Материалы к занятию

Voice - Этот объект представляет голосовое сообщение.

file_id - Уникальный идентификатор файла
duration - Продолжительность аудиофайла, заданная отправителем
mime_type - MIME-тип файла, заданный отправителем
file_size - Размер файла

sendVoice - этот метод для отправки аудиофайлов, если вы хотите, чтобы клиенты Telegram отображали файл в виде воспроизводимого голосового сообщения. Чтобы это сработало, ваш звук должен быть в файле .ogg, закодированном с помощью OPUS (другие форматы могут быть отправлены в виде аудио или документа).

chat_id - Уникальный идентификатор целевого чата или имя пользователя целевого канала
voice - Аудиофайл для отправки. Вы можете либо передать file_id в виде строки для повторной отправки аудио, которое уже есть на серверах Telegram, либо загрузить новый аудиофайл, используя multipart/form-data.
caption - Название аудиосообщения, 0-200 символов
duration - Продолжительность отправленного аудио в секундах
disable_notification - Отправляет сообщение беззвучно. Пользователи iOS не получат уведомление, пользователи Android получат уведомление без звука.
reply_to_message_id - Если сообщение является ответом, идентификатор исходного сообщения
reply_markup - Дополнительные опции интерфейса. Объект, сериализованный в формате JSON для встроенной клавиатуры, пользовательской клавиатуры ответа, инструкций по скрытию клавиатуры ответа или принудительному получению ответа от пользователя.

```
import requests

BASE_URL = 'https://api.telegram.org/bot'

TOKEN = 'Ваш токен'
ADMINS = [ваш id, ]
def pulling():
   count_message = 0
   while True:
       response = requests.get(f'{BASE_URL}{TOKEN}/getUpdates').json()
       if count_message != len(response['result']):
           count_message = len(response['result'])
           message = response['result'][-1]
           file_id = message['message']['voice']['file_id']
           user_id = message['message']['from']['id']
           if user_id in ADMINS:
```

```
requests.get(f'{BASE_URL}{TOKEN}/sendVoice?chat_id={user_id}&voice={file_id
}')

pulling()
```

Contact - Этот объект представляет контакт с номером телефона.

phone_number - Номер телефона
first_name - Имя
last_name - Фамилия
user_id - Идентификатор пользователя в Telegram

sendContant - этот метод для отправки телефонных контактов.

chat_id - Уникальный идентификатор целевого чата или имя пользователя целевого канала
phone_number - Номер телефона контактного лица
first_name - Имя контактного лица
last_name - Фамилия контактного лица
disable_notification - Отправляет сообщение беззвучно. Пользователи iOS не получат уведомление, пользователи Android получат уведомление без звука.
reply_to_message_id - Если сообщение является ответом, идентификатор исходного сообщения
reply_markup - Дополнительные опции интерфейса. Объект, сериализованный в формате JSON для встроенной клавиатуры, пользовательской клавиатуры ответа, инструкций по скрытию клавиатуры или принудительному получению ответа от пользователя.

```
import requests

BASE_URL = 'https://api.telegram.org/bot'

TOKEN = 'Ваш токен'
ADMINS = [Ваш id, ]
def pulling():
   count_message = 0
   while True:
       response = requests.get(f'{BASE_URL}{TOKEN}/getUpdates').json()
       if count_message != len(response['result']):
           count_message = len(response['result'])
           message = response['result'][-1]
           user_id = message['message']['from']['id']
           if user_id in ADMINS:

requests.get(f'{BASE_URL}{TOKEN}/sendContact?chat_id={user_id}&phone_number
=89911904125&first_name=Ivan')

pulling()
```

Location - Этот объект представляет точку на карте.

longitude - Долгота, заданная отправителем
latitude - Широта, заданная отправителем

sendLocation - этот метод для отправки точки на карте.

chat_id - Уникальный идентификатор целевого чата или имя пользователя целевого канала
latitude - Широта, заданная отправителем
longitude - Долгота, заданная отправителем
disable_notification - Отправляет сообщение беззвучно. Пользователи iOS не получат уведомление, пользователи Android получат уведомление без звука.
reply_to_message_id - Если сообщение является ответом, идентификатор исходного сообщения
reply_markup - Дополнительные опции интерфейса. Объект, сериализованный в формате JSON для встроенной клавиатуры, пользовательской клавиатуры ответа, инструкций по скрытию клавиатуры ответа или принудительному получению ответа от пользователя.

```
import requests

BASE_URL = 'https://api.telegram.org/bot'

TOKEN = 'Ваш Токен'
ADMINS = [Ваш id, ]
def pulling():
   count_message = 0
   while True:
       response = requests.get(f'{BASE_URL}{TOKEN}/getUpdates').json()
       if count_message != len(response['result']):
           count_message = len(response['result'])
           message = response['result'][-1]
           user_id = message['message']['from']['id']
           if user_id in ADMINS:

requests.get(f'{BASE_URL}{TOKEN}/sendLocation?chat_id={user_id}&latitude=55
.53932&longitude=37.39892')

pulling()
```

Venue - Этот объект представляет объект на карте.

location - Координаты объекта
title - Название объекта

address - Адрес объекта
foursquare_id - Идентификатор объекта в Foursquare

sendVenue - Используйте этот метод для отправки информации о месте проведения.

chat_id - Уникальный идентификатор целевого чата или имя пользователя целевого канала
latitude - Широта места проведения
longitude - Долгота места проведения
title - Название места проведения
address - Адрес места проведения
foursquare_id - Foursquare идентификатор места проведения
disable_notification - Отправляет сообщение беззвучно. Пользователи iOS не получат уведомление, пользователи Android получат уведомление без звука.
reply_to_message_id - Если сообщение является ответом, идентификатор исходного сообщения
reply_markup - Дополнительные опции интерфейса. Объект, сериализованный в формате JSON для встроенной клавиатуры, пользовательской клавиатуры ответа, инструкций по скрытию клавиатуры ответа или принудительному получению ответа от пользователя.

```
import requests

BASE_URL = 'https://api.telegram.org/bot'

TOKEN = 'Ваш токен'
ADMINS = [Ваш id, ]
def pulling():
   count_message = 0
   while True:
       response = requests.get(f'{BASE_URL}{TOKEN}/getUpdates').json()
       if count_message != len(response['result']):
           count_message = len(response['result'])
           message = response['result'][-1]
           user_id = message['message']['from']['id']
           if user_id in ADMINS:

requests.get(f'{BASE_URL}{TOKEN}/sendVenue?chat_id={user_id}&latitude=55.53
932&longitude=37.39892&title=Moscow')

pulling()
```

UserProfilePhotos - Этот объект содержит фотографии профиля пользователя

total_count - Общее число доступных фотографий профиля
photos - Запрошенные изображения, каждое в 4 разных размерах.

getUserProfilePhotos - Используйте этот метод, чтобы получить список фотографий профиля пользователя. Возвращает объект UserProfilePhotos.

user_id - Уникальный идентификатор целевого пользователя
offset - Порядковый номер первой фотографии, подлежащей возврату. По умолчанию возвращаются все фотографии.
limit - Ограничивает количество извлекаемых фотографий. Принимаются значения в диапазоне от 1 до 100. Значение по умолчанию равно 100.

```
import requests

BASE_URL = 'https://api.telegram.org/bot'

TOKEN = 'Ваш токен'
ADMINS = [Ваш id, ]
def pulling():
   count_message = 0
   while True:
       response = requests.get(f'{BASE_URL}{TOKEN}/getUpdates').json()
       if count_message != len(response['result']):
           count_message = len(response['result'])
           message = response['result'][-1]
           user_id = message['message']['from']['id']
           if user_id in ADMINS:
               r =
requests.get(f'{BASE_URL}{TOKEN}/getUSerProfilePhotos?user_id={user_id}').j
son()
               print(r['result']['photos'])

pulling()
```

File - Этот объект представляет файл, готовый к загрузке. Он может быть скачан по ссылке вида https://api.telegram.org/file/bot<token>/<file_path>. Ссылка будет действительна как минимум в течение 1 часа. По истечении этого срока она может быть запрошена заново с помощью метода getFile.

file_id - Уникальный идентификатор файла
file_size - Размер файла, если известен
file_path - Расположение файла. Для скачивания воспользуйтейсь ссылкой вида *https://api.telegram.org/file/bot<token>/<file_path>*

getFile - Используйте этот метод, чтобы получить основную информацию о файле и подготовить его к загрузке. На данный момент боты могут загружать файлы размером до 20

МБ. При успешном выполнении возвращается объект File. Затем файл можно загрузить по ссылке https://api.telegram.org/file/bot <токен>/<путь к файлу>, где <путь к файлу> берется из ответа. Гарантируется, что ссылка будет действительна не менее 1 часа. Когда срок действия ссылки истечет, можно запросить новую, снова вызвав GetFile.

file_id - Идентификатор файла для получения информации

ReplyKeyboardMarkup - Этот объект представляет клавиатуру с опциями ответа

keyboard - Массив рядов кнопок, каждый из которых является массивом объектов KeyboardButton
resize_keyboard - Указывает клиенту подогнать высоту клавиатуры под количество кнопок (сделать её меньше, если кнопок мало). По умолчанию False, то есть клавиатура всегда такого же размера, как и стандартная клавиатура устройства.
one_time_keyboard - Указывает клиенту скрыть клавиатуру после использования (после нажатия на кнопку). Её по-прежнему можно будет открыть через иконку в поле ввода сообщения. По умолчанию False.
selective - Этот параметр нужен, чтобы показывать клавиатуру только определённым пользователям. Цели: 1) пользователи, которые были @упомянуты в поле text объекта Message; 2) если сообщения бота является ответом (содержит поле reply_to_message_id), авторы этого сообщения.

Пример: Пользователь отправляет запрос на смену языка бота. Бот отправляет клавиатуру со списком языков, видимую только этому пользователю.

KeyboardButton - Этот объект представляет одну кнопку в клавиатуре ответа. Для обычных текстовых кнопок этот объект может быть заменён на строку, содержащую текст на кнопке.

text - Текст на кнопке. Если ни одно из опциональных полей не использовано, то при нажатии на кнопку этот текст будет отправлен боту как простое сообщение.
request_contact - Если значение True, то при нажатии на кнопку боту отправится контакт пользователя с его номером телефона. Доступно только в диалогах с ботом.
request_location - Если значение True, то при нажатии на кнопку боту отправится местоположение пользователя. Доступно только в диалогах с ботом .

```
import requests

BASE_URL = 'https://api.telegram.org/bot'

TOKEN = 'Ваш токен'
ADMINS = [Ваш id, ]
```

```
def pulling():
   count_message = 0
   while True:
       response = requests.get(f'{BASE_URL}{TOKEN}/getUpdates').json()
       if count_message != len(response['result']):
           count_message = len(response['result'])
           message = response['result'][-1]
           user_id = message['message']['from']['id']
           requests.post(f"{BASE_URL}{TOKEN}/",
                         json={'method': 'sendMessage', 'chat_id':
f'{user_id}', 'text': 'Обычная клавиатура',
                               'reply_markup': {'keyboard': [[{'text':
'Yes'}, {'text': 'No'}]],'resize_keyboard': True, 'one_time_keyboard':
True},
                               })


pulling()
```

В данном случае клавиатура у нас будет постоянно появляться, но ничего страшного, в этом примере мы рассматриваем как работать с указанными объектами. Также мы можем сделать несколько рядов кнопок

```
import requests

BASE_URL = 'https://api.telegram.org/bot'

TOKEN = 'Ваш токен'
ADMINS = [Ваш id, ]


def pulling():
   count_message = 0
   while True:
       response = requests.get(f'{BASE_URL}{TOKEN}/getUpdates').json()
       if count_message != len(response['result']):
           count_message = len(response['result'])
           message = response['result'][-1]
           user_id = message['message']['from']['id']
           requests.get(f"{BASE_URL}{TOKEN}/",
                        json={'method': 'sendMessage', 'chat_id':
f'{user_id}', 'text': 'Обычная клавиатура',
                              'reply_markup': {'keyboard': [[{'text':
'Yes'}, {'text': 'No'}],
                                                            ['Второй ряд'],

],'resize_keyboard': True, 'one_time_keyboard': True},
                               })


pulling()
```

Также зная параметры кнопок, мы можем отправить свою локацию или контакт

```
import requests

BASE_URL = 'https://api.telegram.org/bot'

TOKEN = 'Ваш токен'
ADMINS = [Ваш id, ]


def pulling():
   count_message = 0
   while True:
       response = requests.get(f'{BASE_URL}{TOKEN}/getUpdates').json()
       if count_message != len(response['result']):
           count_message = len(response['result'])
           message = response['result'][-1]
           user_id = message['message']['from']['id']
           requests.get(f"{BASE_URL}{TOKEN}/",
                        json={'method': 'sendMessage', 'chat_id':
f'{user_id}', 'text': 'Обычная клавиатура',
                              'reply_markup': {'keyboard': [[{'text':
'Отправить локацию', 'request_location': True},
                                                             {'text':
'Отправить контакт', 'request_contact': True}],

],'resize_keyboard': True, 'one_time_keyboard': True},
                                })


pulling()
```

ReplyKeyboardHide - После получения сообщения с этим объектом, приложение Telegram свернёт клавиатуру бота и отобразит стандартную клавиатуру устройства (с буквами). По умолчанию клавиатуры бота отображаются до тех пор, пока не будет принудительно отправлена новая или скрыта старая клавиатура. Исключение составляют одноразовые клавиатуры, которые скрываются сразу после нажатия на какую-либо кнопку

hide_keyboard - Указание клиенту скрыть клавиатуру бота
selective - Опционально. Используйте этот параметр, чтобы скрыть клавиатуру только у определённых пользователей. Цели: 1) пользователи, которые были @упомянуты в поле text объекта Message; 2) если сообщения бота является ответом (содержит поле reply_to_message_id), авторы этого сообщения.

Пример: Пользователь голосует в опросе, бот отправляет сообщение с подтверждением и скрывает клавиатуру у этого пользователя, в то время как у всех остальных клавиатура видна.

InlineKeyboardMarkup - Этот объект представляет встроенную клавиатуру, которая появляется под соответствующим сообщением.

inline_keyboard - Массив строк, каждая из которых является массивом объектов InlineKeyboardButton.

InlineKeyboardButton - Этот объект представляет одну кнопку встроенной клавиатуры. Вы обязательно должны задействовать ровно одно опциональное поле.

text - Текст на кнопке
url - Опционально. URL, который откроется при нажатии на кнопку
callback_data - Опционально. Данные, которые будут отправлены в callback_query при нажатии на кнопку
switch_inline_query - Опционально. Если этот параметр задан, то при нажатии на кнопку приложение предложит пользователю выбрать любой чат, откроет его и вставит в поле ввода сообщения юзернейм бота и определённый запрос для встроенного режима. Если отправлять пустое поле, то будет вставлен только юзернейм бота.

Примечание: это нужно для того, чтобы быстро переключаться между диалогом с ботом и встроенным режимом с этим же ботом. Особенно полезно в сочетаниями с действиями switch_pm... – в этом случае пользователь вернётся в исходный чат автоматически, без ручного выбора из списка.

switch_inline_query_current_chat - Если задано, нажатие кнопки вставит имя пользователя бота и указанный встроенный запрос в поле ввода текущего чата. Может быть пустым, и в этом случае будет вставлено только имя пользователя бота.

callback_game - Описание игры, которая будет запущена, когда пользователь нажмет на кнопку.

ПРИМЕЧАНИЕ: Кнопка этого типа всегда должна быть первой кнопкой в первом ряду.

```
import requests

BASE_URL = 'https://api.telegram.org/bot'

TOKEN = 'Ваш токен'
ADMINS = [Ваш id, ]


def pulling():
   count_message = 0
   while True:
       response = requests.get(f'{BASE_URL}{TOKEN}/getUpdates').json()
       if count_message != len(response['result']):
```

```
            count_message = len(response['result'])
            message = response['result'][-1]
            user_id = message['message']['from']['id']
            requests.get(f"{BASE_URL}{TOKEN}/",
                         json={'method': 'sendMessage', 'chat_id':
f'{user_id}', 'text': 'Inline клавиатура',
                               'reply_markup': {'keyboard': [[{'text':
'Google', 'url': 'www.google.com'},
                                                              {'text':
'Callback', 'callback_data': 'button'}],
                                                             ]},
                               })


pulling()
```

CallbackQuery - этот объект представляет входящий запрос обратной связи от инлайн-кнопки с заданным callback_data.
Если кнопка, создавшая этот запрос, была привязана к сообщению, то в запросе будет присутствовать поле message.
Если кнопка была показана в сообщении, отправленном при помощи встроенного режима, в запросе будет присутствовать поле inline_message_id.

id - Уникальный идентификатор запроса
from - Отправитель
message - Опционально. Сообщение, к которому была привязана вызвавшая запрос кнопка. Обратите внимание: если сообщение слишком старое, содержание сообщения и дата отправки будут недоступны.
inline_message_id - Опционально. Идентификатор сообщения, отправленного через вашего бота во встроенном режиме
data - Данные, связанные с кнопкой. Обратите внимание, что клиенты могут добавлять свои данные в это поле.

kickChatMember - Используйте этот метод, чтобы удалить пользователя из группы или супергруппы. В случае супергрупп пользователь не сможет вернуться в группу самостоятельно, используя ссылки для приглашения и т.д., Если сначала не будет снят запрет. Чтобы это сработало, бот должен быть администратором в группе. Возвращает значение True при успешном выполнении.

chat_id - Уникальный идентификатор целевой группы или имя пользователя целевой супергруппы (в формате @supergroupusername)
user_id - Уникальный идентификатор целевого пользователя

```
import requests

BASE_URL = 'https://api.telegram.org/bot'
```

```
TOKEN = '5580946066:AAH9SSwpAd_Pyp9pNkE9NrlD7DDZ53igeio'
ADMINS = [1865314469, ]


def pulling():
   count_message = 0
   while True:
       response = requests.get(f'{BASE_URL}{TOKEN}/getUpdates').json()
       if count_message != len(response['result']):
           count_message = len(response['result'])
           message = response['result'][-1]
           user_id = message['message']['from']['id']
           requests.get(f"{BASE_URL}{TOKEN}/",
                        json={'method': 'kickChatMember', 'chat_id': 'id
чата', 'user_id': 'id юзера'})


pulling()
```

unbanChatMember - Используйте этот метод, чтобы отменить блокировку ранее удаленного пользователя в супергруппе. Пользователь не вернется в группу автоматически, но сможет присоединиться по ссылке и т.д. Чтобы это сработало, бот должен быть администратором в группе. Возвращает значение True при успешном выполнении .

chat_id - Уникальный идентификатор целевой группы или имя пользователя целевой супергруппы (в формате @supergroupusername)
user_id - Уникальный идентификатор целевого пользователя

```
import requests

BASE_URL = 'https://api.telegram.org/bot'

TOKEN = 'Ваш токен'
ADMINS = [Ваш id, ]


def pulling():
   count_message = 0
   while True:
       response = requests.get(f'{BASE_URL}{TOKEN}/getUpdates').json()
       if count_message != len(response['result']):
           count_message = len(response['result'])
           message = response['result'][-1]
           user_id = message['message']['from']['id']
           requests.get(f"{BASE_URL}{TOKEN}/",
                        json={'method': 'unbanChatMember', 'chat_id': 'id
чата', 'user_id': 'id юзера'})


pulling()
```