

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по производственной практике
Тема: Скрипт заполнения текстовых шаблонов

Студент гр. 7304

Нгуен К.Х.

Руководитель

Санкт-Петербург

2020

ЗАДАНИЕ НА ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

Студент Нгуен К.Х.

Группа 7304

Тема практики: **Скрипт заполнения текстовых шаблонов**

Задание на практику:

Цель - инструмент, который позволит автоматически заполнять шаблоны документов (docx, odt, pptx, odp) данными из таблицы (каждая колонка - отдельное подставляемое поле, каждая строчка - новый документ).

Сроки прохождения практики: 24.06.2020 – 19.07.2020

Дата сдачи отчета:

Дата защиты отчета:

Студент

Нгуен К.Х.

Руководитель

АННОТАЦИЯ

В этой практике будет представлен инструмент, который будет автоматически заполнять шаблоны документов (docx, odt, pptx, odp ...) данными из таблицы (каждый столбец - это отдельное поле для вставки, каждая строка - новый документ)

В отчете к практике представлены отдельные части сценария и их объяснение, а также скриншоты результатов тестирования скрипта.

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	4
ПОСТАНОВКА ЗАДАЧИ	5
ХОД РАБОТЫ	7
ЗАКЛЮЧЕНИЕ.....	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	15

ПОСТАНОВКА ЗАДАЧИ

Цель - инструмент, который позволит автоматически заполнять шаблоны документов (docx, odt, pptx, odp ...) данными из таблицы (каждая колонка - отдельное подставляемое поле, каждая строчка - новый документ).

Входные данные:

- CSV-файл со значениями подстановок,
- путь к шаблону в формате docx, odt, pptx, odp,
- текстовый шаблон именования файла.

Пример CSV-файла:

```
FirstName,LastName,Address,City,State,ZIP-code

John,Doe,120 jefferson st.,Riverside, NJ, 08075

Jack,McGinnis,220 hobo Av.,Phila, PA,09119

"John ""Da Man""",Repici,120 Jefferson St.,Riverside, NJ,08075

Stephen,Tyler,"7452 Terrace ""At the Plaza"" road",SomeTown,SD, 91234

,Blankman,,SomeTown, SD, 00298

"Joan ""the bone""", Anne",Jet,"9th, at Terrace plc",Desert City,CO,00123
```

Соответственно, полями для подстановки являются: FirstName, LastName, Address, City, State, ZIP-code.

В файле docx, odt, pptx, odp содержатся указанные выше поля подстановки в виде меток следующего вида:

```
{{ Address }}
```

Текстовый шаблон именования файла содержит комбинацию полей подстановки:

```
{{ FirstName }}{{ LastName }}.docx
```

В результате работы должны появиться 6 файлов, по одному для каждой строки CSV файла.

ХОД РАБОТЫ

В проекте есть 3 файла скриптов:

`main.py` содержит код для получения ввода от пользователя, использования рендерера для рендеринга вывода, а также обработки исключений

`exceptions.py` содержит исключения, которые использованы в программе

`custom_odt_renderer.py` содержит класс пользовательского рендерера, полученный из рендерера Секретаря. Это сделано из-за того, что в оригинальном рендерере отсутствуют функции для проверки шаблона (невозможно отобразить список полей, не найденных в шаблоне).

Пояснение к функциям и их параметрам

- `main.py`:
 - `load_csv(csv_path, delimiter)`

Загрузка таблицы данных из CSV-файла и сохранение ее в глобально определенных массивах. Примечание: CSV должен быть написан в кодировке utf-8.

`csv_path`: абсолютный или относительный путь к CSV-файлу, содержащему данные для заполнения в шаблоне

`разделитель`: разделитель, используемый при создании файла CSV

- `verify_template_odt(template_path)`

Поиск пропущенных переменных / полей, используемых в файле csv и шаблоне odt, и отображение сводки. Функция `load_csv` должна быть успешно вызвана перед вызовом этой функции.

`template_path`: путь к файлу шаблона odt

- `render_output_odt(template_path, output_name_template, output_dir)`

Рендеринг выходных файлов с использованием CustomOdtRenderer

template_path: путь к файлу шаблона odt

output_name_template: шаблон для именования сгенерированных файлов;
соглашение о написании такое же, как в шаблоне; неверные символы для имен
файлов будут автоматически удалены

output_dir: путь к каталогу, где будут храниться сгенерированные файлы

- verify_template_docx(template_path)

Поиск пропущенных переменных / полей, используемых в CSV-файле и шаблоне docx и выводе на печать. Функция load_csv должна быть успешно вызвана перед вызовом этой функции.

template_path: путь к файлу шаблона docx

- render_output_docx(template_path, output_name_template, output_dir)

Рендеринг выходных файлов с использованием DocxTemplate из пакета python-docx-template (docxtpl)

template_path: путь к файлу шаблона docx

output_name_template: шаблон для именования сгенерированных файлов;
соглашение о написании такое же, как в шаблоне; неверные символы для имен
файлов будут автоматически удалены

output_dir: каталог, в котором будут храниться сгенерированные файлы

- verify(template_path)

Эта функция решает, какую из двух функций проверки использовать на основе расширения файла, или пишет сообщение, если расширение не поддерживается программой.

`template_path`: путь к файлу шаблона

Функция возвращает количество предупреждений в процессе проверки

- `process(template_path, output_name_template, output_dir)`

Функция создает выходную папку, если она не существует. После этого функция вызывает одну из двух функций рендеринга на основе расширения файла шаблона или пишет сообщение, если расширение не поддерживается программой.

`template_path`: путь к файлу шаблона docx.

`output_name_template`: шаблон для именования сгенерированных файлов; соглашение о написании такое же, как в шаблоне; неверные символы для имен файлов будут автоматически удалены.

`output_dir`: путь к каталогу, где будут храниться сгенерированные файлы.

- `main()`

Функция извлекает данные из параметров командной строки, запускает функции и печатает сообщения об ошибках.

`exceptions.py`

Классы `TemplateTypeNotSupported` и `OutputNameTemplateSyntax` - производные из класса `Exception class`

`custom_odt_renderer.py`

Класс `CustomOdtRenderer` – производный класс рендера `Secretary`.

- `patch_xml(self, src_xml)`

Функция удаляет все ненужные теги xml, чтобы иметь понятный jinja2 необработанный xml. Он заменяет `{{<некоторые теги> материал jinja2 <некоторые другие теги>}}` на `{{материал jinja2}}`. После попытки запуска рендерера по умолчанию с более сложным шаблоном Секретарь обычно не отображался из-за смешивания нескольких тегов `<text: s />` или `<text: span>` между индикаторами шаблона jinja2 `{{}}`. В связи с этим была добавлена эта функция.

self: экземпляр CustomOdtRenderer

src_xml: исходная строка xml

- `render(self, template, **kwargs)`

Переопределение средства визуализации Secretary, чтобы использована функция `patch_xml`.

self: экземпляр CustomOdtRenderer

шаблон: файл шаблона odt (двоичный)

** kwargs: поля данных из CSV-файла для заполнения шаблона

- `render_content_to_xml(self, template, **kwargs)`

Функция делает content.xml файл ODT, используя данные в формате CSV. Вернуть xml после рендеринга, эта функция используется для проверки шаблона. Эта реализация обусловлена тем фактом, что функция рендеринга Secretary будет генерировать и возвращать бинарный файл результата (zip всего xml).

self: экземпляр CustomOdtRenderer

шаблон: файл шаблона odt (двоичный)

**** kwargs:** поля данных из CSV-файла для заполнения шаблона

Создано несколько тестовых наборов для тестирования скриптов:

Каждый набор тестов содержит CSV-файл и 1 или более шаблонов (каждый из них представлен как в odt, так и в docx).

test0:

Задача предоставила пример csv + шаблон визитной карточки

test1:

Шаблон simple очень простой без стилей

Шаблон missing_param такой же, как приведенный выше, но специально написан с несоответствием в полях по сравнению с CSV-файлом, чтобы протестировать функцию предупреждения скрипта

test2:

Титульный лист лабораторной работы

test3:

Флаер с 4 полями

Создан Dockerfile для простой установки скрипта

```
FROM ubuntu:18.04 # На основе изображения Ubuntu

RUN apt-get update

RUN apt-get install -y python3 python3-pip # установка python и pip

ADD app /app # добавление папки приложения к изображению

RUN pip3 install -r app/requirements.txt # установка зависимостей
```

```
ENTRYPOINT ["python3", "/app/main.py"] # настройка точки входа
```

ЗАКЛЮЧЕНИЕ

В конце практики был создан рабочий инструмент в соответствии с заданием. Инструмент работал без ошибок на созданных тестовых наборах.

Шаблон:



CSV:

```
name,title,phone,website,email,content
Eduardo V Harzi,Biological Scientist,214-448-7939,artofprayer.com,althaa2013@hotmail.com,"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque viverra lacus vel sagitt
Evelyn P Cummings,Pipe Fitter and Steamfitter,920-979-8698,shipinthehive.com,bernadine1979@gmail.com,"Aliquam tincidunt magna aem, ut volutpat neque dignissim feugiat. Donec aliquet p
Scott D Williams,Nuclear Monitoring Technician,650-779-7412,detracking.com,finn1994@gmail.com,"Sed semper dictum mollis. Maecenas vitae eleifend mauris. Integer et sapien metus. Sed p
```

Выполнение скрипта:

```
theph@PHANTOM-PC E:\Projects\Python\industrial_practice2020\Nguyen_Q_H
> python app\main.py tests\test4\info.csv tests\test4\cover_letter.docx -o "{{ name }}" -D .\outpu
t
Verifying template...
    OK
Creating output folder...
    OK
Rendering files...
    OK
```

Результат:

	<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque viverra lacus vel sagittis lectus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Mauris quis imperdiet sem, a convallis nisi. Nunc vel ipsum et neque, porttitor varius magna. Nunc porttitor convallis et condimentum. Etiam accumsan urna et orci ornare facilis scelerisque nec arcu. Sed a ultrices leo. Vestibulum consectetur interdum erat non porttitor. Ut pretium nibh neque. ut vulputate tortor commodo ut. Aliquam dapibus molestie lectus. Ut placerat, tellus in varius suscipit, massa dui malesuada dui. In accumsan neque neque ut ante. Integer porttitor ac purus ac tempor. Ut ut, est hendrerit, sollicitudin tortor ut, tempus est. </p>
<h2>EDUARDO V MARTI</h2> <p>Biological Scientist</p>	
<p>CONTACT</p> <p>PHONE: 214-448-7939</p> <p>WEBSITE: artofprayer.com</p> <p>EMAIL: althea2013@hotmail.com</p>	

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документ для библиотеки python-docx-template. URL: <https://github.com/elarouya/python-docx-template>.
2. Документ для библиотеки Secretary. URL: <https://github.com/christopher-ramirez/secretary>.