

Documentation du template pdf complexe

Introduction

Cette documentation est en lien avec le template pour la création de pdf complexes pour le programme `tpl4md`

Structure du template

```
root
|- Makefile
|- generate_files
|- configuration.json
|- dblatex
|- doc
|  |- README_pdf-cplx.md
|  |- README_pdf-cplx.pdf
|  |- README-fr_pdf-cplx.md
|  |- README-fr_pdf-cplx.pdf
|- document.md
|- generator.py
|- images
|  |- hw.jpg
|- make-clean.bat
|- make-pdf-cplx.bat
|- pdf-template-cplx.tex
|- tplformd-cplx.sty
```

Description de ce contenu

Fichier	Description
Makefile	Makefile pour la génération/clean du pdf complexe d'output
doc/README_pdf-cplx.(md/pdf)	La traduction de ce fichier de readme en anglais
doc/README-fr_pdf-cplx.(md/pdf)	Ce fichier de readme
dblax	Feuilles de style nécessaire pour la compilation du template complexe
configuration.json	Fichier de configuration pour le fichier pdf final

generator.py	Fichier permettant la génération du projet basé sur le template
document.md	Le document de base d'exemple pour ce template
images/*	Un ensemble d'images de base
make-clean.bat	Fichier batch à utiliser sous windows pour le nettoyage d'un projet
make-pdf-cplx.bat	Fichier batch à utiliser sous windows pour la génération du pdf complexe
pdf-template-cplx.tex	Fichier de template utilisé pour la création de l'enrobage du fichier pdf
tplformd-cplx.sty	Feuille de style définissant le style complexe

Nous allons aborder les fichiers les plus importants.

Makefile

Fichier de compilation permettant la gestion de la compilation d'un document complexe en pdf :

- pdf-cplx : génération de la version simple en pdf
- clean : pour le nettoyage

pdf-template-cplx.tex

Ce fichier LaTeX décrit l'ensemble des éléments qui forment le template qui permet la création des fichiers complexes en pdf avec notamment l'ensemble des éléments de préambule, de vérification, de relecture, les en-têtes et pieds de pages, etc.

Ce fichier possède des balises de la forme :

!title!

Qui sont automatiquement remplies par le script generate_files

Pour information, voici les variables qui sont utilisées :

Ce fichier est celui qui définit les éléments suivant :

- Le titre avec la balise \title. Ex :

`\title{Exemple de document}`

- L'auteur avec la balise `\author`. Ex :

`\author{John Doe}`

- Le nom de projet avec la balise `\TPLprojectname`. Ex :

`\TPLprojectname{Projet 9999}`

- Le client avec la balise `\TPLclient`. Ex :

`\TPLclient{Client 404}`

- Une liste de collaborateur qui ont travaillé sur le document avec les différents rôle qu'ils ont pu avoir. Ex :

```
\renewcommand{\TPLindexation}{
\begin{TPLindtable}
  % le rédacteur, son nom, la date
  \TPLinditem{\writtenby}{J. Doe}{01 avril 2012}
  % le relecteur, son nom, la date
  \TPLinditem{\verifiedby}{J. Bon}{06 avril 2012}
  % le valideur, son nom, la date
  \TPLinditem{\approvedby}{B. Dupont}{08 avril 2012}
\end{TPLindtable}
}
```

- Une table des révisions du document correspondant. Ex :

```
\renewcommand{\TPLrevhistory}{
\begin{TPLrevtable}
  % numéro de révision, date, description, auteur
  \TPLrevitem{1}{08/03/2012}{Première version}{J. Doe}
  \TPLrevitem{2}{25/03/2012}{Améliorations}{J.Doe, J. Bon}
\end{TPLrevtable}
}
```

- Une table de documents de référence. Ex :

```

\renewcommand{\TPLreferenceTable}{
\begin{TPLreftable}
% référence, nom du document, description du document
\TPLrefitem{ref1}{techDocument}{Ce document}
\TPLrefitem{ref2}{techDocument}{Ce document}
\end{TPLreftable}
}

```

- Une table d'autorisation. Ex :

```

\renewcommand{\TPLauthview}{
\begin{SFauthviewtable}
% Nom du groupe/entreprise, liste des personnels autorisés
\TPLauthviewitem{Client 404}{Tous les membres}
\TPLauthviewitem{Exemple sas}{Tous les membres de Exemple sas}
\end{TPLauthviewtable}
}

```

D'autres options sont disponibles dans le template comme :

- `\TPLreleaseinfo` pour afficher une box en en-tête de page affichant un texte. Ex :

```

\TPLreleaseinfo{Réalisé}

```

- `\showwatermark` pour avoir un filigrane utilisé sur les pages. Vous pouvez utiliser du texte ou mettre un image en utilisant l'environnement LaTeX adapté. Ex :

```

\showwatermark{DRAFT}

```

- `\TPLprojectleader` pour définir un chef de projet. Ex :

```

\TPLprojectleader{Moi}

```

\underline{Note} : Sauf si vous souhaitez modifier le template lui-même, nous n'avez pas à modifier ce fichier lorsque vous souhaitez créer un document.

document.md

Ce fichier est le document maître dans lequel il faudra ajouter l'ensemble des informations qui vont constituer le corps du résultat produit.

À noter que la syntaxe markdown supportée est une extension de la version officielle et vous pouvez retrouver l'ensemble des éléments fournis par Pandoc dont on peut trouver la description sur [la page dédiée](#).

\underline{Note} : Vous pouvez donner le nom que vous voulez à ce fichier tant que vous donnez le nom du fichier source dans le fichier de configuration

configuration.json

Voici le contenu d'un fichier complet de configuration avec les variables correspondantes pour un pdf complexe :

```
{
  "source_file_name":"document.md", // nom du fichier source
  "destination_file_name":"example", // nom du fichier de destination
  "client":"Client", // nom du client
  "title" : "Pretty good document", // titre du document
  "author" : "John Doe", // auteur du document
  "lang":"french", // langue dans laquelle le document est rédigé
  "author_company" : "Example inc.", // nom de la société de l'auteur
  "project_name" : "Projet 999", // nom du projet
  "writtenby":{" // rédacteur du document
    "name":"J. Doe", // son nom
    "date":"01 avril 2012" // la date de rédaction
  },
  "verifiedby":{" // vérificateur du document
    "name":"J. Bon", // son nom
    "date":"06 avril 2012" // la date de vérification
  },
  "approvedby":{" // la personne ayant approuvé le document
    "name":"B. Dupont", // son nom
    "date":"08 avril 2012" // la date
  },
  "revisions":[ // la liste des révisions du document
    {
      "version":"1", // version
      "date":"08/03/2012", // date
      "description":"First version", // description des ajouts de la révision
      "author":"J. Doe" // son auteur
    },
    {
```

```

        "version":"2", // même chose
        "date":"25/03/2012",
        "description":"Improvements",
        "author":"J. Doe, J. Bon"
    }
],
"references":[ // liste de références pour ce document
    {
        "reference":"ref1", // référence
        "document_name":"techDocument1", // nom du fichier
        "description":"Awesome reference" // description de la référence
    },
    {
        "reference":"ref2", // même chose
        "document_name":"techDocument2",
        "description":"Really awesome reference"
    }
],
"authorizations":[ // list des personnes autorisées à accéder au document
    {
        "structure":"Example inc", // expéditeur
        "people":"All members" // liste des personnes
    },
    {
        "structure":"Client", // même chose chez le destinataire
        "people":"All members"
    }
],
"footer":{ // elements allant apparaître dans le pied de page
    "left":[ // pied gauche
        "A marvelous company", // une ligne
        "We are a rock-solid team", // une autre ligne
        "located at the edge of the univers"
    ],
    "right":[ // pied droit
        "Web : \\url{www.example.com}", // on peut utiliser du latex
        "e-mail : \\url{contact@example.com}",
        "Tel : 01 23 45 67 89"
    ]
},
"release_info":"Draft", // information de release.
"water_mark":"DRAFT", // water mark (ex : Draft, confidentiel, etc)
"project_leader":"Me!" // project leader
}

```

generator.py

Ce fichier est utilisé par le script `tpl4md` pour la copie des fichiers d'un template lors de la création d'un projet basé sur ce template. Il déclare deux variables :

- `version` : version du template
- `description` : description du template

Elle serve notamment pour l'affichage de la liste des templates disponibles.

Il implémente aussi deux fonctions :

- `usage_message`: fonction affichant l'usage du template
- `generate` : fonction générant les fichiers spécifique d'une template lors de la création d'un nouveau projet

Structure d'un projet généré

Voici, après génération, le contenu d'un projet pour un document complexe :

```
root
|- Makefile
|- bin
|  |- config_parsing
|  |- generate_files
|- common
|  |- pdf-template-cplx.tex
|  |- tplformd-cplx.sty
|  |- dblatex
|- configuration.json
|- doc
|  |- README.md
|  |- README-fr.md
|  |- README_pdf-cplx.md
|  |- README_pdf-cplx.pdf
|  |- README-fr_pdf-cplx.md
|  |- README-fr_pdf-cplx.pdf
|  |- README.pdf
|  |- README-fr.pdf
|- images
|  |- hw.jpg
|- document.md
|- make-clean.bat
|- make-pdf-cplx.bat
```

README.pdf

Le readme en pdf est générée avec la commande suivante :

```
pandoc -f markdown-raw_tex README-fr_pdf-cplx.md -o README-fr_pdf-cplx.pdf
```