 24
- trakt.objects.person, 26
- trakt.objects.progress, 26
- trakt.objects.rating, 28
- trakt.objects.season, 29
- trakt.objects.show, 30
- trakt.objects.user, 32
- trakt.objects.video, 34

A

- `action` (*trakt.objects.video.Video* attribute), 34
- `action()` (*trakt.interfaces.scrobble.ScrobbleInterface* method), 3
- `Add` (class in *trakt.interfaces.sync.core.mixins*), 40
- `add()` (*trakt.interfaces.sync.collection.SyncCollectionInterface* method), 11
- `add()` (*trakt.interfaces.sync.core.mixins.Add* method), 40
- `add()` (*trakt.interfaces.sync.history.SyncHistoryInterface* method), 11
- `add()` (*trakt.interfaces.sync.ratings.SyncRatingsInterface* method), 11
- `add()` (*trakt.interfaces.sync.watchlist.SyncWatchlistInterface* method), 12
- `add()` (*trakt.interfaces.users.lists.list_UsersListInterface* method), 14
- `add()` (*trakt.objects.list.custom.CustomList* method), 20
- `aired` (*trakt.objects.progress.BaseProgress* attribute), 26
- `aired_episodes` (*trakt.objects.season.Season* attribute), 29
- `aired_episodes` (*trakt.objects.show.Show* attribute), 30
- `airs` (*trakt.objects.show.Show* attribute), 30
- `all()` (*trakt.interfaces.sync.ratings.SyncRatingsInterface* method), 11
- `all()` (*trakt.interfaces.users.ratings.UsersRatingsInterface* method), 14
- `AllCalendarsInterface` (class in *trakt.interfaces.calendars*), 1
- `allow_comments` (*trakt.objects.list.base.List* attribute), 18
- `app()` (*trakt.core.configuration.Configuration* method), 35
- `app()` (*trakt.core.configuration.ConfigurationManager* method), 36
- `append()` (*trakt.core.context_collection.ContextCollection* method), 36
- `append()` (*trakt.core.context_collection.ListCollection* method), 36
- `application()` (in module *trakt.interfaces.base*), 39
- `auth()` (*trakt.core.configuration.Configuration* method), 35
- `auth()` (*trakt.core.configuration.ConfigurationManager* method), 36
- `authenticated()` (in module *trakt.interfaces.base*), 39
- `AuthInterface` (class in *trakt.interfaces.auth*), 1
- `authorize_url()` (*trakt.interfaces.oauth.OAuthInterface* method), 2
- `available_translations` (*trakt.objects.episode.Episode* attribute), 16
- `available_translations` (*trakt.objects.movie.Movie* attribute), 24
- `available_translations` (*trakt.objects.show.Show* attribute), 30

B

- `base_url` (*trakt.client.TraktClient* attribute), 35
- `BaseProgress` (class in *trakt.objects.progress*), 26
- `build()` (*trakt.core.context_collection.ContextCollection* method), 36
- `build_url()` (in module *trakt.helpers*), 39

C

- `certification` (*trakt.objects.movie.Movie* attribute), 24
- `certification` (*trakt.objects.show.Show* attribute), 30
- `clean_username()` (in module *trakt.core.helpers*), 37
- `clear()` (*trakt.core.configuration.OAuthConfiguration* method), 36
- `clear()` (*trakt.core.context_collection.ContextCollection* method), 36
- `client` (*trakt.Trakt* attribute), 35

client() (trakt.core.configuration.Configuration method), 35
 client() (trakt.core.configuration.ConfigurationManager method), 36
 ClientError, 35, 37
 code() (trakt.interfaces.oauth.device.DeviceOAuthInterface method), 3
 collected_at (trakt.objects.video.Video attribute), 34
 CollectionProgress (class in trakt.objects.progress), 27
 Comment (class in trakt.objects.comment), 15
 comment (trakt.objects.comment.Comment attribute), 15
 comment() (trakt.mapper.comment.CommentMapper class method), 42
 comment() (trakt.mapper.CommentMapper class method), 40
 comment_count (trakt.objects.list.base.List attribute), 18
 CommentMapper (class in trakt.mapper), 40
 CommentMapper (class in trakt.mapper.comment), 42
 completed (trakt.objects.progress.BaseProgress attribute), 27
 completed (trakt.objects.progress.EpisodeProgress attribute), 27
 Configuration (class in trakt.core.configuration), 35
 ConfigurationManager (class in trakt.core.configuration), 35
 configure() (trakt.core.http.HttpClient method), 38
 construct() (trakt.client.TraktClient method), 35
 construct() (trakt.mapper.core.base.Mapper class method), 42
 construct() (trakt.Trakt class method), 35
 construct_url() (trakt.core.request.TraktRequest method), 39
 Context (class in trakt.core.context_stack), 36
 ContextCollection (class in trakt.core.context_collection), 36
 ContextStack (class in trakt.core.context_stack), 36
 country (trakt.objects.show.Show attribute), 30
 create() (trakt.interfaces.users.lists.UsersListsInterface method), 13
 created_at (trakt.objects.comment.Comment attribute), 15
 created_at (trakt.objects.list.base.List attribute), 18
 current (trakt.core.configuration.ConfigurationManager attribute), 36
 current (trakt.core.context_collection.ContextCollection attribute), 36
 custom_list() (trakt.mapper.list.ListMapper class method), 42
 custom_list() (trakt.mapper.ListMapper class method), 40
 CustomList (class in trakt.objects.list.custom), 20
 Delete (class in trakt.interfaces.sync.core.mixins), 40
 delete() (trakt.core.http.HttpClient method), 38
 delete() (trakt.interfaces.sync.core.mixins.Delete method), 40
 delete() (trakt.interfaces.sync.playback.SyncPlaybackInterface method), 11
 delete() (trakt.interfaces.users.lists.list_UsersListInterface method), 14
 delete() (trakt.objects.list.custom.CustomList method), 21
 deprecated() (in module trakt.core.helpers), 37
 description (trakt.objects.list.base.List attribute), 18
 DeviceOAuthInterface (class in trakt.interfaces.oauth.device), 3
 dictfilter() (in module trakt.core.helpers), 37
 display_numbers (trakt.objects.list.base.List attribute), 18
E
 emit() (in module trakt.core.emitter), 37
 emit() (trakt.core.emitter.Emitter method), 37
 emit_on() (trakt.core.emitter.Emitter method), 37
 Emitter (class in trakt.core.emitter), 37
 encode_query() (trakt.core.request.TraktRequest class method), 39
 encode_query_parameter() (trakt.core.request.TraktRequest class method), 39
 Episode (class in trakt.objects.episode), 16
 episode() (trakt.interfaces.shows.ShowsInterface method), 10
 episode() (trakt.mapper.list_item.ListItemMapper class method), 42
 episode() (trakt.mapper.ListItemMapper class method), 40
 episode() (trakt.mapper.search.SearchMapper class method), 43
 episode() (trakt.mapper.SearchMapper class method), 41
 episode() (trakt.mapper.summary.SummaryMapper class method), 43
 episode() (trakt.mapper.SummaryMapper class method), 41
 episode() (trakt.mapper.sync.SyncMapper class method), 44
 episode() (trakt.mapper.SyncMapper class method), 41
 episode_count (trakt.objects.season.Season attribute), 29
 EpisodeProgress (class in trakt.objects.progress), 27

episodes (*trakt.objects.progress.SeasonProgress* attribute), 28
 episodes (*trakt.objects.season.Season* attribute), 29
 episodes () (*trakt.interfaces.sync.history.SyncHistoryInterface* method), 11
 episodes () (*trakt.interfaces.sync.playback.SyncPlaybackInterface* attribute), 13
 episodes () (*trakt.interfaces.sync.playback.SyncPlaybackInterface* method), 11
 episodes () (*trakt.interfaces.sync.ratings.SyncRatingsInterface* attribute), 15
 episodes () (*trakt.interfaces.sync.ratings.SyncRatingsInterface* method), 12
 episodes () (*trakt.interfaces.sync.watchlist.SyncWatchlistInterface* attribute), 12
 episodes () (*trakt.interfaces.sync.watchlist.SyncWatchlistInterface* method), 12
 episodes () (*trakt.interfaces.users.history.UsersHistoryInterface* attribute), 13
 episodes () (*trakt.interfaces.users.history.UsersHistoryInterface* method), 13
 episodes () (*trakt.interfaces.users.ratings.UsersRatingsInterface* attribute), 14
 episodes () (*trakt.interfaces.users.ratings.UsersRatingsInterface* method), 14
 episodes () (*trakt.interfaces.users.watchlist.UsersWatchlistInterface* attribute), 15
 episodes () (*trakt.interfaces.users.watchlist.UsersWatchlistInterface* method), 15
 episodes () (*trakt.mapper.list_item.ListItemMapper* class method), 42
 episodes () (*trakt.mapper.ListItemMapper* class method), 40
 episodes () (*trakt.mapper.search.SearchMapper* class method), 43
 episodes () (*trakt.mapper.SearchMapper* class method), 41
 episodes () (*trakt.mapper.summary.SummaryMapper* class method), 43
 episodes () (*trakt.mapper.SummaryMapper* class method), 41
 episodes () (*trakt.mapper.sync.SyncMapper* class method), 44
 episodes () (*trakt.mapper.SyncMapper* class method), 41
 episodes () (*trakt.objects.show.Show* method), 30

F

find_list () (*trakt.core.context_collection.ListCollection* method), 36
 first_aired (*trakt.objects.episode.Episode* attribute), 16
 first_aired (*trakt.objects.season.Season* attribute), 29
 first_aired (*trakt.objects.show.Show* attribute), 30
 flags (*trakt.interfaces.sync.collection.SyncCollectionInterface* attribute), 11
 flags (*trakt.interfaces.sync.core.mixins.Get* attribute), 40
 flags (*trakt.interfaces.sync.history.SyncHistoryInterface* attribute), 11
 flags (*trakt.interfaces.sync.playback.SyncPlaybackInterface* attribute), 11
 flags (*trakt.interfaces.sync.ratings.SyncRatingsInterface* attribute), 12

flags (*trakt.interfaces.sync.watched.SyncWatchedInterface* attribute), 12
 flags (*trakt.interfaces.sync.watchlist.SyncWatchlistInterface* attribute), 12
 flags (*trakt.interfaces.users.history.UsersHistoryInterface* attribute), 13
 flags (*trakt.interfaces.users.history.UsersHistoryInterface* method), 13
 flags (*trakt.interfaces.users.watchlist.UsersWatchlistInterface* attribute), 15
 follow () (*trakt.objects.user.User* method), 32
 followed_at (*trakt.objects.user.User* attribute), 32
 following () (*trakt.objects.user.User* method), 32
 following () (*trakt.objects.user.User* method), 32
 friends_at (*trakt.objects.user.User* attribute), 32
 from_iso8601 () (in module *trakt.core.helpers*), 37
 from_iso8601_date () (in module *trakt.core.helpers*), 37
 from_iso8601_datetime () (in module *trakt.core.helpers*), 38
 from_response () (*trakt.core.configuration.OAuthConfiguration* method), 36

G

genres (*trakt.objects.movie.Movie* attribute), 24
 genres (*trakt.objects.show.Show* attribute), 31
 Get (class in *trakt.interfaces.sync.core.mixins*), 40
 get () (*trakt.core.configuration.Configuration* method), 35
 get () (*trakt.core.configuration.ConfigurationManager* method), 36
 get () (*trakt.core.http.HttpClient* method), 38
 get () (*trakt.core.pagination.PaginationIterator* method), 39
 get () (*trakt.interfaces.calendars.AllCalendarsInterface* method), 1
 get () (*trakt.interfaces.movies.MoviesInterface* method), 2
 get () (*trakt.interfaces.shows.ShowsInterface* method), 10
 get () (*trakt.interfaces.sync.collection.SyncCollectionInterface* method), 11
 get () (*trakt.interfaces.sync.core.mixins.Get* method), 40
 get () (*trakt.interfaces.sync.history.SyncHistoryInterface* method), 11
 get () (*trakt.interfaces.sync.playback.SyncPlaybackInterface* method), 11
 get () (*trakt.interfaces.sync.ratings.SyncRatingsInterface* method), 12
 get () (*trakt.interfaces.sync.watched.SyncWatchedInterface* method), 12
 get () (*trakt.interfaces.sync.watchlist.SyncWatchlistInterface* method), 12
 get () (*trakt.interfaces.users.following.UsersFollowingInterface* method), 13

[get \(\) \(trakt.interfaces.users.friends.UsersFriendsInterface method\), 13](#)
[get \(\) \(trakt.interfaces.users.history.UsersHistoryInterface method\), 13](#)
[get \(\) \(trakt.interfaces.users.lists.list_UsersListInterface method\), 14](#)
[get \(\) \(trakt.interfaces.users.lists.UsersListsInterface method\), 14](#)
[get \(\) \(trakt.interfaces.users.ratings.UsersRatingsInterface method\), 14](#)
[get \(\) \(trakt.interfaces.users.settings.UsersSettingsInterface method\), 14](#)
[get \(\) \(trakt.interfaces.users.watchlist.UsersWatchlistInterface method\), 15](#)
[get_data \(\) \(trakt.interfaces.base.Interface method\), 39](#)
[get_ids \(\) \(trakt.client.TraktClient static method\), 35](#)
[get_ids \(\) \(trakt.mapper.core.base.Mapper static method\), 42](#)
[get_key \(\) \(trakt.objects.episode.Episode method\), 16](#)
[get_key \(\) \(trakt.objects.media.Media method\), 22](#)
[get_key \(\) \(trakt.objects.movie.Movie method\), 24](#)
[get_key \(\) \(trakt.objects.season.Season method\), 29](#)
[get_key \(\) \(trakt.objects.show.Show method\), 31](#)
[get_key \(\) \(trakt.objects.video.Video method\), 34](#)

H

[has_attribute \(\) \(in module trakt.helpers\), 39](#)
[hidden_seasons \(trakt.objects.progress.Progress attribute\), 27](#)
[history \(\) \(trakt.objects.user.User method\), 33](#)
[homepage \(trakt.objects.movie.Movie attribute\), 24](#)
[homepage \(trakt.objects.show.Show attribute\), 31](#)
[http \(trakt.interfaces.base.Interface attribute\), 39](#)
[http \(\) \(trakt.core.configuration.Configuration method\), 35](#)
[http \(\) \(trakt.core.configuration.ConfigurationManager method\), 36](#)
[HttpClient \(class in trakt.core.http\), 38](#)
[HTTPSAdapter \(class in trakt.core.http\), 38](#)

I

[id \(trakt.objects.comment.Comment attribute\), 15](#)
[id \(trakt.objects.list.base.List attribute\), 18](#)
[id \(trakt.objects.list.custom.CustomList attribute\), 21](#)
[id \(trakt.objects.user.User attribute\), 33](#)
[id \(trakt.objects.video.Video attribute\), 34](#)
[images \(trakt.objects.media.Media attribute\), 22](#)
[in_watchlist \(trakt.objects.media.Media attribute\), 22](#)
[index \(trakt.objects.media.Media attribute\), 22](#)
[index \(trakt.objects.person.Person attribute\), 26](#)
[init_poolmanager \(\) \(trakt.core.http.HTTPSAdapter method\), 38](#)
[Interface \(class in trakt.interfaces.base\), 39](#)
[InterfaceProxy \(class in trakt.interfaces.base\), 39](#)
[is_collected \(trakt.objects.video.Video attribute\), 34](#)
[is_watched \(trakt.objects.video.Video attribute\), 34](#)
[item \(\) \(trakt.mapper.sync.SyncMapper class method\), 44](#)
[item \(\) \(trakt.mapper.SyncMapper class method\), 41](#)
[item_count \(trakt.objects.list.base.List attribute\), 18](#)
[items \(\) \(trakt.interfaces.users.lists.list_UsersListInterface method\), 14](#)
[items \(\) \(trakt.objects.list.base.List method\), 18](#)
[items \(\) \(trakt.objects.list.custom.CustomList method\), 21](#)
[iterate_items \(\) \(trakt.mapper.sync.SyncMapper class method\), 44](#)
[iterate_items \(\) \(trakt.mapper.SyncMapper class method\), 41](#)

K

[keys \(trakt.objects.comment.Comment attribute\), 15](#)
[keys \(trakt.objects.list.base.List attribute\), 18](#)
[keys \(trakt.objects.media.Media attribute\), 22](#)
[keys \(trakt.objects.person.Person attribute\), 26](#)
[keys \(trakt.objects.user.User attribute\), 33](#)

L

[language \(trakt.objects.movie.Movie attribute\), 24](#)
[language \(trakt.objects.show.Show attribute\), 31](#)
[last_activities \(\) \(trakt.interfaces.sync.SyncInterface method\), 10](#)
[last_episode \(trakt.objects.progress.Progress attribute\), 27](#)
[last_episode \(\) \(trakt.interfaces.shows.ShowsInterface method\), 10](#)
[last_progress_change \(trakt.objects.progress.Progress attribute\), 27](#)
[last_updated_at \(trakt.objects.media.Media attribute\), 23](#)
[last_watched_at \(trakt.objects.media.Media attribute\), 23](#)
[like \(\) \(trakt.interfaces.users.lists.list_UsersListInterface method\), 14](#)
[like \(\) \(trakt.objects.list.base.List method\), 18](#)
[like \(\) \(trakt.objects.list.custom.CustomList method\), 21](#)
[like_count \(trakt.objects.list.base.List attribute\), 19](#)
[like_count \(trakt.objects.list.custom.CustomList attribute\), 21](#)
[liked_at \(trakt.objects.comment.Comment attribute\), 15](#)

liked_at (*trakt.objects.list.base.List attribute*), 19
 likes (*trakt.objects.comment.Comment attribute*), 15
 likes (*trakt.objects.list.base.List attribute*), 19
 likes() (*trakt.interfaces.users.UsersInterface method*), 12
 List (*class in trakt.objects.list.base*), 18
 list() (*trakt.mapper.list_item.ListItemMapper class method*), 43
 list() (*trakt.mapper.ListItemMapper class method*), 40
 list() (*trakt.mapper.search.SearchMapper class method*), 43
 list() (*trakt.mapper.SearchMapper class method*), 41
 ListCollection (*class in trakt.core.context_collection*), 36
 listed_at (*trakt.objects.media.Media attribute*), 23
 listed_at (*trakt.objects.person.Person attribute*), 26
 ListItemMapper (*class in trakt.mapper*), 40
 ListItemMapper (*class in trakt.mapper.list_item*), 42
 ListMapper (*class in trakt.mapper*), 40
 ListMapper (*class in trakt.mapper.list*), 42
 lists() (*trakt.core.context_collection.ListCollection method*), 36
 log_request_error() (*in module trakt.core.errors*), 37
 login() (*trakt.interfaces.auth.AuthInterface method*), 1
 logout() (*trakt.interfaces.auth.AuthInterface method*), 1
 lookup() (*trakt.interfaces.search.SearchInterface method*), 8

M

map_item() (*trakt.mapper.sync.SyncMapper class method*), 44
 map_item() (*trakt.mapper.SyncMapper class method*), 41
 map_items() (*trakt.mapper.sync.SyncMapper class method*), 44
 map_items() (*trakt.mapper.SyncMapper class method*), 41
 Mapper (*class in trakt.mapper.core.base*), 42
 Media (*class in trakt.objects.media*), 22
 Movie (*class in trakt.objects.movie*), 24
 movie() (*trakt.mapper.list_item.ListItemMapper class method*), 43
 movie() (*trakt.mapper.ListItemMapper class method*), 40
 movie() (*trakt.mapper.search.SearchMapper class method*), 43
 movie() (*trakt.mapper.SearchMapper class method*), 41
 movie() (*trakt.mapper.summary.SummaryMapper class method*), 43
 movie() (*trakt.mapper.SummaryMapper class method*), 41
 movie() (*trakt.mapper.sync.SyncMapper class method*), 44
 movies() (*trakt.mapper.SyncMapper class method*), 41
 movies() (*trakt.interfaces.sync.collection.SyncCollectionInterface method*), 11
 movies() (*trakt.interfaces.sync.core.mixins.Get method*), 40
 movies() (*trakt.interfaces.sync.history.SyncHistoryInterface method*), 11
 movies() (*trakt.interfaces.sync.playback.SyncPlaybackInterface method*), 11
 movies() (*trakt.interfaces.sync.ratings.SyncRatingsInterface method*), 12
 movies() (*trakt.interfaces.sync.watched.SyncWatchedInterface method*), 12
 movies() (*trakt.interfaces.sync.watchlist.SyncWatchlistInterface method*), 12
 movies() (*trakt.interfaces.users.history.UsersHistoryInterface method*), 13
 movies() (*trakt.interfaces.users.ratings.UsersRatingsInterface method*), 14
 movies() (*trakt.interfaces.users.watchlist.UsersWatchlistInterface method*), 15
 movies() (*trakt.mapper.summary.SummaryMapper class method*), 43
 movies() (*trakt.mapper.SummaryMapper class method*), 41
 movies() (*trakt.mapper.sync.SyncMapper class method*), 44
 movies() (*trakt.mapper.SyncMapper class method*), 41
 MoviesInterface (*class in trakt.interfaces.movies*), 2

N

name (*trakt.objects.list.base.List attribute*), 19
 name (*trakt.objects.person.Person attribute*), 26
 name (*trakt.objects.user.User attribute*), 33
 network (*trakt.objects.show.Show attribute*), 31
 new() (*trakt.interfaces.calendars.AllCalendarsInterface method*), 2
 next_episode (*trakt.objects.progress.Progress attribute*), 27
 next_episode() (*trakt.interfaces.shows.ShowsInterface method*), 10
 number_abs (*trakt.objects.episode.Episode attribute*), 16

O

OAuthConfiguration (*class in trakt.core.configuration*), 36
 OAuthInterface (*class in trakt.interfaces.oauth*), 2
 off() (*in module trakt.core.emitter*), 37

off() (*trakt.core.emitter.Emitter* method), 37
 officiallist() (*trakt.mapper.list_item.ListItemMapper* class method), 43
 officiallist() (*trakt.mapper.ListItemMapper* class method), 40
 officiallist() (*trakt.mapper.search.SearchMapper* class method), 43
 officiallist() (*trakt.mapper.SearchMapper* class method), 41
 on() (in module *trakt.core.emitter*), 37
 on() (*trakt.core.emitter.Emitter* method), 37
 once() (in module *trakt.core.emitter*), 37
 once() (*trakt.core.emitter.Emitter* method), 37
 overview (*trakt.objects.media.Media* attribute), 23

P

PaginationIterator (class in *trakt.core.pagination*), 39
 parent_id (*trakt.objects.comment.Comment* attribute), 16
 path (*trakt.interfaces.base.Interface* attribute), 39
 pause() (*trakt.interfaces.scrobble.ScrobbleInterface* method), 4
 paused_at (*trakt.objects.video.Video* attribute), 34
 Person (class in *trakt.objects.person*), 26
 person() (*trakt.mapper.list_item.ListItemMapper* class method), 43
 person() (*trakt.mapper.ListItemMapper* class method), 40
 person() (*trakt.mapper.search.SearchMapper* class method), 43
 person() (*trakt.mapper.SearchMapper* class method), 41
 pin_url() (*trakt.interfaces.oauth.OAuthInterface* method), 2
 PinOAuthInterface (class in *trakt.interfaces.oauth.pin*), 3
 pipe() (*trakt.core.emitter.Emitter* method), 37
 PipeHandler (class in *trakt.core.emitter*), 37
 pk (*trakt.objects.comment.Comment* attribute), 16
 pk (*trakt.objects.episode.Episode* attribute), 17
 pk (*trakt.objects.list.base.List* attribute), 19
 pk (*trakt.objects.list.custom.CustomList* attribute), 21
 pk (*trakt.objects.media.Media* attribute), 23
 pk (*trakt.objects.movie.Movie* attribute), 24
 pk (*trakt.objects.person.Person* attribute), 26
 pk (*trakt.objects.progress.EpisodeProgress* attribute), 27
 pk (*trakt.objects.progress.SeasonProgress* attribute), 28
 pk (*trakt.objects.season.Season* attribute), 29
 pk (*trakt.objects.show.Show* attribute), 31
 pk (*trakt.objects.user.User* attribute), 33
 pk (*trakt.objects.video.Video* attribute), 34
 playback() (*trakt.interfaces.sync.SyncInterface* method), 10
 plays (*trakt.objects.media.Media* attribute), 23
 poll() (*trakt.interfaces.oauth.device.DeviceOAuthInterface* method), 3
 pop() (*trakt.core.context_collection.ContextCollection* method), 36
 pop() (*trakt.core.context_collection.ListCollection* method), 36
 pop() (*trakt.core.context_stack.ContextStack* method), 36
 popular() (*trakt.interfaces.movies.MoviesInterface* method), 2
 popular() (*trakt.interfaces.shows.ShowsInterface* method), 10
 post() (*trakt.core.http.HttpClient* method), 38
 premieres() (*trakt.interfaces.calendars.AllCalendarsInterface* method), 2
 in prepare() (*trakt.core.request.TraktRequest* method), 39
 privacy (*trakt.objects.list.base.List* attribute), 19
 private (*trakt.objects.user.User* attribute), 33
 process() (*trakt.mapper.list_item.ListItemMapper* class method), 43
 process() (*trakt.mapper.ListItemMapper* class method), 40
 process() (*trakt.mapper.search.SearchMapper* class method), 43
 process() (*trakt.mapper.SearchMapper* class method), 41
 process() (*trakt.mapper.sync.SyncMapper* class method), 44
 process() (*trakt.mapper.SyncMapper* class method), 42
 process_many() (*trakt.mapper.list_item.ListItemMapper* class method), 43
 process_many() (*trakt.mapper.ListItemMapper* class method), 40
 process_many() (*trakt.mapper.search.SearchMapper* class method), 43
 process_many() (*trakt.mapper.SearchMapper* class method), 41
 Progress (class in *trakt.objects.progress*), 27
 progress (*trakt.objects.video.Video* attribute), 34
 progress() (*trakt.interfaces.shows.ShowsInterface* method), 10
 progress() (*trakt.mapper.progress.ProgressMapper* class method), 43
 progress() (*trakt.mapper.ProgressMapper* class method), 41
 progress_collection() (*trakt.interfaces.shows.ShowsInterface* method), 10
 progress_timestamp (*trakt.objects.progress.EpisodeProgress* attribute), 27

progress_type(*trakt.objects.progress.CollectionProgress* attribute), 27
 progress_type(*trakt.objects.progress.Progress* attribute), 28
 progress_type(*trakt.objects.progress.WatchedProgress* attribute), 28
 progress_watched()
 (*trakt.interfaces.shows.ShowsInterface* method), 10
 ProgressMapper (class in *trakt.mapper*), 41
 ProgressMapper (class in *trakt.mapper.progress*), 43
 proxies(*trakt.core.http.HttpClient* attribute), 38
 public_list() (*trakt.mapper.list.ListMapper* class method), 42
 public_list() (*trakt.mapper.ListMapper* class method), 40
 public_lists() (*trakt.mapper.list.ListMapper* class method), 42
 public_lists() (*trakt.mapper.ListMapper* class method), 40
 push() (*trakt.core.context_stack.ContextStack* method), 36
 put() (*trakt.core.http.HttpClient* method), 38

Q

query() (*trakt.interfaces.search.SearchInterface* method), 9

R

Rating (class in *trakt.objects.rating*), 28
 rating(*trakt.objects.media.Media* attribute), 23
 ratings() (*trakt.objects.user.User* method), 33
 rebuild() (*trakt.core.http.HttpClient* method), 38
 recommended() (*trakt.interfaces.movies.MoviesInterface* method), 2
 recommended() (*trakt.interfaces.shows.ShowsInterface* method), 10
 released(*trakt.objects.movie.Movie* attribute), 24
 Remove (class in *trakt.interfaces.sync.core.mixins*), 40
 remove() (*trakt.interfaces.sync.collection.SyncCollectionInterface* method), 11
 remove() (*trakt.interfaces.sync.core.mixins.Remove* method), 40
 remove() (*trakt.interfaces.sync.history.SyncHistoryInterface* method), 11
 remove() (*trakt.interfaces.sync.ratings.SyncRatingsInterface* method), 12
 remove() (*trakt.interfaces.sync.watchlist.SyncWatchlistInterface* method), 12
 remove() (*trakt.interfaces.users.lists.list_.UsersListInterface* method), 14
 remove() (*trakt.objects.list.custom.CustomList* method), 21
 replies(*trakt.objects.comment.Comment* attribute), 16
 repr_trim() (in module *trakt.core.emitter*), 37
 request() (*trakt.core.http.HttpClient* method), 38
 RequestError, 35, 37
 RequestFailedError, 37
 reset_at(*trakt.objects.media.Media* attribute), 23
 reset_at(*trakt.objects.progress.Progress* attribute), 28
 resolve() (*trakt.core.pagination.PaginationIterator* method), 39
 review(*trakt.objects.comment.Comment* attribute), 16
 runtime(*trakt.objects.episode.Episode* attribute), 17
 runtime(*trakt.objects.movie.Movie* attribute), 25
 runtime(*trakt.objects.show.Show* attribute), 31

S

score(*trakt.objects.media.Media* attribute), 24
 ScrobbleInterface (class in *trakt.interfaces.scrobble*), 3
 SearchInterface (class in *trakt.interfaces.search*), 8
 SearchMapper (class in *trakt.mapper*), 41
 SearchMapper (class in *trakt.mapper.search*), 43
 Season (class in *trakt.objects.season*), 29
 season(*trakt.objects.episode.Episode* attribute), 17
 season() (*trakt.interfaces.shows.ShowsInterface* method), 10
 season() (*trakt.mapper.list_item.ListItemMapper* class method), 43
 season() (*trakt.mapper.ListItemMapper* class method), 40
 season() (*trakt.mapper.summary.SummaryMapper* class method), 43
 season() (*trakt.mapper.SummaryMapper* class method), 41
 season() (*trakt.mapper.sync.SyncMapper* class method), 44
 season() (*trakt.mapper.SyncMapper* class method), 42
 season_episode() (*trakt.mapper.summary.SummaryMapper* class method), 44
 season_episode() (*trakt.mapper.SummaryMapper* class method), 41
 SeasonProgress (class in *trakt.objects.progress*), 28
 seasons(*trakt.objects.progress.Progress* attribute), 28
 seasons(*trakt.objects.show.Show* attribute), 31
 seasons() (*trakt.interfaces.shows.ShowsInterface* method), 10
 seasons() (*trakt.interfaces.sync.history.SyncHistoryInterface* method), 11
 seasons() (*trakt.interfaces.sync.ratings.SyncRatingsInterface* method), 12
 seasons() (*trakt.interfaces.sync.watchlist.SyncWatchlistInterface* method), 12

[seasons \(\) \(trakt.interfaces.users.history.UsersHistoryInterface method\), 13](#)
[seasons \(\) \(trakt.interfaces.users.ratings.UsersRatingsInterface method\), 14](#)
[seasons \(\) \(trakt.interfaces.users.watchlist.UsersWatchlistInterface method\), 15](#)
[seasons \(\) \(trakt.mapper.list_item.ListItemMapper class method\), 43](#)
[seasons \(\) \(trakt.mapper.ListItemMapper class method\), 41](#)
[seasons \(\) \(trakt.mapper.summary.SummaryMapper class method\), 44](#)
[seasons \(\) \(trakt.mapper.SummaryMapper class method\), 41](#)
[seasons \(\) \(trakt.mapper.sync.SyncMapper class method\), 44](#)
[seasons \(\) \(trakt.mapper.SyncMapper class method\), 42](#)
[send \(\) \(trakt.core.http.HttpClient method\), 38](#)
[ServerError, 35, 37](#)
[setdefault \(\) \(in module trakt.helpers\), 39](#)
[Show \(class in trakt.objects.show\), 30](#)
[show \(trakt.objects.episode.Episode attribute\), 17](#)
[show \(trakt.objects.season.Season attribute\), 29](#)
[show \(\) \(trakt.mapper.list_item.ListItemMapper class method\), 43](#)
[show \(\) \(trakt.mapper.ListItemMapper class method\), 41](#)
[show \(\) \(trakt.mapper.search.SearchMapper class method\), 43](#)
[show \(\) \(trakt.mapper.SearchMapper class method\), 41](#)
[show \(\) \(trakt.mapper.summary.SummaryMapper class method\), 44](#)
[show \(\) \(trakt.mapper.SummaryMapper class method\), 41](#)
[show \(\) \(trakt.mapper.sync.SyncMapper class method\), 44](#)
[show \(\) \(trakt.mapper.SyncMapper class method\), 42](#)
[show_episode \(\) \(trakt.mapper.sync.SyncMapper class method\), 44](#)
[show_episode \(\) \(trakt.mapper.SyncMapper class method\), 42](#)
[show_season \(\) \(trakt.mapper.sync.SyncMapper class method\), 44](#)
[show_season \(\) \(trakt.mapper.SyncMapper class method\), 42](#)
[shows \(\) \(trakt.interfaces.sync.collection.SyncCollectionInterface method\), 11](#)
[shows \(\) \(trakt.interfaces.sync.core.mixins.Get method\), 40](#)
[shows \(\) \(trakt.interfaces.sync.history.SyncHistoryInterface method\), 11](#)
[shows \(\) \(trakt.interfaces.sync.playback.SyncPlaybackInterface method\), 11](#)
[shows \(\) \(trakt.interfaces.sync.ratings.SyncRatingsInterface method\), 12](#)
[shows \(\) \(trakt.interfaces.sync.watched.SyncWatchedInterface method\), 12](#)
[shows \(\) \(trakt.interfaces.sync.watchlist.SyncWatchlistInterface method\), 12](#)
[shows \(\) \(trakt.interfaces.users.history.UsersHistoryInterface method\), 13](#)
[shows \(\) \(trakt.interfaces.users.ratings.UsersRatingsInterface method\), 14](#)
[shows \(\) \(trakt.interfaces.users.watchlist.UsersWatchlistInterface method\), 15](#)
[shows \(\) \(trakt.mapper.summary.SummaryMapper class method\), 44](#)
[shows \(\) \(trakt.mapper.SummaryMapper class method\), 41](#)
[shows \(\) \(trakt.mapper.sync.SyncMapper class method\), 44](#)
[shows \(\) \(trakt.mapper.SyncMapper class method\), 42](#)
[ShowsInterface \(class in trakt.interfaces.shows\), 10](#)
[site_url \(trakt.client.TraktClient attribute\), 35](#)
[sort_by \(trakt.objects.list.base.List attribute\), 19](#)
[sort_how \(trakt.objects.list.base.List attribute\), 20](#)
[spoiler \(trakt.objects.comment.Comment attribute\), 16](#)
[ssl_version \(trakt.core.http.HttpClient attribute\), 38](#)
[start \(\) \(trakt.interfaces.scrobble.ScrobbleInterface method\), 6](#)
[status \(trakt.objects.show.Show attribute\), 31](#)
[stop \(\) \(trakt.interfaces.scrobble.ScrobbleInterface method\), 7](#)
[SummaryMapper \(class in trakt.mapper\), 41](#)
[SummaryMapper \(class in trakt.mapper.summary\), 43](#)
[SyncCollectionInterface \(class in trakt.interfaces.sync.collection\), 11](#)
[SyncHistoryInterface \(class in trakt.interfaces.sync.history\), 11](#)
[synchronized \(\) \(in module trakt.core.helpers\), 38](#)
[SyncInterface \(class in trakt.interfaces.sync\), 10](#)
[SyncMapper \(class in trakt.mapper\), 41](#)
[SyncMapper \(class in trakt.mapper.sync\), 44](#)
[SyncPlaybackInterface \(class in trakt.interfaces.sync.playback\), 11](#)
[SyncRatingsInterface \(class in trakt.interfaces.sync.ratings\), 11](#)
[SyncWatchedInterface \(class in trakt.interfaces.sync.watched\), 12](#)
[SyncWatchlistInterface \(class in trakt.interfaces.sync.watchlist\), 12](#)
[tagline \(trakt.objects.movie.Movie attribute\), 25](#)
[threading \(trakt.core.emitter.Emitter attribute\), 37](#)

threading_workers (*trakt.core.emitter.Emitter* attribute), 37
 timestamp (*trakt.objects.rating.Rating* attribute), 28
 title (*trakt.objects.episode.Episode* attribute), 17
 title (*trakt.objects.movie.Movie* attribute), 25
 title (*trakt.objects.show.Show* attribute), 31
 to_dict() (*trakt.objects.episode.Episode* method), 17
 to_dict() (*trakt.objects.movie.Movie* method), 25
 to_dict() (*trakt.objects.progress.BaseProgress* method), 27
 to_dict() (*trakt.objects.progress.CollectionProgress* method), 27
 to_dict() (*trakt.objects.progress.EpisodeProgress* method), 27
 to_dict() (*trakt.objects.progress.Progress* method), 28
 to_dict() (*trakt.objects.progress.SeasonProgress* method), 28
 to_dict() (*trakt.objects.progress.WatchedProgress* method), 28
 to_dict() (*trakt.objects.season.Season* method), 30
 to_dict() (*trakt.objects.show.Show* method), 32
 to_identifier() (*trakt.objects.episode.Episode* method), 17
 to_identifier() (*trakt.objects.movie.Movie* method), 25
 to_identifier() (*trakt.objects.season.Season* method), 30
 to_identifier() (*trakt.objects.show.Show* method), 32
 to_info() (*trakt.objects.episode.Episode* method), 17
 to_info() (*trakt.objects.movie.Movie* method), 25
 to_info() (*trakt.objects.season.Season* method), 30
 to_info() (*trakt.objects.show.Show* method), 32
 to_iso8601() (in module *trakt.core.helpers*), 38
 to_iso8601_date() (in module *trakt.core.helpers*), 38
 to_iso8601_datetime() (in module *trakt.core.helpers*), 38
 token() (*trakt.interfaces.oauth.device.DeviceOAuthInterface* method), 3
 token() (*trakt.interfaces.oauth.OAuthInterface* method), 2
 token_exchange() (*trakt.interfaces.oauth.OAuthInterface* method), 2
 token_refresh() (*trakt.interfaces.oauth.OAuthInterface* method), 2
 trailer (*trakt.objects.movie.Movie* attribute), 25
 Trakt (class in *trakt*), 35
 trakt (module), 35
 trakt.client (module), 35
 trakt.core (module), 35
 trakt.core.configuration (module), 35
 trakt.core.context_collection (module), 36
 trakt.core.context_stack (module), 36
 trakt.core.emitter (module), 37
 trakt.core.errors (module), 37
 trakt.core.exceptions (module), 37
 trakt.core.helpers (module), 37
 trakt.core.http (module), 38
 trakt.core.pagination (module), 39
 trakt.core.request (module), 39
 trakt.helpers (module), 39
 trakt.interfaces.auth (module), 1
 trakt.interfaces.base (module), 39
 trakt.interfaces.calendars (module), 1
 trakt.interfaces.movies (module), 2
 trakt.interfaces.oauth (module), 2
 trakt.interfaces.oauth.device (module), 3
 trakt.interfaces.oauth.pin (module), 3
 trakt.interfaces.scrobble (module), 3
 trakt.interfaces.search (module), 8
 trakt.interfaces.shows (module), 10
 trakt.interfaces.sync (module), 10
 trakt.interfaces.sync.collection (module), 11
 trakt.interfaces.sync.core (module), 40
 trakt.interfaces.sync.core.mixins (module), 40
 trakt.interfaces.sync.history (module), 11
 trakt.interfaces.sync.playback (module), 11
 trakt.interfaces.sync.ratings (module), 11
 trakt.interfaces.sync.watched (module), 12
 trakt.interfaces.sync.watchlist (module), 12
 trakt.interfaces.users (module), 12
 trakt.interfaces.users.following (module), 13
 trakt.interfaces.users.friends (module), 13
 trakt.interfaces.users.history (module), 13
 trakt.interfaces.users.lists (module), 13
 trakt.interfaces.users.lists.list_ (module), 14
 trakt.interfaces.users.ratings (module), 14
 trakt.interfaces.users.settings (module), 14
 trakt.interfaces.users.watchlist (module), 15
 trakt.mapper (module), 40
 trakt.mapper.comment (module), 42
 trakt.mapper.core (module), 42
 trakt.mapper.core.base (module), 42
 trakt.mapper.list (module), 42
 trakt.mapper.list_item (module), 42

trakt.mapper.progress (module), 43
 trakt.mapper.search (module), 43
 trakt.mapper.summary (module), 43
 trakt.mapper.sync (module), 44
 trakt.mapper.user (module), 44
 trakt.objects.comment (module), 15
 trakt.objects.core (module), 44
 trakt.objects.core.helpers (module), 44
 trakt.objects.episode (module), 16
 trakt.objects.list.base (module), 18
 trakt.objects.list.custom (module), 20
 trakt.objects.media (module), 22
 trakt.objects.movie (module), 24
 trakt.objects.person (module), 26
 trakt.objects.progress (module), 26
 trakt.objects.rating (module), 28
 trakt.objects.season (module), 29
 trakt.objects.show (module), 30
 trakt.objects.user (module), 32
 trakt.objects.video (module), 34
 TraktClient (class in trakt.client), 35
 TraktRequest (class in trakt.core.request), 39
 transform_data () (trakt.core.request.TraktRequest method), 39
 transform_headers () (trakt.core.request.TraktRequest method), 39
 transform_method () (trakt.core.request.TraktRequest method), 39
 transform_parameters () (trakt.core.request.TraktRequest method), 39
 trending () (trakt.interfaces.movies.MoviesInterface method), 2
 trending () (trakt.interfaces.shows.ShowsInterface method), 10
 try_convert () (in module trakt.core.helpers), 38

U

unfollow () (trakt.objects.user.User method), 33
 unlike () (trakt.interfaces.users.lists.list_UsersListInterface method), 14
 unlike () (trakt.objects.list.base.List method), 20
 unlike () (trakt.objects.list.custom.CustomList method), 21
 update () (trakt.interfaces.users.lists.list_UsersListInterface method), 14
 update () (trakt.objects.list.custom.CustomList method), 21
 update_attributes () (in module trakt.objects.core.helpers), 44
 updated_at (trakt.objects.episode.Episode attribute), 17
 updated_at (trakt.objects.list.base.List attribute), 20
 updated_at (trakt.objects.movie.Movie attribute), 25
 updated_at (trakt.objects.show.Show attribute), 32
 url () (trakt.interfaces.oauth.pin.PinOAuthInterface method), 3
 User (class in trakt.objects.user), 32
 user (trakt.objects.comment.Comment attribute), 16
 user (trakt.objects.list.base.List attribute), 20
 user () (trakt.mapper.user.UserMapper class method), 44
 user () (trakt.mapper.UserMapper class method), 42
 user_count (trakt.objects.movie.Movie attribute), 25
 user_count (trakt.objects.show.Show attribute), 32
 user_rating (trakt.objects.comment.Comment attribute), 16
 UserMapper (class in trakt.mapper), 42
 UserMapper (class in trakt.mapper.user), 44
 username (trakt.objects.list.base.List attribute), 20
 username (trakt.objects.list.custom.CustomList attribute), 22
 username (trakt.objects.user.User attribute), 33
 users () (trakt.mapper.user.UserMapper class method), 44
 users () (trakt.mapper.UserMapper class method), 42
 UsersFollowingInterface (class in trakt.interfaces.users.following), 13
 UsersFriendsInterface (class in trakt.interfaces.users.friends), 13
 UsersHistoryInterface (class in trakt.interfaces.users.history), 13
 UsersInterface (class in trakt.interfaces.users), 12
 UsersListInterface (class in trakt.interfaces.users.lists.list_), 14
 UsersListsInterface (class in trakt.interfaces.users.lists), 13
 UsersRatingsInterface (class in trakt.interfaces.users.ratings), 14
 UsersSettingsInterface (class in trakt.interfaces.users.settings), 14
 UsersWatchlistInterface (class in trakt.interfaces.users.watchlist), 15

V

validate () (trakt.core.http.HttpClient method), 38
 value (trakt.objects.rating.Rating attribute), 29
 version (trakt.client.TraktClient attribute), 35
 video (class in trakt.objects.video), 34
 vip (trakt.objects.user.User attribute), 33
 vip_ep (trakt.objects.user.User attribute), 33
 votes (trakt.objects.rating.Rating attribute), 29

W

watched_at (trakt.objects.video.Video attribute), 34

WatchedProgress (*class in trakt.objects.progress*),
28
watchers (*trakt.objects.movie.Movie attribute*), 25
watchers (*trakt.objects.show.Show attribute*), 32
watchlist() (*trakt.objects.user.User method*), 33
with_mapper() (*trakt.core.pagination.PaginationIterator
method*), 39

Y

year (*trakt.objects.movie.Movie attribute*), 25
year (*trakt.objects.show.Show attribute*), 32