

Tp 06 Sécurité et site administration

AMONA Birewa Audrey

8 décembre 2023

Table des matières

1	Mise en route	1
2	Création du module controlleur et du template	1
3	Définition de vues django	3

Résumé

Ce Tp a pour objectif de générer les documents pdf à partir du moteur jinja2 et du template Latex

1 Mise en route

Nous allons installer les paquet suivant :

```
pip install latex
pip install jinja2
sudo apt install texlive-lang-franch
```

2 Création du module controlleur et du template

1. Créons le dossier **templating_ifnti** à la racine de notre projet. Créons le fichier **controlleur.py** dans ce dossier. Ensuite, on édite le fichier comme suit :

```

1  from posixpath import abspath
2  from jinja2 import Environment
3
4  from latex import build_pdf
5  from os.path import dirname, abspath
6
7
8  def generate_pdf(context):
9      """INSTACIATION D'UN NOUVEL ENVIRONNEMENT AVEC DES OPTIONS DE BALISES PERSONNALISÉES"""
10     j2_env = Environment(
11         variable_start_string="{{VAR{",
12         variable_end_string="}}",
13         block_start_string="\BLOCK{",
14         block_end_string="}",
15         comment_start_string="\COMMENT{",
16         comment_end_string="}",
17     )
18     """DECLARATION DE FICHER"""
19     #Fichier à lire contenant le template avec les balises
20     fichier_in = open("ifnti/liste_eleves.text", "r")
21
22     #Fichier en sortie accueillant les données fournies
23     fichier_out = open("out/template_out.tex", "w")
24
25     template = fichier_in.read()#Lecture du template
26     monContext = context
27     monContext["image_path"] = dirname(abspath(__file__))+"out/images"
28
29     """APPLICATION DE L'ENVIRONNEMENT ÉDITÉ SUR LE TEMPLATE"""
30     j2_template = j2_env.from_string(template)
31
32     #Écriture dans le fichier en sortie
33     fichier_out.write(j2_template.render(monContext))
34     fichier_out.close()
35
36     mon_pdf = build_pdf(open("out/template_out.tex", "r"))
37     mon_pdf.save_to("out/liste_eleves.pdf")
38     """FERMETURE DE CANAUX"""
39     fichier_in.close()

```

Ce code permet de définir des variables d'environnement qui seront utiliser pour injecter des variables ou des structures de controles dans notre fichier de template. Il permet également de définir les fichier templates et de sorties ainsi que le pdf qui sera produit.

2. Créons les dossiers nommés **ifnti** et **out**. Créons ensuite les fichiers **liste_eleves.tex** dans le dossier **ifnti**. Puis on y place le code suivant :

```

1
2 \documentclass[10pt,a4paper]{article}
3 \usepackage[utf8]{inputenc}
4 \usepackage[français]{babel}
5 \usepackage[T1]{fontenc}
6 \usepackage{amsmath}
7 \usepackage{amsfonts}
8 \usepackage{amssymb}
9 \usepackage{makeidx}
10 \usepackage{graphicx}
11 \usepackage[left=2cm, top=3cm, bottom=2cm, right=2cm]{geometry}
12 \usepackage{multirow}
13 \usepackage{lmodern}
14 \usepackage{fmtcount}
15 \usepackage{tcolorbox}
16 \usepackage{fancybox}
17 \usepackage{color}
18 \usepackage[autolanguage,np]{numprint}
19 \usepackage{longtable}
20 \usepackage{phonenumbers}
21 \usepackage{telprint}
22 \graphicspath{ {\VAR{image_path}} }
23 \begin{document}
24 \begin{minipage}{12cm}
25   \textsf{REGION CENTRALE} \\\hspace{1cm}
26 \end{minipage}
27 \begin{minipage}{5cm}
28   REPUBLIQUE TOGOLAISE\\
29   Travail-Liberté-Patrie
30 \end{minipage}
31 \begin{minipage}{3cm}
32   \begin{center}
33     \includegraphics[scale=0.8]{logoIfnti.jpg} \\\hspace{1cm}
34     IFNTI-Sokod'e TOGO\300 BP 40
35   \end{center}
36 \end{minipage}
37 \begin{center}
38   \underline{\textbf{LISTE DES ETUDIANTS}}
39 \end{center}
40 \begin{longtable}{} \hline
41   \textbf{Matricule} & \textbf{Nom} & \textbf{Prénom} & \\
42   \textbf{Sexe} & \textbf{Date Naissance} & & \\\hline
43   \BLOCK{for elv in eleves}
44   \VAR{elv.matricule} & \VAR{elv.nom} & \VAR{elv.prenom} & \\
45   \VAR{elv.sexe} & \VAR{elv.dateNais} & & \\\hline
46   \BLOCK{endfor}
47 \end{longtable}
48 \end{document}

```

Nous allons créer le dossier **images** dans notre dossier **out** et y placer le logo de l'ifnti avec le même nom que celui dans le template.

```

(app) audrey@LEFREREDUCOUZIN:~/Documents/django
/TP/tp7/my_first_django_project$ tree Templatin
g_ifnti/
Templating_ifnti/
├── controleur.py
├── ifnti
│   ├── indent.log
│   └── liste_eleves.tex
└── out
    ├── images
    │   └── logoIfnti.jpg
    └──
3 directories, 4 files

```

3 Définition de vues django

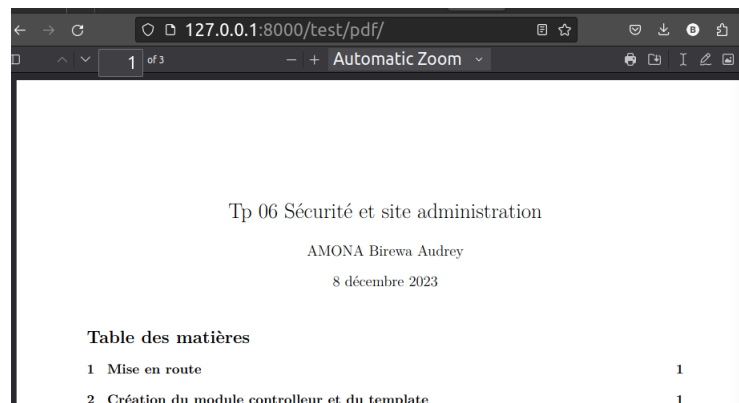
Dans cette section, nous allons définir des vues qui devront retourner des dictionnaires. Ces dictionnaires serviront d'argument à la fonction `generate_pdf()`

1. Définissons la vue **listEleves** permettant d'ouvrir un fichier pdf quelconque.

```

1 def listeEleves(request):
2     with open( dirname(abspath(__file__)) + "/pdf/mon.pdf", "rb") as my_file:
3         return HttpResponse(my_file.read(), content_type = "application/pdf")

```

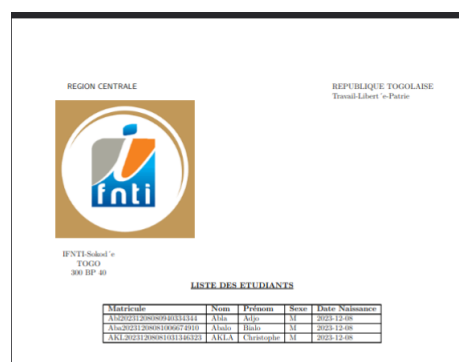


2. Nous allons modifier la vue précédente pour générer la liste des élèves

```

1 def listeEleves(request):
2     eleves = Eleve.objects.all()
3     liste = {"eleves":eleves}
4     print(liste)
5     generate_pdf(liste)

```



3. Nous allons à présent créer la vue `liste_niveauElv` permettant de générer la liste des élèves d'un niveau donné :

```

1
2 def liste_niveauElv(request, niveau_id):
3     niveau = get_object_or_404(Niveau, pk= niveau_id)
4     print(niveau)
5     eleves = niveau.eleve_set.values()
6     liste = {"eleves":list(eleves.values())}
7     generate_pdf(liste)
8     with open( dirname(dirname(abspath(__file__))) + "/Templating_ifnti/out/liste_eleves.pdf", "rb") as my_file:
9         return HttpResponse(my_file.read(), content_type = "application/pdf")

```

