

Documentation du template pdf complexe

Introduction

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

Structure du template

```
root
|- Makefile
|- configuration.json
|- dblatex
|- doc
|  |- README_pdf-cplx.md
|  |- README_pdf-cplx.pdf
|- document.md
|- generator.py
|- images
|  |- hw.jpg
|- make-clean.bat
|- make-pdf-cplx.bat
|- pdf-template-cplx.tex
|- sysfera-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) | Ce fichier de readme |
| dblatex | 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 complexe |
| sysfera-cplx.sty | 1 Feuille de style définissant le style complexe SysFera |

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 au style SysFera avec notamment l'ensemble des éléments de préambule, de vérification, de relecture, les en-têtes et pieds de pages, le background, 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 SysFera}
```

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

```
\author{John Doe}
```

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

```
\SFprojectname{Projet 9999}
```

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

```
\SFclient{Client 404}
```

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

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

- Une table des révisions du document correspondant. Ex :

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

- Une table de documents de référence. Ex :

```
\renewcommand{\SFreferenceTable}{
\begin{SFreferenceTable}
% référence, nom du document, description du document
\SFrefitem{ref1}{techDocument}{Ce document}
\SFrefitem{ref2}{techDocument}{Ce document}
\end{SFreferenceTable}
}
```

- Une table d'autorisation. Ex :

```
\renewcommand{\SFauthview}{
\begin{SFauthviewtable}
% Nom du groupe/entreprise, liste des personnels autorisés
\SFauthviewitem{Client 404}{Tous les membres}
\SFauthviewitem{SysFera}{Tous les membres de SysFera}
\end{SFauthviewtable}
}
```

D'autres options sont disponibles dans le template comme :

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

```
\SFreleaseinfo{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}
```

- `\SFsetheadlogo` pour avoir un logo dans l'en-tête des pages. Le premier paramètre est la taille de l'image relativement à la dimension `\textwidth`, le second paramètre est le chemin vers l'image à afficher. Ex :

```
\SFsetheadlogo{.25}{common/logosysfera.pdf}
```

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

```
\SFprojectleader{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" : "Document entre SysFera et un client", // titre du document
  "author" : "John Doe", // auteur du document
  "project_name" : "Projet 999", // nom du projet
  "writtenby": { // personne ayant écrit le document avec la date correspondante
    "name": "J. Doe",
    "date": "01 avril 2012"
  },
  "verifiedby": { // personne ayant vérifié le document avec la date correspondante
    "name": "J. Bon",
    "date": "06 avril 2012"
  },
  "approvedby": { // personne ayant approuvé le document avec la date correspondante
    "name": "B. Dupont",
    "date": "08 avril 2012"
  },
  "revisions": [ // liste des révisions du document
    {
      "version": "1",
      "date": "08/03/2012",
      "description": "Première version",
      "author": "J. Doe"
    },
    {
      "version": "2",
      "date": "25/03/2012",
      "description": "Amélioration",
      "author": "J. Doe, J. Bon"
    }
  ],
  "references": [ // liste des documents de référence
    {
      "reference": "ref1",
      "document_name": "techDocument1",
      "description": "Awesome reference"
    },
    {
      "reference": "ref2",
```

```

        "document_name":"techDocument2",
        "description":"Really awesome reference"
    }
],
"authorizations":[ // liste des personnes autorisées par structure
    {
        "structure":"SysFera",
        "people":"Tous les membres"
    },
    {
        "structure":"Client",
        "people":"Tous les membres"
    }
],
"release_info":"Réalisé héhé", // information de release du document
"water_mark":"DRAFT", // water mark à utiliser sur le document
"project_leader":"Me!" // leader du projet
}

```

generator.py

Ce fichier est utilisé par le script `marksf_gen` 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 sert 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
|   |- bgsysfera.png
|   |- logosysfera.pdf
|   |- pdf-template-cplx.tex
|   |- sysferacplx.sty
|   |- dblatex
|- configuration.json
|- doc
|   |- README.md
|   |- README_pdf-cplx.md
|   |- README_pdf-cplx.pdf
|   |- README.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_pdf-cplx.md -o README_pdf-cplx.pdf
```