

UnoGenerator documentation

Version: 0.41.0



1 Introduction

UnoGenerator uses Libreoffice UNO API python bindings to generate documents. So in order to use, you need to launch a --headless LibreOffice instance. We recomend to launch then easily with 'unogenerator_start' script.

UnoGenerator has standard templates to help you with edition, although you can use your own templates. You can edit this one or create your own. This document has been created with 'standard.odt' files that you can find inside this python module.

1.1 Installation

You can use pip to install this python package:

```
pip install unogenerator
```

1.2 ODT 'Hello World' example

This is a Hello World example. You get the example in odt, docx and pdf formats:

```
from unogenerator import ODT_Standard
with ODT_Standard() as doc:
    doc.addParagraph("Hello World", "Heading 1")
    doc.addParagraph("Easy, isn't it", "Standard")
    doc.save("hello_world.odt")
    doc.export_docx("hello_world.docx")
    doc.export_pdf("hello_world.pdf")
```

2 ODT

ODT files can be quickly generated with UnoGenerator. There is a predefined template in code called 'standard.odt' to help you with edition.

2.1 Calling the ODT constructor

You can call ODT constructor in this ways:

- ODT with standard template (Recomended). There is a predefined template in code called 'standard.odt', inside this python module, to help you with edition, although you can use your own ones. With this mode you can create new documents

```
from unogenerator import ODT_Standard
doc=ODT_Standard()
```

- ODT with template or file (Recomended). With this mode you can read your files to overwrite them or use your file as a new template to create new documents

```
from unogenerator import ODT
doc=ODT('yourdocument.odt')
```

- ODT without template. With this mode you can write your files with Libreoffice default styles. If you want to create new ones, you should write them using Libreoffice API code

```
from unogenerator import ODT
doc=ODT()
```

2.2 Styles

To call default Libreoffice paragraph styles you must use their english name. You can see their names with this method:

```
doc.print_styles()
```

2.3 Tables

We can create tables with diferent font sizes and formats:[15, 70, 15]

Concept	Value	Comment
Text	This is a text	Good
Datetime	2024-07-10 21:12:59.467922	Good
Date	2024-07-10	Good
Float	12.121	Good
Currency	-12.12 €	Good
Percentage	33.33 %	Good

Concept	Value	Comment
Text	This is a text	Good
Datetime	2024-07-10 21:12:59.467922	Good
Date	2024-07-10	Good
Float	12.121	Good

Currency	-12.12 €	Good
Percentage	33.33 %	Good

2.4 Lists and numbered lists

- Simple list
 - Simple list
 - Simple list
- Simple list
 - Simple list
- Simple list

2.5 Hyperlinks

If you want to go to Google, click on this [link](#). That's all folks!
[Other link](#)


2.6 HTML code

- This is a html list

This is a html paragraph.

2.7 Images

This is an 'image as char' example:  . Now I keep writing without problems.

This is an image loaded from bytes: 

As you can see, I can reuse it one hundred times. File size will not be increased because I used reference names. 

The next paragraph is generated with the illustration method



The next paragraph is generated with the illustration method



You can play with image width and height:



width are set to None.

Image default size. Height and

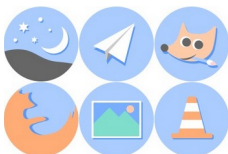


Image width 3cm of width and height automatically set.

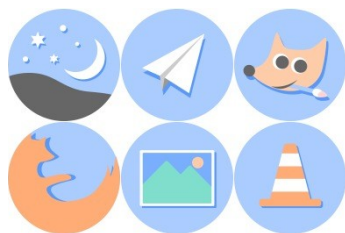


Image height 3cm of width and width automatically set.



Image width and height set to 3cm.

You can trim image border white space when needed:



This is a camerawith gray space around it: . You can trim that gray space when needed with 'bytes_after_trim_image' method. To use this method you need Imagemagick



installed to use 'convert' command. This is the result:

2.8 Search and Replace

Below this paragraph is a paragraph with a % REPLACEME % (Without white spaces) text and it's going to be replaced after all document is been generated

This paragraph was set after replacement.

This paragraph was set after a page break.

This paragraph was set after a page break with Landscape style.

This is a pair of brackets).NOW)(

This is a set of symbols: .,:;?0a-/(). REPLACEDNOW)(

This paragraph was set at the end of the code after a find and replace command.

3 ODS

3.1 ODS 'Hello World' example

This is a Hello World example. You'll get the example in ods, xlsx and pdf formats:

```
from unogenerator import ODS_Standard
with ODS_Standard() as doc:
    doc.addCellMergedWithStyle("A1:E1", "Hello world", style="BoldCenter")
    doc.save("hello_world.ods")
    doc.export_xlsx("hello_world.xlsx")
    doc.export_pdf("hello_world.pdf")
```