

---

**toktokkie**  
*Release 0.18.0*

**Feb 06, 2019**



# CONTENTS

<b>1</b>	<b>toktokkie package</b>	<b>3</b>
1.1	Subpackages . . . . .	3
1.2	Submodules . . . . .	19
1.3	toktokkie.Directory module . . . . .	19
1.4	toktokkie.exceptions module . . . . .	20
1.5	Module contents . . . . .	20
<b>2</b>	<b>toktokkie</b>	<b>21</b>
<b>3</b>	<b>Indices and tables</b>	<b>23</b>
	<b>Python Module Index</b>	<b>25</b>
	<b>Index</b>	<b>27</b>



Contents:



## TOKTOKKIE PACKAGE

### 1.1 Subpackages

#### 1.1.1 toktokkie.check package

##### Submodules

##### toktokkie.check.BookChecker module

Copyright 2015 Hermann Krumrey <[hermann@krumreyh.com](mailto: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)
```

```
    Bases: toktokkie.check.Checker.Checker
```

```
    Class that check Book media for consistency
```

##### toktokkie.check.BookSeriesChecker module

Copyright 2015 Hermann Krumrey <[hermann@krumreyh.com](mailto: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)

Bases: toktokkie.check.Checker.Checker

Class that check Book Series media for consistency
```

### toktokkie.check.Checker module

Copyright 2015 Hermann Krumrey <[hermann@krumreyh.com](mailto: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)

Bases: object

Class that performs checks on Metadata objects

__init__ (metadata: toktokkie.metadata.Metadata.Metadata, show_warnings: bool, fix_interactively:
          bool)
    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

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.MovieChecker module

Copyright 2015 Hermann Krumrey <[hermann@krumreyh.com](mailto: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)
```

Bases: *toktokkie.check.Checker.Checker*

Class that check Movie media for consistency

## **toktokkie.check.TvSeriesChecker module**

Copyright 2015 Hermann Krumrey <[hermann@krumreyh.com](mailto: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)
```

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.map module**

Copyright 2015 Hermann Krumrey <[hermann@krumreyh.com](mailto: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 = {<MediaTypes.TV_SERIES: 'tv'>: <class 'toktokkie.check.TvSeriesChecker'>}
    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](mailto: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

```

## 1.1.3 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 something 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

**ids**

**Returns** A dictionary containing lists of IDs mapped to ID types

**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

**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

**season\_number**

**Returns** The season number of the season

**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.MediaType
```

```
    Bases: enum.Enum
```

```
    Enumeration that defines all possible media types
```

```
    BOOK = 'book'
```

```
    BOOK_SERIES = 'book_series'
```

```
    MOVIE = 'movie'
```

```
    TV_SERIES = 'tv'
```

```
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'
```

## Module contents

### toktokkie.metadata.helper package

#### Submodules

#### toktokkie.metadata.helper.functions module

toktokkie.metadata.helper.functions.**create\_metadata** (*directory: str, media\_type: str*) → tok-

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-

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`

**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  
**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

#### volumes

**Returns** A list of book volumes for this book series

### 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

#### 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

#### 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

**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

### 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

### 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

### season\_start\_overrides

**Returns** A dictionary mapping episodes that override a season starting point to ID types Form: {idtype: {season: episode}}

### seasons

**Returns** A list of TV seasons

### 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

## 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: 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

**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 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

**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

**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: toktokkie.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

**bot**

**Returns** The bot to use for updating

**episode\_offset**

**Returns** The amount of episodes offset from 1 when updating

**p\_resolution**

**Returns** The resolution in P-notation (1080p)

**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

**resolution**

**Returns** The resolution in which to update the series

**search\_engine**

**Returns** The search engine to use

**search\_name**

**Returns** The name of the series for searching purposes

**search\_pattern**

**Returns** The search pattern to use

**season**

**Returns** The season to update

**update ()**  
 Executes the XDCC Update procedure :return: None

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



# INDEX

## Symbols

- `__init__()` (*toktokkie.Directory.Directory* method), 19
  - `__init__()` (*toktokkie.check.Checker.Checker* method), 4
  - `__init__()` (*toktokkie.iconizing.Iconizer.Iconizer* method), 7
  - `__init__()` (*toktokkie.metadata.Metadata.Metadata* method), 13
  - `__init__()` (*toktokkie.metadata.components.MetadataPart.MetadataPart* method), 8
  - `__init__()` (*toktokkie.metadata.components.TvEpisode.TvEpisode* method), 9
  - `__init__()` (*toktokkie.metadata.components.TvEpisodeRange.TvEpisodeRange* method), 9
  - `__init__()` (*toktokkie.metadata.prompt.PromptType.PromptType* method), 12
  - `__init__()` (*toktokkie.renaming.RenameOperation.RenameOperation* method), 16
  - `__init__()` (*toktokkie.renaming.Renamer.Renamer* method), 16
  - `__init__()` (*toktokkie.xdcc\_update.XDCCUpdater.XDCCUpdater* method), 18
- ## A
- `add_exclude()` (*toktokkie.metadata.TvSeries.TvSeries* method), 14
  - `add_multi_episode()` (*toktokkie.metadata.TvSeries.TvSeries* method), 15
- ## ANILIST
- (*toktokkie.metadata.components.enums.BookIdType* attribute), 10
  - (*toktokkie.metadata.components.enums.MovieIdType* attribute), 10
  - (*toktokkie.metadata.components.enums.TvIdType* attribute), 10
- ## B
- Book* (class in *toktokkie.metadata.Book*), 12
  - BOOK* (*toktokkie.metadata.components.enums.MediaType* attribute), 10
- BOOK\_SERIES* (*toktokkie.metadata.components.enums.MediaType* attribute), 10
  - BookChecker* (class in *toktokkie.check.BookChecker*), 3
  - BookIdType* (class in *toktokkie.metadata.components.enums*), 10
  - BookSeries* (class in *toktokkie.metadata.BookSeries*), 12
  - BookSeriesChecker* (class in *toktokkie.check.BookSeriesChecker*), 4
  - BookVolume* (class in *toktokkie.metadata.components.BookVolume*), 8
  - bot* (*toktokkie.xdcc\_update.XDCCUpdater.XDCCUpdater* attribute), 18
- ## C
- `check()` (*toktokkie.check.Checker.Checker* method), 4
  - `check()` (*toktokkie.check.TvSeriesChecker.TvSeriesChecker* method), 5
  - `check()` (*toktokkie.Directory.Directory* method), 19
  - `checker_map` (in module *toktokkie.check.map*), 5
  - `cleanup()` (*toktokkie.test.TestFramework.TestFramework* method), 17
- ## CommaList
- (class in *toktokkie.metadata.prompt.CommaList*), 11
- ## create\_metadata()
- (in module *toktokkie.metadata.helper.functions*), 11
- ## D
- `default_procedure()` (in module *toktokkie.iconizing*), 7
  - Directory* (class in *toktokkie.Directory*), 19
- ## E
- `episode_offset` (*toktokkie.xdcc\_update.XDCCUpdater.XDCCUpdater* attribute), 18
  - `episodes` (*toktokkie.metadata.components.TvEpisodeRange.TvEpisodeRange* attribute), 9
  - `error()` (*toktokkie.check.Checker.Checker* method), 4

- excludes (*toktokkie.metadata.TvSeries.TvSeries* attribute), 15
- ## F
- from\_directory() (*toktokkie.metadata.Metadata.Metadata* class method), 13
- ## G
- generate\_metadata() (*toktokkie.Directory.Directory* method), 19
- generate\_tv\_episode\_filename() (*toktokkie.renaming.Renamer.Renamer* static method), 16
- get() (*toktokkie.test.TestFramework.TestFramework* method), 17
- get\_episode\_files() (*toktokkie.metadata.TvSeries.TvSeries* method), 15
- get\_ext() (*toktokkie.renaming.Renamer.Renamer* static method), 16
- get\_icon\_file() (*toktokkie.metadata.Metadata.Metadata* method), 13
- get\_metadata() (in module *toktokkie.metadata.helper.functions*), 11
- get\_metadata\_class() (in module *toktokkie.metadata.helper.functions*), 11
- get\_season() (*toktokkie.metadata.TvSeries.TvSeries* method), 15
- GnomeProcedure (class in *toktokkie.iconizing.procedures.GnomeProcedure*), 6
- ## I
- iconize() (*toktokkie.Directory.Directory* method), 19
- iconize() (*toktokkie.iconizing.Iconizer.Iconizer* method), 7
- iconize() (*toktokkie.iconizing.procedures.GnomeProcedure.GnomeProcedure* class method), 6
- iconize() (*toktokkie.iconizing.procedures.Procedure.Procedure* class method), 6
- Iconizer (class in *toktokkie.iconizing.Iconizer*), 7
- id\_type() (*toktokkie.metadata.Book.Book* class method), 12
- id\_type() (*toktokkie.metadata.Metadata.Metadata* class method), 13
- id\_type() (*toktokkie.metadata.Movie.Movie* class method), 14
- id\_type() (*toktokkie.metadata.TvSeries.TvSeries* class method), 15
- ids (*toktokkie.metadata.components.MetadataPart.MetadataPart* class attribute), 8
- ids (*toktokkie.metadata.Metadata.Metadata* attribute), 13
- IMDB (*toktokkie.metadata.components.enums.MovieIdType* attribute), 10
- input() (*toktokkie.metadata.Metadata.Metadata* class method), 13
- IntCommaList (class in *toktokkie.metadata.prompt.CommaList*), 11
- InvalidMetadata, 20
- InvalidXDCCInstructions, 20
- is\_applicable() (*toktokkie.iconizing.procedures.GnomeProcedure.GnomeProcedure* class method), 6
- is\_applicable() (*toktokkie.iconizing.procedures.Procedure.Procedure* class method), 6
- is\_spinoff() (*toktokkie.metadata.components.TvSeason.TvSeason* method), 9
- ISBN (*toktokkie.metadata.components.enums.BookIdType* attribute), 10
- ## J
- json\_parameter() (in module *toktokkie.metadata.helper.wrappers*), 11
- ## K
- KITSU (*toktokkie.metadata.components.enums.BookIdType* attribute), 10
- KITSU (*toktokkie.metadata.components.enums.MovieIdType* attribute), 10
- KITSU (*toktokkie.metadata.components.enums.TvIdType* attribute), 10
- ## L
- load\_tvdb\_episode\_name() (*toktokkie.renaming.Renamer.Renamer* static method), 16
- ## M
- media\_type() (*toktokkie.metadata.Book.Book* class method), 12
- media\_type() (*toktokkie.metadata.BookSeries.BookSeries* class method), 12
- media\_type() (*toktokkie.metadata.Metadata.Metadata* class method), 13
- media\_type() (*toktokkie.metadata.Movie.Movie* class method), 14
- media\_type() (*toktokkie.metadata.TvSeries.TvSeries* class method), 15
- MediaType (class in *toktokkie.metadata.components.enums*), 10



- Metadata (class in *toktokkie.metadata.Metadata*), 13
- MetadataMismatch, 20
- MetadataPart (class in *toktokkie.metadata.components.MetadataPart*), 8
- MissingMetadata, 20
- MissingXDCCInstructions, 20
- Movie (class in *toktokkie.metadata.Movie*), 14
- MOVIE (*toktokkie.metadata.components.enums.MediaType* attribute), 10
- MovieChecker (class in *toktokkie.check.MovieChecker*), 5
- MovieIdType (class in *toktokkie.metadata.components.enums*), 10
- multi\_episodes (*toktokkie.metadata.TvSeries.TvSeries* attribute), 15
- MYANIMELIST (*toktokkie.metadata.components.enums.BookIdType* attribute), 10
- MYANIMELIST (*toktokkie.metadata.components.enums.MovieIdType* attribute), 10
- MYANIMELIST (*toktokkie.metadata.components.enums.TvIdType* attribute), 10
- ## N
- name (*toktokkie.iconizing.procedures.Procedure.Procedure* attribute), 7
- name (*toktokkie.metadata.components.MetadataPart.MetadataPart* attribute), 8
- name (*toktokkie.metadata.Metadata.Metadata* attribute), 13
- ## P
- p\_resolution (*toktokkie.xdcc\_update.XDCCUpdater.XDCCUpdater* attribute), 18
- primitive (*toktokkie.metadata.prompt.CommaList.CommaList* attribute), 11
- primitive (*toktokkie.metadata.prompt.CommaList.IntCommaList* attribute), 11
- print\_folder\_icon\_source() (*toktokkie.metadata.Metadata.Metadata* method), 13
- Procedure (class in *toktokkie.iconizing.procedures.Procedure*), 6
- prompt() (*toktokkie.metadata.Book.Book* class method), 12
- prompt() (*toktokkie.metadata.BookSeries.BookSeries* class method), 12
- prompt() (*toktokkie.metadata.Metadata.Metadata* class method), 14
- prompt() (*toktokkie.metadata.Movie.Movie* class method), 14
- prompt() (*toktokkie.metadata.TvSeries.TvSeries* class method), 15
- prompt() (*toktokkie.xdcc\_update.XDCCUpdater.XDCCUpdater* class method), 18
- prompt\_for\_ids() (*toktokkie.metadata.Metadata.Metadata* class method), 14
- PromptType (class in *toktokkie.metadata.prompt.PromptType*), 12
- ## R
- reload() (*toktokkie.Directory.Directory* method), 19
- rename() (*toktokkie.Directory.Directory* method), 19
- rename() (*toktokkie.renaming.RenameOperation.RenameOperation* method), 16
- rename() (*toktokkie.renaming.Renamer.Renamer* method), 17
- renameOperation (class in *toktokkie.renaming.RenameOperation*), 16
- renamer (class in *toktokkie.renaming.Renamer*), 16
- Resolution (class in *toktokkie.xdcc\_update.enums*), 19
- resolution (*toktokkie.xdcc\_update.XDCCUpdater.XDCCUpdater* attribute), 18
- resources (*toktokkie.test.TestFramework.TestFramework* attribute), 17
- ## S
- sanitize() (*toktokkie.renaming.RenameOperation.RenameOperation* static method), 16
- search\_engine (*toktokkie.xdcc\_update.XDCCUpdater.XDCCUpdater* attribute), 18
- search\_name (*toktokkie.xdcc\_update.XDCCUpdater.XDCCUpdater* attribute), 18
- search\_pattern (*toktokkie.xdcc\_update.XDCCUpdater.XDCCUpdater* attribute), 18
- search\_pattern (*toktokkie.xdcc\_update.XDCCUpdater.XDCCUpdater* attribute), 18
- season\_number (*toktokkie.metadata.components.TvSeason.TvSeason* attribute), 9
- season\_start\_overrides (*toktokkie.metadata.TvSeries.TvSeries* attribute), 15
- seasons (*toktokkie.metadata.TvSeries.TvSeries* attribute), 15
- setUp() (*toktokkie.test.TestFramework.TestFramework* method), 17
- ## T
- tags (*toktokkie.metadata.Metadata.Metadata* attribute), 14

- tearDown() (*toktokkie.test.TestFramework.TestFramework* method), 17
- test\_check() (*toktokkie.test.TestCheck.TestCheck* method), 17
- test\_renaming() (*toktokkie.test.TestRenaming.TestRenaming* method), 18
- test\_res (*toktokkie.test.TestFramework.TestFramework* attribute), 17
- TestCheck (class in *toktokkie.test.TestCheck*), 17
- TestFramework (class in *toktokkie.test.TestFramework*), 17
- TestRenaming (class in *toktokkie.test.TestRenaming*), 18
- toktokkie (module), 20
- toktokkie.check (module), 6
- toktokkie.check.BookChecker (module), 3
- toktokkie.check.BookSeriesChecker (module), 3
- toktokkie.check.Checker (module), 4
- toktokkie.check.map (module), 5
- toktokkie.check.MovieChecker (module), 4
- toktokkie.check.TvSeriesChecker (module), 5
- toktokkie.Directory (module), 19
- toktokkie.exceptions (module), 20
- toktokkie.iconizing (module), 7
- toktokkie.iconizing.Iconizer (module), 7
- toktokkie.iconizing.procedures (module), 7
- toktokkie.iconizing.procedures.GnomeProcedure (module), 6
- toktokkie.iconizing.procedures.ProcedureTvEpisodeRange (module), 6
- toktokkie.metadata (module), 16
- toktokkie.metadata.Book (module), 12
- toktokkie.metadata.BookSeries (module), 12
- toktokkie.metadata.components (module), 11
- toktokkie.metadata.components.BookVolume (module), 8
- toktokkie.metadata.components.enums (module), 10
- toktokkie.metadata.components.MetadataPart (module), 8
- toktokkie.metadata.components.TvEpisode (module), 9
- toktokkie.metadata.components.TvEpisodeRange (module), 9
- toktokkie.metadata.components.TvSeason (module), 9
- toktokkie.metadata.helper (module), 11
- toktokkie.metadata.helper.functions (module), 11
- toktokkie.metadata.helper.wrappers (module), 11
- toktokkie.metadata.Metadata (module), 13
- toktokkie.metadata.Movie (module), 14
- toktokkie.metadata.prompt (module), 12
- toktokkie.metadata.prompt.CommaList (module), 11
- toktokkie.metadata.prompt.PromptType (module), 12
- toktokkie.metadata.TvSeries (module), 14
- toktokkie.renaming (module), 17
- toktokkie.renaming.RenameOperation (module), 16
- toktokkie.renaming.Renamer (module), 16
- toktokkie.test (module), 18
- toktokkie.test.TestCheck (module), 17
- toktokkie.test.TestFramework (module), 17
- toktokkie.test.TestRenaming (module), 18
- toktokkie.xdcc\_update (module), 19
- toktokkie.xdcc\_update.enums (module), 19
- toktokkie.xdcc\_update.XDCCUpdater (module), 18
- TV\_SERIES (*toktokkie.metadata.components.enums.MediaType* attribute), 10
- TVDB (*toktokkie.metadata.components.enums.TvIdType* attribute), 10
- tvdb\_id (*toktokkie.metadata.components.TvSeason.TvSeason* attribute), 9
- tvdb\_id (*toktokkie.metadata.TvSeries.TvSeries* attribute), 15
- TvEpisode (class in *toktokkie.metadata.components.TvEpisode*), 9
- TvEpisodeRange (class in *toktokkie.metadata.components.TvEpisodeRange*), 9
- TvIdType (class in *toktokkie.metadata.components.enums*), 10
- TvSeason (class in *toktokkie.metadata.components.TvSeason*), 9
- TvSeries (class in *toktokkie.metadata.TvSeries*), 14
- TvSeriesChecker (class in *toktokkie.check.TvSeriesChecker*), 5
- ## U
- update() (*toktokkie.xdcc\_update.XDCCUpdater.XDCCUpdater* method), 18
- ## V
- validate() (*toktokkie.metadata.components.BookVolume.BookVolume* method), 8
- validate() (*toktokkie.metadata.components.MetadataPart.MetadataPart* method), 8
- validate() (*toktokkie.metadata.components.TvSeason.TvSeason* method), 9

`validate_json()` (*toktokkie.metadata.Metadata.Metadata* method), 14  
`validate_json()` (*toktokkie.metadata.TvSeries.TvSeries* method), 15  
`value` (*toktokkie.metadata.prompt.CommaList.CommaList* attribute), 11  
`value` (*toktokkie.metadata.prompt.PromptType.PromptType* attribute), 12  
`verify()` (*toktokkie.test.TestCheck.TestCheck* method), 17  
`volumes` (*toktokkie.metadata.BookSeries.BookSeries* attribute), 13

## W

`warn()` (*toktokkie.check.Checker.Checker* method), 4  
`write()` (*toktokkie.metadata.Metadata.Metadata* method), 14  
`write_metadata()` (*toktokkie.Directory.Directory* method), 19

## X

`X1080p` (*toktokkie.xdcc\_update.enums.Resolution* attribute), 19  
`X480p` (*toktokkie.xdcc\_update.enums.Resolution* attribute), 19  
`X720p` (*toktokkie.xdcc\_update.enums.Resolution* attribute), 19  
`x_resolution` (*toktokkie.xdcc\_update.XDCCUpdater.XDCCUpdater* attribute), 19  
`xdcc_update()` (*toktokkie.Directory.Directory* method), 19  
`XDCCUpdater` (*class in toktokkie.xdcc\_update.XDCCUpdater*), 18