Руководство разработчика к приложению для работы с данными ЦБ по форме 102

Разработчики:

Сафрыгин Владислав

Козлов Даниил

Автор: Козлов Даниил **Редактор: Сафрыгин Владислав**

Технические требования

64-битная операционная система Windows, на которую возможна установка интерпретатора Python 3.7 (и версий выше 3.7) (https://www.python.org/downloads/)

Версии Библиотек

Данное приложение использует небольшой набор популярных библиотек питона. Ниже приведена таблица которая описывает использованную версию каждой такой библиотеки (Табл. 1).

Библиотека	Версия
mysqlclient	1.3.13
pandas	1.0.4
tokenize	4.0

re	3.0.2
requests	2.24.0
django	3.0.6
matplotlib	3.2.2
Bs4	4.9.0
csv	13.1
io	3.7
dbfread	2.0.7

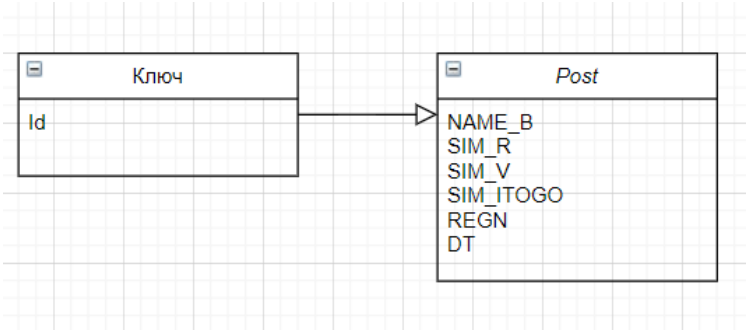
Табл. 1. Версии библиотек.

Работа с СУБД

№	REGN	NAME_B	SIM_R	SIM_V	SIM_ITOGO	DT
1	3185	Эс-Би-Ай Банк ООО	5656060	450345	35435435	01.07.2019
…	…	…	…	…	…	…

№ - это ключ, с помощью которого идёт обращение к элементу

Схема СУБД



Описание типов каждой переменной:

```
class Post(models.Model):
    """
    Автор: Козлов Даниил
    Цель: Создание внутренней БД на основе класса Post
    """
    NAME_B = models.CharField(max_length=200, default='1')
    SIM_R = models.BigIntegerField(default=1)
    SIM_V = models.BigIntegerField(default=1)
    SIM_ITOGO = models.BigIntegerField(default=1)
    REGN = models.CharField(max_length=200, default='1')
    DT = models.DateField(default=2019 / 10 / 12)
```

Архитектура Приложения

Главными директориями приложения являются work/bank-master и work/bank-master/Bank, которые содержат управляющие программы приложения. Все основные функции и методы, используемые в приложении находятся в модуле views.py. В этом же модуле находятся подключённые к проекту библиотеки. В директории templates находятся все модули и файлы, отвечающие за фронт-энд составляющую проекта (html файлы и т.д.).

Модуль manage.py находится в директории bank-master и позволяет запустить веб-приложение с помощью вызова в командной строке:

“python manage.py runserver”

Модуль	Местонахождение	Функция
manage.py	Work/bank-master	Запускает приложение django
admin.py	Work/ bank-master /Bank	Содержит информацию и доступ к администрированию сайта
apps.py	Work/ bank-master /Bank	

models.py	Work/ bank-master / Bank	Содержит информацию о базе данных проекта
tests.py	Work/ bank-master / Bank	
urls.py	Work/ bank-master /Bank	Регистрирует функции из views.py
views.py	Work/ bank-master /Bank	Строит логику приложения
wsgi.py	Work/ bank-master / Bank	
settings.py	Work/ bank-master /Bank	
asgi.py	Work/ bank-master /Bank	

Табл. 2. Модули приложения.

Структура Каталогов

Данное приложение использует следующую систему каталогов (Табл. 2).

Первый уровень	Второй уровень	Объяснение
Untitled2		Основной каталог
	Bank	Содержит базу данных, основные скрипты, функции

	Templates	Содержит копии графических отчетов
	Notes	Содержит документацию
	Untitled2	Содержит настройки приложения

Табл. 2. Каталоги приложения

Листинг Скрипта

Ниже приведён список функций и docstrings каждого модуля.

Модуль	Функции с Докстрингами

Views.py	index """ Автор: Козлов Даниил Цель: подключить основную html страницу :param request: :return: рендер шаблона 'index.html' """ new_report """ Автор: Козлов Даниил Цель: проверить наличие новой отчетности :param request: :return: в зависимости от наличия новой отчености возвращает рендер шаблона, а также показатель ее наличия """ parser """ Автор:Козлов Даниил Цель: Обновление БД :param request: :return: обновленную базу данных в фромате DataFrame """ choises """ Автор: Сафрыгин Владислав Цель: передать значение выбора пользователя по виду отчета :param request: :return: рендер шаблона 'index.html', словарь с выбранным типом отчета и списком банков """ input_bank """ Автор: Сафрыгин Владислав Цель: Получения названия банка, проверка наличия такого в БД, загрузка отчета по банку в случае наличия :param request: :return: рендер нужного шаблона в зависимости от наличия банка в БД, название введенного пользователем банка """ input_date """ Автор: Сафрыгин Владислав Цель: загрузка отчета по всем банкам за определенную дату :param request: :return: рендер нужного шаблона в зависимости от резульата, выбранная пользователем дата """ graphic """ Автор: Сафрыгин Владислав Цель: Построить график на основе показателя выбранного пользователем :param request:

	`` :return: рендер шаблона 'graphic,html' """ profile_upload """ Автор: Козлов Даниил Цель: Загрузка БД :param request: :return: рендер шаблона template """ my_image """ Автор: Сафрыгин Владислав Цель: Вывод графика пользователю :param request: :return: HttpResponce """ ```
Models.py	``` class Post """ Автор: Козлов Даниил Цель: Создание внутренней БД на основе класса Post """ ```

Admin.py	""" Автор: Козлов Даниил Цель: регистрация классов в админ-панели """ class PostAdmin class form
Urls.py	""" Автор: Сафрыгин Владислав Цель: регистрация всех функций, запускающихся при вводе пользователя """ urlpatterns = [path('admin/', admin.site.urls), path('', include('bank.urls')), path('upload-csv/', profile_upload, name="profile_upload"), url(r'^choises/', views.choises, name='choises'), url(r'^input_bank/', views.input_bank, name='input_bank'), url(r'^input_date/', views.input_date, name='input_date'), url(r'^graphic/', views.graphic, name='graphic'), url(r'^bank-autocomplete/$', BankAutocomplete.as_view(), name='bank-autocomplete'), url(r'^new_report/', views.new_report, name='new_report'), url(r'^my_image/', views.my_image, name='my_image'), url(r'^parser/', views.parser, name='parser'),]
Templates (содержит все html файлы)	""" Автор: Сафрыгин Владислав Цель: Построение логики приложения за счёт работы с пользователем """