
edeposit.amqp.downloader

Release 0.3.0

Sep 27, 2017

Contents

1	API	3
2	AMQP communication	7
3	Installation	9
4	Source code	11
5	Indices and tables	13
	Python Module Index	15

This module is basically just simple worker, which listens at AMQP queue and asynchronously downloads data from internet. Downloaded data are then returned as AMQP structure.

downloader package

Subpackages

downloader.structures package

Submodules

downloader.structures.requests module

class `downloader.structures.requests.Download`

Bases: `downloader.structures.requests.Download`

Download data from *url*.

url

str – URL of the internet resource.

Returns `DownloadedFile`.

Return type `obj`

Create new instance of `Download(url,)`

class `downloader.structures.requests.ProgressDownload`

Bases: `downloader.structures.requests.ProgressDownload`

Download data from *url* and report back the progress.

url

str – URL of the internet resource.

steps

int – Number of steps used to track progress.

Progress is reported using *Progress* structure.

Returns *DownloadedFile*.

Return type obj

Create new instance of ProgressDownload(url, steps)

class `downloader.structures.requests.CheckExistence`

Bases: *downloader.structures.requests.CheckExistence*

Send HEAD request to given *url* and check it's existence.

url

str – URL of the internet resource.

Returns *Exists*.

Return type obj

Create new instance of CheckExistence(url,)

downloader.structures.responses module

class `downloader.structures.responses.DownloadedFile`

Bases: *downloader.structures.responses.DownloadedFile*

Response to *Download* and *ProgressDownload*.

url

str – URL of the internet resource.

b64_data

str – Downloaded data encoded as base64 string.

Create new instance of DownloadedFile(url, b64_data)

class `downloader.structures.responses.Progress`

Bases: *downloader.structures.responses.Progress*

Response to *ProgressDownload*.

url

str – URL of the internet resource.

step

int – Number of current step.

downloaded

int – How many bytes was downloaded to this step.

content_length

int – How big is the whole file (in bytes).

Create new instance of Progress(url, step, downloaded, content_length)

class `downloader.structures.responses.Exists`

Bases: *downloader.structures.responses.Exists*

Response to *CheckExistence*.

url

str – URL of the internet resource.

result

bool – True if the file exists, False if not.

headers

dict – HTTP headers returned to this requests.

Create new instance of Exists(url, result, headers)

Module contents

Submodules

downloader.downloader module

`downloader.downloader.head_request(url)`

Send HEAD request to given *url*.

Parameters *url* (*str*) – URL of the internet resource.

Returns Dictionary with *headers*. Important headers: *content-length* and *content-type*.

Return type *dict*

`downloader.downloader.download(url)`

Download resource at *url*.

Parameters *url* (*str*) – URL of the internet resource.

Returns Content of the downloaded resource.

Return type *str*

`downloader.downloader.progress_download(url, steps, callback)`

Download resource at *url* and call *callback* after each step. The number of steps is defined by *steps* parameter.

Parameters

- *url* (*str*) – URL of the internet resource.
- *steps* (*int*) – Number of steps (how many times will be *callback* called).
- *callback* (*fn reference*) – Reference to function expecting three arguments: *step* (number of step), *downloaded* (number of downloaded bytes), *content_len* (size of downloaded resource).

Module contents

`downloader.get_progress_reporter(send_back)`

Construct progress reporter callback from *send_back* function.

Parameters *send_back* (*fn reference*) – Reference to function for sending messages back using AMQP.

Returns Function taking 3 parameters as is required by *progress_download()*.

Return type *fn reference*

`downloader.reactToAMQPMessage(message, send_back)`

React to given (AMQP) message. *message* is usually expected to be `collections.namedtuple()` structure filled with all necessary data.

Parameters

- **message** (**Request class*) – only `ConversionRequest` class is supported right now
- **send_back** (*fn reference*) – Reference to function for responding. This is useful for progress monitoring for example. Function takes one parameter, which may be response structure/namedtuple, or string or whatever would be normally returned.

Returns response filled with data about conversion and converted file.

Return type `ConversionResponse`

Raises `ValueError` – if bad type of *message* structure is given.

CHAPTER 2

AMQP communication

CHAPTER 3

Installation

Installation at debian systems is really easy:

```
pip install edeposit.amqp.downloader
```


CHAPTER 4

Source code

This project is released as opensource (GPL) and source codes can be found at GitHub:

- <https://github.com/edeposit/edeposit.amqp.downloader>

CHAPTER 5

Indices and tables

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

d

- `downloader`, 5
- `downloader.downloader`, 5
- `downloader.structures`, 5
- `downloader.structures.requests`, 3
- `downloader.structures.responses`, 4

B

b64_data (downloader.structures.responses.DownloadedFile attribute), 4

C

CheckExistence (class in downloader.structures.requests), 4

content_length (downloader.structures.responses.Progress attribute), 4

D

Download (class in downloader.structures.requests), 3

download() (in module downloader.downloader), 5

downloaded (downloader.structures.responses.Progress attribute), 4

DownloadedFile (class in downloader.structures.responses), 4

downloader (module), 5

downloader.downloader (module), 5

downloader.structures (module), 5

downloader.structures.requests (module), 3

downloader.structures.responses (module), 4

E

Exists (class in downloader.structures.responses), 4

G

get_progress_reporter() (in module downloader), 5

H

head_request() (in module downloader.downloader), 5

headers (downloader.structures.responses.Exists attribute), 5

P

Progress (class in downloader.structures.responses), 4

progress_download() (in module downloader.downloader), 5

ProgressDownload (class in downloader.structures.requests), 3

R

reactToAMQPMessage() (in module downloader), 5

result (downloader.structures.responses.Exists attribute), 4

S

step (downloader.structures.responses.Progress attribute), 4

steps (downloader.structures.requests.ProgressDownload attribute), 3

U

url (downloader.structures.requests.CheckExistence attribute), 4

url (downloader.structures.requests.Download attribute), 3

url (downloader.structures.requests.ProgressDownload attribute), 3

url (downloader.structures.responses.DownloadedFile attribute), 4

url (downloader.structures.responses.Exists attribute), 4

url (downloader.structures.responses.Progress attribute), 4