Сопроводительная документация к проекту

«Умный цифровой помощник Главы региона»

Созданный: команда Вектор

Для определения типа вопроса и его смысла мы реализовали ф-цию **get\_response (приложение 1)**, которая: разбивает введённую строку на слова, проверяет данный массив слов на схожесть с массивами ключ.слов **required\_words(файл bot.json – приложение 2)**. По итогу, словарь, в котором процент совпадения **key word** и привязанного к нему **required\_words – наибольший,** то бот возвращает **bot\_response данного словаря**. Если же таких сходств нет, то возвращает случайный ответ из **random\_response.json (приложение 3)**.

Само заполнение **bot.json** осуществлялось с помощью файла **inputDataInBot.py(приложение 4)**, где берется строка вопроса, желаемого ответа и ключ слов для него, после разбивается по словно и с помощью шаблона **q** перезаписывается в **bot.json**.

За основу работы бота (файл **bot.py** – приложение 5) была взята библиотека **vk\_api** (**python**).

В функции main осуществляться основная проверка событий, который приходят с **longpoll** сервера бота, с помощью цикла **for** и прослушки сервера (**bot\_longpoll.listen()**). В самом блоке цикла имеются проверки на первичный старт бота от лица пользователя и дальнейшее описание взаимодействия с ним.

Функция **question** отвечает за обработку вопросов к администрации/главы региона со стороны пользователя.

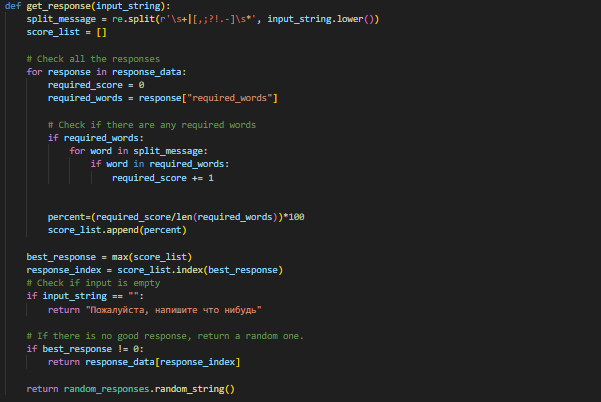
Основной функц-л бота описан в файле **botlib.py**, где находятся основные ф-ции и объекты для его дальнейшего масштабирования. Также в боте предусмотрено ведение логов в файл **BotLog.txt (приложение 6).**

Чтобы прикреплять документацию, в зависимости от смысла вопроса – было реализован парсер, который записал все ссылки на указы в файл **ref.json** (2500 записей). В самом **botlib.py** описана функция **attachUrl**, которая принимает ссылку на нужный указ и может быть приложена к сообщению бота (в виде доп строки). Также, в **main.py** имеется функция **writeRefInFile**, с помощью которой можно обновить ссылки на указы (если они обновились или появились новые)

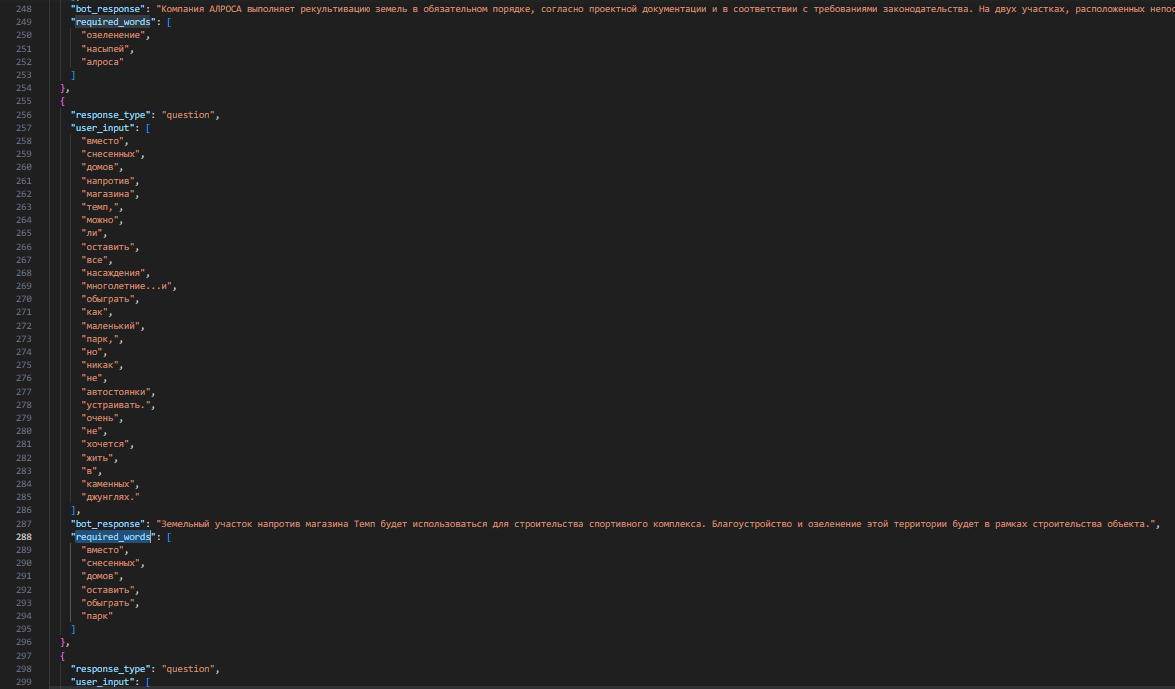
**Web api** организован на базе трех основных окон, с возможностью расширения под будующие задачи:

* Dashboard Окно для сбора информации о работе бота, количестве пользователей и просмотра последних поступивших запросов. (приложение )
* Newsletter Форма для отправки сообщений, с возможность прикреплять файлы, всем пользователям, например, произошло обновление документов и нужно сообщить всем об этом, это окно поможет это сделать.
* Settings Окно для редактирования (создания / удаления / изменения / прикрепления файлов) запросов и ответов на них, также с редактированием ключевых слов по которым, определяется наиболее подходящий ответ для пользователя. В разработке Web api использовалась базовая верстка на html с использованием таблицы каскадных стилей. Обработка действий на странице реализована на JavaScript. Данный стек технологий позволяет легко и быстро модифицировать продукт под новые нужды.

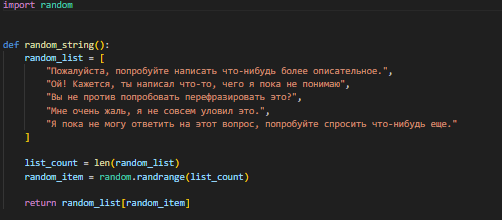
Приложение 1



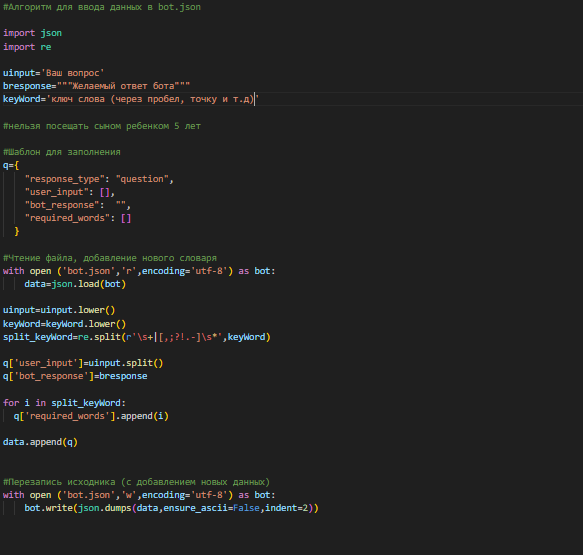
Приложение 2



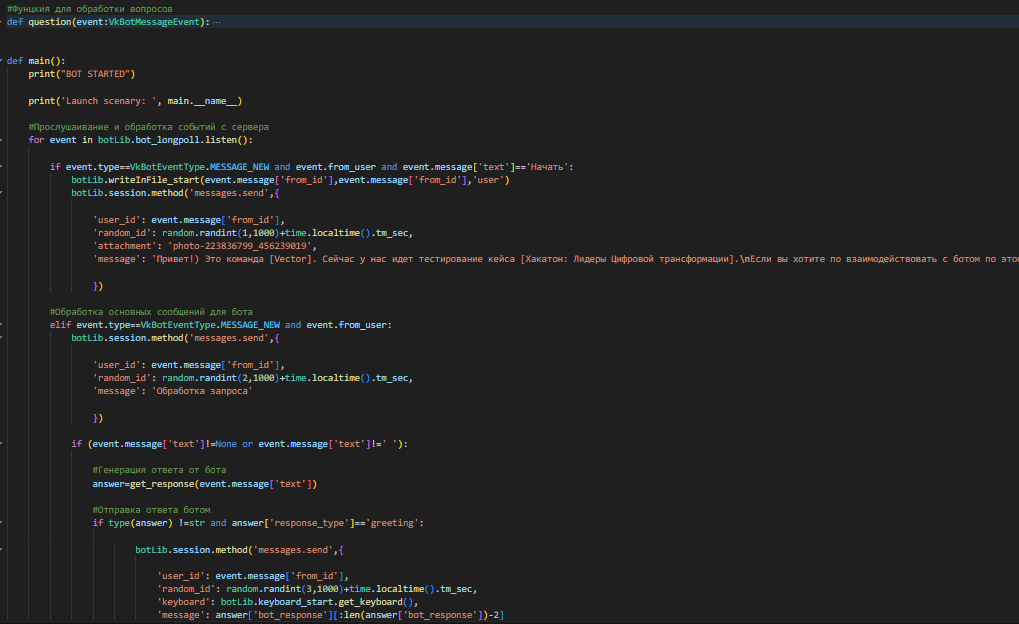
Приложение 3



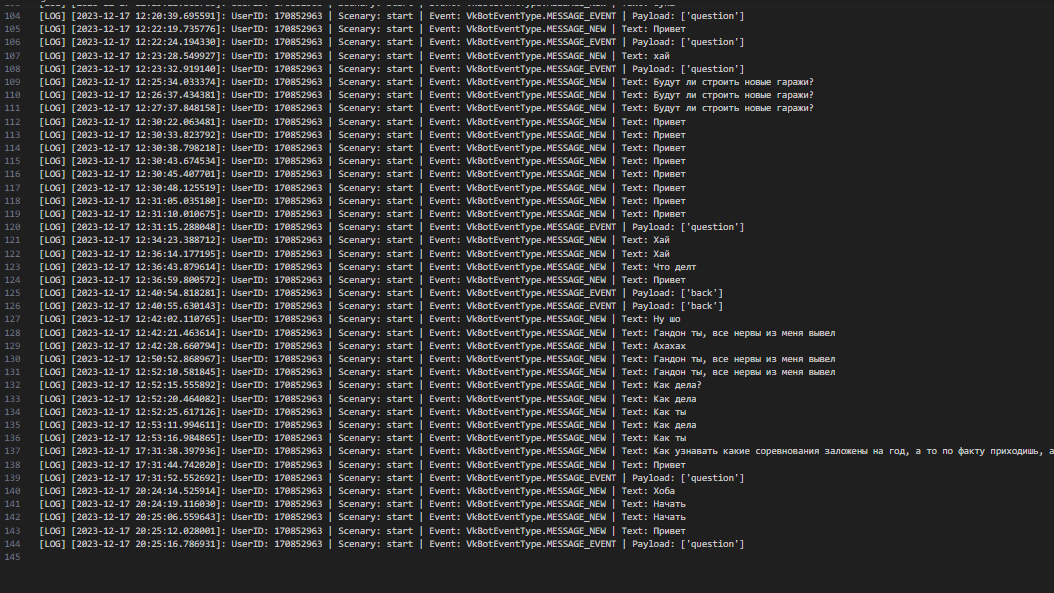
Приложение 4



Приложение 5



Приложение 6



Приложение 7

