
MediaCollectionDB Documentation

Release 0.1

Commonists

October 15, 2015

1	Modules	3
1.1	mediacollectiondb package	3
2	Authors	7
3	Indices and tables	9
	Python Module Index	11

MediaCollectionDB is a set of tools to play with Media Collections from MediaWiki.

- **Repo:** <https://github.com/Commonists/MediaCollectionDB>

1.1 mediacollectiondb package

Media Collection DB for program evaluation.

1.1.1 Infrastructure

Media and MediaCollection representations.

mediacollectiondb database define the media collection database and provides tools to manage the database.

```
class mediacollectiondb.media.Media (pageid, filename, uploader=None, width=None,  
height=None, size=None, quality_image=None, fea-  
tured_picture=None, valued_image=None, times-  
tamp=None)
```

Bases: `object`

Media representation in database.

pageid

str – MediaWiki pageid

filename

unicode – MediaWiki filename

uploader

unicode – Username of the uploader

width

int – width of the image

height

int – height of the image

size

int – Size in pixels of the image (width x height)

quality_image

bool – Whether the image is quality image or not

featured_picture

bool – Whether the image is featured picture or not

valued_image

bool – Whether the image is valued image or not

timestamp

date – Datetime of upload

classmethod fromtuple (*media*)

Return a Media from a tuple.

totuple ()

Return the object as a tuple of all values.

class `mediacollectiondb.media.MediaCollection` (*filename*)

Bases: `object`

Persistent media collection, i.e. collection saved into SQLite database.

connection

sqlite connection.

cursor

sqlite cursor.

COLLECTIONS_TABLE = 'media_collections'

exists (*media*)

Test whether a media is already in database or not.

In order to test whether the media is already in the database we count whether there is more than 0 element in the table with the same pageid

Parameters *media* (*dict*) – a media to check if it is in the database.

Returns True if a media with the same pageid exists in the database False otherwise

Return type `bool`

insert (*media*)

Insert a media into the database.

listall ()

List all elements of the collection.

reset ()

Delete all entry in the database.

save (*media*)

Add media to the persistent media collection.

Parameters *media* – media object to save to the into the persistent collection.

update (*media*)

Update a media already in database.

1.1.2 Metrics

Metrics module provide metrics for the Media Collections.

class `mediacollectiondb.metrics.MediaCollectionsMetrics` (*collections*)

Bases: `object`

Media Collections Metrics.

collections

list – list of MediaCollections

cache_allmedia

list – cache list of all the media.Media from collections. None until cache is filled.

fpcount ()

Amount of featured pictures in the set of collections.

listall ()

List all Media objects from the media collections.

mediacount ()

Count of media in the collections.

pixelcount ()

Amount of pixels within all media collections.

qicount ()

Amount of quality images in the set of collections.

qualitypixelcount ()

Amount of pixels from quality images in all media collections.

uploadercount ()

Amount of distinct uploader.

vicount ()

Amount of valued images in the set of collections.

widthcount (width)

Amount of pictures with a wider border greater than minimal width.

Parameters **width** (*int*) – minimal width.

Returns amount of picture with a wider border greater than minimal width

Return type *int*

Authors

Commonists

Indices and tables

- `genindex`
- `modindex`
- `search`

m

mediacollectiondb, 3
mediacollectiondb.media, 3
mediacollectiondb.metrics, 4

C

cache_allmedia (mediacollectiondb.metrics.MediaCollectionsMetrics attribute), 4

collections (mediacollectiondb.metrics.MediaCollectionsMetrics attribute), 4

COLLECTIONS_TABLE (mediacollectiondb.media.MediaCollection attribute), 4

connection (mediacollectiondb.media.MediaCollection attribute), 4

cursor (mediacollectiondb.media.MediaCollection attribute), 4

E

exists() (mediacollectiondb.media.MediaCollection method), 4

F

featured_picture (mediacollectiondb.media.Media attribute), 3

filename (mediacollectiondb.media.Media attribute), 3

fpcount() (mediacollectiondb.metrics.MediaCollectionsMetrics method), 5

fromtuple() (mediacollectiondb.media.Media class method), 4

H

height (mediacollectiondb.media.Media attribute), 3

I

insert() (mediacollectiondb.media.MediaCollection method), 4

L

listall() (mediacollectiondb.media.MediaCollection method), 4

listall() (mediacollectiondb.metrics.MediaCollectionsMetrics method), 5

M

Media (class in mediacollectiondb.media), 3

MediaCollection (class in mediacollectiondb.media), 4

mediacollectiondb (module), 3

mediacollectiondb.media (module), 3

mediacollectiondb.metrics (module), 4

MediaCollectionsMetrics (class in mediacollectiondb.metrics), 4

mediacount() (mediacollectiondb.metrics.MediaCollectionsMetrics method), 5

P

pageid (mediacollectiondb.media.Media attribute), 3

pixelcount() (mediacollectiondb.metrics.MediaCollectionsMetrics method), 5

Q

qicount() (mediacollectiondb.metrics.MediaCollectionsMetrics method), 5

quality_image (mediacollectiondb.media.Media attribute), 3

qualitypixelcount() (mediacollectiondb.metrics.MediaCollectionsMetrics method), 5

R

reset() (mediacollectiondb.media.MediaCollection method), 4

S

save() (mediacollectiondb.media.MediaCollection method), 4

size (mediacollectiondb.media.Media attribute), 3

T

timestamp (mediacollectiondb.media.Media attribute), 3
totuple() (mediacollectiondb.media.Media method), 4

U

update() (mediacollectiondb.media.MediaCollection
method), 4
uploader (mediacollectiondb.media.Media attribute), 3
uploadercount() (mediacollectiondb.metrics.MediaCollectionsMetrics
method), 5

V

valued_image (mediacollectiondb.media.Media attribute), 3
vicount() (mediacollectiondb.metrics.MediaCollectionsMetrics
method), 5

W

width (mediacollectiondb.media.Media attribute), 3
widthcount() (mediacollectiondb.metrics.MediaCollectionsMetrics
method), 5