
toktokkie
Release 0.19.0

Hermann Krumrey

Oct 24, 2019

CONTENTS

1	toktokkie package	3
1.1	Subpackages	3
1.2	Submodules	37
1.3	toktokkie.Directory module	37
1.4	toktokkie.exceptions module	38
1.5	Module contents	39
2	toktokkie	41
3	Indices and tables	43
	Python Module Index	45
	Index	47

Contents:

TOKTOKKIE PACKAGE

1.1 Subpackages

1.1.1 toktokkie.anithemes package

Submodules

toktokkie.anithemes.AniParam module

class toktokkie.anithemes.AniParam.**AniParam** (*show_name: str, mal_id: int, theme_type: str, song_name: str, episodes: str, media_url: str*)

Bases: object

Class that contains all relevant information for an anime theme song. Uses reddit and myanimelist to retrieve all the information # TODO Integrate AniDB, since it has way more info on artists etc

__init__ (*show_name: str, mal_id: int, theme_type: str, song_name: str, episodes: str, media_url: str*)

Initializes the anime theme. Loads missing data from the internet while initializing, so this might take up to a couple of seconds. :param show_name: The name of the show this song is a theme song for :param mal_id: The myanimelist ID of that show :param theme_type: The type of theme (example: OP, ED) :param song_name: The name of the song :param episodes: During which episodes the theme song played :param media_url: The URL to the video of this theme song

convert_to_mp3 ()

Converts the previously downloaded webm file to mp3 :return: None

download_webm ()

Downloads the .webm video of the anime theme song to the tmp directory :return: None

classmethod load_reddit_anithemes_wiki_info (*year: int, season: str, whitelist: Optional[List[str]] = None*) → List[toktokkie.anithemes.AniParam.AniParam]

Loads all theme songs for a specific season and year CAUTION: Currently season does nothing, it is recommended to use the whitelist parameter to limit the requests done :param year: The year for which to fetch the theme songs :param season: The season for which to fetch the theme songs :param whitelist: If provided, will ignore any series that are not

contained in the whitelist

Returns The fetched AniParam objects

```
logger = <Logger toktokkie.anithemes.AniTheme (WARNING)>
    Logger for this class
mal_cache = {}
    Cache for repeated myanimelist requests
```

Module contents

1.1.2 toktokkie.check package

Submodules

toktokkie.check.BookChecker module

Copyright 2015 Hermann Krumrey <hermann@krumreyh.com>

This file is part of toktokkie.

toktokkie is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

toktokkie is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with toktokkie. If not, see <<http://www.gnu.org/licenses/>>.

```
class toktokkie.check.BookChecker.BookChecker (metadata:          tok-
                                             tokkie.metadata.Metadata.Metadata,
                                             show_warnings: bool, fix_interactively:
                                             bool, config: Dict[str, Any])
```

Bases: *toktokkie.check.Checker.Checker*

Class that check Book media for consistency

toktokkie.check.BookSeriesChecker module

Copyright 2015 Hermann Krumrey <hermann@krumreyh.com>

This file is part of toktokkie.

toktokkie is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

toktokkie is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with toktokkie. If not, see <<http://www.gnu.org/licenses/>>.

```
class toktokkie.check.BookSeriesChecker.BookSeriesChecker (metadata:          tok-
                                             tokkie.metadata.Metadata.Metadata,
                                             show_warnings: bool,
                                             fix_interactively: bool,
                                             config: Dict[str, Any])
```

Bases: *toktokkie.check.Checker.Checker*

Class that check Book Series media for consistency

check () → bool

Performs sanity checks and prints out anything that's wrong :return: The result of the check

toktokkie.check.Checker module

Copyright 2015 Hermann Krumrey <hermann@krumreyh.com>

This file is part of toktokkie.

toktokkie is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

toktokkie is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with toktokkie. If not, see <<http://www.gnu.org/licenses/>>.

```
class toktokkie.check.Checker.Checker (metadata: toktokkie.metadata.Metadata.Metadata,
                                         show_warnings: bool, fix_interactively: bool, config:
                                         Dict[str, Any])
```

Bases: object

Class that performs checks on Metadata objects

```
__init__ (metadata: toktokkie.metadata.Metadata.Metadata, show_warnings: bool, fix_interactively:
          bool, config: Dict[str, Any])
```

Initializes the checker :param metadata: The metadata to check :param show_warnings: Whether or not to show warnings :param fix_interactively: Can be set to True to

interactively fix some errors

Parameters config – A dictionary containing configuration options

check () → bool

Performs sanity checks and prints out anything that's wrong :return: The result of the check

error (*text: str*) → bool

Prints a black-on-red error message :param text: The text to print :return: False

warn (*text: str*) → bool

Prints a black-on-yellow warning message :param text: The text to print :return: True if not showing warnings, else False

toktokkie.check.MangaChecker module

Copyright 2015 Hermann Krumrey <hermann@krumreyh.com>

This file is part of toktokkie.

toktokkie is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

toktokkie is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with toktokkie. If not, see <<http://www.gnu.org/licenses/>>.

```
class toktokkie.check.MangaChecker.MangaChecker (metadata: tok-  
tokkie.metadata.Metadata.Metadata,  
show_warnings: bool,  
fix_interactively: bool, config: Dict[str,  
Any])
```

Bases: *toktokkie.check.Checker.Checker*

Class that check Manga media for consistency

check () → bool

Performs sanity checks and prints out anything that's wrong :return: The result of the check

toktokkie.check.MovieChecker module

Copyright 2015 Hermann Krumrey <hermann@krumreyh.com>

This file is part of toktokkie.

toktokkie is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

toktokkie is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with toktokkie. If not, see <<http://www.gnu.org/licenses/>>.

```
class toktokkie.check.MovieChecker.MovieChecker (metadata: tok-  
tokkie.metadata.Metadata.Metadata,  
show_warnings: bool,  
fix_interactively: bool, config: Dict[str,  
Any])
```

Bases: *toktokkie.check.Checker.Checker*

Class that check Movie media for consistency

toktokkie.check.TvSeriesChecker module

Copyright 2015 Hermann Krumrey <hermann@krumreyh.com>

This file is part of toktokkie.

toktokkie is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

toktokkie is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with toktokkie. If not, see <<http://www.gnu.org/licenses/>>.

```
class toktokkie.check.TvSeriesChecker.TvSeriesChecker (metadata:          tok-
                                                    tokkie.metadata.Metadata.Metadata,
                                                    show_warnings:          bool,
                                                    fix_interactively:      bool,
                                                    config: Dict[str, Any])
```

Bases: *toktokkie.check.Checker.Checker*

Class that check TV Series media for consistency

check () → bool

Performs sanity checks and prints out anything that's wrong :return: The result of the check

toktokkie.check.VisualNovelChecker module

Copyright 2015 Hermann Krumrey <hermann@krumreyh.com>

This file is part of toktokkie.

toktokkie is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

toktokkie is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with toktokkie. If not, see <<http://www.gnu.org/licenses/>>.

```
class toktokkie.check.VisualNovelChecker.VisualNovelChecker (metadata:          tok-
                                                    tokkie.metadata.Metadata.Metadata,
                                                    show_warnings:
                                                    bool,
                                                    fix_interactively:
                                                    bool,          config:
                                                    Dict[str, Any])
```

Bases: *toktokkie.check.Checker.Checker*

Class that checks Visual Novels for consistency

check () → bool

Performs sanity checks and prints out anything that's wrong :return: The result of the check

toktokkie.check.map module

Copyright 2015 Hermann Krumrey <hermann@krumreyh.com>

This file is part of toktokkie.

toktokkie is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

toktokkie is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with toktokkie. If not, see <<http://www.gnu.org/licenses/>>.

```
toktokkie.check.map.checker_map = {<MediaType.BOOK: 'book':>: <class 'toktokkie.check.Book
```

A dictionary mapping media types to checker classes

Module contents

Copyright 2015 Hermann Krumrey <hermann@krumreyh.com>

This file is part of toktokkie.

toktokkie is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

toktokkie is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with toktokkie. If not, see <<http://www.gnu.org/licenses/>>.

1.1.3 toktokkie.gui package

Subpackages

toktokkie.gui.pyuic package

Submodules

toktokkie.gui.pyuic.book_widget module

```
class toktokkie.gui.pyuic.book_widget.Ui_BookWidget
    Bases: object

    retranslateUi (BookWidget)
    setupUi (BookWidget)
```

toktokkie.gui.pyuic.main module

```
class toktokkie.gui.pyuic.main.Ui_MainWindow
    Bases: object

    retranslateUi (MainWindow)
    setupUi (MainWindow)
```

toktokkie.gui.pyuic.movie_widget module

```
class toktokkie.gui.pyuic.movie_widget.Ui_MovieWidget
    Bases: object

    retranslateUi (MovieWidget)
    setupUi (MovieWidget)
```

toktokkie.gui.pyuic.tv_season_widget module

```
class toktokkie.gui.pyuic.tv_season_widget.Ui_TvSeasonWidget
    Bases: object
    retranslateUi (TvSeasonWidget)
    setupUi (TvSeasonWidget)
```

toktokkie.gui.pyuic.tv_show_widget module

```
class toktokkie.gui.pyuic.tv_show_widget.Ui_TvSeriesWidget
    Bases: object
    retranslateUi (TvSeriesWidget)
    setupUi (TvSeriesWidget)
```

Module contents**toktokkie.gui.qt_designer package****Module contents****toktokkie.gui.widgets package****Submodules****toktokkie.gui.widgets.BookWidget module**

```
class toktokkie.gui.widgets.BookWidget.BookWidget (parent:
    PyQt5.QtWidgets.QMainWindow)
    Bases: PyQt5.QtWidgets.QWidget, toktokkie.gui.pyuic.book_widget.Ui_BookWidget
    Class that defines the behaviour of a Book widget
    __init__ (parent: PyQt5.QtWidgets.QMainWindow)
        Initializes the widget :param parent: The parent window
    initialize_buttons ()
        Initializes the widget buttons :return: None
    open_directory ()
        Opens the directory of the metadata :return: None
    set_icon_image ()
        Sets the icon image of a Book Widget :return: None
    set_metadata (metadata: toktokkie.metadata.types.Book.Book)
        Displays the information about a metadata :param metadata: The metadata to display :return: None
    write_changes ()
        Writes the changes to the metadata to disk :return: None
```

toktokkie.gui.widgets.MovieWidget module

class toktokkie.gui.widgets.MovieWidget.**MovieWidget** (*parent:*
PyQt5.QtWidgets.QMainWindow)
Bases: PyQt5.QtWidgets.QWidget, *toktokkie.gui.pyuic.movie_widget.*
Ui_MovieWidget

Class that defines the behaviour of a Movie widget

__init__ (*parent: PyQt5.QtWidgets.QMainWindow*)
Initializes the widget :param parent: The parent window

initialize_buttons ()
Initializes the widget buttons :return: None

open_directory ()
Opens the directory of the metadata :return: None

set_icon_image ()
Sets the icon image of a Movie Widget :return: None

set_metadata (*metadata: toktokkie.metadata.types.Movie.Movie*)
Displays the information about a metadata :param metadata: The metadata to display :return: None

write_changes ()
Writes the changes to the metadata to disk :return: None

toktokkie.gui.widgets.TvSeasonWidget module

class toktokkie.gui.widgets.TvSeasonWidget.**TvSeasonWidget** (*parent:*
PyQt5.QtWidgets.QMainWindow)
Bases: PyQt5.QtWidgets.QWidget, *toktokkie.gui.pyuic.tv_season_widget.*
Ui_TvSeasonWidget

Class that defines the behaviour of a TV Series widget

__init__ (*parent: PyQt5.QtWidgets.QMainWindow*)
Initializes the widget :param parent: The parent window

initialize_buttons ()
Initializes the widget buttons :return: None

load_episode_files ()
Displays the episode files :return: None

load_tvdb_info ()
Loads TVDB data. :return: None

open_directory ()
Opens the directory of the season metadata :return: None

open_tvdb_url ()
Opens the tvdb URL in a web browser :return: None

set_icon_image ()
Sets the icon image of a TV Season Widget :return: None

set_metadata (*metadata: toktokkie.metadata.types.TvSeries.TvSeries, season: str*)
Displays the information about a metadata :param metadata: The metadata of the series to display :param
season: The name of the season to display :return: None

write_changes ()
Writes the changes to the metadata to disk :return: None

toktokkie.gui.widgets.TvSeriesWidget module

class toktokkie.gui.widgets.TvSeriesWidget.**TvSeriesWidget** (*parent: PyQt5.QtWidgets.QMainWindow*)
PyQt5.QtWidgets.QMainWindow
 Bases: PyQt5.QtWidgets.QWidget, *toktokkie.gui.pyuic.tv_show_widget.Ui_TvSeriesWidget*
 Class that defines the behaviour of a TV Series widget

__init__ (*parent: PyQt5.QtWidgets.QMainWindow*)
Initializes the widget :param parent: The parent window

initialize_buttons ()
Initializes the widget buttons :return: None

load_tvdb_info ()
Loads TVDB data. :return: None

open_directory ()
Opens the directory of the metadata :return: None

open_tvdb_url ()
Opens the tvdb URL in a web browser :return: None

set_icon_image ()
Sets the icon image of a TV Series Widget :return: None

set_metadata (*metadata: toktokkie.metadata.types.TvSeries.TvSeries*)
Displays the information about a metadata :param metadata: The metadata to display :return: None

write_changes ()
Writes the changes to the metadata to disk :return: None

Module contents

Submodules

toktokkie.gui.MainWindow module

class toktokkie.gui.MainWindow.**MainWindow**
 Bases: PyQt5.QtWidgets.QMainWindow, *toktokkie.gui.pyuic.main.Ui_MainWindow*
 The Main Window of the application

__init__ ()
Initializes the main window

reload ()
Refreshes the data shown :return: None

write_config ()
Writes the config file to disk :return:

Module contents

1.1.4 toktokkie.iconizing package

Subpackages

toktokkie.iconizing.procedures package

Submodules

toktokkie.iconizing.procedures.GnomeProcedure module

class toktokkie.iconizing.procedures.GnomeProcedure.**GnomeProcedure**

Bases: *toktokkie.iconizing.procedures.Procedure.Procedure*

Iconizing Procedure that uses the gio command line tool to set the icon file in gnome-based systems

classmethod **iconize** (*directory: str, icon_path_no_ext*)

Iconizes the directory using gio :param directory: The directory to iconize :param icon_path_no_ext: The icon file without a file extension.

.png will be appended

Returns None

classmethod **is_applicable** () → bool

Checks if this procedure is applicable to the current system. The Gnome procedure is applicable if the system is running Linux as well as a Gnome environment, like the Gnome DE or Cinnamon :return: True if applicable, else False

toktokkie.iconizing.procedures.NoopProcedure module

class toktokkie.iconizing.procedures.NoopProcedure.**NoopProcedure**

Bases: *toktokkie.iconizing.procedures.Procedure.Procedure*

Iconizing Procedure that does nothing. Used as a fallback when no other procedures are available

classmethod **iconize** (*directory: str, icon_path_no_ext*)

Doesn't do anything :param directory: The directory to iconize :param icon_path_no_ext: The icon file without a file extension.

.png will be appended

Returns None

classmethod **is_applicable** () → bool

Checks if this procedure is applicable to the current system. NOOP Procedure is always applicable :return: True if applicable, else False

toktokkie.iconizing.procedures.Procedure module

class toktokkie.iconizing.procedures.Procedure.**Procedure**

Bases: object

Interface that defines the methods an iconizing procedure must provide

```
classmethod iconize (directory: str, icon_path_no_ext: str)
    Iconizes a directory with an icon file :param directory: The directory to iconize :param icon_path_no_ext:
    The icon file without a file extension. :return: None

classmethod is_applicable () → bool
    Checks if this procedure is applicable to the current system :return: True if applicable, else False

name = 'procedure'
    The name/identifier of the procedure
```

Module contents

Copyright 2015 Hermann Krumrey <hermann@krumreyh.com>

This file is part of toktokkie.

toktokkie is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

toktokkie is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with toktokkie. If not, see <<http://www.gnu.org/licenses/>>.

Submodules

toktokkie.iconizing.Iconizer module

```
class toktokkie.iconizing.Iconizer.Iconizer (path: str, icon_location: str, procedure: tok-
tokkie.iconizing.procedures.Procedure.Procedure
= None)
```

Bases: object

Class that handles iconizing directories

```
__init__ (path: str, icon_location: str, procedure: toktokkie.iconizing.procedures.Procedure.Procedure
= None)
```

Initializes the iconizer :param path: The path to the directory to iconize :param icon_location: The location of the icons :param procedure: The procedure to use for iconizing.

```
iconize ()
```

Iconizes the directory :return: None

Module contents

```
toktokkie.iconizing.default_procedure () → Type[toktokkie.iconizing.procedures.Procedure.Procedure]
    Checks all available procedures for eligibility :return: The eligible procedure or None if none were found
```

1.1.5 toktokkie.metadata package

Subpackages

toktokkie.metadata.components package

Submodules

toktokkie.metadata.components.MetadataPart module

```
class toktokkie.metadata.components.MetadataPart.MetadataPart (parent: tok-  
tokkie.metadata.Metadata.Metadata,  
json_data:  
Dict[str, Any])
```

Bases: object

Class that defines various functionality of a metadata part

```
__init__ (parent: toktokkie.metadata.Metadata.Metadata, json_data: Dict[str, Any])
```

Initializes the MetadataPart object using JSON data :param parent: The parent metadata :param json_data: The JSON data used to generate the MetadataPart :raises InvalidMetadataException: If any errors were encountered

while generating the object

property ids

Returns A dictionary containing lists of IDs mapped to ID types

validate ()

Validates the JSON data of the metadata part :return: None :raises InvalidMetadataException: If something is wrong

with the JSON data

toktokkie.metadata.components.TvEpisode module

```
class toktokkie.metadata.components.TvEpisode.TvEpisode (json_data: Dict[str, Any])
```

Bases: object

Class that models a TV Episode

```
__init__ (json_data: Dict[str, Any])
```

Initializes the TvEpisode object :param json_data: The JSON data from which to generate the episode from :raises InvalidMetadataException: If the provided JSON is invalid

toktokkie.metadata.components.TvEpisodeRange module

```
class toktokkie.metadata.components.TvEpisodeRange.TvEpisodeRange (json_data:  
Dict[str,  
Any])
```

Bases: object

Class that models a TV Episode Range

```
__init__ (json_data: Dict[str, Any])
```

Initializes the TvEpisodeRange object :param json_data: The JSON data from which to generate the episode range from

Raises InvalidMetadataException – If the provided JSON is invalid

property episodes

Returns A list of episodes included in this episode range

toktokkie.metadata.components.TvSeason module

class toktokkie.metadata.components.TvSeason.**TvSeason** (*parent:* *toktokkie.metadata.Metadata.Metadata*, *json_data:* *Dict[str, Any]*)

Bases: *toktokkie.metadata.components.MetadataPart.MetadataPart*

Class that models a single tv season

is_spinoff () → bool

Returns Whether or not this season is a spinoff

property name

Returns The name of the season directory

property path

Returns The path to the season directory

property season_number

Returns The season number of the season

property tvdb_id

Returns The TVDB ID of the TV season

validate ()

Validates the JSON data of the tv season. :return: None :raises InvalidMetadataException: If something is wrong

with the JSON data

Module contents**toktokkie.metadata.ids package****Submodules****toktokkie.metadata.ids.IdFetcher module**

class toktokkie.metadata.ids.IdFetcher.**IdFetcher** (*name:* *str*, *media_type:* *toktokkie.metadata.MediaType.MediaType*)

Bases: object

__init__ (*name:* *str*, *media_type:* *toktokkie.metadata.MediaType.MediaType*)

Initialize self. See help(type(self)) for accurate signature.

fetch_ids (*id_type:* *toktokkie.metadata.ids.IdType.IdType*, *other_ids:* *Dict[toktokkie.metadata.ids.IdType.IdType, List[str]]*) → Optional[List[str]]

logger = <Logger toktokkie.metadata.ids.IdFetcher (WARNING)>

toktokkie.metadata.ids.IdType module

```
class toktokkie.metadata.ids.IdType.IdType
    Bases: enum.Enum

    Enumeration of all possible ID types

    ANILIST = 'anilist'

    IMDB = 'imdb'

    ISBN = 'isbn'

    KITSU = 'kitsu'

    MANGADEX = 'mangadex'

    MUSICBRAINZ = 'musicbrainz'

    MYANIMELIST = 'myanimelist'

    TVDB = 'tvdb'

    VNDB = 'vndb'
```

toktokkie.metadata.ids.functions module

```
toktokkie.metadata.ids.functions.fill_ids (ids: Dict[toktokkie.metadata.ids.IdType.IdType,
                                                List[str]],
                                             valid_ids: List[toktokkie.metadata.ids.IdType.IdType],
                                             _parent_ids: Optional[Dict[toktokkie.metadata.ids.IdType.IdType,
                                                                    List[str]]] = None) →
                                                Dict[toktokkie.metadata.ids.IdType.IdType,
                                                List[str]]
```

Fills in any missing id type key for an ID dictionary :param ids: The ID dictionary to fill :param valid_ids: Any valid ID types :param _parent_ids: Optionally provided dictionary of parent IDs.

The values of this dictionary will be entered into the filled dictionary if they aren't defined already

Returns The filled dictionary

```
toktokkie.metadata.ids.functions.minimize_ids (ids: Dict[toktokkie.metadata.ids.IdType.IdType,
                                                         List[str]],
                                                _parent_ids: Optional[Dict[toktokkie.metadata.ids.IdType.IdType,
                                                                           List[str]]] = None) →
                                                Dict[toktokkie.metadata.ids.IdType.IdType,
                                                List[str]]
```

Removes redundant and empty IDs from a dictionary of IDs :param ids: The dictionary to minimize :param _parent_ids: Optionally provide a parent's IDs. These IDs will

be removed from the child dictionary if the values are the same

Returns The minimized dictionary

`toktokkie.metadata.ids.functions.objectify_ids` (*ids*: `Dict[str, List[str]]`) → `Dict[toktokkie.metadata.ids.IdType.IdType, List[str]]`
 Converts the keys in an ID dictionary using string keys to `IdType` objects :param *ids*: The ID dictionary to convert :return: The converted ID dictionary

`toktokkie.metadata.ids.functions.stringify_ids` (*ids*: `Dict[toktokkie.metadata.ids.IdType.IdType, List[str]]`) → `Dict[str, List[str]]`
 Converts the keys in an ID dictionary to strings :param *ids*: The ID dictionary to convert :return: The converted ID dictionary

toktokkie.metadata.ids.mappings module

`toktokkie.metadata.ids.mappings.id_prompt_order` = [`<IdType.TVDB: 'tvdb'>`, `<IdType.IMDB: 'imdb'>`]
 The order in which ID types should be prompted

`toktokkie.metadata.ids.mappings.int_id_types` = [`<IdType.MYANIMELIST: 'myanimelist'>`, `<IdType.ISBN: 'isbn'>`]
 ID Types that are always integer values

`toktokkie.metadata.ids.mappings.literature_media_types` = [`<MediaType.BOOK: 'book'>`, `<MediaType.MANGA: 'manga'>`]
 Media types that represent literature

`toktokkie.metadata.ids.mappings.required_id_types` = {`<MediaType.BOOK: 'book'>`: [], `<MediaType.MANGA: 'manga'>`: []}
 Required ID Types for the various Media Types

`toktokkie.metadata.ids.mappings.theme_song_ids` = [`<IdType.TVDB: 'tvdb'>`, `<IdType.MYANIMELIST: 'myanimelist'>`]
 ID types that can be associated with theme songs

`toktokkie.metadata.ids.mappings.valid_id_types` = {`<MediaType.BOOK: 'book'>`: [`<IdType.ISBN: 'isbn'>`], `<MediaType.MANGA: 'manga'>`: [`<IdType.TVDB: 'tvdb'>`]}
 Valid ID types for the various Media types

Module contents

toktokkie.metadata.prompt package

Submodules

toktokkie.metadata.prompt.Prompter module

class `toktokkie.metadata.prompt.Prompter.Prompter` (*directory*: `str`, *media_type*: `toktokkie.metadata.MediaType.MediaType`)
 Bases: `object`
 Class that handles prompting for metadata information

__init__ (*directory*: `str`, *media_type*: `toktokkie.metadata.MediaType.MediaType`)
 Initializes the Prompter object :param *directory*: The directory for which to prompt for metadata for :param *media_type*: The media type of the directory's content

logger = `<Logger toktokkie.metadata.prompt.Prompter (WARNING)>`
 Logger for this class

prompt () → `Dict[str, Any]`
 Prompts the user for metadata information :return: The metadata JSON that resulted from the user's input

Module contents

toktokkie.metadata.schema package

Submodules

toktokkie.metadata.schema.SchemaBuilder module

class toktokkie.metadata.schema.SchemaBuilder.**SchemaBuilder** (*media_type: tok-*
tokkie.metadata.MediaType.MediaType)

Bases: object

Class that defines JSON schemas for the various media types

__init__ (*media_type: toktokkie.metadata.MediaType.MediaType*)

Initializes the SchemaBuilder using a media type :param media_type: The media type for which to generate
a JSON schema

build_schema () → Dict[str, Any]

Generates the JSON schema :return: The JSON schema

logger = <Logger toktokkie.metadata.schema.SchemaBuilder (WARNING)>

Logger object for this class

Module contents

toktokkie.metadata.types package

Subpackages

toktokkie.metadata.types.components package

Submodules

toktokkie.metadata.types.components.BookVolume module

class toktokkie.metadata.types.components.BookVolume.**BookVolume** (*volume_number:*
int, volume_name:
str, path: str,
parent_ids:
Dict[toktokkie.metadata.ids.IdType.IdType,
List[str]],
json_data:
Dict[str,
Dict[str,
List[str]]])

Bases: object

Class that models a Volume in a Book Series

```

__init__(volume_number: int, volume_name: str, path: str, parent_ids:
         Dict[toktokkie.metadata.ids.IdType.IdType, List[str]], json_data: Dict[str, Dict[str,
         List[str]]])

```

Initializes the Book Volume :param volume_number: The volume number :param volume_name: The volume name :param path: The path to the volume file :param parent_ids: The IDs of the parent BookSeries object :param json_data: The JSON data for the book volume containing the IDs for this specific volume

property json

Returns A JSON-compatible dictionary representing this object

toktokkie.metadata.types.components.MusicAlbum module

toktokkie.metadata.types.components.MusicThemeSong module

toktokkie.metadata.types.components.TvSeason module

Module contents

Submodules

toktokkie.metadata.types.Book module

```

class toktokkie.metadata.types.Book.Book(directory_path: str, json_data: Optional[Dict[str, Any]] = None, no_validation: bool = False)

```

Bases: *toktokkie.metadata.Metadata.Metadata*

Metadata class that model a Book

```

classmethod media_type() → toktokkie.metadata.MediaType.MediaType

```

Returns The media type of the Metadata class

toktokkie.metadata.types.BookSeries module

```

class toktokkie.metadata.types.BookSeries.BookSeries(directory_path: str, json_data: Optional[Dict[str, Any]] = None, no_validation: bool = False)

```

Bases: *toktokkie.metadata.types.Book.Book*

Metadata class that models a Book series

```

classmethod media_type() → toktokkie.metadata.MediaType.MediaType

```

Returns The media type of the Metadata class

property volumes

Returns A list of book volumes for this book series

toktokkie.metadata.types.Manga module

class toktokkie.metadata.types.Manga.**Manga** (*directory_path: str, json_data: Optional[Dict[str, Any]] = None, no_validation: bool = False*)

Bases: *toktokkie.metadata.Metadata.Metadata*

Metadata class that models a Manga series

property main_path

The path to the main manga directory :return: The path

classmethod media_type () → toktokkie.metadata.MediaType.MediaType

Returns The media type of the Metadata class

property special_chapters

Returns A list of special chapter identifiers for this series

property special_path

The path to the special manga directory :return: The path or None if it does not exist

toktokkie.metadata.types.Movie module

class toktokkie.metadata.types.Movie.**Movie** (*directory_path: str, json_data: Optional[Dict[str, Any]] = None, no_validation: bool = False*)

Bases: *toktokkie.metadata.Metadata.Metadata*

Metadata class that model a Movie

classmethod media_type () → toktokkie.metadata.MediaType.MediaType

Returns The media type of the Metadata class

toktokkie.metadata.types.MusicArtist module

class toktokkie.metadata.types.MusicArtist.**MusicArtist** (*directory_path: str, json_data: Optional[Dict[str, Any]] = None, no_validation: bool = False*)

Bases: *toktokkie.metadata.Metadata.Metadata*

Metadata class that model a music artist

add_album (*album_data: Dict[str, Any]*)

Adds an album to the metadata :param album_data: The album metadata to add :return: None

property albums

Returns All 'album' album metadata

property all_albums

Returns All album metadata

classmethod media_type () → toktokkie.metadata.MediaType.MediaType

Returns The media type of the Metadata class

property singles**Returns** All 'single' album metadata**property theme_songs****Returns** All theme songs for this music artist**toktokkie.metadata.types.TvSeries module**

```
class toktokkie.metadata.types.TvSeries.TvSeries (directory_path: str, json_data:
Optional[Dict[str, Any]] = None,
no_validation: bool = False)
```

Bases: *toktokkie.metadata.Metadata.Metadata*

Metadata class that model a TV Series

add_exclude (id_type: toktokkie.metadata.ids.IdType.IdType, season: int, episode: int)Adds an excluded episode :param id_type: The ID type :param season: The season of the excluded episode
:param episode: The excluded episode number :return: None**add_multi_episode** (id_type: toktokkie.metadata.ids.IdType.IdType, season: int, start_episode: int,
end_episode: int)Adds a multi-episode :param id_type: The ID type :param season: The season of the multi episode :param
start_episode: The start episode :param end_episode: The end episode :return: None**property excludes**Generates data for episodes to be excluded during renaming etc. :return A dictionary mapping episode
anithemes to seasons and id types

Form: {idtype: {season: [ep1, ep2]}}

get_episode_files () → Dict[str, Dict[int, List[str]]]Generates a dictionary categorizing internal episode files for further processing. A current limitation is,
that only a single tvdb ID per season is supported. It's currently not planned to lift this limitation, as no
valid use case for more than one tvdb ID per season has come up. The episode lists are sorted by their
episode name. :return: The generated dictionary. It will have the following form:

{tvdb_id: {season_number: [episode_files]}}

get_season (season_name: str) → toktokkie.metadata.components.TvSeason.TvSeasonRetrieves a single season for a provided season name :param season_name: The name of the season :return:
The season :raises KeyError: If the season could not be found**classmethod media_type** () → toktokkie.metadata.MediaType.MediaType**Returns** The media type of the Metadata class**property multi_episodes****Returns** A dictionary mapping lists of multi-episodes to id types Form: {idtype: {season: {start:
end}}}**property season_start_overrides****Returns** A dictionary mapping episodes that override a season starting point to ID types Form:
{idtype: {season: episode}}**property seasons****Returns** A list of TV seasons**property tvdb_id**

Returns The TVDB ID of the TV Series

validate()

Validates the JSON data to make sure everything has valid values :raises InvalidMetadataException: If any errors were encountered :return: None

toktokkie.metadata.types.VisualNovel module

```
class toktokkie.metadata.types.VisualNovel.VisualNovel (directory_path: str,  
json_data: Optional[Dict[str, Any]] =  
None, no_validation: bool =  
False)
```

Bases: *toktokkie.metadata.Metadata.Metadata*

Metadata class that model a visual novel

property cgs

Generates a dictionary of paths to CG images :return: A dictionary mapping the various CG images to their

respective directory identifiers

property eds

Returns a list of ending theme videos

property has_cgs

Returns Whether or not the Visual Novel has a CG gallery

property has_ed

Returns Whether or not the Visual Novel has an ending theme

property has_op

Returns Whether or not the Visual Novel has an opening theme

property has_ost

Returns Whether or not the Visual Novel has an OST

classmethod media_type() → toktokkie.metadata.MediaType.MediaType

Returns The media type of the Metadata class

property ops

Returns a list of opening theme videos

property ost

Returns a list of files for the OST of a visual novel

Module contents

Submodules

toktokkie.metadata.MediaType module

class toktokkie.metadata.MediaType.**MediaType**

Bases: enum.Enum

Enumeration that defines all possible media types

BOOK = 'book'

BOOK_SERIES = 'book_series'

MANGA = 'manga'

MOVIE = 'movie'

MUSIC_ARTIST = 'music'

TV_SERIES = 'tv'

VISUAL_NOVEL = 'visual_novel'

toktokkie.metadata.Metadata module

class toktokkie.metadata.Metadata.**Metadata** (*directory_path: str, json_data: Optional[Dict[str, Any]] = None, no_validation: bool = False*)

Bases: object

Class that acts as the base class for all possible metadata types

__init__ (*directory_path: str, json_data: Optional[Dict[str, Any]] = None, no_validation: bool = False*)

Initializes the metadata object using JSON data :param directory_path: The directory of the media for which to

generate the metadata

Parameters

- **json_data** – Optional metadata JSON. Will be used instead of info.json metadata if provided
- **no_validation** – Skips JSON validation if True

Raises InvalidMetadataException – if the metadata could not be parsed correctly

get_icon_file (*name: str*) → Optional[str]

Retrieves the path to an icon file, if it exists :param name: The name of the icon. :return: The path to the icon file or None, if the file does not exist

property ids

Returns A dictionary containing lists of IDs mapped to ID types

logger = <Logger toktokkie.metadata.Metadata (WARNING)>

Logger for the metadata class

classmethod media_type () → toktokkie.metadata.MediaType.MediaType

Returns The media type of the Metadata class

property name

Returns The name of the media

print_folder_icon_source()

Prints a message with a URL for possible folder icons on deviantart :return: None

classmethod prompt (*directory_path: str*) → toktokkie.metadata.Metadata.Metadata

Generates a new Metadata object using prompts for a directory :param directory_path: The path to the directory for which to generate

the metadata object

Returns The generated metadata object

set_ids (*id_type: toktokkie.metadata.ids.IdType.IdType, ids: List[str]*)

Sets IDs for one ID type to the metadata :param id_type: The id type :param ids: The IDs to set :return: None

property tags

Returns A list of tags

validate()

Validates the JSON data to make sure everything has valid values :raises InvalidMetadataException: If any errors were encountered :return: None

write()

Writes the metadata to the metadata file :return: None

toktokkie.metadata.functions module

toktokkie.metadata.functions.**create_metadata** (*directory: str, media_type: Union[str, toktokkie.metadata.MediaType.MediaType]*)
→ toktokkie.metadata.Metadata.Metadata

Generates a new metadata object using user prompts :param directory: The directory for which to generate the metadata :param media_type: The media type of the metadata :return: The generated metadata

toktokkie.metadata.functions.**get_metadata** (*directory: str*) → toktokkie.metadata.Metadata.Metadata

Automatically resolves the metadata of a directory :param directory: The directory for which to generate the metadata :return: The generated metadata :raises InvalidMetadataException: If the metadata is invalid

toktokkie.metadata.functions.**get_metadata_class** (*media_type: Union[str, toktokkie.metadata.MediaType.MediaType]*)
→ Type[toktokkie.metadata.Metadata.Metadata]

Retrieves the metadata class for a given media type :param media_type: The media type for which to get the metadata class :return: The metadata class

Module contents

1.1.6 toktokkie.renaming package

Submodules

toktokkie.renaming.RenameOperation module

class toktokkie.renaming.RenameOperation.**RenameOperation** (*source_path: str, new_name: str*)

Bases: object

Class that models a renaming operation

```

__init__(source_path: str, new_name: str)
    Initializes the RenameOperation object :param source_path: The currently existing path to the
    file/directory :param new_name: The new name of the file/directory

rename()
    Renames the episode file to the new name :return: None

static sanitize(parent: str, filename: str) → str
    Replaces all illegal file system characters with valid ones. Also, limits the length of the resulting file path
    to 250 characters, if at all possible :param parent: The parent directory :param filename: The filename to
    sanitize :return: The sanitized string

```

toktokkie.renaming.Renamer module

```

class toktokkie.renaming.Renamer.Renamer(metadata: tok-
                                             tokkie.metadata.Metadata.Metadata)
    Bases: object
    Class that handles the renaming of a media directory's content

    __init__(metadata: toktokkie.metadata.Metadata.Metadata)
        Initializes the Renamer. ValueErrors will be thrown if an error is encountered at any point during the
        initialization process :param metadata: The metadata to use for information

    get_active_operations() → List[toktokkie.renaming.RenameOperation.RenameOperation]
        Returns Any rename operations whose source and destination paths differ.

    load_title_name(id_override: Optional[Dict[toktokkie.metadata.ids.IdType.IdType, List[str]]] =
        None) → str
        Loads the title name for the metadata object :param id_override: Uses other IDs than in the metadata
        :return: The title the metadata should have

    property path
        Returns The path to the media directory

    rename(noconfirm: bool)
        Renames the contained files according to the naming schema. :param noconfirm: Skips the confirmation
        phase if True :return: None

```

toktokkie.renaming.functions module

```

toktokkie.renaming.functions.generate_tv_episode_filename(original_file: str,
                                                         series_name: str,
                                                         season_number: int,
                                                         episode_number:
                                                         int, episode_name:
                                                         str, multi_end: Op-
                                                         tional[int] = None)
    Generates an episode name for a given episode :param original_file: The original file. Used to get the file
    extension :param series_name: The name of the series :param season_number: The season number :param
    episode_name: The episode name :param episode_number: The episode number :param multi_end: Can be
    provided to create a multi-episode range :return: The generated episode name

toktokkie.renaming.functions.load_tvdb_episode_name(tvdb_id: str, season_number:
                                                     int, episode_number: int,
                                                     multi_end: Optional[int] =
                                                     None) → str

```

Loads an episode name from TVDB :param tvdb_id: The TVDB ID for the episode's series :param season_number: The season number :param episode_number: The episode number :param multi_end: If provided, will generate a name for a range of episodes

Returns The TVDB name

Module contents

1.1.7 toktokkie.scripts package

Submodules

toktokkie.scripts.AlbumArtFetchCommand module

class toktokkie.scripts.AlbumArtFetchCommand.**AlbumArtFetchCommand** (*args: argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the album-art-fetch command

execute ()
Executes the commands :return: None

classmethod name () → str
Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)
Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.AnilistOpenCommand module

class toktokkie.scripts.AnilistOpenCommand.**AnilistOpenCommand** (*args: argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the anilist-open command

execute ()
Executes the command :return: None

classmethod name () → str
Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)
Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.AnimeThemeDlCommand module

class toktokkie.scripts.AnimeThemeDlCommand.**AnimeThemeDlCommand** (*args: argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the anitheme-dl command

execute ()
Executes the commands :return: None

generate_artist_album_structure (*selected_songs: List[toktokkie.anithemes.AniTheme.AniTheme]*)

Generates a folder structure for OPs and EDs following the schema: Artist - Album

- Song

Songs are copied from the mp3 directory. :param selected_songs: The song data :return: None

handle_excludes (*selected_songs: List[toktokkie.anithemes.AniTheme.AniTheme]*) → List[toktokkie.anithemes.AniTheme.AniTheme]

Allows the user to exclude certain songs from being downloaded Deletes any files that may already exist for excluded songs :param selected_songs: All currently selected songs :return: The selected songs minus any excluded songs

load_titles (*year: int, season: str, include_previous_season: bool = True*) → List[str]

Loads a list of titles which can then be selected by the user :param year: The year for which to fetch titles :param season: The season for which to fetch titles :param include_previous_season: Whether to include the previous season :return: The list of titles

classmethod name () → str

Returns The command name

post_commands ()

Commands executed after the core anitheme-dl functionality has been completed :return: None

classmethod prepare_parser (*parser: argparse.ArgumentParser*)

Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

prompt_selection (*shows: List[str]*) → List[str]

Prompts the user for a selection of series for which to download songs :param shows: All series that are up for selection :return: A list of series names that were selected

static resolve_selected_songs (*selected_series: List[str], data: Dict[str, List[Dict[str, str]]]*) → List[Dict[str, str]]

Retrieves a list of all songs that are included in a selection of series :param selected_series: The selection of series :param data: The song data from reddit :return: The list of selected songs

static segmentize (*titles: List[str]*) → List[List[str]]

Segments a list of titles into segments :param titles: The titles to segmentize :return: The segments

toktokkie.scripts.ArchiveCommand module

class toktokkie.scripts.ArchiveCommand.**ArchiveCommand** (*args: argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the archive command

archive (*source: str, dest: str*)

Creates a low-filesize archive of a directory into a new directory :param source: The source directory :param dest: The destination directory :return: None

execute ()

Executes the commands :return: None

classmethod name () → str

Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)

Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.CheckCommand module

class `toktokkie.scripts.CheckCommand.CheckCommand` (*args: argparse.Namespace*)
Bases: `toktokkie.scripts.Command.Command`

Class that encapsulates behaviour of the check command

execute ()
Executes the commands :return: None

classmethod name () → str
Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)
Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.Command module

class `toktokkie.scripts.Command.Command` (*args: argparse.Namespace*)
Bases: object

Class that encapsulates behaviour of a command for the toktokkie script

__init__ (*args: argparse.Namespace*)
Initializes the command object :param args: The command line arguments

static add_directories_arg (*parser: argparse.ArgumentParser*)
Adds a directories argument to a parser :param parser: The parser to which to add the argument :return: None

execute ()
Executes the commands :return: None

load_directories (*paths: List[str], restrictions: Optional[List[toktokkie.metadata.MediaType.MediaType]] = None*) → List[toktokkie.Directory.Directory]
Loads directory objects from a list of file paths :param paths: The paths to convert into objects :param restrictions: Limits the media type of directories :return: The converted directories

classmethod name () → str
Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)
Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.GuiCommand module

class `toktokkie.scripts.GuiCommand.GuiCommand` (*args: argparse.Namespace*)
Bases: `toktokkie.scripts.Command.Command`

Class that encapsulates behaviour of the gui command

execute ()
Executes the commands :return: None

classmethod name () → str
Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)
Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.IconizeCommand module

class toktokkie.scripts.IconizeCommand.**IconizeCommand** (*args: argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the iconize command

execute ()

Executes the commands :return: None

classmethod name () → str

Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)

Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.MangaCreateCommand module

class toktokkie.scripts.MangaCreateCommand.**MangaCreateCommand** (*args: argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the manga-create command

execute ()

Executes the commands :return: None

classmethod name () → str

Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)

Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.MangaUpdateCommand module

class toktokkie.scripts.MangaUpdateCommand.**MangaUpdateCommand** (*args: argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the manga-update command

check_latest_chapter_completeness (*metadata: toktokkie.metadata.types.Manga.Manga, chapters: List[manga_dl.entities.Chapter.Chapter]*)

Checks the latest regular chapter for completeness. This is necessary since some groups release their chapters in parts (i.e. 10.1 and then 10.2). manga-dl merges these chapter parts, so we only need to check if the local files has less pages. :param metadata: The metadata of the series to check :param chapters: The chapters for the series :return: None

execute ()

Executes the commands :return: None

execute_rename (*directory: toktokkie.Directory.Directory*)

Renames the current directory content. Automatically checks if the dry-run flag is set. If it is set, prints out any chapters that would have been renamed :param directory: The directory whose content should be renamed :return: None

static load_chapters (*metadata: toktokkie.metadata.types.Manga.Manga*) → Optional[List[manga_dl.entities.Chapter.Chapter]]
Loads chapter information from mangadex.org :param metadata: The metadata for which to load the chapters :return: Either a list of chapters or None if no mangadex ID was found

classmethod name () → str
Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)
Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

update_main_chapters (*metadata: toktokkie.metadata.types.Manga.Manga, chapters: List[manga_dl.entities.Chapter.Chapter]*)
Updates the regular chapters of the series :param metadata: The metadata of the series :param chapters: The chapters of the series :return: None

update_special_chapters (*metadata: toktokkie.metadata.types.Manga.Manga, chapters: List[manga_dl.entities.Chapter.Chapter]*)
Updates the special chapters of the series :param metadata: The metadata of the series :param chapters: The chapters of the series :return: None

toktokkie.scripts.MetadataAddCommand module

class toktokkie.scripts.MetadataAddCommand.**MetadataAddCommand** (*args: argparse.Namespace*)
Bases: *toktokkie.scripts.Command.Command*
Class that encapsulates behaviour of the metadata-add command

execute ()
Executes the commands :return: None

classmethod name () → str
Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)
Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.MetadataGenCommand module

class toktokkie.scripts.MetadataGenCommand.**MetadataGenCommand** (*args: argparse.Namespace*)
Bases: *toktokkie.scripts.Command.Command*
Class that encapsulates behaviour of the metadata-gen command

execute ()
Executes the commands :return: None

classmethod name () → str
Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)
Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.MusicMergeCommand module

class toktokkie.scripts.MusicMergeCommand.**MusicMergeCommand** (*args:* *argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the music-merge command

execute ()

Executes the commands :return: None

classmethod name () → str

Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)

Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.MusicTagCommand module

class toktokkie.scripts.MusicTagCommand.**MusicTagCommand** (*args:* *argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the music-tag command

execute ()

Executes the commands :return: None

classmethod name () → str

Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)

Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.PlaylistCreateCommand module

class toktokkie.scripts.PlaylistCreateCommand.**PlaylistCreateCommand** (*args:* *argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the playlist-create command

execute ()

Executes the commands :return: None

classmethod name () → str

Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)

Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.PrintCommand module

class toktokkie.scripts.PrintCommand.**PrintCommand** (*args: argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the print command

execute ()

Executes the commands :return: None

classmethod name () → str

Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)

Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.RenameCommand module

class toktokkie.scripts.RenameCommand.**RenameCommand** (*args: argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the rename command

execute ()

Executes the commands :return: None

classmethod name () → str

Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)

Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.SetMangaCoverCommand module

class toktokkie.scripts.SetMangaCoverCommand.**SetMangaCoverCommand** (*args: argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the set-manga-cover command

execute ()

Executes the commands :return: None

classmethod name () → str

Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)

Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

toktokkie.scripts.SuperCutCommand module

class toktokkie.scripts.SuperCutCommand.**SuperCutCommand** (*args: argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the supercut command

static adjust_chapter_names (*supercut_result: str; chapters: List[str]*)

Changes the chapter names of a file to a given list of chapter names :param supercut_result: The file of which the chapters should be

renamed

Parameters chapters – The chapter names

Returns None

static calculate_clip_duration (*start: str, end: str*) → int

Calculates the time delta in seconds between two timestamps :param start: The starting timestamp :param end: The ending timestamp :return: The time delta in seconds

create_supercut (*directory: toktokkie.Directory.Directory*)

Creates the supercut file :param directory: The directory for which to create a supercut :return: None

cut_part (*supercut_dir: str, instruction: Dict[str, str]*) → Dict[str, str]

Cuts a part of a video file based on a supercut instruction using ffmpeg :param supercut_dir: The supercut directory in which to

store the output

Parameters instruction – The instruction to use

Returns A dictionary detailing the path to the cut clip and its title

execute ()

Executes the commands :return: None

static generate_instructions (*directory: toktokkie.Directory.Directory*)

Generates a supercut instructions file based on the the existing season metadata and episode files :param directory: The directory for which to generate the instructions :return: None

static is_installed (*command: str*) → bool

Checks whether or not a command is installed/in the path :param command: The command to check :return: True if installed, else False

static merge_parts (*supercut_dir: str, files: List[str]*) → str

Merges a list of MKV files and stores them in supercut.mkv :param supercut_dir: The supercut directory :param files: The files to combine :return: The path to the combined file

classmethod name () → str

Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)

Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

read_supercut_instructions (*metadata: toktokkie.metadata.types.TvSeries.TvSeries, supercut_file: str*) → List[Dict[str, str]]

Reads the supercut instructions file and generates an easily processable configuration list :param metadata: The metadata used for additional information :param supercut_file: The supercut file to parse :return: A list of dictionaries modelling the individual parts of the

supercut in order

toktokkie.scripts.XdccUpdateCommand module

class toktokkie.scripts.XdccUpdateCommand.XdccUpdateCommand (*args: argparse.Namespace*)

Bases: *toktokkie.scripts.Command.Command*

Class that encapsulates behaviour of the xdcc-update command

execute ()

Executes the commands :return: None

classmethod name () → str

Returns The command name

classmethod prepare_parser (*parser: argparse.ArgumentParser*)

Prepares an argumentparser for this command :param parser: The parser to prepare :return: None

Module contents

`toktokkie.scripts.toktokkie_commands = [<class 'toktokkie.scripts.PrintCommand.PrintCommand']`

A list of commands for the toktokkie script

1.1.8 toktokkie.test package

Subpackages

toktokkie.test.metadata package

Submodules

toktokkie.test.metadata.TestBook module

class `toktokkie.test.metadata.TestBook.TestBook` (*methodName='runTest'*)

Bases: `toktokkie.test.metadata.TestMetadata._TestMetadata`

Class that tests the Book metadata class

test_checking ()

Tests if the checking mechanisms work correctly :return: None

test_prompt ()

Tests generating a new metadata object using user prompts :return: None

test_renaming ()

Tests renaming files associated with the metadata type :return: None

test_validation ()

Tests if the validation of metadata works correctly :return: None

toktokkie.test.metadata.TestBookSeries module

class `toktokkie.test.metadata.TestBookSeries.TestBookSeries` (*methodName='runTest'*)

Bases: `toktokkie.test.metadata.TestMetadata._TestMetadata`

Class that tests the BookSeries metadata class

test_checking ()

Tests if the checking mechanisms work correctly :return: None

test_prompt ()

Tests generating a new metadata object using user prompts :return: None

test_renaming ()

Tests renaming files associated with the metadata type :return: None

test_validation ()

Tests if the validation of metadata works correctly :return: None

toktokkie.test.metadata.TestManga module

class toktokkie.test.metadata.TestManga.**TestManga** (*methodName='runTest'*)

Bases: toktokkie.test.metadata.TestMetadata._TestMetadata

Class that tests the Manga metadata class

test_checking ()

Tests if the checking mechanisms work correctly :return: None

test_prompt ()

Tests generating a new metadata object using user prompts :return: None

test_renaming ()

Tests renaming files associated with the metadata type :return: None

test_validation ()

Tests if the validation of metadata works correctly :return: None

toktokkie.test.metadata.TestMetadata module**toktokkie.test.metadata.TestMovie module**

class toktokkie.test.metadata.TestMovie.**TestMovie** (*methodName='runTest'*)

Bases: toktokkie.test.metadata.TestMetadata._TestMetadata

Class that tests the Movie metadata class

test_checking ()

Tests if the checking mechanisms work correctly :return: None

test_prompt ()

Tests generating a new metadata object using user prompts :return: None

test_renaming ()

Tests renaming files associated with the metadata type :return: None

test_validation ()

Tests if the validation of metadata works correctly :return: None

toktokkie.test.metadata.TestTvSeries module

class toktokkie.test.metadata.TestTvSeries.**TestTvSeries** (*methodName='runTest'*)

Bases: toktokkie.test.metadata.TestMetadata._TestMetadata

Test class for TV metadata

test_checking ()

Tests if the checking mechanisms work correctly :return: None

test_prompt ()

Tests generating a new metadata object using user prompts :return: None

test_renaming ()

Tests renaming files associated with the metadata type :return: None

test_validation ()

Tests if the validation of metadata works correctly :return: None

toktokkie.test.metadata.TestVisualNovel module

class toktokkie.test.metadata.TestVisualNovel.**TestVisualNovel** (*methodName='runTest'*)
Bases: toktokkie.test.metadata.TestMetadata._TestMetadata

Class that tests the VisualNovel metadata class

test_checking ()
Tests if the checking mechanisms work correctly :return: None

test_prompt ()
Tests generating a new metadata object using user prompts :return: None

test_renaming ()
Tests renaming files associated with the metadata type :return: None

test_validation ()
Tests if the validation of metadata works correctly :return: None

Module contents

Submodules

toktokkie.test.TestFramework module

Module contents

1.1.9 toktokkie.xdcc_update package

Submodules

toktokkie.xdcc_update.XDCCUpdater module

class toktokkie.xdcc_update.XDCCUpdater.**XDCCUpdater** (*metadata: tok-*
tokkie.metadata.types.TvSeries.TvSeries,
throttle: int = -1, timeout: int =
120)

Bases: object

Class that handles the configuration and execution of a xdcc-update

__init__ (*metadata: toktokkie.metadata.types.TvSeries.TvSeries, throttle: int = -1, timeout: int = 120*)
Initializes the XDCC Updater object :param metadata: The metadata for the series for which to execute
an xdcc-update

Parameters

- **throttle** – The throttle value
- **timeout** – The timeout value

property bot

Returns The bot to use for updating

property episode_offset

Returns The amount of episodes offset from 1 when updating

property p_resolution**Returns** The resolution in P-notation (1080p)**predefined_patterns** = {'horriblesubs': '[HorribleSubs] @{NAME} - @{EPI-2} [@{RES-P}] .1

A collection of predefined patterns

classmethod prompt (*metadata: toktokkie.metadata.types.TvSeries.TvSeries*)

Creates a new XDCC Update instruction set for a given metadata :param metadata: The metadata for which to generate the instructions :return: The generated XDCCUpdater object

property resolution**Returns** The resolution in which to update the series**property search_engine****Returns** The search engine to use**property search_name****Returns** The name of the series for searching purposes**property search_pattern****Returns** The search pattern to use**property season****Returns** The season to update**update** ()

Executes the XDCC Update procedure :return: None

property x_resolution**Returns** The resolution in X-notation (1920x1080)**toktokkie.xdcc_update.enums module****class** toktokkie.xdcc_update.enums.**Resolution**

Bases: enum.Enum

Enum that models the different resolution options

x1080p = '1080p'**x480p** = '480p'**x720p** = '720p'**Module contents****1.2 Submodules****1.3 toktokkie.Directory module****class** toktokkie.Directory.**Directory** (*path: str, generate_metadata: bool = False, meta-
data_type: str = None*)

Bases: object

Class that encapsulates all of toktokkie's functionality

__init__ (*path: str, generate_metadata: bool = False, metadata_type: str = None*)
Initializes the metadata of the directory :param path: The directory's path :except MissingMetadataException, InvalidMetadataException, MetadataMismatch

check (*show_warnings: bool, fix_interactively: bool, config: Dict[str, Any]*) → bool
Performs a check, making sure that everything in the directory is configured correctly and up-to-date :param show_warnings: Whether or not to show warnings :param fix_interactively: Whether or not to enable interactive fixing :param config: Configuration dictionary for checks :return: The check result

generate_metadata (*metadata_type: str*)
Prompts the user for metadata information :param metadata_type: The metadata type to generate :return: None

iconize (*procedure: toktokkie.iconizing.procedures.Procedure.Procedure*)
Applies the directory's icons :param procedure: The iconizing procedure to use :return: None

reload ()
Reloads the metadata from the metadata file :return: None

rename (*noconfirm: bool = False*)
Renames the contained files. :param noconfirm: Skips the confirmation phase :return: None

write_metadata ()
Updates the metadata file with the current contents of the metadata :return: None

xdcc_update (*throttle: int, timeout: int*)
Performs an XDCC Update Action :param throttle: The throttle value :param timeout: The timeout value :return: None

1.4 toktokkie.exceptions module

exception toktokkie.exceptions.InvalidMetadata

Bases: Exception

Exception that is raised whenever metadata is invalid

exception toktokkie.exceptions.InvalidXDCCInstructions

Bases: Exception

Exception that is raised whenever xdcc update instructions are invalid

exception toktokkie.exceptions.MetadataMismatch

Bases: Exception

Exception that is raised whenever the metadata type in the JSON data is in conflict with the actual metadata class

exception toktokkie.exceptions.MissingMetadata

Bases: Exception

Exception that is raised whenever metadata is not found

exception toktokkie.exceptions.MissingXDCCInstructions

Bases: Exception

Exception that is raised whenever xdcc update instructions don't exist

1.5 Module contents

`toktokkie.sentry_dsn = 'https://77992bd58d9a46fc812ad491ba460a7e@sentry.namibsun.net/10'`
Sentry DSN used for exception logging

`toktokkie.version = '0.19.0'`
The current version of the package

TOKTOKKIE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

t

toktokkie, 39

toktokkie.anithemes, 4

toktokkie.anithemes.AniTheme, 3

toktokkie.check, 8

toktokkie.check.BookChecker, 4

toktokkie.check.BookSeriesChecker, 4

toktokkie.check.Checker, 5

toktokkie.check.MangaChecker, 5

toktokkie.check.map, 7

toktokkie.check.MovieChecker, 6

toktokkie.check.TvSeriesChecker, 6

toktokkie.check.VisualNovelChecker, 7

toktokkie.Directory, 37

toktokkie.exceptions, 38

toktokkie.gui, 12

toktokkie.gui.MainWindow, 11

toktokkie.gui.pyuic, 9

toktokkie.gui.pyuic.book_widget, 8

toktokkie.gui.pyuic.main, 8

toktokkie.gui.pyuic.movie_widget, 8

toktokkie.gui.pyuic.tv_season_widget, 9

toktokkie.gui.pyuic.tv_show_widget, 9

toktokkie.gui.qt_designer, 9

toktokkie.gui.widgets, 11

toktokkie.gui.widgets.BookWidget, 9

toktokkie.gui.widgets.MovieWidget, 10

toktokkie.gui.widgets.TvSeasonWidget, 10

toktokkie.gui.widgets.TvSeriesWidget, 11

toktokkie.iconizing, 13

toktokkie.iconizing.Iconizer, 13

toktokkie.iconizing.procedures, 13

toktokkie.iconizing.procedures.GnomeProcedure, 12

toktokkie.iconizing.procedures.NoopProcedure, 12

toktokkie.iconizing.procedures.Procedure, 12

toktokkie.metadata, 24

toktokkie.metadata.components, 15

toktokkie.metadata.components.MetadataPart, 14

toktokkie.metadata.components.TvEpisode, 14

toktokkie.metadata.components.TvEpisodeRange, 14

toktokkie.metadata.components.TvSeason, 15

toktokkie.metadata.functions, 24

toktokkie.metadata.ids, 17

toktokkie.metadata.ids.functions, 16

toktokkie.metadata.ids.IdFetcher, 15

toktokkie.metadata.ids.IdType, 16

toktokkie.metadata.ids.mappings, 17

toktokkie.metadata.MediaType, 23

toktokkie.metadata.Metadata, 23

toktokkie.metadata.prompt, 18

toktokkie.metadata.prompt.Prompter, 17

toktokkie.metadata.schema, 18

toktokkie.metadata.schema.SchemaBuilder, 18

toktokkie.metadata.types, 22

toktokkie.metadata.types.Book, 19

toktokkie.metadata.types.BookSeries, 19

toktokkie.metadata.types.components, 19

toktokkie.metadata.types.components.BookVolume, 18

toktokkie.metadata.types.components.MusicAlbum, 19

toktokkie.metadata.types.components.MusicThemeSong, 19

toktokkie.metadata.types.components.TvSeason, 19

toktokkie.metadata.types.Manga, 20

toktokkie.metadata.types.Movie, 20

toktokkie.metadata.types.MusicArtist, 20

toktokkie.metadata.types.TvSeries, 21

toktokkie.metadata.types.VisualNovel, 22

toktokkie.renaming, 26

toktokkie.renaming.functions, 25

- toktokkie.renaming.RenameOperation, 24
- toktokkie.renaming.Renamer, 25
- toktokkie.scripts, 34
- toktokkie.scripts.AlbumArtFetchCommand, 26
- toktokkie.scripts.AnilistOpenCommand, 26
- toktokkie.scripts.AnimeThemeDlCommand, 26
- toktokkie.scripts.ArchiveCommand, 27
- toktokkie.scripts.CheckCommand, 28
- toktokkie.scripts.Command, 28
- toktokkie.scripts.GuiCommand, 28
- toktokkie.scripts.IconizeCommand, 29
- toktokkie.scripts.MangaCreateCommand, 29
- toktokkie.scripts.MangaUpdateCommand, 29
- toktokkie.scripts.MetadataAddCommand, 30
- toktokkie.scripts.MetadataGenCommand, 30
- toktokkie.scripts.MusicMergeCommand, 31
- toktokkie.scripts.MusicTagCommand, 31
- toktokkie.scripts.PlaylistCreateCommand, 31
- toktokkie.scripts.PrintCommand, 31
- toktokkie.scripts.RenameCommand, 32
- toktokkie.scripts.SetMangaCoverCommand, 32
- toktokkie.scripts.SuperCutCommand, 32
- toktokkie.scripts.XdccUpdateCommand, 33
- toktokkie.test, 36
- toktokkie.test.metadata, 36
- toktokkie.test.metadata.TestBook, 34
- toktokkie.test.metadata.TestBookSeries, 34
- toktokkie.test.metadata.TestManga, 35
- toktokkie.test.metadata.TestMetadata, 35
- toktokkie.test.metadata.TestMovie, 35
- toktokkie.test.metadata.TestTvSeries, 35
- toktokkie.test.metadata.TestVisualNovel, 36
- toktokkie.test.TestFramework, 36
- toktokkie.xdcc_update, 37
- toktokkie.xdcc_update.enums, 37
- toktokkie.xdcc_update.XDCCUpdater, 36

INDEX

Symbols

- `__init__()` (*toktokkie.Directory.Directory* method), 37
 - `__init__()` (*toktokkie.anithemes AniTheme AniTheme* method), 3
 - `__init__()` (*toktokkie.check.Checker.Checker* method), 5
 - `__init__()` (*toktokkie.gui.MainWindow.MainWindow* method), 11
 - `__init__()` (*toktokkie.gui.widgets.BookWidget.BookWidget* method), 9
 - `__init__()` (*toktokkie.gui.widgets.MovieWidget.MovieWidget* method), 10
 - `__init__()` (*toktokkie.gui.widgets.TvSeasonWidget.TvSeasonWidget* method), 10
 - `__init__()` (*toktokkie.gui.widgets.TvSeriesWidget.TvSeriesWidget* method), 11
 - `__init__()` (*toktokkie.iconizing.Iconizer.Iconizer* method), 13
 - `__init__()` (*toktokkie.metadata.Metadata.Metadata* method), 23
 - `__init__()` (*toktokkie.metadata.components.MetadataPart.MetadataPart* method), 14
 - `__init__()` (*toktokkie.metadata.components.TvEpisode.TvEpisode* method), 14
 - `__init__()` (*toktokkie.metadata.components.TvEpisodeRange.TvEpisodeRange* method), 14
 - `__init__()` (*toktokkie.metadata.ids.IdFetcher.IdFetcher* method), 15
 - `__init__()` (*toktokkie.metadata.prompt.Prompter.Prompter* method), 17
 - `__init__()` (*toktokkie.metadata.schema.SchemaBuilder.SchemaBuilder* method), 18
 - `__init__()` (*toktokkie.metadata.types.components.BookVolume.BookVolume* method), 18
 - `__init__()` (*toktokkie.renaming.RenameOperation.RenameOperation* method), 24
 - `__init__()` (*toktokkie.renaming.Renamer.Renamer* method), 25
 - `__init__()` (*toktokkie.scripts.Command.Command* method), 28
 - `__init__()` (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater* method), 36
- ## A
- `add_album()` (*toktokkie.metadata.types.MusicArtist.MusicArtist* method), 20
 - `add_directories_arg()` (*toktokkie.scripts.Command.Command* static method), 28
 - `add_exclude()` (*toktokkie.metadata.types.TvSeries.TvSeries* method), 21
 - `add_multi_episode()` (*toktokkie.metadata.types.TvSeries.TvSeries* method), 21
 - `adjust_chapter_names()` (*toktokkie.scripts.SuperCutCommand.SuperCutCommand* static method), 32
 - `AlbumArtFetchCommand` (class in *toktokkie.scripts.AlbumArtFetchCommand*), 26
 - `albums()` (*toktokkie.metadata.types.MusicArtist.MusicArtist* property), 20
 - `all_albums()` (*toktokkie.metadata.types.MusicArtist.MusicArtist* property), 20
 - `ANILIST_EPISODE_RANGE` (*toktokkie.metadata.ids.IdType.IdType* attribute), 16
 - `AnilistOpenCommand` (class in *toktokkie.scripts.AnilistOpenCommand*), 26
 - `AnimeThemeDlCommand` (class in *toktokkie.scripts.AnimeThemeDlCommand*), 26
 - `AniTheme` (class in *toktokkie.anithemes.AniTheme*), 3
 - `ArchiveCommand` (class in *toktokkie.scripts.ArchiveCommand*), 27
 - `ArchiveCommand` (class in *toktokkie.scripts.ArchiveCommand*), 27
- ## B
- `Book` (class in *toktokkie.metadata.types.Book*), 19
 - `BOOK` (*toktokkie.metadata.MediaType.MediaType* attribute), 23

BOOK_SERIES (*toktokkie.metadata.MediaType.MediaType* attribute), 23

BookChecker (*class in toktokkie.check.BookChecker*), 4

BookSeries (*class in toktokkie.metadata.types.BookSeries*), 19

BookSeriesChecker (*class in toktokkie.check.BookSeriesChecker*), 4

BookVolume (*class in toktokkie.metadata.types.components.BookVolume*), 18

BookWidget (*class in toktokkie.gui.widgets.BookWidget*), 9

bot () (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater* property), 36

build_schema () (*toktokkie.metadata.schema.SchemaBuilder.SchemaBuilder* method), 18

C

calculate_clip_duration () (*toktokkie.scripts.SuperCutCommand.SuperCutCommand* static method), 33

cgs () (*toktokkie.metadata.types.VisualNovel.VisualNovel* property), 22

check () (*toktokkie.check.BookSeriesChecker.BookSeriesChecker* method), 5

check () (*toktokkie.check.Checker.Checker* method), 5

check () (*toktokkie.check.MangaChecker.MangaChecker* method), 6

check () (*toktokkie.check.TvSeriesChecker.TvSeriesChecker* method), 7

check () (*toktokkie.check.VisualNovelChecker.VisualNovelChecker* method), 7

check () (*toktokkie.Directory.Directory* method), 38

check_latest_chapter_completeness () (*toktokkie.scripts.MangaUpdateCommand.MangaUpdateCommand* method), 29

CheckCommand (*class in toktokkie.scripts.CheckCommand*), 28

Checker (*class in toktokkie.check.Checker*), 5

checker_map (*in module toktokkie.check.map*), 7

Command (*class in toktokkie.scripts.Command*), 28

convert_to_mp3 () (*toktokkie.anithemes.AniTheme.AniTheme* method), 3

create_metadata () (*in module toktokkie.metadata.functions*), 24

create_supercut () (*toktokkie.scripts.SuperCutCommand.SuperCutCommand* method), 33

cut_part () (*toktokkie.scripts.SuperCutCommand.SuperCutCommand* method), 33

default_procedure () (*in module toktokkie.iconizing*), 13

Directory (*class in toktokkie.Directory*), 37

download_webm () (*toktokkie.anithemes.AniTheme.AniTheme* method), 3

E

eds () (*toktokkie.metadata.types.VisualNovel.VisualNovel* property), 22

episode_offset () (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater* property), 36

episodes () (*toktokkie.metadata.components.TvEpisodeRange.TvEpisodeRange* property), 14

error () (*toktokkie.check.Checker.Checker* method), 5

excludes () (*toktokkie.metadata.types.TvSeries.TvSeries* property), 21

execute () (*toktokkie.scripts.AlbumArtFetchCommand.AlbumArtFetchCommand* method), 26

execute () (*toktokkie.scripts.AnilistOpenCommand.AnilistOpenCommand* method), 26

execute () (*toktokkie.scripts.AnimeThemeDICommand.AnimeThemeDICommand* method), 26

execute () (*toktokkie.scripts.ArchiveCommand.ArchiveCommand* method), 27

execute () (*toktokkie.scripts.CheckCommand.CheckCommand* method), 28

execute () (*toktokkie.scripts.Command.Command* method), 28

execute () (*toktokkie.scripts.GuiCommand.GuiCommand* method), 28

execute () (*toktokkie.scripts.IconizeCommand.IconizeCommand* method), 29

execute () (*toktokkie.scripts.MangaCreateCommand.MangaCreateCommand* method), 29

execute () (*toktokkie.scripts.MangaUpdateCommand.MangaUpdateCommand* method), 29

execute () (*toktokkie.scripts.MetadataAddCommand.MetadataAddCommand* method), 30

execute () (*toktokkie.scripts.MetadataGenCommand.MetadataGenCommand* method), 30

execute () (*toktokkie.scripts.MusicMergeCommand.MusicMergeCommand* method), 31

execute () (*toktokkie.scripts.MusicTagCommand.MusicTagCommand* method), 31

execute () (*toktokkie.scripts.PlaylistCreateCommand.PlaylistCreateCommand* method), 31

execute () (*toktokkie.scripts.PrintCommand.PrintCommand* method), 31

execute () (*toktokkie.scripts.RenameCommand.RenameCommand* method), 32

- execute () (*toktokkie.scripts.SetMangaCoverCommand.SetMangaCoverCommand* method), 32
- execute () (*toktokkie.scripts.SuperCutCommand.SuperCutCommand* method), 33
- execute () (*toktokkie.scripts.XdccUpdateCommand.XdccUpdateCommand* method), 33
- execute_rename () (*toktokkie.scripts.MangaUpdateCommand.MangaUpdateCommand* method), 29
- F**
- fetch_ids () (*toktokkie.metadata.ids.IdFetcher.IdFetcher* method), 15
- fill_ids () (in module *toktokkie.metadata.ids.functions*), 16
- G**
- generate_artist_album_structure () (*toktokkie.scripts.AnimeThemeDICommand.AnimeThemeDICommand* method), 27
- generate_instructions () (*toktokkie.scripts.SuperCutCommand.SuperCutCommand* static method), 33
- generate_metadata () (*toktokkie.Directory.Directory* method), 38
- generate_tv_episode_filename () (in module *toktokkie.renaming.functions*), 25
- get_active_operations () (*toktokkie.renaming.Renamer.Renamer* method), 25
- get_episode_files () (*toktokkie.metadata.types.TvSeries.TvSeries* method), 21
- get_icon_file () (*toktokkie.metadata.Metadata.Metadata* method), 23
- get_metadata () (in module *toktokkie.metadata.functions*), 24
- get_metadata_class () (in module *toktokkie.metadata.functions*), 24
- get_season () (*toktokkie.metadata.types.TvSeries.TvSeries* method), 21
- GnomeProcedure (class in *toktokkie.iconizing.procedures.GnomeProcedure*), 12
- GuiCommand (class in *toktokkie.scripts.GuiCommand*), 28
- H**
- handle_excludes () (*toktokkie.scripts.AnimeThemeDICommand.AnimeThemeDICommand* method), 27
- has_ost () (*toktokkie.metadata.types.VisualNovel.VisualNovel* property), 22
- iconize () (*toktokkie.Directory.Directory* method), 38
- iconize () (*toktokkie.iconizing.Iconizer.Iconizer* method), 13
- iconize () (*toktokkie.iconizing.procedures.GnomeProcedure.GnomeProcedure* class method), 12
- iconize () (*toktokkie.iconizing.procedures.NoopProcedure.NoopProcedure* class method), 12
- iconize () (*toktokkie.iconizing.procedures.Procedure.Procedure* class method), 13
- IconizeCommand (class in *toktokkie.scripts.IconizeCommand*), 29
- Iconizer (class in *toktokkie.iconizing.Iconizer*), 13
- id_prompt_order (in module *toktokkie.metadata.ids.mappings*), 17
- IdFetcher (class in *toktokkie.metadata.ids.IdFetcher*), 15
- ids () (*toktokkie.metadata.components.MetadataPart.MetadataPart* property), 14
- ids () (*toktokkie.metadata.Metadata.Metadata* property), 23
- IdType (class in *toktokkie.metadata.ids.IdType*), 16
- IMDB (*toktokkie.metadata.ids.IdType.IdType* attribute), 16
- initialize_buttons () (*toktokkie.gui.widgets.BookWidget.BookWidget* method), 9
- initialize_buttons () (*toktokkie.gui.widgets.MovieWidget.MovieWidget* method), 10
- initialize_buttons () (*toktokkie.gui.widgets.TvSeasonWidget.TvSeasonWidget* method), 10
- initialize_buttons () (*toktokkie.gui.widgets.TvSeriesWidget.TvSeriesWidget* method), 11
- int_id_types (in module *toktokkie.metadata.ids.mappings*), 17
- InvalidMetadata, 38
- InvalidXDCCInstructions, 38
- is_applicable () (*toktokkie.iconizing.procedures.GnomeProcedure.GnomeProcedure* class method), 12
- is_applicable () (*toktokkie.iconizing.procedures.NoopProcedure.NoopProcedure* class method), 12

- class method), 12
- is_applicable() (toktokkie.iconizing.procedures.Procedure.Procedure class method), 13
- is_installed() (toktokkie.scripts.SuperCutCommand.SuperCutCommand static method), 33
- is_spinoff() (toktokkie.metadata.components.TvSeason.TvSeason method), 15
- ISBN (toktokkie.metadata.ids.IdType.IdType attribute), 16
- ## J
- json() (toktokkie.metadata.types.components.BookVolume.BookVolume property), 19
- ## K
- KITSU (toktokkie.metadata.ids.IdType.IdType attribute), 16
- ## L
- literature_media_types (in module toktokkie.metadata.ids.mappings), 17
- load_chapters() (toktokkie.scripts.MangaUpdateCommand.MangaUpdateCommand static method), 29
- load_directories() (toktokkie.scripts.Command.Command method), 28
- load_episode_files() (toktokkie.gui.widgets.TvSeasonWidget.TvSeasonWidget method), 10
- load_reddit_anithemes_wiki_info() (toktokkie.anithemes.AniTheme.AniTheme class method), 3
- load_title_name() (toktokkie.renaming.Renamer.Renamer method), 25
- load_titles() (toktokkie.scripts.AnimeThemeDICCommand.AnimeThemeDICCommand method), 27
- load_tvdb_episode_name() (in module toktokkie.renaming.functions), 25
- load_tvdb_info() (toktokkie.gui.widgets.TvSeasonWidget.TvSeasonWidget method), 10
- load_tvdb_info() (toktokkie.gui.widgets.TvSeriesWidget.TvSeriesWidget method), 11
- logger (toktokkie.anithemes.AniTheme.AniTheme attribute), 3
- logger (toktokkie.metadata.ids.IdFetcher.IdFetcher attribute), 15
- logger (toktokkie.metadata.Metadata.Metadata attribute), 23
- logger (toktokkie.metadata.prompt.Prompter.Prompter attribute), 17
- logger (toktokkie.metadata.schema.SchemaBuilder.SchemaBuilder attribute), 18
- ## M
- main_path() (toktokkie.metadata.types.Manga.Manga property), 20
- MainWindow (class in toktokkie.gui.MainWindow), 11
- mal_cache (toktokkie.anithemes.AniTheme.AniTheme attribute), 4
- Manga (class in toktokkie.metadata.types.Manga), 20
- media_type() (toktokkie.metadata.MediaType.MediaType attribute), 23
- MangaChecker (class in toktokkie.check.MangaChecker), 6
- MangaCreateCommand (class in toktokkie.scripts.MangaCreateCommand), 29
- MANGADEX (toktokkie.metadata.ids.IdType.IdType attribute), 16
- MangaUpdateCommand (class in toktokkie.scripts.MangaUpdateCommand), 29
- media_type() (toktokkie.metadata.Metadata.Metadata class method), 23
- media_type() (toktokkie.metadata.types.Book.Book class method), 19
- media_type() (toktokkie.metadata.types.BookSeries.BookSeries class method), 19
- media_type() (toktokkie.metadata.types.Manga.Manga class method), 20
- media_type() (toktokkie.metadata.types.Movie.Movie class method), 20
- media_type() (toktokkie.metadata.types.MusicArtist.MusicArtist class method), 20
- media_type() (toktokkie.metadata.types.TvSeries.TvSeries class method), 21
- media_type() (toktokkie.metadata.types.VisualNovel.VisualNovel class method), 22
- MediaType (class in toktokkie.metadata.MediaType), 23
- merge_parts() (toktokkie.scripts.SuperCutCommand.SuperCutCommand static method), 33
- Metadata (class in toktokkie.metadata.Metadata), 23

MetadataAddCommand (class in toktokkie.scripts.MetadataAddCommand), 30
 MetadataGenCommand (class in toktokkie.scripts.MetadataGenCommand), 30
 MetadataMismatch, 38
 MetadataPart (class in toktokkie.metadata.components.MetadataPart), 14
 minimize_ids() (in module toktokkie.metadata.ids.functions), 16
 MissingMetadata, 38
 MissingXDCCInstructions, 38
 Movie (class in toktokkie.metadata.types.Movie), 20
 MOVIE (toktokkie.metadata.MediaType.MediaType attribute), 23
 MovieChecker (class in toktokkie.check.MovieChecker), 6
 MovieWidget (class in toktokkie.gui.widgets.MovieWidget), 10
 multi_episodes() (toktokkie.metadata.types.TvSeries.TvSeries property), 21
 MUSIC_ARTIST (toktokkie.metadata.MediaType.MediaType attribute), 23
 MusicArtist (class in toktokkie.metadata.types.MusicArtist), 20
 MUSICBRAINZ (toktokkie.metadata.ids.IdType.IdType attribute), 16
 MusicMergeCommand (class in toktokkie.scripts.MusicMergeCommand), 31
 MusicTagCommand (class in toktokkie.scripts.MusicTagCommand), 31
 MYANIMELIST (toktokkie.metadata.ids.IdType.IdType attribute), 16

N

name (toktokkie.iconizing.procedures.Procedure.Procedure attribute), 13
 name () (toktokkie.metadata.components.TvSeason.TvSeason property), 15
 name () (toktokkie.metadata.Metadata.Metadata property), 23
 name () (toktokkie.scripts.AlbumArtFetchCommand.AlbumArtFetchCommand class method), 26
 name () (toktokkie.scripts.AnilistOpenCommand.AnilistOpenCommand class method), 26
 name () (toktokkie.scripts.AnimeThemeDICommand.AnimeThemeDICommand class method), 27
 name () (toktokkie.scripts.ArchiveCommand.ArchiveCommand class method), 27
 name () (toktokkie.scripts.CheckCommand.CheckCommand class method), 28
 name () (toktokkie.scripts.Command.Command class method), 28
 name () (toktokkie.scripts.GuiCommand.GuiCommand class method), 28
 name () (toktokkie.scripts.IconizeCommand.IconizeCommand class method), 29
 name () (toktokkie.scripts.MangaCreateCommand.MangaCreateCommand class method), 29
 name () (toktokkie.scripts.MangaUpdateCommand.MangaUpdateCommand class method), 30
 name () (toktokkie.scripts.MetadataAddCommand.MetadataAddCommand class method), 30
 name () (toktokkie.scripts.MetadataGenCommand.MetadataGenCommand class method), 30
 name () (toktokkie.scripts.MusicMergeCommand.MusicMergeCommand class method), 31
 name () (toktokkie.scripts.MusicTagCommand.MusicTagCommand class method), 31
 name () (toktokkie.scripts.PlaylistCreateCommand.PlaylistCreateCommand class method), 31
 name () (toktokkie.scripts.PrintCommand.PrintCommand class method), 32
 name () (toktokkie.scripts.RenameCommand.RenameCommand class method), 32
 name () (toktokkie.scripts.SetMangaCoverCommand.SetMangaCoverCommand class method), 32
 name () (toktokkie.scripts.SuperCutCommand.SuperCutCommand class method), 33
 name () (toktokkie.scripts.XdccUpdateCommand.XdccUpdateCommand class method), 33
 NoopProcedure (class in toktokkie.iconizing.procedures.NoopProcedure), 12

O

objectify_ids() (in module toktokkie.metadata.ids.functions), 16
 open_directory () (toktokkie.gui.widgets.BookWidget.BookWidget method), 9
 open_directory () (toktokkie.gui.widgets.MovieWidget.MovieWidget method), 10
 open_directory () (toktokkie.gui.widgets.TvSeasonWidget.TvSeasonWidget method), 10
 open_directory () (toktokkie.gui.widgets.TvSeriesWidget.TvSeriesWidget method), 11
 open_tvdb_url () (toktokkie.gui.widgets.TvSeasonWidget.TvSeasonWidget method), 10
 open_tvdb_url () (toktokkie.gui.widgets.TvSeriesWidget.TvSeriesWidget method), 10

method), 11

ops () (toktokkie.metadata.types.VisualNovel.VisualNovel property), 22

ost () (toktokkie.metadata.types.VisualNovel.VisualNovel property), 22

P

p_resolution () (toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater property), 36

path () (toktokkie.metadata.components.TvSeason.TvSeason property), 15

path () (toktokkie.renaming.Renamer.Renamer property), 25

PlaylistCreateCommand (class in toktokkie.scripts.PlaylistCreateCommand), 31

post_commands () (toktokkie.scripts.AnimeThemeDICommand.AnimeThemeDICommand method), 27

predefined_patterns (toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater attribute), 37

prepare_parser () (toktokkie.scripts.AlbumArtFetchCommand.AlbumArtFetchCommand class method), 26

prepare_parser () (toktokkie.scripts.AnilistOpenCommand.AnilistOpenCommand class method), 26

prepare_parser () (toktokkie.scripts.AnimeThemeDICommand.AnimeThemeDICommand class method), 27

prepare_parser () (toktokkie.scripts.ArchiveCommand.ArchiveCommand class method), 27

prepare_parser () (toktokkie.scripts.CheckCommand.CheckCommand class method), 28

prepare_parser () (toktokkie.scripts.Command.Command class method), 28

prepare_parser () (toktokkie.scripts.GuiCommand.GuiCommand class method), 28

prepare_parser () (toktokkie.scripts.IconizeCommand.IconizeCommand class method), 29

prepare_parser () (toktokkie.scripts.MangaCreateCommand.MangaCreateCommand class method), 29

prepare_parser () (toktokkie.scripts.MangaUpdateCommand.MangaUpdateCommand class method), 30

prepare_parser () (toktokkie.scripts.MetadataAddCommand.MetadataAddCommand class method), 30

prepare_parser () (toktokkie.scripts.MetadataGenCommand.MetadataGenCommand class method), 30

prepare_parser () (toktokkie.scripts.MusicMergeCommand.MusicMergeCommand class method), 31

prepare_parser () (toktokkie.scripts.MusicTagCommand.MusicTagCommand class method), 31

prepare_parser () (toktokkie.scripts.PlaylistCreateCommand.PlaylistCreateCommand class method), 31

prepare_parser () (toktokkie.scripts.PrintCommand.PrintCommand class method), 32

prepare_parser () (toktokkie.scripts.RenameCommand.RenameCommand class method), 32

prepare_parser () (toktokkie.scripts.SetMangaCoverCommand.SetMangaCoverCommand class method), 32

prepare_parser () (toktokkie.scripts.SuperCutCommand.SuperCutCommand class method), 33

prepare_parser () (toktokkie.scripts.XdccUpdateCommand.XdccUpdateCommand class method), 34

prepare_parser () (toktokkie.scripts.ArchiveCommand.ArchiveCommand class method), 27

PrintCommand (class in toktokkie.scripts.PrintCommand), 31

Procedure (class in toktokkie.iconizing.procedures.Procedure), 12

prompt () (toktokkie.metadata.Metadata.Metadata class method), 24

prompt () (toktokkie.metadata.prompt.Prompter.Prompter method), 17

prompt () (toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater class method), 37

prompt_selection () (toktokkie.scripts.AnimeThemeDICommand.AnimeThemeDICommand method), 27

Prompter (class in toktokkie.metadata.prompt.Prompter), 17

R

read_supercut_instructions () (toktokkie.scripts.SuperCutCommand.SuperCutCommand method), 33

reload () (toktokkie.Directory.Directory method), 38

reload() (*toktokkie.gui.MainWindow.MainWindow* method), 11
 rename() (*toktokkie.Directory.Directory* method), 38
 rename() (*toktokkie.renaming.RenameOperation.RenameOperation* method), 25
 rename() (*toktokkie.renaming.Renamer.Renamer* method), 25
 RenameCommand (class in *toktokkie.scripts.RenameCommand*), 32
 RenameOperation (class in *toktokkie.renaming.RenameOperation*), 24
 Renamer (class in *toktokkie.renaming.Renamer*), 25
 required_id_types (in module *toktokkie.metadata.ids.mappings*), 17
 Resolution (class in *toktokkie.xdcc_update.enums*), 37
 resolution() (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater* property), 37
 resolve_selected_songs() (*toktokkie.scripts.AnimeThemeDICommand.AnimeThemeDICommand* static method), 27
 retranslateUi() (*toktokkie.gui.pyuic.book_widget.Ui_BookWidget* method), 8
 retranslateUi() (*toktokkie.gui.pyuic.main.Ui_MainWindow* method), 8
 retranslateUi() (*toktokkie.gui.pyuic.movie_widget.Ui_MovieWidget* method), 8
 retranslateUi() (*toktokkie.gui.pyuic.tv_season_widget.Ui_TvSeasonWidget* method), 9
 retranslateUi() (*toktokkie.gui.pyuic.tv_show_widget.Ui_TvSeriesWidget* method), 9
 season() (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater* property), 37
 season_number() (*toktokkie.metadata.components.TvSeason.TvSeason* property), 15
 season_start_overrides() (*toktokkie.metadata.types.TvSeries.TvSeries* property), 21
 seasons() (*toktokkie.metadata.types.TvSeries.TvSeries* property), 21
 segmentize() (*toktokkie.scripts.AnimeThemeDICommand.AnimeThemeDICommand* static method), 27
 sentry_dsn (in module *toktokkie*), 39
 set_icon_image() (*toktokkie.gui.widgets.BookWidget.BookWidget* method), 9
 set_icon_image() (*toktokkie.gui.widgets.MovieWidget.MovieWidget* method), 10
 set_image() (*toktokkie.gui.widgets.TvSeasonWidget.TvSeasonWidget* method), 10
 set_icon_image() (*toktokkie.gui.widgets.TvSeriesWidget.TvSeriesWidget* method), 11
 set_ids() (*toktokkie.metadata.Metadata.Metadata* method), 24
 set_metadata() (*toktokkie.gui.widgets.BookWidget.BookWidget* method), 9
 set_metadata() (*toktokkie.gui.widgets.MovieWidget.MovieWidget* method), 10
 set_metadata() (*toktokkie.gui.widgets.TvSeasonWidget.TvSeasonWidget* method), 10
 set_metadata() (*toktokkie.gui.widgets.TvSeriesWidget.TvSeriesWidget* method), 11
 SetMangaCoverCommand (class in *toktokkie.scripts.SetMangaCoverCommand*), 32
 setupUi() (*toktokkie.gui.pyuic.book_widget.Ui_BookWidget* method), 8
 setupUi() (*toktokkie.gui.pyuic.main.Ui_MainWindow* method), 8
 setupUi() (*toktokkie.gui.pyuic.movie_widget.Ui_MovieWidget* method), 8
 setupUi() (*toktokkie.gui.pyuic.tv_season_widget.Ui_TvSeasonWidget* method), 9
 setupUi() (*toktokkie.gui.pyuic.tv_show_widget.Ui_TvSeriesWidget* method), 9
 singles() (*toktokkie.metadata.types.MusicArtist.MusicArtist* property), 37

S

sanitize() (*toktokkie.renaming.RenameOperation.RenameOperation* static method), 25
 SchemaBuilder (class in *toktokkie.metadata.schema.SchemaBuilder*), 18
 search_engine() (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater* property), 37
 search_name() (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater* property), 37
 search_pattern() (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater* property), 37

- property), 20
- special_chapters() (toktokkie.metadata.types.Manga.Manga property), 20
- special_path() (toktokkie.metadata.types.Manga.Manga property), 20
- stringify_ids() (in module toktokkie.metadata.ids.functions), 17
- SuperCutCommand (class in toktokkie.scripts.SuperCutCommand), 32
- ## T
- tags() (toktokkie.metadata.Metadata.Metadata property), 24
- test_checking() (toktokkie.test.metadata.TestBook.TestBook method), 34
- test_checking() (toktokkie.test.metadata.TestBookSeries.TestBookSeries method), 34
- test_checking() (toktokkie.test.metadata.TestManga.TestManga method), 35
- test_checking() (toktokkie.test.metadata.TestMovie.TestMovie method), 35
- test_checking() (toktokkie.test.metadata.TestTvSeries.TestTvSeries method), 35
- test_checking() (toktokkie.test.metadata.TestVisualNovel.TestVisualNovel method), 36
- test_prompt() (toktokkie.test.metadata.TestBook.TestBook method), 34
- test_prompt() (toktokkie.test.metadata.TestBookSeries.TestBookSeries method), 34
- test_prompt() (toktokkie.test.metadata.TestManga.TestManga method), 35
- test_prompt() (toktokkie.test.metadata.TestMovie.TestMovie method), 35
- test_prompt() (toktokkie.test.metadata.TestTvSeries.TestTvSeries method), 35
- test_prompt() (toktokkie.test.metadata.TestVisualNovel.TestVisualNovel method), 36
- test_renaming() (toktokkie.test.metadata.TestBook.TestBook method), 34
- test_renaming() (toktokkie.test.metadata.TestBookSeries.TestBookSeries method), 34
- test_renaming() (toktokkie.test.metadata.TestManga.TestManga method), 35
- test_renaming() (toktokkie.test.metadata.TestMovie.TestMovie method), 35
- test_renaming() (toktokkie.test.metadata.TestTvSeries.TestTvSeries method), 35
- test_renaming() (toktokkie.test.metadata.TestVisualNovel.TestVisualNovel method), 36
- test_validation() (toktokkie.test.metadata.TestBook.TestBook method), 34
- test_validation() (toktokkie.test.metadata.TestBookSeries.TestBookSeries method), 34
- test_validation() (toktokkie.test.metadata.TestManga.TestManga method), 35
- test_validation() (toktokkie.test.metadata.TestMovie.TestMovie method), 35
- test_validation() (toktokkie.test.metadata.TestTvSeries.TestTvSeries method), 35
- test_validation() (toktokkie.test.metadata.TestVisualNovel.TestVisualNovel method), 36
- TestBook (class in toktokkie.test.metadata.TestBook), 34
- TestBookSeries (class in toktokkie.test.metadata.TestBookSeries), 34
- TestManga (class in toktokkie.test.metadata.TestManga), 35
- TestMovie (class in toktokkie.test.metadata.TestMovie), 35
- TestTvSeries (class in toktokkie.test.metadata.TestTvSeries), 35
- TestVisualNovel (class in toktokkie.test.metadata.TestVisualNovel), 36
- theme_song_ids (in module toktokkie.metadata.ids.mappings), 17
- theme_songs() (toktokkie.metadata.types.MusicArtist.MusicArtist property), 21
- toktokkie (module), 39
- toktokkie.anithemes (module), 4
- toktokkie.anithemes AniTheme (module), 3
- toktokkie.check (module), 8

toktokkie.check.BookChecker (*module*), 4
 toktokkie.check.BookSeriesChecker (*module*), 4
 toktokkie.check.Checker (*module*), 5
 toktokkie.check.MangaChecker (*module*), 5
 toktokkie.check.map (*module*), 7
 toktokkie.check.MovieChecker (*module*), 6
 toktokkie.check.TvSeriesChecker (*module*), 6
 toktokkie.check.VisualNovelChecker (*module*), 7
 toktokkie.Directory (*module*), 37
 toktokkie.exceptions (*module*), 38
 toktokkie.gui (*module*), 12
 toktokkie.gui.MainWindow (*module*), 11
 toktokkie.gui.pyuic (*module*), 9
 toktokkie.gui.pyuic.book_widget (*module*), 8
 toktokkie.gui.pyuic.main (*module*), 8
 toktokkie.gui.pyuic.movie_widget (*module*), 8
 toktokkie.gui.pyuic.tv_season_widget (*module*), 9
 toktokkie.gui.pyuic.tv_show_widget (*module*), 9
 toktokkie.gui.qt_designer (*module*), 9
 toktokkie.gui.widgets (*module*), 11
 toktokkie.gui.widgets.BookWidget (*module*), 9
 toktokkie.gui.widgets.MovieWidget (*module*), 10
 toktokkie.gui.widgets.TvSeasonWidget (*module*), 10
 toktokkie.gui.widgets.TvSeriesWidget (*module*), 11
 toktokkie.iconizing (*module*), 13
 toktokkie.iconizing.Iconizer (*module*), 13
 toktokkie.iconizing.procedures (*module*), 13
 toktokkie.iconizing.procedures.GnomeProcedure (*module*), 12
 toktokkie.iconizing.procedures.NoopProcedure (*module*), 12
 toktokkie.iconizing.procedures.Procedure (*module*), 12
 toktokkie.metadata (*module*), 24
 toktokkie.metadata.components (*module*), 15
 toktokkie.metadata.components.MetadataParser (*module*), 14
 toktokkie.metadata.components.TvEpisode (*module*), 14
 toktokkie.metadata.components.TvEpisodeRange (*module*), 14
 toktokkie.metadata.components.TvSeason (*module*), 15
 toktokkie.metadata.functions (*module*), 24
 toktokkie.metadata.ids (*module*), 17
 toktokkie.metadata.ids.functions (*module*), 16
 toktokkie.metadata.ids.IdFetcher (*module*), 15
 toktokkie.metadata.ids.IdType (*module*), 16
 toktokkie.metadata.ids.mappings (*module*), 17
 toktokkie.metadata.MediaType (*module*), 23
 toktokkie.metadata.Metadata (*module*), 23
 toktokkie.metadata.prompt (*module*), 18
 toktokkie.metadata.prompt.Prompter (*module*), 17
 toktokkie.metadata.schema (*module*), 18
 toktokkie.metadata.schema.SchemaBuilder (*module*), 18
 toktokkie.metadata.types (*module*), 22
 toktokkie.metadata.types.Book (*module*), 19
 toktokkie.metadata.types.BookSeries (*module*), 19
 toktokkie.metadata.types.components (*module*), 19
 toktokkie.metadata.types.components.BookVolume (*module*), 18
 toktokkie.metadata.types.components.MusicAlbum (*module*), 19
 toktokkie.metadata.types.components.MusicThemeSong (*module*), 19
 toktokkie.metadata.types.components.TvSeason (*module*), 19
 toktokkie.metadata.types.Manga (*module*), 20
 toktokkie.metadata.types.Movie (*module*), 20
 toktokkie.metadata.types.MusicArtist (*module*), 20
 toktokkie.metadata.types.TvSeries (*module*), 21
 toktokkie.metadata.types.VisualNovel (*module*), 22
 toktokkie.renaming (*module*), 26
 toktokkie.renaming.functions (*module*), 25
 toktokkie.renaming.RenameOperation (*module*), 24
 toktokkie.renaming.Renamer (*module*), 25
 toktokkie.scripts (*module*), 34
 toktokkie.scripts.AlbumArtFetchCommand (*module*), 26
 toktokkie.scripts.AnilistOpenCommand (*module*), 26
 toktokkie.scripts.AnimeThemeDlCommand (*module*), 26

- toktokkie.scripts.ArchiveCommand (*module*), 27
- toktokkie.scripts.CheckCommand (*module*), 28
- toktokkie.scripts.Command (*module*), 28
- toktokkie.scripts.GuiCommand (*module*), 28
- toktokkie.scripts.IconizeCommand (*module*), 29
- toktokkie.scripts.MangaCreateCommand (*module*), 29
- toktokkie.scripts.MangaUpdateCommand (*module*), 29
- toktokkie.scripts.MetadataAddCommand (*module*), 30
- toktokkie.scripts.MetadataGenCommand (*module*), 30
- toktokkie.scripts.MusicMergeCommand (*module*), 31
- toktokkie.scripts.MusicTagCommand (*module*), 31
- toktokkie.scripts.PlaylistCreateCommand (*module*), 31
- toktokkie.scripts.PrintCommand (*module*), 31
- toktokkie.scripts.RenameCommand (*module*), 32
- toktokkie.scripts.SetMangaCoverCommand (*module*), 32
- toktokkie.scripts.SuperCutCommand (*module*), 32
- toktokkie.scripts.XdccUpdateCommand (*module*), 33
- toktokkie.test (*module*), 36
- toktokkie.test.metadata (*module*), 36
- toktokkie.test.metadata.TestBook (*module*), 34
- toktokkie.test.metadata.TestBookSeries (*module*), 34
- toktokkie.test.metadata.TestManga (*module*), 35
- toktokkie.test.metadata.TestMetadata (*module*), 35
- toktokkie.test.metadata.TestMovie (*module*), 35
- toktokkie.test.metadata.TestTvSeries (*module*), 35
- toktokkie.test.metadata.TestVisualNovel (*module*), 36
- toktokkie.test.TestFramework (*module*), 36
- toktokkie.xdcc_update (*module*), 37
- toktokkie.xdcc_update.enums (*module*), 37
- toktokkie.xdcc_update.XDCCUpdater (*module*), 36
- toktokkie_commands (*in module toktokkie.scripts*), 34
- TV_SERIES (*toktokkie.metadata.MediaType.MediaType attribute*), 23
- TVDB (*toktokkie.metadata.ids.IdType.IdType attribute*), 16
- tvdb_id() (*toktokkie.metadata.components.TvSeason.TvSeason property*), 15
- tvdb_id() (*toktokkie.metadata.types.TvSeries.TvSeries property*), 21
- TvEpisode (*class in toktokkie.metadata.components.TvEpisode*), 14
- TvEpisodeRange (*class in toktokkie.metadata.components.TvEpisodeRange*), 14
- TvSeason (*class in toktokkie.metadata.components.TvSeason*), 15
- TvSeasonWidget (*class in toktokkie.gui.widgets.TvSeasonWidget*), 10
- TvSeries (*class in toktokkie.metadata.types.TvSeries*), 21
- TvSeriesChecker (*class in toktokkie.check.TvSeriesChecker*), 6
- TvSeriesWidget (*class in toktokkie.gui.widgets.TvSeriesWidget*), 11
- ## U
- Ui_BookWidget (*class in toktokkie.gui.pyuic.book_widget*), 8
- Ui_MainWindow (*class in toktokkie.gui.pyuic.main*), 8
- Ui_MovieWidget (*class in toktokkie.gui.pyuic.movie_widget*), 8
- Ui_TvSeasonWidget (*class in toktokkie.gui.pyuic.tv_season_widget*), 9
- Ui_TvSeriesWidget (*class in toktokkie.gui.pyuic.tv_show_widget*), 9
- update() (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater method*), 37
- update_main_chapters() (*toktokkie.scripts.MangaUpdateCommand.MangaUpdateCommand method*), 30
- update_special_chapters() (*toktokkie.scripts.MangaUpdateCommand.MangaUpdateCommand method*), 30
- ## V
- valid_id_types (*in module toktokkie.metadata.ids.mappings*), 17
- validate() (*toktokkie.metadata.components.MetadataPart.MetadataPart method*), 14
- validate() (*toktokkie.metadata.components.TvSeason.TvSeason method*), 15

validate() (*toktokkie.metadata.Metadata.Metadata* XDCCUUpdater (class in tok-
method), 24 *tokkie.xdcc_update.XDCCUUpdater*), 36
validate() (*toktokkie.metadata.types.TvSeries.TvSeries*
method), 22
version (in module *toktokkie*), 39
VISUAL_NOVEL (*tok-*
tokkie.metadata.MediaType.MediaType at-
tribute), 23
VisualNovel (class in tok-
tokkie.metadata.types.VisualNovel), 22
VisualNovelChecker (class in tok-
tokkie.check.VisualNovelChecker), 7
VNDB (*toktokkie.metadata.ids.IdType.IdType* attribute),
 16
volumes() (*toktokkie.metadata.types.BookSeries.BookSeries*
property), 19

W

warn() (*toktokkie.check.Checker.Checker* method), 5
write() (*toktokkie.metadata.Metadata.Metadata*
method), 24
write_changes() (*tok-*
tokkie.gui.widgets.BookWidget.BookWidget
method), 9
write_changes() (*tok-*
tokkie.gui.widgets.MovieWidget.MovieWidget
method), 10
write_changes() (*tok-*
tokkie.gui.widgets.TvSeasonWidget.TvSeasonWidget
method), 10
write_changes() (*tok-*
tokkie.gui.widgets.TvSeriesWidget.TvSeriesWidget
method), 11
write_config() (*tok-*
tokkie.gui.MainWindow.MainWindow method),
 11
write_metadata() (*toktokkie.Directory.Directory*
method), 38

X

X1080p (*toktokkie.xdcc_update.enums.Resolution* at-
tribute), 37
X480p (*toktokkie.xdcc_update.enums.Resolution* at-
tribute), 37
X720p (*toktokkie.xdcc_update.enums.Resolution* at-
tribute), 37
x_resolution() (*tok-*
tokkie.xdcc_update.XDCCUUpdater.XDCCUUpdater
property), 37
xdcc_update() (*toktokkie.Directory.Directory*
method), 38
XdccUpdateCommand (class in tok-
tokkie.scripts.XdccUpdateCommand), 33