

Python Documentation

version

April 23, 2020

Sommaire

Welcome to Juradinfo IA's documentation!	1
Contents:	1
Data pipeline documentation	1
Ocr module	1
Base module	2
Entities module	2
Indices and tables	2
Index	3
Index des modules Python	5

Welcome to Juradinfo IA's documentation!

Contents:

Data pipeline documentation

Ocr module

Module containing the definition of the Ocr class.

```
class Ocr.Ocr
```

Class to manage OCR actions.

classmethod `create_folder (folder)`

Create a folder if it doesn't exist

classmethod `nb_jpg_in_folder (folder)`

Return the number of jpg images in a specify folder

classmethod `ocr_image (img)`

Extract text from an image

classmethod `ocr_pdf_image (folder_of_pdf, filename, prefix='.')`

Extract a text from a pdf of images

classmethod `ocr_pdf_text (filename)`

Extract text from pdf of text

classmethod `pdf2images (filename, folder_of_pdf, folder_of_images)`

Convert a pdf file into set of jpg images, one image for one page of the pdf file

classmethod `preprocess_image (img)`

Preprocess an image before OCR operation

Contains definition of OcrConcurrent class.

```
class Ocr_batch.OcrConcurrent
```

Class to manage parallel execution of the OCR pipeline

classmethod `concurrent_ocr (nb_thread, lst_filename, outdir='data', bool_db=0)`

Run several threads, each thread for an image file

classmethod `concurrent_pdf2image (nb_thread, lst_path, img_folder='data')`

Run several threads, each thread for an image file

classmethod `worker (id_thread)`

Instructions runs by each thread

classmethod `worker_df (id_thread)`

Instructions runs by each thread

classmethod `worker_pdf2image (id_thread, img_folder)`

Instructions runs by each thread

TODO : the goal of this scrip

Execute OCR pipeline on files inside the input directory

Base module

`class base.ConfigReader.Config`

Class to manage the conf file.

Contains definition of DataBase class.

`class base.db.DataBase`

Manage database actions

Module to define usefull functions across the project;

`base.utils.get_logger` (logging_conf_path, logger_name, log_handler_path)

Returns the logger from the conf file after having created the log file handler. :param logging_conf_path: the logging config file path :param logger_name: the name of the logger defined inside the config file :param log_handler_path: the file which will handle the log or sys.stdout

Entities module

`class entities.page.Page` (**kwargs)

Indices and tables

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

Index

B

base.ConfigReader

module

base.db

module

base.utils

module

C

concurrent_ocr() (méthode de la classe Ocr_batch.OcrConcurrent)

concurrent_pdf2image() (méthode de la classe Ocr_batch.OcrConcurrent)

Config (classe dans base.ConfigReader)

create_folder() (méthode de la classe Ocr.Ocr)

D

DataBase (classe dans base.db)

E

entities.page

module

G

get_logger() (dans le module base.utils)

M

module

base.ConfigReader

base.db

base.utils

entities.page

Ocr

Ocr_batch

pdf2imageBatch

pipelineOcerisation

N

nb_jpg_in_folder() (méthode de la classe Ocr.Ocr)

O

Ocr

module

Ocr (classe dans Ocr)

Ocr_batch

module

ocr_image() (méthode de la classe Ocr.Ocr)

ocr_pdf_image() (méthode de la classe Ocr.Ocr)

ocr_pdf_text() (méthode de la classe Ocr.Ocr)

OcrConcurrent (classe dans Ocr_batch)

P

Page (classe dans entities.page)

pdf2imageBatch

module

pdf2images() (méthode de la classe Ocr.Ocr)

pipelineOcerisation

module

preprocess_image() (méthode de la classe Ocr.Ocr)

W

worker() (méthode de la classe Ocr_batch.OcrConcurrent)

worker_df() (méthode de la classe Ocr_batch.OcrConcurrent)

worker_pdf2image() (méthode de la classe Ocr_batch.OcrConcurrent)

Index des modules Python

b

[base](#)

[base.ConfigReader](#)

[base.db](#)

[base.utils](#)

e

[entities](#)

[entities.page](#)

o

[Ocr](#)

[Ocr_batch](#)

p

[pdf2imageBatch](#)

[pipelineOcerisation](#)