tpl4md : Markdown templates to ease the creation of Markdown documents

Object

`tpl4md` provide markdown templates for widely used documents such as simple pdf documents, complex pdf documents, letters, invoices, orders, or even slides. The goal is to be able to focus on the content that will be written in markdown. The rest is handle by pandoc, latex etc.

Translation

This `README` file is also available in french

Prerequisite

Linux/Mac OSX

tpl4md is using different technologies to generate documents:

- Pandoc used to convert Markdown documents to various document formats is available on all the major platforms. The pandoc version must be greater than 1.9.4.2 or the latest available.
- The LaTeX document typesetting system. Don't worry, you won't have to write LaTeX code! Well perhaps a little bit ... (The following packages are mandatory : `texlive`, `texlive-latex-recommended`, `texlive-xetex`, `ttf-bitstream-vera` and `texlive-latex-extra`)
- The Markdown format for the writing of documents. Some extensions are provided by pandoc to support tables and other cool stuff and you can find the documentation on the related page
- The compilation tool : Make
- A simple or more advanced editor (providing syntax highlightning) to write markdown (Mou available for the Mac OSX platform is pretty good with a decent WYSIWYG support)
- A recent version of the Python framework (v2.7 is ok) with `python-tk` available

Windows

- Pandoc used to convert Markdown documents to various document formats is available on all the major platforms. The pandoc version must be greater than 1.9.4.2 or the latest available.

- The LaTeX document typesetting system. Don't worry, you won't have to write LaTeX code! Well perhaps a little bit ... On windows platforms my preffered one is the TeXLive (worked for me at least) or the MikTeX distribution John Mac Farlane is (le créateur de Pandoc) sur son site.
- Une version récente de Python (celle de la communauté ou celle d'ActiveState)
- Le format Markdown pour l'écriture du document avec les extensions fournies par Pandoc dont on peut trouver la description sur la page dédiée
- Un éditeur de texte pour taper le markdown (Notepad ++ semple aller, à voir s'il existe un éditeur WYSIWIG sympa)
- Les logiciels make/sed et l'ensemble des coreutils de gnu portés sur Windows
- Les polices d'écriture présentes dans le svn (Marketing/03-Print/fonts) ajoutées au système d'exploitation

Prerequisites installation walkthrough for Windows Operating Systems

- Pandoc's installation : you just have to go on the download page and get the **msi** installer : https://code.google.com/p/pandoc/downloads/list
- Get the mandatory unix ports on Windows and install them in a easy to remember directory (e.g. `C:\gnuwin32`) :
- Make : http://gnuwin32.sourceforge.net/downlinks/make.php
- Sed : http://sourceforge.net/projects/gnuwin32/files//sed/4.2.1/sed-4.2.1-setup.exe/download
- CoreUtils : http://gnuwin32.sourceforge.net/downlinks/coreutils.php
- Install a flavor of the Python framework (2.7.* or higher) : this one http://python.org/ftp/python/2.7.5/python-2.7.5.msi or that one http://www.activestate.com/activepython/downloads/thank-you?dl=http://downloads.activestate.com/ActivePython/releases/2.7.2.5/ActivePython-2.7.2.5-win32-x86.msi
- Install the TexLive LaTeX distribution http://mirror.ctan.org/systems/texlive/tlnet/install-tl.zip, uncompress it and then execute `install-tl-advanced.bat` that will allow you to install all the LaTeX packages available. Remember the path where the TexLive is installed (something like `C:\texlive2012` for example).

Once you're done just add the path to the executable binaries in the `PATH` (global) and `Path` (user specific) variables.

How to install

From the github repository

To install `tpl4md` from the github repository :

```
$ cd output_path
$ git clone git@github.com:dloureiro/tpl4md.git .
```

Then you just have to add the `bin` directory to your `PATH` variable.

Using pip (not available)

To install `tpl4md` using the pip installer :

```
pip install tpl4md
```

It will automatically be added to your `PATH` variable.

Content of the project

Repository Structure

```
root
|- bin
|  |- marksf_gen
|  |- marksf_gen.bat
|- share
|  |- common
|  |  |- bin
|  |  |  |- config_parsing
|  |  |  |- generate_files
|  |- doc
|  |  |- README.pdf
|  |  |- README-fr.pdf
|  |- templates              // The list of available templates.
|     |- epub                // Take a look at their README.md for more info
|     |- pdf-cplx
|     |- pdf-lettre
|     |- pdf-simple
|     |- invoice-fr
```

```
|     |- order-fr
|     |- slides-reveal
|- README.md
|- README-fr.md
```

Content description

- **bin/tpl4md**: Main script generating the base project depending on the chosen template
- **bin/tpl4md.bat**: Corresponding batch script for windows
- **README(-fr).md**: This readme file
- **share/doc/README(-fr).pdf** : This readme file (and the french translation) in pdf version
- **share/common/bin/config_parser**: Script used to retrieve some variables from the configuration script
- **share/templates**: The directory containing template generation files
- **share/templates/epub**: Template folder for the ebooks
- **share/templates/pdf-simple**: Template folder for simple pdf documents
- **share/templates/pdf-cplx**: Template folder for complex pdf documents
- **share/templates/pdf-lettre**: Template folder for letters
- **share/templates/invoice-fr**: Template folder for invoices (in french for the moment)
- **share/templates/order-fr**: Template folder for orders (in french for the moment)
- **share/templates/slides-reveal**: Template folder for slides powered by reveal.js

`tpl4md`

This script is the principal elements of the project: it allows the generation of empty projects for documents following existing templates. Here is the script signature:

```
tpl4md [-h] [-l] [-o OUTPUT] [-i] [-t TYPE]
```

- **-l** or –**list** : Option used to list of the avaiable templates (provided in the default distribution and the one available in your local home directory
- **-t** or –**type** : Option used to select the template type to use (one of those provided by the **-l** option)
- **-o** or –**output** : Option used to define the name output directory (it will be created and must not exist before the execution of `tpl4md`). It will be filled with the elements of the selected template

- **-i** or –**interface** : This option provide a graphical usr interface to `tpl4md` that will allow you to define :
- the type of template to use
- the name of the directory to create
- the path where this directory will be created

tpl4md.bat

This file is a shortcut for Windows users for the folowing command : `tpl4md -i`

config_parser

This script is used to retrieve variables from the configuration file. It is mainly use by the `Makefile` to get some useful information such as the name of the source file or the name of the destination file.

It is called as follows:

```
python ./bin/config_parser key
```

key being the key you want the value for.

** Warning: there is no reason to use that script outsite of the `Makefile`. Its sole usage is the generation of a temporary file **

Template

Main description

A template is pretty free in is implementation. Some rules must still be observed for their use by `tpl4md`.

Here are the main principles for a well written template. You can refer to the specific documentation of each template for more details.

Each template is currently composed of the following elements:

- A `Makefile` with two targets:
- the main one used for the compilation
- `clean` used for the clean up

The main target does not currently follows any pattern regarding its name (the standardisation is in the **TODO** list) and is provided as a batch file for the Windows users.

- A configuration file in the *JSON* format named `configuration.json` containing the information used during the generation by each template (those info will change for each template, even if some of them are mandatory for each template)
- A file called `generator.py` that will deal with the copy of the template files to the destination folder and that will be the entry point for a template

And other files could then be added in each template to perform the needed operations.

Global and user specific templates

Templates can be located in two places :

- in the `share/templates` folder near the bin directory where `tpl4md` is installed. This folder will hold the global templates, mainly provided with the distribution of `tpl4md`
- in the `$HOME/.tpl4md/templates` folder for the user specific templates. You can put templates of your own in that directory. they will be available through `tpl4md`. It provided a way to use new templates without modifying the distribution of `tpl4md`

README.pdf

The readme file in the pdf format is generated with the following command:

```
pandoc -f markdown-raw_tex README.md -o share/doc/README.pdf
```

TODO list

A lot of enhancements can be added to the existing project. Here is a non-exhaustive list:

- Template standardisation regarding the copy of the template files
- Factorization of the template elements
- Génération of a HTML document
- **DONE** Addition of Reveal.js presentation in a template
- Makefile targets standardization
- **DONE**: `pdflatex` -> `XeTeX` for a better support of UTF-8 and custom fonts
- Better simple and complex pdf templates (regarding the watermark support)

- User's guide for the creation of templates
- **DONE**: Separation of the file adapting the templates depending on the `configuration.json`
- Addition of a CSS stylesheet and a cover image for the epub
- **DONE**: Definition of a non-technic document template
- **DONE**: Preprocessing that will take a input file in the *JSON* format that would allow the adaptation of the templates