PyMDb

Release 0.2.1

Dec 10, 2019

Contents:

PyMDb is a package for both parsing the datasets provided by IMDb and scraping information from their web pages.

This package is able to gather information on people, titles, and companies provided by IMDb and is split into two separate modules: one for parsing the IMDb datasets, and one for scraping webpages on imdb.com.

CHAPTER 1

Installation

The latest release of PyMDb can be installed from PyPI with:

```
pip install py-mdb
```

If downloading the source from GitHub, PyMDb requires the following packages:

- requests
- selectolax

CHAPTER 2

Usage

```
>>> import pymdb
>>> from collections import defaultdict
>>>
>>> parser = pymdb.PyMDbParser(gunzip_files=True)
>>> genre_count = defaultdict(int)
>>> for title in parser.get_title_basics("path/to/files"):
...     for genre in title.genres:
...             genre_count[genre] += 1
...
>>> for genre in genre_count:
...     print(f"{genre}: {genre_count[genre]}")
...
Documentary: 600184
Short: 837912
Animation: 312227
    ...
Talk-Show: 584252
Reality-TV: 307037
Adult: 178493
>>>
>>> scraper = pymdb.PyMDbScraper()
>>> title = scraper.get_title("tt0076759")
>>> print(f"{title.display_title} came out in {title.release_date.year}!")
Star Wars: Episode IV - A New Hope came out in 1977!
```

CHAPTER 3

Documentation

Full documentation can be found at the PyMDb Read the Docs page.

CHAPTER 4

Disclaimer

PyMDb is still in a pre-release state and has only been tested with a small amount of data found on imdb.com. If any bugs or issues are found, please do not hesitate to create an issue or make a pull request on GitHub. Suggestions for features to be added to PyMDb in future releases are also welcome!

CHAPTER 5

License

This project is licensed under the MIT License. Please see the LICENSE file for details.

5.1 PyMDb modules

5.1.1 pymdb.exceptions module

Module containing exception classes unique to PyMDb.

InvalidCompanyId

class pymdb.exceptions.**InvalidCompanyId**
Raised when an invalid company ID has been used in an IMDb request.

InvalidParseFormat

class pymdb.exceptions.**InvalidParseFormat**
Raised when PyMDbParser runs into a row with an incorrect column size.

5.1.2 pymdb.models.company module

The classes used to represent various information about companies on IMDb.

All information for the classes here will be scraped from IMDb web pages.

CompanyScrape

class pymdb.models.company.**CompanyScrape**(*company_id*, *title_id*, *start_year*, *end_year*, *notes*)

Stores a title a company is credited for on IMDb.

This information is taken from IMDb's search by company ID for titles that include it in their credits.

Parameters

- **company_id** (str) – The companys's ID used by IMDb prefixed with *co*.
- **title_id** (str) – The titles's ID used by IMDb prefixed with *tt*.
- **start_year** (int) – The year the title released, or the year the company started being credited for a TV series.
- **end_year** (int) – The year the company stopped being credited for a TV series, or *None* otherwise.
- **notes** (list of str) – A list of further notes IMDb gives about the credit.

CompanyCreditScrape

class pymdb.models.company.**CompanyCreditScrape**(*company_id*, *title_id*, *company_name*, *category*, *notes*)

Stores a company that is credited on a title's IMDb page.

This information is taken from a title's IMDb company credits page, and contains more information on what a company is credited on a title for.

Parameters

- **company_id** (str) – The company's ID used by IMDb prefixed with *co*.
- **title_id** (str) – The titles's ID used by IMDb prefixed with *tt*.
- **company_name** (str) – The company's name it was credited under.
- **category** (str) – The category the company was credited for.
- **notes** (list of str) – A list of further notes IMDb gives about the credit.

5.1.3 pymdb.models.name module

The classes used to represent various information about persons on IMDb.

This will contain classes for both information gathered from the datasets provided by IMDb and information scraped from IMDb web pages. Class names ending with "*Scrape*" are scraped from the web pages. Otherwise, they are gathered from the datasets.

NameBasics

class pymdb.models.name.**NameBasics**(*name_id*, *primary_name*, *birth_year*, *death_year*, *primary_professions*, *known_for_titles*)

Class to store the row information from IMDb's "*name.basics.tsv*" dataset.

Parameters

- **name_id** (str) – The person's ID used by IMDb prefixed with *nm*.

- **primary_name** (str) – The person's name.
- **birth_year** (int) – The person's birth year.
- **death_year** (int) – The person's death year, or *None* otherwise.
- **primary_professions** (list of str) – A list of all the person's primary professions.
- **known_for_titles** (list of str) – A list of title IDs for each title the person is known for.

NameScrape

class pymdb.models.name.**NameScrape**(*name_id*, *display_name*, *known_for_titles*, *birth_name*, *birth_date*, *birth_city*, *death_date*, *death_city*, *death_cause*, *nicknames*, *height*)

Specific information on a person scraped from IMDb.

This information is taken from IMDb's *bio* web page on a person to find detailed information.

Parameters

- **name_id** (str) – The person's ID used by IMDb prefixed with *nm*.
- **display_name** (str) – The name IMDb lists the person having currently. Usually how they are well known or credited.
- **known_for_titles** (list of str) – A list of title IDs for each title the person is known for.
- **birth_name** (str) – The name IMDb lists the person born as.
- **birth_date** (datetime) – The date the person was born.
- **birth_city** (str) – The city the person was born in.
- **death_date** (datetime) – The date the person died, or *None* otherwise.
- **death_city** (str) – The city the person died in, or *None* otherwise.
- **death_cause** (str) – The person's cause of death, or *None* otherwise.
- **nicknames** (list of str) – All of the person's nicknames.
- **height** (float) – How tall the person is in meters.

NameCreditScrape

class pymdb.models.name.**NameCreditScrape**(*name_id*, *title_id*, *category*, *start_year*, *end_year*, *role*, *title_notes*)

Stores credit information from a person's *full filmography* on IMDb.

This information is taken from IMDb's *full filmography* section of a person's personal web page.

Parameters

- **name_id** (str) – The person's ID used by IMDb prefixed with *nm*.
- **title_id** (str) – The titles's ID used by IMDb prefixed with *tt*.
- **category** (str) – The category this credit is listed under in the filmography section.
- **start_year** (int) – The year the title released, or the starting year they were credited for on a TV series.

- **end_year** (int) – The year the person stopped being credited on a TV series, or *None* otherwise.
- **role** (str) – A string of the role the person is credited for the title, such as character.
- **title_notes** (list of str) – A list of further notes for a person's credit on a title.

CreditScrape

class pymdb.models.name.**CreditScrape**(*name_id*, *title_id*, *job_title*, *credit*, *episode_count*, *episode_year_start*, *episode_year_end*)

Object to represent information for each person scraped from IMDb's *fullcredits* page for a title.

This information is scraped from the *fullcredits* IMDb web page, and will either represent an actor or another crew member.

Parameters

- **name_id** (str) – The person's ID used by IMDb prefixed with *nm*.
- **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.
- **job_title** (str) – The job title the person is credited for on the title.
- **credit** (str) – Further credits for the person on the title.
- **episode_count** (int) – How many episodes the person is credited for if a TV series, otherwise *None*.
- **episode_year_start** (int) – The year the person began being credited in the title if the title is a TV series, otherwise *None*.
- **episode_year_end** (int) – The year the person stopped being credited in the title if the title is a TV series, otherwise *None*.

5.1.4 pymdb.models.search module

The classes used to represent search results on IMDb.

This information is gathered from the JSON responses IMDb uses for various GET requests for their search results.

SearchResult

class pymdb.models.search.**SearchResult**(*imdb_id*, *search_rank*)

The base search result class.

Contains the basic search result information both *SearchResultName* and *SearchResultTitle* use.

Parameters

- **imdb_id** (str) – The ID used by IMDb. Prefixed with *nm* for names and *tt* for titles.
- **search_rank** (int) – The ranking of the result by IMDb.

SearchResultName

class pymdb.models.search.**SearchResultName**(*imdb_id*, *search_rank*, *name*, *known_for*)

Search result for a person within IMDb.

Contains the rest of the information IMDb provides within a search result when the ID is for a person.

Parameters

- **imdb_id** (str) – The ID used by IMDb prefixed with *nm*.
- **search_rank** (int) – The ranking of the result by IMDb.
- **name** (str) – The name of the person.
- **known_for** (str) – The blurb IMDb provides for what the person is known for to build the search results.

SearchResultTitle

class pymdb.models.search.**SearchResultTitle**(*imdb_id*, *search_rank*, *display_title*, *title_type*, *starring*, *start_year*, *end_year*)

Search result for a title within IMDb.

Contains the rest of the information IMDb provides within a search result when the ID is for a title. Note: If the *title_type* is a "*video game*", the *starring* list will be empty as IMDb provides the title's genres in its place.

Parameters

- **imdb_id** (str) – The ID used by IMDb prefixed with *tt*.
- **search_rank** (int) – The ranking of the result by IMDb.
- **display_title** (str) – The title used by IMDb within search results.
- **title_type** (str) – The type of title (ex: TV series, feature, etc).
- **starring** (list of str) – A list of two actor names that are known for starring in the film.
- **start_year** (int) – The year the title was released, or the start year if a TV series.
- **end_year** (int) – The year a TV series was ended, or *None* if the series has not ended or is not a TV series.

5.1.5 pymdb.models.title module

The classes used to represent various information about titles on IMDb.

This will contain classes for both information gathered from the datasets provided by IMDb and information scraped from IMDb web pages. Class names ending with "*Scrape*" are scraped from the web pages. Otherwise, they are gathered from the datasets.

TitleAkas

class pymdb.models.title.**TitleAkas**(*title_id*, *ordering*, *localized_title*, *region*, *language*, *types*, *attributes*, *is_original_title*)

Class to store the row information from IMDb's "*title.akas.tsv*" dataset.

Provides additional information for a title.

Parameters

- **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.
- **ordering** (int) – Uniquely identifies the row for a given *title_id*.
- **localized_title** (str) – The localized title for this version.

- **region** (str) – The region for this version of the title.
- **language** (str) – The language of this title.
- **types** (list of str) – A list of the different types attributed to this title.
- **attributes** (list of str) – A list of terms to describe this title.
- **is_original_title** (bool) – Determines if this version is the original title.

TitleBasics

class pymdb.models.title.**TitleBasics**(*title_id*, *title_type*, *primary_title*, *original_title*, *is_adult*, *start_year*, *end_year*, *runtime*, *genres*)

Class to store the row information from IMDb's "*title.basics.tsv*" dataset.

Basic information for a title.

Parameters

- **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.
- **title_type** (str) – The type/format of the title (ex: *movie*, *short*, *tvseries*, *tvepisode*, *video*, etc).
- **primary_title** (str) – The popular title used for this title.
- **original_title** (str) – The original title in the original langauge.
- **is_adult** (bool) – Determines if this is an adult title.
- **start_year** (int) – The year this title was released or the year the series started.
- **end_year** (int) – The year this TV series ended, otherwise *None*.
- **runtime** (int) – The length of the title's runtime in minutes.
- **genres** (list of str) – A list of up to three genres associated with this title.

TitleCrew

class pymdb.models.title.**TitleCrew**(*title_id*, *director_ids*, *writer_ids*)

Class to store the row information from IMDb's "*title.crew.tsv*" dataset.

Director(s) and writer(s) for a title.

Parameters

- **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.
- **director_ids** (list of str) – A list of IDs used by IMDb for people, prefixed with *nm*, representing the director(s) of the title.
- **writer_ids** (list of str) – A list of IDs used by IMDb for people, prefixed with *nm*, representing the writer(s) of the title.

TitleEpisode

class pymdb.models.title.**TitleEpisode**(*title_id*, *parent_title_id*, *season_number*, *episode_number*)

Class to store the row information from IMDb's "*title.episodes.tsv*" dataset.

Parameters

- **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.
- **parent_title_id** (str) – The title's ID used by IMDb prefixed with *tt*. Represents the parent TV series to the episode.
- **season_number** (int) – The season this episode was aired in.
- **episode_number** (int) – The episode number it was aired as in the season.

TitlePrincipalCrew

class pymdb.models.title.**TitlePrincipalCrew**(*title_id*, *ordering*, *name_id*, *category*, *job*, *characters*)

Class to store the row information from IMDb's "*title.principals.tsv*" dataset.

Principal cast/crew for a title.

Parameters

- **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.
- **ordering** (int) – Uniquely identifies the row for a given *title_id*.
- **name_id** (str) – The person's ID used by IMDb prefixed with *nm*.
- **category** (str) – The category of job the person was in.
- **job** (str) – The specific job title if available, otherwise *None*.
- **characters** (list of str) – A list of characters the person played, if applicable.

TitleRating

class pymdb.models.title.**TitleRating**(*title_id*, *average_rating*, *num_votes*)

Class to store the row information from IMDb's "*title.ratings.tsv*" dataset.

Parameters

- **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.
- **average_rating** (float) – The weighted average of all user ratings on IMDb.
- **num_votes** (int) – The total number of user votes the title has received on IMDb.

TitleScrape

class pymdb.models.title.**TitleScrape**(*title_id*, *display_title*, *title_parent_id*, *mpaa_rating*, *country*, *language*, *release_date*, *end_year*, *season_number*, *episode_number*, *taglines*, *plot*, *storyline*, *production_companies*, *top_cast*, *budget*, *budget_denomination*, *opening_weekend_gross*, *opening_weekend_date*, *usa_gross*, *worldwide_gross*)

Object to represent detailed information for a title on its IMDb web page.

This information is scraped from the main page for a title's IMDb web page.

Parameters

- **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.
- **display_title** (str) – The title's name on its IMDb page.

- **title_parent_id** (str) – The episode's TV series parent ID used by IMDb prefixed with *tt*, otherwise *None*.
- **mpaa_rating** (str) – The title's MPAA rating.
- **country** (str) – The title's country of origin.
- **language** (str) – The title's original language.
- **release_date** (datetime) – The title's original release date, or when the TV series began airing.
- **end_year** (int) – The year the TV series stopped airing, otherwise *None*.
- **season_number** (int) – The season number the episode is in, otherwise *None*.
- **episode_number** (int) – The episode's number in the season, otherwise *None*.
- **taglines** (list of str) – A list of all of the title's taglines.
- **plot** (str) – The title's plot.
- **storyline** (str) – The title's storyline.
- **production_companies** (list of str) – A list of company IDs, used by IMDb and prefixed with *co*, that are credited with producing the title.
- **top_cast** (list of *CreditScrape*) – A list of *CreditScrape* objects that are the top cast in the title.
- **budget** (int) – The estimated budget for the title.
- **budget_denomination** (str) – The denomination the budget value is listed as
- **opening_weekend_gross** (int) – The title's gross in USD on its opening weekend.
- **opening_weekend_date** (datetime) – The title's opening weekend date.
- **usa_gross** (int) – The title's gross in the USA in USD.
- **worldwide_gross** (int) – The title's gross worldwide in USD.

TitleTechSpecsScrape

class pymdb.models.title.**TitleTechSpecsScrape**(*title_id*, *runtime*, *sound_mix*, *color*, *aspect_ratio*, *camera*, *laboratory*, *negative_format*, *cinematographic_process*, *printed_film_format*)

Object to represent information for a title's technical specifications.

This information is scraped from the *technical* IMDb web page for a title.

Parameters

- **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.
- **runtime** (int) – The length of the title's runtime in minutes.
- **sound_mix** (list of str) – A list of the title's sound mixes.
- **color** (str) – The color used in the title.
- **aspect_ratio** (list of str) – A list of each aspect ratio the title is available in.
- **camera** (list of str) – A list of the cameras used in the title.
- **laboratory** (list of str) – A list of all the title's laboratories.

- **negative_format** (str) – The title's negative format.
- **cinematographic_process** (list of str) – A list of each of the title's cinematographic processes.
- **printed_film_format** (str) – The title's printed film format.

5.1.6 pymdb.parser module

Module containing the PyMDbParser class.

PyMDbParser

class pymdb.parser.**PyMDbParser**(*use_default_filenames=True*, *gunzip_files=False*, *delete_gzip_files=False*)

Object used to parse the *tsv* datasets provided by IMDb.

Parses each row in the *tsv* file into a specific PyMDb object.

Parameters

- **use_default_filenames** (bool, optional) – Determine whether the filenames for each dataset are the same as the names provided by IMDb.
- **gunzip_files** (bool, optional) – Determine if the files are gzipped or not.
- **delete_gzip_files** (bool, optional) – Determine if gzip files should be deleted after being gunzipped.

_build_path(*path*, *default_filename*)

Private function to combine a system path with a default filename.

This method will append the default filename of a dataset to the given path it is located in. If the files are to be gunzipped, it will also append the correct gzip extension used by IMDb.

Parameters

- **path** (str) – The system path to the directory where the dataset is located.
- **default_filename** (str) – The default filename of the dataset.

Returns The path and default filename combined correctly.

Return type str

get_name_basics(*path*, *contains_headers=True*)

Parse the "*name.basics.tsv*" dataset provided by IMDb.

Parameters

- **path** (str) – The system path to the dataset file. If not using default filenames, this string will include the dataset file.
- **contains_headers** (bool, optional) – Determine if the first line is column titles or a data row.

Yields A *NameBasics* object for each row in the dataset.

Raises InvalidParseFormat – If a row has an incorrect column size.

get_title_akas(*path*, *contains_headers=True*)

Parse the "*title.akas.tsv*" dataset provided by IMDb.

Parameters

- **path** (str) – The system path to the dataset file. If not using default filenames, this string will include the dataset file.
- **contains_headers** (bool, optional) – Determine if the first line is column titles or a data row.

Yields A *TitleAkas* object for each row in the dataset.

Raises InvalidParseFormat – If a row has an incorrect column size.

get_title_basics(*path*, *contains_headers=True*)
Parse the "*title.basics.tsv*" dataset provided by IMDb.

Parameters

- **path** (str) – The system path to the dataset file. If not using default filenames, this string will include the dataset file.
- **contains_headers** (bool, optional) – Determine if the first line is column titles or a data row.

Yields A *TitleBasics* object for each row in the dataset.

Raises InvalidParseFormat – If a row has an incorrect column size.

get_title_crew(*path*, *contains_headers=True*)
Parse the "*title.crew.tsv*" dataset provided by IMDb.

Parameters

- **path** (str) – The system path to the dataset file. If not using default filenames, this string will include the dataset file.
- **contains_headers** (bool, optional) – Determine if the first line is column titles or a data row.

Yields A *TitleCrew* object for each row in the dataset.

Raises InvalidParseFormat – If a row has an incorrect column size.

get_title_episodes(*path*, *contains_headers=True*)
Parse the "*title.episodes.tsv*" dataset provided by IMDb.

Parameters

- **path** (str) – The system path to the dataset file. If not using default filenames, this string will include the dataset file.
- **contains_headers** (bool, optional) – Determine if the first line is column titles or a data row.

Yields A *TitleEpisode* object for each row in the dataset.

Raises InvalidParseFormat – If a row has an incorrect column size.

get_title_principals(*path*, *contains_headers=True*)
Parse the "*title.principals.tsv*" dataset provided by IMDb.

Parameters

- **path** (str) – The system path to the dataset file. If not using default filenames, this string will include the dataset file.
- **contains_headers** (bool, optional) – Determine if the first line is column titles or a data row.

Yields A *TitlePrincipalCrew* object for each row in the dataset.

Raises `InvalidParseFormat` – If a row has an incorrect column size.

get_title_ratings(*path*, *contains_headers=True*)
Parse the "*title.ratings.tsv*" dataset provided by IMDb.

Parameters

- **path** (`str`) – The system path to the dataset file. If not using default filenames, this string will include the dataset file.
- **contains_headers** (`bool`, optional) – Determine if the first line is column titles or a data row.

Yields A *TitleRating* object for each row in the dataset.

Raises `InvalidParseFormat` – If a row has an incorrect column size.

5.1.7 pymdb.scraper module

Module containing the PyMDbScraper class.

PyMDbScraper

class pymdb.scraper.**PyMDbScraper**
Scrapes various information from IMDb web pages.

Contains functions for various IMDb pages and scrapes information into Python classes.

_get_tree(*request*)
Get the selectolax HTML tree given a request.

Parameters **request** (`str`) – The HTTP GET request.

Returns The HTML tree from the GET request.

Return type `HTMLTree`

Raises `HTTPError` – If a non successful response was returned.

get_company(*company_id*)
Scrapes all titles a company is credited for on IMDb.

Will scrape all titles listed under a company on IMDb by going through each page in IMDb's *company search*. This only gives the year(s) the company was involved with each title and *notes* for each listed on IMDb.

Parameters **company_id** (`str`) – The company's ID used by IMDb prefixed with *co*.

Yields *CompanyScrape* – An object for each title the company is credited for.

Raises

- `HTTPError` – If a request failed.
- `InvalidCompanyId` – If an invalid company ID was given.

get_company_credits(*title_id*)
Gets all companies credited for a title.

Scrapes a title's company credits page on IMDb to find information for each company that was credited. Each company creates a new *CompanyCreditScrape* object.

Parameters **title_id** (`str`) – The title's ID used by IMDb prefixed with *tt*.

Yields *CompanyCreditScrape* – An object for each company.

Raises HTTPError – If the request failed.

get_full_cast(*title_id*, *include_episodes=False*)
Scrapes the full cast of actors for a specified title.

Will scrape the full cast of actors for a title, each into their own *CreditScrape* object. An optional argument *include_episodes* will also scrape each episode an actor is in if the title is a TV series.

Parameters

- **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.
- **include_episodes** (bool, optional) – Specify if individual episodes of a TV series should also be scraped.

Yields CreditScrape – An object for each cast member in the title.

Raises HTTPError – If a request failed.

get_full_credits(*title_id*, *include_episodes=False*)
Scrapes the full list of credited people for a title.

Will scrape all the cast and crew for a title by returning both *get_full_cast* and *get_full_crew* as a single generator. An optional argument *include_episodes* will also scrape each episode an actor is in if the title is a TV series.

Parameters

- **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.
- **include_episodes** (bool, optional) – Specify if individual episodes of a TV series should also be scraped.

Yields CreditScrape – An object for each credited crew member in the title.

Raises HTTPError – If the request failed.

get_full_credits_as_dict(*title_id*, *include_episodes=False*)
Scrapes the full list of credited people for a title into a dictionary.

Builds a dictionary with *job_title* as key of lists of CreditScrape objects. Uses the results of the *get_full_credits* method to gather the objects. An optional argument *include_episodes* will also scrape each episode an actor is in if the title is a TV series.

Parameters

- **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.
- **include_episodes** (bool, optional) – Specify if individual episodes of a TV series should also be scraped.

Returns

A dictionary where each key is a str of a *job_title* and the value is a list of CreditScrape objects who's *job_title* value is the same as the key.

Return type dict of list of CreditScrape

Raises HTTPError – If the request failed.

get_full_crew(*title_id*)
Scrapes the full list of credited crew people for a title, not including actors.

Will scrape all the credited crew members of a title, without the actors. For example, this will include all directors, writers, producers, cinematographers, etc.

Parameters **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.

Yields CreditScrape – An object for each credited crew member in the title.

Raises HTTPError – If the request failed.

get_name(*name_id*, *include_known_for_titles=False*)
Scrapes detailed information from a person's personal IMDb web page.

Will scrape detailed information on a person's IMDb *bio* page into a new *NameScrape* object.

Parameters

- **name_id** (str) – The person's ID used by IMDb prefixed with *nm*.
- **include_known_for_titles** (bool, optional) – Determines if an second request should be sent to get the known for titles on a person's default IMDb page.

Returns An object with the person's information.

Return type *NameScrape*

Raises HTTPError – If the request failed.

get_name_credits(*name_id*, *include_episodes=False*)
Scrapes all title credits a person is included in.

Scrapes the *full filmography* from a person's IMDb page to get each title they are credited in, and what category that credit is under. An optional argument *include_episodes* will also scrape each episode an actor is in if the title is a TV series. Each credit is created with a new *NameCreditScrape* object.

Parameters

- **name_id** (str) – The person's ID used by IMDb prefixed with *nm*.
- **include_episodes** (bool, optional) – Specify if individual episodes of a TV series should also be scraped.

Yields *NameCreditScrape* – An object for each credit in the person's filmography.

Raises HTTPError – If a request failed.

get_search_results(*keyword*)
Gets search results for a given keyword.

Uses IMDb's GET requests for searches to retrieve a JSON response containing the search result information. A list of *SearchResult* objects is created for each result that is either a name or title. If the result is a name, the object is a *SearchResultName*. If it is a title, the object is a *SearchResultTitle*.

Parameters **keyword** (str) – The keyword to search for. IMDb caps keywords at 20 characters.

Returns

A list of either *SearchResultName* and/or *SearchResultTitle* objects.

Return type list of *SearchResult*

Raises HTTPError – If the request failed.

get_tech_specs(*title_id*)
Gets information for all tech specs for a title.

Uses a title's *technical* web page on IMDb to scrape all technical specifications listed. A new *TitleTechSpecScrape* object is created for the tech specs.

Parameters **title_id** (str) – The title's ID used by IMDb prefixed with *tt*.

Returns An object containing the information.

Return type `TitleTechSpecScrape`

Raises `HTTPError` – If the request failed.

get_title(*title_id*, *include_taglines=False*)
Scrapes information from the IMDb web page for the specified title.

Uses the given title ID to request the IMDb page for the title and scrapes the page's information into a new *TitleScrape* object. An optional argument *include_taglines* allows an additional request to be made to gather all taglines IMDb has for the title.

Parameters

- **title_id** (`str`) – The title's ID used by IMDb prefixed with *tt*.
- **include_taglines** (`bool`, optional) – Specify if an extra request should be made to get all the taglines for the title

Returns An object containing the page's information.

Return type *TitleScrape*

Raises `HTTPError` – If the request failed.

5.1.8 pymdb.utils module

Module containing various utility functions used within other PyMDb modules.

The functions within here are not intended to be used outside of the PyMDb package.

append_filename_to_path

pymdb.utils.**append_filename_to_path**(*path*, *filename*)
Append a filename to a system file path.

This method correctly appends a filename to a file path with the correct path separators used within the path string.

Parameters

- **path** (`str`) – The system file path.
- **filename** (`str`) – The filename to append.

Returns The filename correctly appended to the file path.

Return type `str`

is_float

pymdb.utils.**is_float**(*f*)
Check if a variable is a *float* type.

Parameters **f** – The object to check.

Returns If the object can be converted to a `float`.

Return type `bool`

is_int

pymdb.utils.**is_int**(*i*)
Check if a variable is an *int* type.

Parameters **i** – The object to check.

Returns If the object can be converted to an `int`.

Return type `bool`

get_company_id

pymdb.utils.**get_company_id**(*node*)
Find the IMDb company ID within a selectolax *Node*.

Expects the ID to be within the *Node*'s "*href*" attribute.

Parameters **node** (`Node`) – A *Node* containing the ID.

Returns The IMDb company ID.

Return type `str`

get_name_id

pymdb.utils.**get_name_id**(*node*)
Find the IMDb name ID within a selectolax *Node*.

Expects the ID to be within the *Node*'s "*href*" attribute.

Parameters **node** (`Node`) – A *Node* containing the ID.

Returns The IMDb name ID.

Return type `str`

get_title_id

pymdb.utils.**get_title_id**(*node*)
Find the IMDb title ID within a selectolax *Node*.

Expects the ID to be within the *Node*'s "*href*" attribute.

Parameters **node** (`Node`) – A *Node* containing the ID.

Returns The IMDb title ID.

Return type `str`

get_category

pymdb.utils.**get_category**(*node*)
Gets the category value from a selectolax *Node*.

Grabs the value from the *Node*'s "*onclick*" attribute.

Parameters **node** (`Node`) – A *Node* containing the "*onclick*" attribute.

Returns The category.

Return type str

get_ref_marker

pymdb.utils.**get_ref_marker**(*node*)

Gets the ref marker value from a selectolax *Node*.

Grabs the value from the *Node*'s "*onclick*" attribute.

Parameters **node** (Node) – A *Node* containing the "*onclick*" attribute.

Returns The ref marker.

Return type str

get_episode_info

pymdb.utils.**get_episode_info**(*node*)

Gets the episode count, episode year start, and episode year end for an actor.

Gets the episode information for an actor's credit within an IMDb TV series. The format the information is expected is: "*episode count* episodes, *episode year start-episode year end*". Single episodes/years are also handled. For example:

- 124 episodes, 1999-2013
- 2 episodes, 2010
- 1 episode

Parameters **node** (Node) – A *Node* containing the episode information.

Returns The episode count, episode start year, and episode end year, or *None* if a value is not found.

Return type (int, int, int)

gunzip_file

pymdb.utils.**gunzip_file**(*infile*, *outfile=None*, *delete_infile=False*)

Unzips a gzip file and returns the unzipped filename.

Unzips the given gzipped file into the specified outfile, or a default outfile name. If the infile's filename ends with "*.gz*", the oufile will be the same filename with the gzip extension removed. The function is also capable of deleteing the gzipped infile afterwards.

Parameters

- **infile** (str) – The gzipped file's filename.
- **outfile** (str, optional) – The filename to unzip the infile to, or *None* to use the default filename.
- **delete_infile** (bool, optional) – Determine if the gzipped infile should be deleted after it is unzipped to the outfile.

Returns The outfile's filename for the case when the default filename was used.

Return type str

preprocess_list

pymdb.utils.**preprocess_list**(*lst*)
Process a row of data from the IMDb datasets.

Replaces all "\N" characters in the IMDb dataset with *None*.

Parameters **lst** (list of str) – A list of strings to process.

Returns A list of strings with all "\N" strings being set to *None*.

Return type list of str

remove_tags

pymdb.utils.**remove_tags**(*s*, *tag*)
Removes the specified opening and closing tags of the given type.

This method does not remove content between the tags, rather just the tags themselves. For example: "*td*" to remove all table column tags.

Parameters

- **s** (str) – The HTML to parse.
- **tag** (str) – The tag to be removed.

Returns A string with all of the given tags removed, but other HTML information intact.

Return type str

remove_tags_and_content

pymdb.utils.**remove_tags_and_content**(*s*, *tag*)
Removes all of the specified tags from the string including their children.

Greedily finds an opening and closing of specified tag and removes all content between the two. **Note**: Not intended to remove multiple sibling nodes with content in between.

Parameters

- **s** (str) – The HTML to parse.
- **tag** (str) – The tag to be removed.

Returns A string with all of the specified tags and their content removed.

Return type str

split_by_br

pymdb.utils.**split_by_br**(*s*)
Split a string by <*br*> tags.

Splits by replacing each <*br*> tag with a "\t" character and then splitting.

Parameters **s** (str) – A string containing <*br*> tags.

Returns A list of strings split around the <*br*> tags.

Return type list

trim_name

pymdb.utils.**trim_name**(*name*)
Used to trim roman numerals from names.

IMDb differentiates people's names that are the same with the format: *<name> (<Roman numeral>)*. This function removes the roman numerals and returns only the name.

Parameters **name** (str) – The name and roman numeral combination.

Returns The name with roman numerals removed, or *None* if name was *None*.

Return type str

trim_year

pymdb.utils.**trim_year**(*year*)
Used to trim roman numerals from year values.

IMDb differentiates movies of the same title and the same year with the format: *YYYY/<Roman numeral>*. This function removes the roman numerals and returns just the year value.

Parameters **year** (str) – The year and roman numeral combination.

Returns The year with roman numerals removed, or *None* if year was *None*.

Return type str

is_money_string

pymdb.utils.**is_money_string**(*s*)
Determine if a string is in a money format.

Determines if the string represents a monetary value, for example: $123,456,789.

Parameters **s** (str) – The monetary amount to check.

Returns If the string does represent a monetary value for not.

Return type bool

trim_money_string

pymdb.utils.**trim_money_string**(*s*)
Trims excess characters from a monetary value.

Only keeps the digits within a monetary value, such as trimming *$123,456* to *123456*. Trims dollar signs and commas.

Parameters **s** (str) – The monetary amount to trim.

Returns The same monetary amount with excess characters removed.

Return type str

get_denomination

pymdb.utils.**get_denomination**(*s*)

Returns the monetary denomination for the given monetary value.

Checks if the monetary value has one of the supported denominations. In the case it is a US dollar ($), the dollar sign character is replaced with "*USD*". Currently supported denominations:

- *GBP*
- *USD* ($)

Parameters **s** (str) – The monetary amount to retrieve the denomination from.

Returns The denomination type, or *None* if not a monetary value or supported denomination.

Return type str

to_bool

pymdb.utils.**to_bool**(*b*)

Convert a variable to a *boolean* type.

Parameters **b** – The object to convert.

Returns The *boolean* representation of the object.

Return type bool

to_datetime

pymdb.utils.**to_datetime**(*d*)

Convert a variable to a *datetime* object.

Checks various formats used in IMDb to convert the variable to a *datetime* object under those formats. The formats include:

- *%d %B %Y*
- *%Y*
- *%Y-%m-%d*

Parameters **d** (str) – A string to convert to a *datetime* object.

Returns A *datetime* object that was represented by the string, or *None* if *d* is *None*.

Return type datetime

Raises ValueError – If the string could not be converted.

5.2 Release Notes

- Release 0.2.1 (10 December 2019)
  - Fixed import of utils methods within the pymdb.models modules (**'#8'**_)
- Release 0.2.0 (29 November 2019)
  - Mapped *CreditScrape*'s member variable *job_title* into key values (#5)

  - Added module *search* to store different IMDb search results (#1)
  - Added method *~.scraper.PyMDbScraper.get_search_results* to *PyMDbScraper* to retrieve search results from IMDb based on keywords (#1)
  - Added utils method *trim_name* to trim names within IMDb search results (#1)
  - Added method *get_full_credits_as_dict* to *PyMDbScraper* to return a dictionary with *job_title* as the key for a list of *CreditScrape* objects (#4)
  - Added support for Python 3.8 by updating *selectolax* version to 0.2.3 (**'#7'**_)

- Release 0.1.1 (21 November 2019)
  - Changed the type of *TitleScrape*'s member variable *end_year* from `datetime` to `int` to be more consistent with other classes
  - Changed the name of *TitleScrape*'s member variable *title_text* to *display_title*
  - Added the member variable *known_for_titles* to *NameScrape* to store a person's known for titles as listed on their IMDb page
  - Added an optional parameter *include_known_for_titles* to *PyMDbScraper*'s method *get_name* since an extra request needs to be sent to retrieve data for *known_for_titles*
  - Renamed `get_full_credits` to *get_full_crew* (#3)
  - Combined *get_full_cast* and *get_full_crew* into a single helper method of the previously used name *get_full_credits* (#3)
  - Added new utils function *get_episode_info* to parse the episode information for an actor's credits on a TV series' IMDb page
  - Modified *TitleScrape*'s *top_cast* member variable to be a list of *CreditScrape* objects instead of a list of name IDs (#6)
- Release 0.1.0 (14 November 2019)
  - Initial pre-release of PyMDb
  - Introduction of Travis-CI at https://travis-ci.com/zembrodt/pymdb
  - Modification of unit tests to execute more quickly
- Release 0.0.3 (14 November 2019)
  - Renaming of *TitleBasics*'s member variable *runtime_minutes* to *runtime* to be more consistent with the naming in *TitleTechSpecsScrape*
- Release 0.0.2 (13 Novemer 2019)
  - Introduction of member variable *budget_denomination* in *TitleScrape* to specify the monetary denomination of member variable *budget*
  - Additional *utils* method *get_denomination*
  - Introduction of full documentation via Sphinx
- Release 0.0.1 (11 November 2019)
  - Initial stable release of PyMDb to PyPI with completed unit tests

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

Index

Symbols

A

C

G

I

N

P

R

S

T