

Template for Testing Secretary

1. CONTROL FLOW & VARIABLE PRINTING

Let's try a classical country-capital table:
If used properly, Secretary will always respect the original format in the template.

Country	Capital
United States	Washington
	OTHER IMPORTANT CITIES
	Miami New York California Texas Atlanta
England	London
	OTHER IMPORTANT CITIES
	Gales
Japan	Tokio
	OTHER IMPORTANT CITIES
	Hiroshima Nagazaki
Nicaragua	Managua
	OTHER IMPORTANT CITIES
	Leon Granada Masaya
Argentina	Buenos Aires
	OTHER IMPORTANT CITIES
	– not other cities known –

Country	Capital
Chile	Santiago
	OTHER IMPORTANT CITIES
	– not other cities known –
Mexico	Mexico City
	OTHER IMPORTANT CITIES
	Puebla Cancun
Total Countries printed 7	

Control-Flow using “before::” and “after::”.

Countries
United States
England
Japan
Nicaragua
Argentina
Chile
Mexico

This template file test secretary's markdown filter, flow control and variable printing

2. MARKDOWN FILTER

(content of README.md should be visible below)

python-odt-template

\n

[!IMPORTANT]\n This package currently contains minimal features and is a work-in-progress

Table of Contents

- [python-odt-template\n\nTable of Contents\nInstallation\nUsage\n\nJinja2\nDjango\n\nAlternatives\nCredits\nLicense\n](#)

Installation

console\npip install python-odt-template\n

Usage

Jinja2

```
``python\nfrom pythonodttemplate import ODTTemplate\nfrom pythonodttemplate.jinja import\nenablemarkdown\nfrom pythonodttemplate.jinja import getodtrenderer\nfrom\npythonodttemplate.libreoffice import convertto_pdf\nodtrenderer = getodtrenderer(mediapath="inputs")\nwith ODTTemplate("inputs/simpletemplate.odt") as template:\n    odtrenderer.render(\n        template,\n        context={"document": document, "countries": countries},\n    )\n    template.pack("simpletemplaterendered.odt")\n    converttopdf("simpletemplaterendered.odt", "outputs")\n``
```

Django

```
``python
```

settings.py

Add at least one staticfiles dirs, this is what the imgae filter will use to find images

```
STATICFILES_DIRS = [BASEDIR / "example" / "static"]
```

Add the image filter to the builtins templates config

```
TEMPLATES = [\n    {\n        "BACKEND": "django.template.backends.django.DjangoTemplates",\n        ...\n    }\n]
```

This document was rendered on 2024-06-25 21:21:48.556079 with Secretary.
7 countries printed.

```
"OPTIONS": {\n ...\n "builtins": ["pythonodttemplate.django"],\n },\n },\n ]
```

views.py

```
from pythonodttemplate import ODTTemplate\nfrom pythonodttemplate.django import getodtrenderer\nfrom pythonodttemplate.libreoffice import converttopdf\nodtrenderer = getodt_renderer()\ndef renderodt(request):\n    with ODTTemplate("template.odt") as template:\n        odtrenderer.render(\n            template,\n            {"image": "writer.png"},\n        )\n        template.pack("templaterendered.odt")\n        converttopdf("templaterendered.odt", "outputs")\n        return FileResponse(\n            open("outputs/templaterendered.pdf", "rb"),\n            as_attachment=True,\n            filename="template_rendered.pdf"\n        )
```

Alternatives

- [python-docx-template](#)

Credits

Thanks to [secretary](#) for the enormous amount of integration work on Jinja2 and ODT.

License

python-odt-template is distributed under the terms of the [MIT](#) license.