depdf

Release 0.2.1

Jun 04, 2020

Contents:

CHAPTER 1

depdf

An ultimate pdf file disintegration tool. DePDF is designed to extract tables and paragraphs into structured markup language [eg. html] from embedding pdf pages. You can also use it to convert page/pdf to html.

depdf.**convert_pdf_to_html**(*pdf*, ***kwargs*)

Parameters

- **pdf** – pdf file path
- **kwargs** – config keyword arguments

Returns pdf html string

depdf.**convert_page_to_html**(*pdf*, *pid*, ***kwargs*)

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page html string

depdf.**extract_page_tables**(*pdf*, *pid*, ***kwargs*)

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page tables list

depdf.**extract_page_paragraphs**(*pdf*, *pid*, ***kwargs*)

Parameters

- **pdf** – pdf file path

- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page paragraphs list

CHAPTER 2

depdf

2.1 depdf package

2.1.1 Subpackages

depdf.components package

Submodules

depdf.components.image module

class depdf.components.image.**Image**(*bbox=None*, *src=''*, *percent=100*, *pid='1'*, *img_idx=1*, *scan=False*, *config=None*)

Bases: *depdf.base.Base*, *depdf.base.Box*

object_type = 'image'

depdf.components.paragraph module

class depdf.components.paragraph.**Paragraph**(*bbox=None*, *text=''*, *pid='1'*, *para_idx=1*, *config=None*, *inner_objects=None*, *style=None*, *align=None*)

Bases: *depdf.base.InnerWrapper*, *depdf.base.Box*

object_type = 'paragraph'

save_html()

depdf.components.span module

class depdf.components.span.**Span**(*bbox=None*, *span_text=''*, *config=None*, *style=None*)
Bases: *depdf.base.Base*, *depdf.base.Box*

object_type = 'span'

depdf.components.table module

class depdf.components.table.**Cell**(*bbox=None*, *text=''*, *inner_objects=None*)
Bases: *depdf.base.InnerWrapper*, *depdf.base.Box*

object_type = 'cell'

to_dict

class depdf.components.table.**Table**(*rows*, *pid='1'*, *tid=1*, *config=None*, *bbox=None*)
Bases: *depdf.base.Base*, *depdf.base.Box*

html

object_type = 'table'

save_html()

to_dict

to_html

depdf.components.table.**convert_table_to_html**(*table_dict*, *pid='1'*, *tid=1*, *tc_mt=5*, *table_class='pdf-table'*, *skip_et=False*)

depdf.components.table.**gen_column_cell_sizes**(*t*)

depdf.components.text module

class depdf.components.text.**Text**(*bbox=''*, *text=''*)
Bases: *depdf.base.Base*, *depdf.base.Box*

object_type = 'text'

Module contents

class depdf.components.**Paragraph**(*bbox=None*, *text=''*, *pid='1'*, *para_idx=1*, *config=None*, *inner_objects=None*, *style=None*, *align=None*)
Bases: *depdf.base.InnerWrapper*, *depdf.base.Box*

object_type = 'paragraph'

save_html()

class depdf.components.**Table**(*rows*, *pid='1'*, *tid=1*, *config=None*, *bbox=None*)
Bases: *depdf.base.Base*, *depdf.base.Box*

html

object_type = 'table'

save_html()

to_dict

to_html

class depdf.components.**Span**(*bbox=None*, *span_text=''*, *config=None*, *style=None*)
Bases: *depdf.base.Base*, *depdf.base.Box*

object_type = 'span'

class depdf.components.**Text**(*bbox=''*, *text=''*)
Bases: *depdf.base.Base*, *depdf.base.Box*

object_type = 'text'

class depdf.components.**Cell**(*bbox=None*, *text=''*, *inner_objects=None*)
Bases: *depdf.base.InnerWrapper*, *depdf.base.Box*

object_type = 'cell'

to_dict

class depdf.components.**Image**(*bbox=None*, *src=''*, *percent=100*, *pid='1'*, *img_idx=1*, *scan=False*, *config=None*)
Bases: *depdf.base.Base*, *depdf.base.Box*

object_type = 'image'

2.1.2 Submodules

2.1.3 depdf.api module

depdf.api.**api_load_pdf**(*api_func*)

depdf.api.**convert_page_to_html**(*pdf*, *pid*, ***kwargs*)

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page html string

depdf.api.**convert_pdf_to_html**(*pdf*, ***kwargs*)

Parameters

- **pdf** – pdf file path
- **kwargs** – config keyword arguments

Returns pdf html string

depdf.api.**extract_page_paragraphs**(*pdf*, *pid*, ***kwargs*)

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page paragraphs list

depdf.api.**extract_page_tables**(*pdf*, *pid*, ***kwargs*)

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page tables list

2.1.4 depdf.base module

class depdf.base.**Base**
Bases: object

html

refresh()

reset()

soup

to_dict

to_soup(*parser*)

write_to(*file_name*)

class depdf.base.**Box**
Bases: object

bbox

bottom = Decimal('0')

height

static normalize_bbox(*bbox*)

top = Decimal('0')

width

x0 = Decimal('0')

x1 = Decimal('0')

class depdf.base.**InnerWrapper**
Bases: *depdf.base.Base*

inner_objects

to_dict

2.1.5 depdf.config module

class depdf.config.**Config**(***kwargs*)
Bases: object

add_horizontal_line_tolerance = Decimal('0.1')

add_horizontal_lines_flag = False

```
add_line_flag = False
add_vertical_lines_flag = False
char_overlap_size = 3
char_size_lower = Decimal('3')
char_size_upper = Decimal('30')
column_region_half_width = 4
```

copy(***kwargs*)

```
curved_line_flag = False
debug_flag = False
default_char_size = Decimal('12')
default_head_tail_page_offset_percent = 0.1
dotted_line_flag = True
header_footer_flag = True
image_class = 'pdf-image'
image_flag = True
image_resolution = 300
log_level = 30
logo_flag = True
main_frame_tolerance = None
max_columns = 3
max_double_line_tolerance = 3
min_column_region_objects = 1
min_double_line_tolerance = Decimal('0.05')
min_image_size = 80
mini_page_class = 'pdf-mini-page'
multiple_columns_flag = True
page_class = 'pdf-page'
page_num_left_fraction = Decimal('0.44')
page_num_right_fraction = Decimal('0.56')
page_num_top_fraction = Decimal('0.75')
paragraph_class = 'pdf-paragraph'
paragraph_flag = True
pdf_class = 'pdf-content'
resolution = 144
skip_empty_table = False
snap_flag = False
```

span_class = 'pdf-span'

table_cell_merge_tolerance = 5

table_class = 'pdf-table'

table_flag = True

temp_dir_prefix = 'temp_depdf'

to_dict

unique_prefix = None

update(***kwargs*)

verbose_flag = False

vertical_double_line_tolerance = Decimal('2')

x_tolerance = None

y_tolerance = None

depdf.config.**check_config**(*func*)

depdf.config.**check_config_type**(*config*)

2.1.6 depdf.error module

exception depdf.error.**BoxValueError**(*value*)
Bases: ValueError

exception depdf.error.**ConfigTypeError**(*value*)
Bases: TypeError

exception depdf.error.**PDFTypeError**(*value*)
Bases: TypeError

exception depdf.error.**PageTypeError**(*value*)
Bases: TypeError

2.1.7 depdf.log module

depdf.log.**logger_init**(*name*)

2.1.8 depdf.page module

class depdf.page.**DePage**(*page*, *pid='1'*, *same=None*, *logo=None*, *config=None*, *columns=1*, *mini=False*)
Bases: *depdf.base.Base*

analyze_lines()

analyze_main_frame()

analyze_page_attributes()

analyze_paragraph_border()

ave_cs = 0

ave_lh = 3

```
border = (0, 0, 0, 0)
chars
check_if_toc_page()
check_multi_column_page()
config
debug = False
extract_images()
extract_paragraph()
extract_phrases()
extract_tables()
frame_bottom = 0
frame_top = 0
h_edges = []
height
html
images
images_raw
min_cs = 0
new_para_end_flag = None
new_para_start_flag = None
object_key_list = ['_tables', '_paragraphs', '_images']
objects
orientation = ''
page
pagination_phrases = []
paragraphs
phrases = None
pid
prefix = UUID('0479b05d-265b-401c-b327-b24755c37464')
process_mini_page()
process_page()
refresh()
save_html()
screenshot
set_global()
tables
```

tables_raw

temp_dir = 'temp'

to_html

to_screenshot()

toc_flag = False

v_edges = []

verbose = False

width

x_tolerance = 3

y_tolerance = 3

class depdf.page.**MiniDePage**(*page*, *pid='1'*, *same=None*, *logo=None*, *config=None*, *columns=1*, *mini=False*)

Bases: *depdf.page.DePage*

save_html()

to_html

depdf.page.**check_page_type**(*page*)

depdf.page.**convert_plumber_table**(*pdf_page*, *table*, *pid='1'*, *tid=1*, *config=None*, *min_cs=1*)

depdf.page.**extract_cell_region**(*cell_region*, *bbox*, *config=None*, *pid='1'*, *tid=1*, *cid=1*)

2.1.9 depdf.page_tools module

depdf.page_tools.**add_horizontal_lines**(*v_lines*, *h_lines*, *vlts_tolerance=0.1*)

depdf.page_tools.**add_vertical_lines**(*v_lines*, *h_lines*, *page_rects*, *page*, *ave_cs*)

depdf.page_tools.**analyze_char_size**(*chars*, *char_size_upper=30*, *char_size_lower=3*, *default_char_size=12*)

depdf.page_tools.**analyze_page_num_word**(*phrases*, *page_height*, *page_width*, *top_fraction=Decimal('0.7')*, *left_fraction=Decimal('0.4')*, *right_fraction=Decimal('0.6')*)

depdf.page_tools.**analyze_page_orientation**(*plumber_page*)

Parameters **plumber_page** – pdfplumber.page.Page class

Returns

depdf.page_tools.**calculate_paragraph_border**(*depdf_page_object*)

depdf.page_tools.**curve_to_lines**(*curves*)

depdf.page_tools.**edges_to_lines**(*edges*)

depdf.page_tools.**format_text**(*text*)

depdf.page_tools.**merge_page_figures**(*pdf_page*, *tables_raw=None*, *logo=None*, *min_width=3*, *min_height=3*, *pid='1'*)

depdf.page_tools.**remove_duplicate_chars**(*chars*, *overlap_size=3*)

depdf.page_tools.**remove_single_lines**(*lines*, *max_double=3*, *min_double=0.05*, *vertical_double=2*, *m='h'*)

2.1.10 depdf.pdf module

class depdf.pdf.**DePDF**(*pdf*, *config=None*, ***kwargs*)

Bases: *depdf.base.Base*

close()

config

extract_html_pages()

generate_pages()

get_prefix()

html

html_pages

classmethod load(*file_name*, *config=None*, ***kwargs*)

logo

classmethod open(**args*, ***kwargs*)

page_num

pages

pdf

refresh()

same

save_html()

to_html

depdf.pdf.**check_pdf_type**(*pdf*)

2.1.11 depdf.pdf_tools module

depdf.pdf_tools.**check_page_orientation**(*pdf*, *pid*)

Parameters

- **pdf** – pdfplumber class
- **pid** – page number starts from 0

Returns

depdf.pdf_tools.**pdf_head_tail**(*pdf*, *config=None*)

Parameters

- **pdf** – plumber pdf object
- **config** – depdf config class

Returns PDF

depdf.pdf_tools.**pdf_logo**(*pdf*)

2.1.12 depdf.settings module

2.1.13 depdf.utils module

depdf.utils.**calc_bbox**(*objects*)

depdf.utils.**calc_overlap**(*a*, *b*)
:param a: [a_lower, a_upper] :param b: [b_lower, b_upper] :return: overlapping length

depdf.utils.**construct_style**(*style=None*)

depdf.utils.**convert_html_to_soup**(*html*, *parser='html.parser'*)

depdf.utils.**convert_soup_to_html**(*soup*)

depdf.utils.**repr_str**(*text*, *max_length=5*)

2.1.14 depdf.version module

2.1.15 Module contents

depdf

An ultimate pdf file disintegration tool. DePDF is designed to extract tables and paragraphs into structured markup language [eg. html] from embedding pdf pages. You can also use it to convert page/pdf to html.

class depdf.**Config**(***kwargs*)
Bases: object

add_horizontal_line_tolerance = Decimal('0.1')

add_horizontal_lines_flag = False

add_line_flag = False

add_vertical_lines_flag = False

char_overlap_size = 3

char_size_lower = Decimal('3')

char_size_upper = Decimal('30')

column_region_half_width = 4

copy(***kwargs*)

curved_line_flag = False

debug_flag = False

default_char_size = Decimal('12')

default_head_tail_page_offset_percent = 0.1

dotted_line_flag = True

header_footer_flag = True

image_class = 'pdf-image'

```
image_flag = True
image_resolution = 300
log_level = 30
logo_flag = True
main_frame_tolerance = None
max_columns = 3
max_double_line_tolerance = 3
min_column_region_objects = 1
min_double_line_tolerance = Decimal('0.05')
min_image_size = 80
mini_page_class = 'pdf-mini-page'
multiple_columns_flag = True
page_class = 'pdf-page'
page_num_left_fraction = Decimal('0.44')
page_num_right_fraction = Decimal('0.56')
page_num_top_fraction = Decimal('0.75')
paragraph_class = 'pdf-paragraph'
paragraph_flag = True
pdf_class = 'pdf-content'
resolution = 144
skip_empty_table = False
snap_flag = False
span_class = 'pdf-span'
table_cell_merge_tolerance = 5
table_class = 'pdf-table'
table_flag = True
temp_dir_prefix = 'temp_depdf'
to_dict
unique_prefix = None
update(**kwargs)
verbose_flag = False
vertical_double_line_tolerance = Decimal('2')
x_tolerance = None
y_tolerance = None
```

class depdf.**DePDF**(*pdf*, *config=None*, ***kwargs*)

Bases: *depdf.base.Base*

close()

config

extract_html_pages()

generate_pages()

get_prefix()

html

html_pages

classmethod load(*file_name*, *config=None*, ***kwargs*)

logo

classmethod open(**args*, ***kwargs*)

page_num

pages

pdf

refresh()

same

save_html()

to_html

class depdf.**DePage**(*page*, *pid='1'*, *same=None*, *logo=None*, *config=None*, *columns=1*, *mini=False*)

Bases: *depdf.base.Base*

analyze_lines()

analyze_main_frame()

analyze_page_attributes()

analyze_paragraph_border()

ave_cs = 0

ave_lh = 3

border = (0, 0, 0, 0)

chars

check_if_toc_page()

check_multi_column_page()

config

debug = False

extract_images()

extract_paragraph()

extract_phrases()

extract_tables()

frame_bottom = 0

```
    frame_top = 0
    h_edges = []
    height
    html
    images
    images_raw
    min_cs = 0
    new_para_end_flag = None
    new_para_start_flag = None
    object_key_list = ['_tables', '_paragraphs', '_images']
    objects
    orientation = ''
    page
    pagination_phrases = []
    paragraphs
    phrases = None
    pid
    prefix = UUID('0479b05d-265b-401c-b327-b24755c37464')
    process_mini_page()
    process_page()
    refresh()
    save_html()
    screenshot
    set_global()
    tables
    tables_raw
    temp_dir = 'temp'
    to_html
    to_screenshot()
    toc_flag = False
    v_edges = []
    verbose = False
    width
    x_tolerance = 3
    y_tolerance = 3
```

depdf.**convert_pdf_to_html**(*pdf*, ***kwargs*)

Parameters

- **pdf** – pdf file path
- **kwargs** – config keyword arguments

Returns pdf html string

depdf.**convert_page_to_html**(*pdf*, *pid*, ***kwargs*)

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page html string

depdf.**extract_page_tables**(*pdf*, *pid*, ***kwargs*)

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page tables list

depdf.**extract_page_paragraphs**(*pdf*, *pid*, ***kwargs*)

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page paragraphs list

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

d

Index

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W

X

Y