
toktokkie

Release 0.18.0

Hermann Krumrey

Sep 11, 2019

CONTENTS

1	toktokkie package	3
1.1	Subpackages	3
1.2	Submodules	22
1.3	toktokkie.Directory module	22
1.4	toktokkie.exceptions module	22
1.5	Module contents	23
2	toktokkie	25
3	Indices and tables	27
	Python Module Index	29
	Index	31

Contents:

TOKTOKKIE PACKAGE

1.1 Subpackages

1.1.1 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)

Prints a black-on-yellow warning message :param text: The text to print :return: None

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/>>.

1.1.2 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.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: toktokkie.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

1.1.3 toktokkie.metadata package

Subpackages

toktokkie.metadata.components package

Submodules

toktokkie.metadata.components.BookVolume module

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

Class that models a single book volume

```
validate ()
    Validates the JSON data of the book volume. :return: None :raises InvalidMetadataException: If some-
    thing is wrong
    with the JSON data
```

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

property name

Returns The name of the part

```
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 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

toktokkie.metadata.components.enums module

```
class toktokkie.metadata.components.enums.BookIdType
    Bases: enum.Enum

    Enumeration that defines various ID types for Books

    ANILIST = 'anilist'

    ISBN = 'isbn'

    KITSU = 'kitsu'

    MYANIMELIST = 'myanimelist'

class toktokkie.metadata.components.enums.MangaIdType
    Bases: enum.Enum

    Enumeration that defines various ID types for Manga

    ANILIST = 'anilist'

    KITSU = 'kitsu'

    MYANIMELIST = 'myanimelist'

class toktokkie.metadata.components.enums.MediaType
    Bases: enum.Enum

    Enumeration that defines all possible media types

    BOOK = 'book'

    BOOK_SERIES = 'book_series'

    MANGA = 'manga'

    MOVIE = 'movie'

    TV_SERIES = 'tv'

    VISUAL_NOVEL = 'visual_novel'

class toktokkie.metadata.components.enums.MovieIdType
    Bases: enum.Enum

    Enumeration that defines various ID types for Movies

    ANILIST = 'anilist'

    IMDB = 'imdb'

    KITSU = 'kitsu'

    MYANIMELIST = 'myanimelist'

class toktokkie.metadata.components.enums.TvIdType
    Bases: enum.Enum

    Enumeration that defines various ID types for TV Series

    ANILIST = 'anilist'

    KITSU = 'kitsu'

    MYANIMELIST = 'myanimelist'

    TVDB = 'tvdb'
```

```
class toktokkie.metadata.components.enums.VisualNovelIdType
    Bases: enum.Enum

    Enumeration that defines various ID types for visual novels

    VNDB = 'vndb'
```

Module contents

toktokkie.metadata.helper package

Submodules

toktokkie.metadata.helper.functions module

```
toktokkie.metadata.helper.functions.create_metadata (directory: str, media_type: str) → tok-
    tokkie.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.helper.functions.get_metadata (directory: str) → tok-
    tokkie.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.helper.functions.get_metadata_class (media_type: str) → type
    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
```

toktokkie.metadata.helper.wrappers module

```
toktokkie.metadata.helper.wrappers.json_parameter (func: Callable) → Callable
    Decorator that implements additional error-catching for JSON parameters :param func: The function to wrap
    :return: The wrapped function
```

Module contents

toktokkie.metadata.prompt package

Submodules

toktokkie.metadata.prompt.CommaList module

```
class toktokkie.metadata.prompt.CommaList.CommaList (value: str)
    Bases: toktokkie.metadata.prompt.PromptType.PromptType

    Class that allows the automatic conversion of user-provided strings to values

    primitive
        alias of builtins.str
```


property value

Converts the string value into its actual value :return: The generated value

class toktokkie.metadata.prompt.CommaList.**IntCommaList** (*value: str*)

Bases: *toktokkie.metadata.prompt.CommaList.CommaList*

A comma list using integer values

primitive

alias of `builtins.int`

toktokkie.metadata.prompt.PromptType module

class toktokkie.metadata.prompt.PromptType.**PromptType** (*value: str*)

Bases: `object`

Class that allows the automatic conversion of user-provided strings to values

__init__ (*value: str*)

Initializes the prompt type :param value: The string value to parse

property value

Converts the string value into its actual value :return: The generated value

Module contents**Submodules****toktokkie.metadata.Book module**

class toktokkie.metadata.Book.**Book** (*directory_path: str, json_data: Optional[Dict[str, Any]] = None*)

Bases: *toktokkie.metadata.Metadata.Metadata*

Metadata class that model a Book

classmethod **id_type** () → `enum.EnumMeta`

Returns The ID type used by this metadata object

classmethod **media_type** () → `toktokkie.metadata.components.enums.MediaType`

Returns The media type of the Metadata class

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

toktokkie.metadata.BookSeries module

class toktokkie.metadata.BookSeries.**BookSeries** (*directory_path: str, json_data: Optional[Dict[str, Any]] = None*)

Bases: *toktokkie.metadata.Book.Book*

Metadata class that models a Book series

classmethod `media_type ()` → toktokkie.metadata.components.enums.MediaType

Returns The media type of the Metadata class

classmethod `prompt (directory_path: str)` → toktokkie.metadata.Book.Book

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

property `volumes`

Returns A list of book volumes for this book series

toktokkie.metadata.Manga module

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

Bases: *toktokkie.metadata.Metadata.Metadata*

Metadata class that models a Manga series

classmethod `id_type ()` → enum.EnumMeta

Returns The ID type used by this metadata object

property `main_path`

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

classmethod `media_type ()` → toktokkie.metadata.components.enums.MediaType

Returns The media type of the Metadata class

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

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.Metadata module

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

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*)

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

Raises `InvalidMetadataException` – if the metadata could not be parsed correctly

classmethod `from_directory` (*directory_path: str*)

Generates metadata for an existing media directory :param directory_path: The path to the media directory :raises InvalidMetadataException: if the metadata could not be

parsed correctly

Raises `MissingMetadataException` – if no metadata file was found

Returns The generated metadata object

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

classmethod `id_type` () → enum.EnumMeta

Returns The ID type used by this metadata object

property `ids`

Returns A dictionary containing lists of IDs mapped to ID types

classmethod `input` (*prompt_text: str, default: Optional[toktokkie.metadata.prompt.PromptType.PromptType], _type: type, required: bool = False*) → Any

Creates a user prompt that supports default options and automatic type conversions. :param prompt_text: The text to prompt the user :param default: The default value to use :param _type: The type of the prompted value :param required: Whether or not a response is required :return: The user's response

classmethod `media_type` () → toktokkie.metadata.components.enums.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*)

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

classmethod `prompt_for_ids` (*defaults: Optional[Dict[str, List[str]]] = None, required: Optional[List[enum.Enum]] = None*) → Dict[str, List[str]]

Prompts the user for IDs :param defaults: The default values to use, mapped to id type names :param required: A list of required ID types :return: The generated IDs. At least one ID will be included

property `tags`

Returns A list of tags

validate_json()

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.Movie module

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

Bases: *toktokkie.metadata.Metadata.Metadata*

Metadata class that model a Movie

classmethod id_type() → enum.EnumMeta

Returns The ID type used by this metadata object

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

Returns The media type of the Metadata class

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

toktokkie.metadata.TvSeries module

class toktokkie.metadata.TvSeries.**TvSeries** (*directory_path: str, json_data: Optional[Dict[str, Any]] = None*)

Bases: *toktokkie.metadata.Metadata.Metadata*

Metadata class that model a TV Series

add_exclude (*id_type: toktokkie.metadata.components.enums.TvIdType, 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.components.enums.TvIdType, 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 info 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.TvSeason
 Retrieves 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 id_type () → enum.EnumMeta
Returns The ID type used by this metadata object

classmethod media_type () → toktokkie.metadata.components.enums.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}}}

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

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

toktokkie.metadata.VisualNovel module

class toktokkie.metadata.VisualNovel.**VisualNovel** (*directory_path: str, json_data: Optional[Dict[str, Any]] = None*)
 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 id_type () → enum.EnumMeta

Returns The ID type used by this metadata object

classmethod media_type () → toktokkie.metadata.components.enums.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

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

Module contents

1.1.4 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:
                                         toktokkie.metadata.Metadata.Metadata)
    tok-
    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
    static generate_tv_episode_filename (original_file: str, series_name: str, sea-
                                         son_number: int, episode_number: int,
                                         episode_name: str, multi_end: Optional[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
    static get_ext (filename: str) → Optional[str]
        Gets the file extension of a file :param filename: The filename for which to get the file extension :return:
        The file extension or None if the file has no extension
    static 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 sea-
        son_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
    rename (noconfirm: bool)
        Renames the contained files according to the naming scheme. :param noconfirm: Skips the confirmation
        phase if True :return: None

```

Module contents

1.1.5 toktokkie.test package

Submodules

toktokkie.test.TestCheck module

```

class toktokkie.test.TestCheck.TestCheck (methodName='runTest')
    Bases: toktokkie.test.TestFramework.TestFramework
    Class that checks whether or not the validity checks work as intended
    config = {'tvdb_api': {}}
    test_check ()
        Returns None

```

verify (*should_dir: str, is_dir: str*)

Makes sure that two directories share the same content. :param should_dir: Directory :param is_dir: :return:

toktokkie.test.TestFramework module

class toktokkie.test.TestFramework.**TestFramework** (*methodName='runTest'*)

Bases: unittest.case.TestCase

A class that implements standard setUp and tearDown methods for unit tests

cleanup ()

Deletes any generated resources :return:

get (*directory: str*) → str

Retrieves the path to a test resource :param directory: The directory to get :return: The path to the directory

resources = 'toktokkie/test/res'

The directory containing the original test resources. The contents of this directory should not be modified during tests.

setUp ()

Sets up the test resources :return: None

tearDown ()

Deletes the test resources :return: None

test_res = 'test-res'

The directory containing a fresh copy of the test resources. May be modified during tests. Regenerated in setUp method.

toktokkie.test.TestRenaming module

class toktokkie.test.TestRenaming.**TestRenaming** (*methodName='runTest'*)

Bases: *toktokkie.test.TestFramework.TestFramework*

Test class that tests renaming and metadata structure using the 'Game of Thrones' test directory

test_renaming ()

Tests renaming the contents of the Game of Thrones directory :return: None

Module contents

1.1.6 toktokkie.xdcc_update package

Submodules

toktokkie.xdcc_update.XDCCUpdater module

class toktokkie.xdcc_update.XDCCUpdater.**XDCCUpdater** (*metadata: tok-*
tokkie.metadata.TvSeries.TvSeries)

Bases: object

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

__init__ (*metadata: toktokkie.metadata.TvSeries.TvSeries*)

Initializes the XDCC Updater object :param metadata: The metadata for the series for which to execute

an xdcc-update

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}}] . . .
A collection of predefined patterns

classmethod prompt (*metadata: toktokkie.metadata.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, metadata_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 ()

Performs an XDCC Update Action :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

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

t

- toktokkie, 23
- toktokkie.check, 7
 - toktokkie.check.BookChecker, 3
 - toktokkie.check.BookSeriesChecker, 3
 - toktokkie.check.Checker, 4
 - toktokkie.check.MangaChecker, 4
 - toktokkie.check.map, 6
 - toktokkie.check.MovieChecker, 5
 - toktokkie.check.TvSeriesChecker, 5
 - toktokkie.check.VisualNovelChecker, 6
- toktokkie.Directory, 22
- toktokkie.exceptions, 22
- toktokkie.iconizing, 8
 - toktokkie.iconizing.Iconizer, 8
 - toktokkie.iconizing.procedures, 8
 - toktokkie.iconizing.procedures.GnomeProcedure, 7
 - toktokkie.iconizing.procedures.Procedure, 8
- toktokkie.metadata, 18
 - toktokkie.metadata.Book, 13
 - toktokkie.metadata.BookSeries, 13
 - toktokkie.metadata.components, 12
 - toktokkie.metadata.components.BookVolume, 9
 - toktokkie.metadata.components.enums, 11
 - toktokkie.metadata.components.MetadataPart, 9
 - toktokkie.metadata.components.TvEpisode, 10
 - toktokkie.metadata.components.TvEpisodeRange, 10
 - toktokkie.metadata.components.TvSeason, 10
 - toktokkie.metadata.helper, 12
 - toktokkie.metadata.helper.functions, 12
 - toktokkie.metadata.helper.wrappers, 12
 - toktokkie.metadata.Manga, 14
 - toktokkie.metadata.Metadata, 14
 - toktokkie.metadata.Movie, 16
 - toktokkie.metadata.prompt, 13
 - toktokkie.metadata.prompt.CommaList, 12
 - toktokkie.metadata.prompt.PromptType, 13
 - toktokkie.metadata.TvSeries, 16
 - toktokkie.metadata.VisualNovel, 17
- toktokkie.renaming, 19
 - toktokkie.renaming.RenameOperation, 18
 - toktokkie.renaming.Renamer, 19
- toktokkie.test, 20
 - toktokkie.test.TestCheck, 19
 - toktokkie.test.TestFramework, 20
 - toktokkie.test.TestRenaming, 20
- toktokkie.xdcc_update, 22
 - toktokkie.xdcc_update.enums, 21
 - toktokkie.xdcc_update.XDCCUpdater, 20

Symbols

- `__init__()` (*toktokkie.Directory.Directory* method), 22
- `__init__()` (*toktokkie.check.Checker.Checker* method), 4
- `__init__()` (*toktokkie.iconizing.Iconizer.Iconizer* method), 8
- `__init__()` (*toktokkie.metadata.Metadata.Metadata* method), 14
- `__init__()` (*toktokkie.metadata.components.MetadataPart.MetadataPart* method), 9
- `__init__()` (*toktokkie.metadata.components.TvEpisode.TvEpisode* method), 10
- `__init__()` (*toktokkie.metadata.components.TvEpisodeRange.TvEpisodeRange* method), 10
- `__init__()` (*toktokkie.metadata.prompt.PromptType.PromptType* method), 13
- `__init__()` (*toktokkie.renaming.RenameOperation.RenameOperation* method), 18
- `__init__()` (*toktokkie.renaming.Renamer.Renamer* method), 19
- `__init__()` (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater* method), 20
- A**
- `add_exclude()` (*toktokkie.metadata.TvSeries.TvSeries* method), 16
- `add_multi_episode()` (*toktokkie.metadata.TvSeries.TvSeries* method), 16
- ANILIST (*toktokkie.metadata.components.enums.BookIdType* attribute), 11
- ANILIST (*toktokkie.metadata.components.enums.MangaIdType* attribute), 11
- ANILIST (*toktokkie.metadata.components.enums.MovieIdType* attribute), 11
- ANILIST (*toktokkie.metadata.components.enums.TvIdType* attribute), 11
- B**
- Book (*class in toktokkie.metadata.Book*), 13
- BOOK (*toktokkie.metadata.components.enums.MediaType* attribute), 11
- BOOK_SERIES (*toktokkie.metadata.components.enums.MediaType* attribute), 11
- BookChecker (*class in toktokkie.check.BookChecker*), 3
- BookIdType (*class in toktokkie.metadata.components.enums*), 11
- BookSeries (*class in toktokkie.metadata.BookSeries*), 13
- BookSeriesChecker (*class in toktokkie.check.BookSeriesChecker*), 4
- BookVolume (*class in toktokkie.metadata.components.BookVolume*), 9
- `bot()` (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater* property), 21
- C**
- `cgs()` (*toktokkie.metadata.VisualNovel.VisualNovel* property), 17
- `cleanup()` (*toktokkie.check.BookSeriesChecker.BookSeriesChecker* method), 4
- `check()` (*toktokkie.check.Checker.Checker* method), 4
- `check()` (*toktokkie.check.MangaChecker.MangaChecker* method), 5
- `check()` (*toktokkie.check.TvSeriesChecker.TvSeriesChecker* method), 6
- `check()` (*toktokkie.check.VisualNovelChecker.VisualNovelChecker* method), 6
- `check()` (*toktokkie.Directory.Directory* method), 22
- Checker (*class in toktokkie.check.Checker*), 4
- `checker_map` (*in module toktokkie.check.map*), 7
- `cleanup()` (*toktokkie.test.TestFramework.TestFramework* method), 20
- CommaList (*class in toktokkie.metadata.prompt.CommaList*), 12
- `config` (*toktokkie.test.TestCheck.TestCheck* attribute), 19
- `create_metadata()` (*in module toktokkie.metadata.helper.functions*), 12

D

default_procedure() (in module toktokkie.iconizing), 8
 Directory (class in toktokkie.Directory), 22

E

eds() (toktokkie.metadata.VisualNovel.VisualNovel property), 17
 episode_offset() (toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater property), 21
 episodes() (toktokkie.metadata.components.TvEpisodeRange.TvEpisodeRange property), 10
 error() (toktokkie.check.Checker.Checker method), 4
 excludes() (toktokkie.metadata.TvSeries.TvSeries property), 16

F

from_directory() (toktokkie.metadata.Metadata.Metadata class method), 15

G

generate_metadata() (toktokkie.Directory.Directory method), 22
 generate_tv_episode_filename() (toktokkie.renaming.Renamer.Renamer static method), 19
 get() (toktokkie.test.TestFramework.TestFramework method), 20
 get_episode_files() (toktokkie.metadata.TvSeries.TvSeries method), 16
 get_ext() (toktokkie.renaming.Renamer.Renamer static method), 19
 get_icon_file() (toktokkie.metadata.Metadata.Metadata method), 15
 get_metadata() (in module toktokkie.metadata.helper.functions), 12
 get_metadata_class() (in module toktokkie.metadata.helper.functions), 12
 get_season() (toktokkie.metadata.TvSeries.TvSeries method), 17
 GnomeProcedure (class in toktokkie.iconizing.procedures.GnomeProcedure), 7

H

has_cgs() (toktokkie.metadata.VisualNovel.VisualNovel property), 17
 has_ed() (toktokkie.metadata.VisualNovel.VisualNovel property), 17

has_op() (toktokkie.metadata.VisualNovel.VisualNovel property), 18
 has_ost() (toktokkie.metadata.VisualNovel.VisualNovel property), 18

I

iconize() (toktokkie.Directory.Directory method), 22
 iconize() (toktokkie.iconizing.Iconizer.Iconizer method), 8
 iconize() (toktokkie.iconizing.procedures.GnomeProcedure.GnomeProcedure class method), 7
 iconize() (toktokkie.iconizing.procedures.Procedure.Procedure class method), 8
 Iconizer (class in toktokkie.iconizing.Iconizer), 8
 id_type() (toktokkie.metadata.Book.Book class method), 13
 id_type() (toktokkie.metadata.Manga.Manga class method), 14
 id_type() (toktokkie.metadata.Metadata.Metadata class method), 15
 id_type() (toktokkie.metadata.Movie.Movie class method), 16
 id_type() (toktokkie.metadata.TvSeries.TvSeries class method), 17
 id_type() (toktokkie.metadata.VisualNovel.VisualNovel class method), 18
 ids() (toktokkie.metadata.components.MetadataPart.MetadataPart property), 9
 ids() (toktokkie.metadata.Metadata.Metadata property), 15
 IMDB (toktokkie.metadata.components.enums.MovieIdType attribute), 11
 input() (toktokkie.metadata.Metadata.Metadata class method), 15
 IntCommaList (class in toktokkie.metadata.prompt.CommaList), 13
 InvalidMetadata, 22
 InvalidXDCCInstructions, 22
 is_applicable() (toktokkie.iconizing.procedures.GnomeProcedure.GnomeProcedure class method), 7
 is_applicable() (toktokkie.iconizing.procedures.Procedure.Procedure class method), 8
 is_spinoff() (toktokkie.metadata.components.TvSeason.TvSeason method), 10
 ISBN (toktokkie.metadata.components.enums.BookIdType attribute), 11

J

json_parameter() (in module toktokkie.metadata.helper.wrappers), 12

K

KITSU (*toktokkie.metadata.components.enums.BookIdType* attribute), 11

KITSU (*toktokkie.metadata.components.enums.MangaIdType* attribute), 11

KITSU (*toktokkie.metadata.components.enums.MovieIdType* attribute), 11

KITSU (*toktokkie.metadata.components.enums.TvIdType* attribute), 11

L

`load_tvdb_episode_name()` (*toktokkie.renaming.Renamer.Renamer* static method), 19

M

`main_path()` (*toktokkie.metadata.Manga.Manga* property), 14

Manga (class in *toktokkie.metadata.Manga*), 14

MANGA (*toktokkie.metadata.components.enums.MediaType* attribute), 11

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

MangaIdType (class in *toktokkie.metadata.components.enums*), 11

`media_type()` (*toktokkie.metadata.Book.Book* class method), 13

`media_type()` (*toktokkie.metadata.BookSeries.BookSeries* class method), 13

`media_type()` (*toktokkie.metadata.Manga.Manga* class method), 14

`media_type()` (*toktokkie.metadata.Metadata.Metadata* class method), 15

`media_type()` (*toktokkie.metadata.Movie.Movie* class method), 16

`media_type()` (*toktokkie.metadata.TvSeries.TvSeries* class method), 17

`media_type()` (*toktokkie.metadata.VisualNovel.VisualNovel* class method), 18

MediaType (class in *toktokkie.metadata.components.enums*), 11

Metadata (class in *toktokkie.metadata.Metadata*), 14

MetadataMismatch, 22

MetadataPart (class in *toktokkie.metadata.components.MetadataPart*), 9

MissingMetadata, 23

MissingXDCCInstructions, 23

Movie (class in *toktokkie.metadata.Movie*), 16

MOVIE (*toktokkie.metadata.components.enums.MediaType* attribute), 11

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

MovieIdType (class in *toktokkie.metadata.components.enums*), 11

`multi_episodes()` (*toktokkie.metadata.TvSeries.TvSeries* property), 17

MYANIMELIST (*toktokkie.metadata.components.enums.BookIdType* attribute), 11

MYANIMELIST (*toktokkie.metadata.components.enums.MangaIdType* attribute), 11

MYANIMELIST (*toktokkie.metadata.components.enums.MovieIdType* attribute), 11

MYANIMELIST (*toktokkie.metadata.components.enums.TvIdType* attribute), 11

N

`name()` (*toktokkie.iconizing.procedures.Procedure.Procedure* attribute), 8

`name()` (*toktokkie.metadata.components.MetadataPart.MetadataPart* property), 9

`name()` (*toktokkie.metadata.Metadata.Metadata* property), 15

O

`ops()` (*toktokkie.metadata.VisualNovel.VisualNovel* property), 18

`ost()` (*toktokkie.metadata.VisualNovel.VisualNovel* property), 18

P

`p_resolution()` (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater* property), 21

`predefined_patterns` (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater* attribute), 21

`primitive()` (*toktokkie.metadata.prompt.CommaList.CommaList* attribute), 12

`primitive()` (*toktokkie.metadata.prompt.CommaList.IntCommaList* attribute), 13

`print_folder_icon_source()` (*toktokkie.metadata.Metadata.Metadata* method), 15

Procedure (class in *toktokkie.iconizing.procedures.Procedure*), 8

`prompt()` (*toktokkie.metadata.Book.Book* class method), 13

`prompt()` (*toktokkie.metadata.BookSeries.BookSeries* class method), 14

`prompt()` (*toktokkie.metadata.Manga.Manga* class method), 14

`prompt()` (*toktokkie.metadata.Metadata.Metadata* class method), 15

prompt () (*toktokkie.metadata.Movie.Movie class method*), 16
 prompt () (*toktokkie.metadata.TvSeries.TvSeries class method*), 17
 prompt () (*toktokkie.metadata.VisualNovel.VisualNovel class method*), 18
 prompt () (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater class method*), 21
 prompt_for_ids () (*toktokkie.metadata.Metadata.Metadata class method*), 15
 PromptType (*class in toktokkie.metadata.prompt.PromptType*), 13

R

reload () (*toktokkie.Directory.Directory method*), 22
 rename () (*toktokkie.Directory.Directory method*), 22
 rename () (*toktokkie.renaming.RenameOperation.RenameOperation method*), 18
 rename () (*toktokkie.renaming.Renamer.Renamer method*), 19
 RenameOperation (*class in toktokkie.renaming.RenameOperation*), 18
 Renamer (*class in toktokkie.renaming.Renamer*), 19
 Resolution (*class in toktokkie.xdcc_update.enums*), 21
 resolution () (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater property*), 21
 resources (*toktokkie.test.TestFramework.TestFramework attribute*), 20

S

sanitize () (*toktokkie.renaming.RenameOperation.RenameOperation static method*), 18
 search_engine () (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater property*), 21
 search_name () (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater property*), 21
 search_pattern () (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater property*), 21
 season () (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater property*), 21
 season_number () (*toktokkie.metadata.components.TvSeason.TvSeason property*), 10
 season_start_overrides () (*toktokkie.metadata.TvSeries.TvSeries property*), 17
 seasons () (*toktokkie.metadata.TvSeries.TvSeries property*), 17

setUp () (*toktokkie.test.TestFramework.TestFramework method*), 20
 special_chapters () (*toktokkie.metadata.Manga.Manga property*), 14
 special_path () (*toktokkie.metadata.Manga.Manga property*), 14

T

tags () (*toktokkie.metadata.Metadata.Metadata property*), 15
 tearDown () (*toktokkie.test.TestFramework.TestFramework method*), 20
 test_check () (*toktokkie.test.TestCheck.TestCheck method*), 19
 test_renaming () (*toktokkie.test.TestRenaming.TestRenaming method*), 20
 test_res (*toktokkie.test.TestFramework.TestFramework attribute*), 20
 TestCheck (*class in toktokkie.test.TestCheck*), 19
 TestFramework (*class in toktokkie.test.TestFramework*), 20
 TestRenaming (*class in toktokkie.test.TestRenaming*), 20
 toktokkie (*module*), 23
 toktokkie.check (*module*), 7
 toktokkie.check.BookChecker (*module*), 3
 toktokkie.check.BookSeriesChecker (*module*), 3
 toktokkie.check.Checker (*module*), 4
 toktokkie.check.MangaChecker (*module*), 4
 toktokkie.check.map (*module*), 6
 toktokkie.check.MovieChecker (*module*), 5
 toktokkie.check.TvSeriesChecker (*module*), 5
 toktokkie.check.VisualNovelChecker (*module*), 6
 toktokkie.Directory (*module*), 22
 toktokkie.exceptions (*module*), 22
 toktokkie.iconizing (*module*), 8
 toktokkie.iconizing.Iconizer (*module*), 8
 toktokkie.iconizing.procedures (*module*), 8
 toktokkie.iconizing.procedures.GnomeProcedure (*module*), 7
 toktokkie.iconizing.procedures.Procedure (*module*), 8
 toktokkie.metadata (*module*), 18
 toktokkie.metadata.Book (*module*), 13
 toktokkie.metadata.BookSeries (*module*), 13
 toktokkie.metadata.components (*module*), 12
 toktokkie.metadata.components.BookVolume (*module*), 9

toktokkie.metadata.components.enums (module), 11
 toktokkie.metadata.components.MetadataPart (module), 9
 toktokkie.metadata.components.TvEpisode (module), 10
 toktokkie.metadata.components.TvEpisodeRange (module), 10
 toktokkie.metadata.components.TvSeason (module), 10
 toktokkie.metadata.helper (module), 12
 toktokkie.metadata.helper.functions (module), 12
 toktokkie.metadata.helper.wrappers (module), 12
 toktokkie.metadata.Manga (module), 14
 toktokkie.metadata.Metadata (module), 14
 toktokkie.metadata.Movie (module), 16
 toktokkie.metadata.prompt (module), 13
 toktokkie.metadata.prompt.CommaList (module), 12
 toktokkie.metadata.prompt.PromptType (module), 13
 toktokkie.metadata.TvSeries (module), 16
 toktokkie.metadata.VisualNovel (module), 17
 toktokkie.renaming (module), 19
 toktokkie.renaming.RenameOperation (module), 18
 toktokkie.renaming.Renamer (module), 19
 toktokkie.test (module), 20
 toktokkie.test.TestCheck (module), 19
 toktokkie.test.TestFramework (module), 20
 toktokkie.test.TestRenaming (module), 20
 toktokkie.xdcc_update (module), 22
 toktokkie.xdcc_update.enums (module), 21
 toktokkie.xdcc_update.XDCCUpdater (module), 20
 TV_SERIES (toktokkie.metadata.components.enums.MediaType attribute), 11
 TVDB (toktokkie.metadata.components.enums.TvIdType attribute), 11
 tvdb_id() (toktokkie.metadata.components.TvSeason.TvSeason property), 10
 tvdb_id() (toktokkie.metadata.TvSeries.TvSeries property), 17
 TvEpisode (class in toktokkie.metadata.components.TvEpisode), 10
 TvEpisodeRange (class in toktokkie.metadata.components.TvEpisodeRange), 10
 TvIdType (class in toktokkie.metadata.components.enums), 11
 TvSeason (class in toktokkie.metadata.components.TvSeason), 10
 TvSeries (class in toktokkie.metadata.TvSeries), 16
 TvSeriesChecker (class in toktokkie.check.TvSeriesChecker), 6
 U
 update() (toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater method), 21
 V
 validate() (toktokkie.metadata.components.BookVolume.BookVolume method), 9
 validate() (toktokkie.metadata.components.MetadataPart.MetadataPart method), 9
 validate() (toktokkie.metadata.components.TvSeason.TvSeason method), 10
 validate_json() (toktokkie.metadata.Metadata.Metadata method), 15
 validate_json() (toktokkie.metadata.TvSeries.TvSeries method), 17
 value() (toktokkie.metadata.prompt.CommaList.CommaList property), 12
 value() (toktokkie.metadata.prompt.PromptType.PromptType property), 13
 verify() (toktokkie.test.TestCheck.TestCheck method), 19
 VISUAL_NOVEL (toktokkie.metadata.components.enums.MediaType attribute), 11
 VisualNovel (class in toktokkie.metadata.VisualNovel), 17
 VisualNovelChecker (class in toktokkie.check.VisualNovelChecker), 6
 VisualNovelIdType (class in toktokkie.metadata.components.enums), 11
 VNDB (toktokkie.metadata.components.enums.VisualNovelIdType attribute), 12
 volumes() (toktokkie.metadata.BookSeries.BookSeries property), 14
 W
 warn() (toktokkie.check.Checker.Checker method), 4
 write() (toktokkie.metadata.Metadata.Metadata method), 16
 write_metadata() (toktokkie.Directory.Directory method), 22
 X
 X1080p (toktokkie.xdcc_update.enums.Resolution attribute), 21

X480p (*toktokkie.xdcc_update.enums.Resolution* attribute), 21

X720p (*toktokkie.xdcc_update.enums.Resolution* attribute), 21

x_resolution() (*toktokkie.xdcc_update.XDCCUpdater.XDCCUpdater* property), 21

xdcc_update() (*toktokkie.Directory.Directory* method), 22

XDCCUpdater (class in *toktokkie.xdcc_update.XDCCUpdater*), 20