ПОЛЬЗОВАТЕЛЬСКАЯ ДОКУМЕНТАЦИЯ ПО ГЕНЕРАТОРУ ПРИКАЗОВ

Репозиторий GitHub: https://github.com/cadnev/decree_gen

Оглавление

1. СТРУКТУРА ПРОЕКТА

Проект содержит в себе:

- `raw/img` – директория, в которой содержатся изображения, используемые в `README.md`;
- `samples` – директория, в которой содержатся размеченные данные для генерации приказов;
- `.gitignore`;
- `README.md`;
- `auxil.py` – модуль, содержащий вспомогательные функции;
- `change_case.py` – модуль, позволяющий склонять инструкции и ответственных по падежам;
- `consts.py` – модуль, содержащий основные константные параметры, используемые при генерации;
- `gen.py` – основной файл проекта;
- `requirements.txt`;
- `russian_datetime.py` – надстройка над классом `date` встроенного модуля `datetime`;
- `write.py` – модуль, содержащий функции для создания файлов с приказами.

2. УСТАНОВКА И ИСПОЛЬЗОВАНИЕ

2.1 Установка

1. Клонировать репозиторий;
2. Установить библиотеки из `requirementst.txt`;
3. Установить abiword для конвертации приказов в pdf.

2.2 Использование

```
gen.py [-h] [-i] [-f format] [-s path] [-o path] [-v] size
```

- `-h/--help`: вывод справки;

- `-i/--image`: использование картинок в документах (логотип, подпись, печать). `action = store_true`;
- `-f/--format`: определяет форматы файлов (d – docx, p – pdf, j – jpg). По умолчанию имеет значение `dp`. Использование `j` возможно только в комбинации с `dp`, т. к. скрипт сначала генерирует docx, потом конвертирует его в pdf, высчитывает координаты картинок в документе из pdf, потом конвертирует pdf в jpg;
- `-s/--samples`: пусть к папке с образцами. По умолчанию `./samples/`;
- `-o/--out`: пусть к папке с выходными файлами. По умолчанию `./decrees`. Если указанной папки не существует, она автоматически создается;
- `-v/--verbose`: задает уровень логгера. `action = count`. Таким образом, по умолчанию уровень логгера warning, при `-v` – info, при `-vv` – debug.
- `size`: размер сгенерированных файлов, при которой скрипт останавливает работу. Комбинация числа и единицы измерения: 5KB, 10MB, 20GB.

2.3 Примеры

```
python3 gen.py 50MB -f dpj -s samples -o decrees -vv
python3 gen.py 5MB
python3 gen.py 1GB -i
```

3. ОПИСАНИЕ ФАЙЛОВ

3.1 `auxil.py`

Модуль с дополнительными функциями, используемыми при работе скрипта.

- `logger_config(v: int) -> None`

Конфигурирует логгер.

- `generate_date(standart_format=False, unixtime=False) -> str`

Генерирует случайную дату.

`standard_format` задает формат даты.

При `unixtime=True` возвращает массив, где 1-ый элемент – `str`, второй – `float`.

- `check_size_format(size: str, pat=re.compile(r"^\d*[KMG]B$")) -> str`

Проверяет формат параметра (аргумента) скрипта. Например, 1KB, 5MB, 10GB. При несоответствии вызывает `ArgumentTypeError`.

- `size_to_bytes(size: str) -> int`

Переводит 1KB в 1024.

- `getsize(out: str) -> int`

Возвращает размер файлов в директории out в байтах.

- `to_roman(n: int) -> str`

Переводит арабское число в римское.

- `add_numbering(instruction: list) -> list`

Нумерует каждую инструкцию, рандомно определяет уровень вложенности инструкции.

- `check_abiword() -> int`

Проверяет наличие abiword в системе. Возвращает 0 при успешном выполнении или вызывает исключение.

- `check_os() -> str`

Возвращает название используемой ОС.

- `parse_formats(fmts: str) -> str`

Проверяет расширение файлов в аргументе (параметре) скрипта на соответствие заданному формату.

- `mm_to_px(mm: int, dpi=300) -> int`

Переводит из миллиметров в пиксели.

- `PDFunits_to_px(units: int, dpi=300) -> int`

Переводит из pdf units в пиксели.

3.2 `change_case.py`

Модуль для изменения падежей в тексте ответственных.

3.3 `consts.py`

Модуль с основными константными параметрами, использующимися при генерации приказов.

Массив `formats` содержит форматы дат, аналогичные форматам в библиотеке `datetime`.

3.4 `gen.py`

Основной файл проекта, запускается для генерации приказов.

- `load_samples(samples_dir: str) -> tuple`

Загружает образцы из указанной директории. Возвращает кортеж с данными.

- `generate(data: tuple, out: str, formats: str, size: int, samples_dir: str, is_image: bool) ->None`

Запускает процесс генерации приказов, вызывая функции из других модулей.

- `get_args() -> argparse.Namespace`

Парсит флаги скрипта.

- `main() -> None`

Главная функция скрипта.

3.5 `russian_datetime.py`

Модуль, переопределяющий метод класса `date` стандартной библиотеки `datetime`. Нужен, чтобы метод `strftime` переводил месяц из числа в строку на русском языке.

3.6 `write.py`

Модуль, содержащий функции записи приказов в файл.

- `extend_instruction(instruction: list, samples_dir: str) -> list`

Добавляет ответственных и дедлайн в задачу с шансом 25%, если они отсутствуют.

- `write_docx(header: str, name: str, intro: str, instruction: list, responsible: str, creator: str, date: str, out: str, count: int, logo: str, sign: str, seal: str) -> str`

Создает docx документ с приказом. Через аргументы функция принимает полный текст с приказом (шапка, название, введение…), директорию с выходными файлами, порядковый номер приказа, пути к изображениям (логотип, подпись, печать). Возвращает путь к сгенерированному документу.

- `write_json(instruction: list, responsible_arr: list, date: list, out: str, count: int) -> str`

Создает разметку для приказа. Возвращает путь к сгенерированному файлу.

- `write_pdf_linux(docx_path: str, out: str, count: int) -> str`

Конвертирует docx в pdf. Возвращает путь к сгенерированному документу.

- `write_jpg(out: str, count: int) -> None`

Конвертирует pdf в jpg.

- `write_coords(json_path: str, pdf_path: str) -> None`

Добавляет координаты изображений в pdf файле в json разметку.

4. ФОРМАТ ФАЙЛОВ С ОБРАЗЦАМИ

4.1 `execution_control.txt`

Контроль над исполнением распоряжения оставить за {ablt} – на место фигурных скобок будет подставляться ответственный в заданном падеже (ablt – творительный, accs – винительный).

4.2 `responsible.json`

```
[
    "{Министр} {генерал} полиции Российской Федерации В.{{КОЛОКОЛЬЦЕВ}}",
```

```
    "КОЛОКОЛЬЦЕВ",
    "В.",
    "",
    "Министр генерал полиции Российской Федерации"
]
```

Первый элемент массива – строка с ответственным, которая пойдет в приказ. В одинарных фигурных скобках – слово из должности (профессии), у которого нужно изменить падеж. В двойных фигурных скобках – слово из имени, у которого нужно изменить падеж. Это сделано так, потому что профессии и имена склоняют две разные библиотеки.

Следующие элементы заносятся в разметку:

Второй элемент массива – фамилия.

Третий элемент массива – имя.

Четвертый элемент массива – отчество.

Пятый элемент массива – должность.