

# Описание конечного результата.

## Описание задачи:

Вк бот, работающий от лица сообщества, который пересылает помеченные сообщения из одной беседы(source) в ряд других бесед (targets). Управлять ботом нужно с использованием команд. Бот поддерживает роль godlike, открывающую доступ к особнным командам.

## Описание команд:

Формат команды	Где прописывать	Описание
«/ <b>reg_target</b> [число_идентификатор] [«title»]»	В беседе, кто угодно	Беседа добавляется в спиок targets. Беседа ассоциируется с число_идентификатор. Число_идентификатор должно быть уникальным для каждой беседы. Если этот параметр не передан, то он генерируется программой. Title будет отображаться при пересылке сообщений из этой беседы(на будущее).
«/ <b>reg_source</b> [«title»]»	В беседе, кто угодно	Беседа становится source. Title будет отображаться при пересылке сообщений из этой беседы.
«/ <b>del_target</b> »	В target беседе, кто угодно	Удаляет зарегистрированную беседу из списка targets.
«/ <b>del_source</b> »	В source беседе, кто угодно	Удаляет зарегистрированную беседу из списка source.
«/ <b>reg_godlike</b> айди»	Где угодно, права godlike	Добавляет айди в список godlike.
«/ <b>delete_target_by_num</b> число_идентификатор»	Где угодно, права godlike	Удаляет target беседу по идентификатору.
«/ <b>change_mode</b> »	Где угодно, права godlike	Переключает режим работы между config и mode.
«/ <b>status</b> »	Где угодно, кто угодно	Бот отправляет строку подтверждающую жизнеспособность.
«/ <b>help</b> »	Где угодно, кто угодно	Бот отправляет список доступных команд(на будущее).

## Пометки сообщений:

- «\_**всем**\_ [«title»]» - сообщение содержащее такую строчку будет отправлено всем targets. Title будет помещен в тексте сообщения бота.
- «\_**всем\_важно**\_ [«title»]» - сообщение содержащее такую строчку будет переслано всем targets с пометкой @all. Title будет помещен в тексте сообщения бота.

# Детальные требования.

## Описание работы программы:

При запуске бот считывает файл **config.json** и начинает слушание порта, указанного в файле. Если файл не удалось открыть или порт недоступен — Fatal Error. Бот находит режим работы в поле **«mode»**. Если поля нет, или оно не содержит **«work»** или **«config»** — Fatal Error. На заданный порт от будут приходить json файлы с event-ами. Подробнее в разделе приложение. Дальше в зависимости от режима.

### Режим «config»:

В этом режиме происходит настройка бота с помощью всех команд. Пересылка сообщений не работает.

- **«reg\_target** [число\_идентификатор] [«title»]» - надо чтобы и число\_идентификатор и id беседы были свободны в таблице(см. конфиг).  
Если id беседы является source, то сообщение об ошибке в данную беседу.  
Если id беседы используется с другим числом идентификатом, то сообщение об ошибке в данную беседу.  
Если число\_идентификатор уже занято, то сообщение об ошибке в данную беседу.
- **«reg\_source** [«title»]» - надо чтобы source был не занят и id не лежал в targets.  
Если source уже существует, то сообщение об ошибке в данную беседу.  
Если id находится в targets, то сообщение об ошибке в данную беседу.
- **«del\_target»** - прописывается в target беседе.  
Если беседа не является target, то сообщение об ошибке.
- **«del\_source»** - прописывается в source беседе.  
Если беседа не является source, то сообщение об ошибке
- **«set\_title»** - на будущее.
- **«del\_title»** - на будущее.

### Все режимы:

- **«status»** - Прописывается кем угодно, где угодно, где есть бот. Если всё хорошо, бот присылает в ответ что всё хорошо.
- **«change\_mode»** - команда для Godlikes. Этой командой пользователь может переключать бота между режимами.
- **«reg\_godlike айди\_пользователя»** - команда для Godlikes. Данный айди добавляется в список godlike.
- **«del\_target\_by\_num число\_идентификатор»** - эксклюзивная команда в личку бота. Доступна только Godlike пользователям(см. конфиг).

Удаляет беседу связанную с число\_идентификатор из targets.

### Режим «work»:

Обязательно наличие source беседы, а также хотя бы одной target беседы. Иначе Fatal error.

В этом режиме бот просто пересылает сообщения с тегами из source беседы в targets.

## Приложение:

### VK confirmation:

Для подключения Callback API сервер должен вернуть заданную строку.

Адрес:

Для получения уведомлений нужно подтвердить адрес сервера. На него будет отправлен POST-запрос, содержащий JSON:  
{ "type": "confirmation", "group\_id": 220604034 }

Строка, которую должен вернуть сервер: 3сac4с26

Подтвердить

Секретный ключ:

Сохранить

## VK event json:

В таком формате vk будет присылать уведомления на сервер.

```
JSON
{
  "type": <тип события>,
  "object": <объект, инициировавший событие>,
  "group_id": <ID сообщества, в котором произошло событие>
}
```

Например:

```
JSON
{
  "type": "group_join",
  "object": {
    "user_id": 1,
    "join_type": "approved"
  },
  "group_id": 1
}
```

Нас интересует событие «message\_new». «client\_info» не содержит ничего важного для нашего бота, так что будет игнорироваться.

Тип события:

- `message_new` — входящее сообщение.

Формат поля `object`:

- Для версий API 5.103 и выше: объект, содержащий следующие поля:
- `message` — личное сообщение;
- `client_info` — информация о функциях, доступных пользователю.

Личное сообщение содержит важные для нас поля

`int id` — уникальный идентификатор сообщения.  
`int peer_id` — уникальный идентификатор назначения(кому отправили).  
`int from_id` — уникальный идентификатор отправителя.  
`string text` — текст сообщения.

## Работа с VK API:

URL — <https://api.vk.com/method/<METHOD>?<PARAMS>>

METHOD может быть например `status.get`, `message.send` и прочие (см. <https://dev.vk.com/api/api-requests>).

PARAMS обязательно должен содержать минимум два параметра — «**access\_token**» и «**v**».

«**v**» - указывает с какой версией API мы работаем. На данный момент актуальная «5.131».

«**access\_token**» см далее.

## Access\_token:

Вот что написано на сайте вк. «Для работы с методами API вам необходимо передавать в запросах **специальный ключ доступа** — `access_token`. Он представляет собой строку из латинских букв и цифр и может соответствовать отдельному пользователю, сообществу или вашему приложению». **access\_token** можно будет получить из настроек нашего сообщества(см. **конфиг**). Будет действовать до тех пор, пока не удалён вручную.

## Коды ошибок VK:

При попытке выполнить запрос сайт может вернуть ошибку.

<https://dev.vk.com/reference/errors> — здесь можно посмотреть все коды. Рассмотрим основные из них.

- Код ошибки 5. Неверный токен. Может возникнуть при работе с API. Нужно попытаться снова через **1, 2, 5 секунд** — если не получилось то прервать выполнение операции. Такое бывает, но редко, что вк не узнает токены. Как правило быстро проходит (~5 минут).
- Код ошибки 15. Доступ запрещен. Нужно попытаться снова через **1, 2, 5 секунд** — если не получилось то прервать выполнение операции.
- Код ошибки 1 или 10. Неизвестная ошибка. Может возникнуть при работе с API. Нужно попытаться снова через **1, 2, 5 секунд** — если не получилось то прервать выполнение операции (одноразовая ошибка).
- Код ошибки 6. Слишком много запросов в секунду. Может возникнуть при работе с API. Нужно выдержать паузу в 2 секунды и повторить снова. И вообще таких случаев лучше не допускать. Будут тормозить программу. Использовать метод **messages.send peer\_ids** чтобы избежать проблемы.
- При остальных кодах нужно попытаться снова через 1, 2, 5 секунд — если не получилось то прервать выполнение операции.

### Логирование:

Бот должен складывать в лог файл и в консоль информацию для дебага о каждом этапе выполнения обработки event-ов с помощью Boost.Log. Максимальный размер лог файла 10 мегабайт.

### Структура config.json:

- «mode» - «work» or «config». **(обязательное)**
- «v» - версия VK API с которой работаем. **(обязательное)**
- «token» - токен сообщества бота. **(обязательное)**
- «secret\_string» - строка для подтверждения связи с вк. **(обязательное)**
- «port» - порт который будет слушать сервер. **(обязательное)**
- «group\_id» - ай ди группы, от лица которой работаем. Нужно чтобы отбирать сообщения от самого себя. **(обязательное)**
- «base\_url» - по умолчанию <https://api.vk.com/method>. Поле нужно для работы с мок объектом. (необязательное)
- «target\_chats» - массив с объектами {num идентификатор, peer\_id беседы, title беседы(необязательное)}. **(обязательное для work)**
- «source\_chat» - peer\_id чата. **(обязательное для work)**
- «godlike\_ids» - id пользователей с эксклюзивными правами. (необязательное)

### Используемые библиотеки:

- "jsoncpp" чтобы работать с json.
- "Boost.Log" ну тут понятно да.
- "libhttpserver" простая и вроде как приятная библиотека для поднятия сервера.
- "cpr" для легкого использования http запросов.