



## Урок 8

[Введение](#)[Новые возможности](#)[URL-кнопки](#)[Callback-кнопки и редактирование сообщений](#)[Switch-кнопки](#)

# Bot API v2: Кнопки и редактирование сообщений

## Введение

В начале апреля 2016 года [вышло](#) первое по-настоящему крупное обновление API для ботов. Изменений довольно много, поэтому материал я разобью на несколько частей. Сегодня поговорим об inline-кнопках и редактировании сообщений, а затем обсудим новые инлайн-режимы вместе со специальными кнопками для отправки геолокации и номера телефона.

## Новые возможности

Начнём с двух важных изменений:

1. Каждая кнопка, будь то обычная или инлайн, это теперь самостоятельный объект `KeyboardButton` или `InlineKeyboardButton`, не забудьте обновить своих ботов!
2. В Inline-режиме все текстовые поля теперь представлены отдельными объектами `InputMessageContent`, которые, в свою очередь могут быть аж 4-х типов (подробности [тут](#)).

## URL-кнопки

Итак, **инлайн-кнопки**. Что это такое? Это специальные объекты, которые “цепляются” к конкретным сообщениям и распространяют своё действие, в общем случае, только на них. Делятся такие кнопки на три типа: URL-кнопки, Callback-кнопки и Switch-кнопки. Самыми простыми являются кнопки-ссылки (URL). Как видно из названия, их цель - просто перекидывать пользователей по определенным веб-адресам. Давайте сразу напишем обработчик, который будет на любое сообщение отвечать каким-либо текстом и предложением перейти, например, на Яндекс.

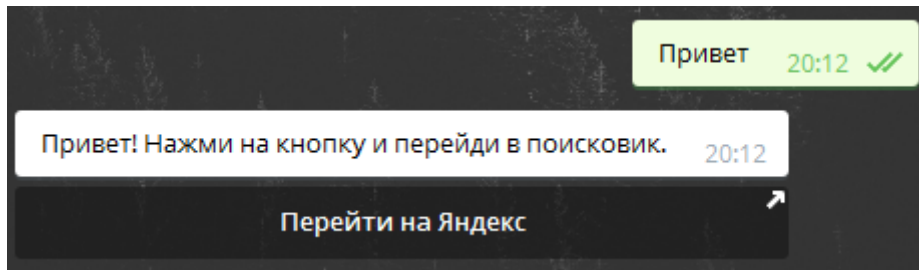
```
:types=["text"])
```

```
boardMarkup()
```

```
keyboardButton(text="Перейти на Яндекс", url="https://ya.ru")
```

```
chat.id, "Привет! Нажми на кнопку и перейди в поисковик.", reply_markup=keyboard)
```

Инлайн-клавиатура представляет собой объект `InlineKeyboardMarkup`, а каждая инлайн-кнопка – это объект `InlineKeyboardButton`. Чтобы получилась URL-кнопка, нужно указать значения параметров `text` (текст на кнопке) и `url` (валидный веб-адрес). В результате бот пришлет нам такое сообщение (см. рис.). В целях обеспечения безопасности, перед переходом по URL-кнопкам появляется всплывающее окно, в котором видна ссылка целиком.



*URL-кнопка*

## Callback-кнопки и редактирование сообщений

Прежде, чем мы перейдем к другим кнопкам, давайте познакомимся с функциями редактирования сообщений, коих тоже три: `editMessageText` (редактирование текста), `editMessageCaption` (редактирование подписи к медиа) и `editMessageReplyMarkup` (редактирование инлайн-клавиатуры). В рамках этого урока рассмотрим только первую функцию, остальные работают аналогично и предлагаются для самостоятельного изучения.

Чтобы отредактировать сообщение, нам надо знать, про какое именно идёт речь. В случае, если оно было отправлено самим ботом, идентификаторами служит связка `chat_id + message_id`. Но если сообщение было отправлено в инлайн-режиме, то ориентироваться надо по параметру `inline_message_id`.

И вот теперь вернемся к нашим ~~баранам~~ кнопкам. На очереди – **Callback**. Это, на мой взгляд, самая крутая фишка нового обновления. Колбэк-кнопки позволяют выполнять произвольные действия по их нажатию. Всё зависит от того, какие параметры каждая кнопка в себе несёт. Соответственно, все нажатия будут приводить к отправке боту объекта `CallbackQuery`, содержащему поле `data`, в котором написана некоторая строка, заложенная в кнопку, а также либо объект `Message`, если сообщение

отправлено ботом в обычном режиме, либо поле `inline_message_id`, если сообщение отправлено в инлайн-режиме.

Приведу пример, после которого все вопросы должны отпасть: пусть, например, если сообщение отправлено ботом в обычном режиме, то нажатие на кнопку заменит текст сообщения на "Пыщь", если в инлайн – то "Бдыщь". При этом в обоих случаях значение `callback_data` будет равно `test`. Что для этого нужно сделать: во-первых, написать простейший хэндлер для всех входящих сообщений, во-вторых, написать простейший хэндлер для инлайн-сообщений, в-третьих, написать простейший хэндлер для колбэка, который определит, из какого режима пришло сообщение.

#### # Обычный режим

```
@bot.message_handler(content_types=["text"])
def any_msg(message):
    keyboard = types.InlineKeyboardMarkup()
    callback_button = types.InlineKeyboardButton(text="Нажми меня", callback_data="test")
    keyboard.add(callback_button)
    bot.send_message(message.chat.id, "Я – сообщение из обычного режима", reply_to_message_id=message.id)
```

#### # Инлайн-режим с непустым запросом

```
@bot.inline_handler(lambda query: len(query.query) > 0)
def query_text(query):
    kb = types.InlineKeyboardMarkup()
    # Добавляем колбэк-кнопку с содержимым "test"
    kb.add(types.InlineKeyboardButton(text="Нажми меня", callback_data="test"))
    results = []
    single_msg = types.InlineQueryResultArticle(
        id="1", title="Press me",
        input_message_content=types.InputTextMessageContent(message_text="Я – сообщение из инлайн-режима"),
        reply_markup=kb
    )
    results.append(single_msg)
    bot.answer_inline_query(query.id, results)
```

#### # В большинстве случаев целесообразно разбить этот хэндлер на несколько маленьких

```
@bot.callback_query_handler(func=lambda call: True)
def callback_inline(call):
    # Если сообщение из чата с ботом
```

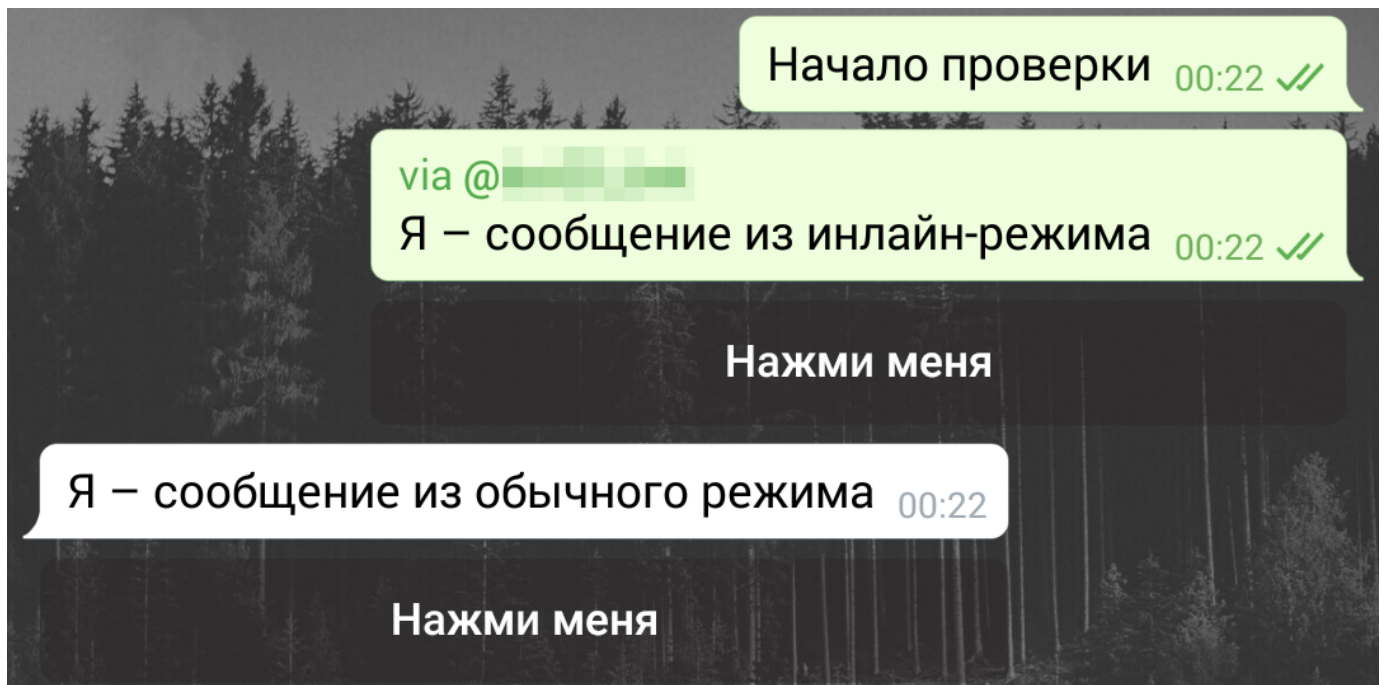
```

if call.message:
    if call.data == "test":
        bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.m
# Если сообщение из инлайн-режима
elif call.inline_message_id:
    if call.data == "test":
        bot.edit_message_text(inline_message_id=call.inline_message_id, text:

if __name__ == '__main__':
    bot.infinity_polling()

```

Запускаем бота, отправляем инлайн-сообщение, которое, в свою очередь, вызовет обычное:



Нажмем на обе кнопки, результат правильный:

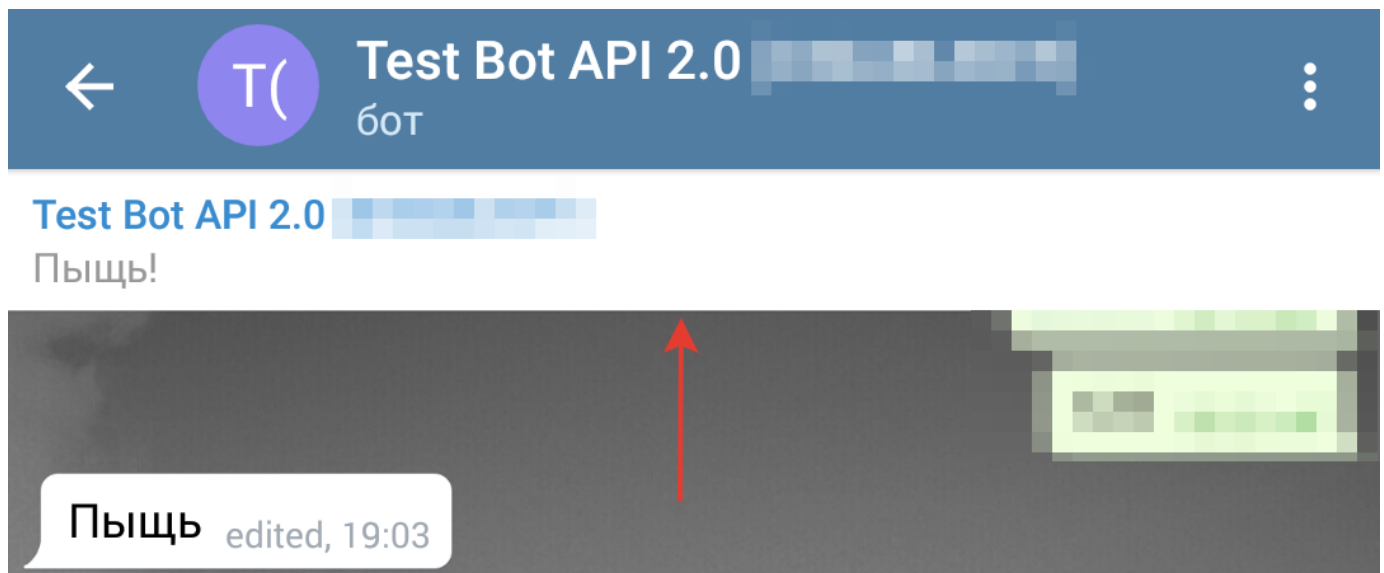


*После проверки*

Таким образом, callback-кнопки – это очень мощный инструмент для взаимодействия пользователей с ботом, а редактирование сообщений дополнительно помогает в этом. Более того, нажатие на колбэк-кнопку может дополнительно триггернуть либо уведомление в верхней части экрана, либо всплывающее окно. Покажу первый вариант. Пускай помимо изменения сообщения на “Пыщь”, аналогичное слово показывается уведомлением. Для этого перепишем первое if-условие в хендлере колбэков:

```
if call.message:
    if call.data == "test":
        bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message_id, text="Пыщь")
        bot.answer_callback_query(callback_query_id=call.id, show_alert=True, text="Пыщь")
```

Результат – на скриншоте. Попробуйте, кстати, изменить аргумент `show_alert` на `True` и посмотрите, что получится.



*Всплывающее уведомление*

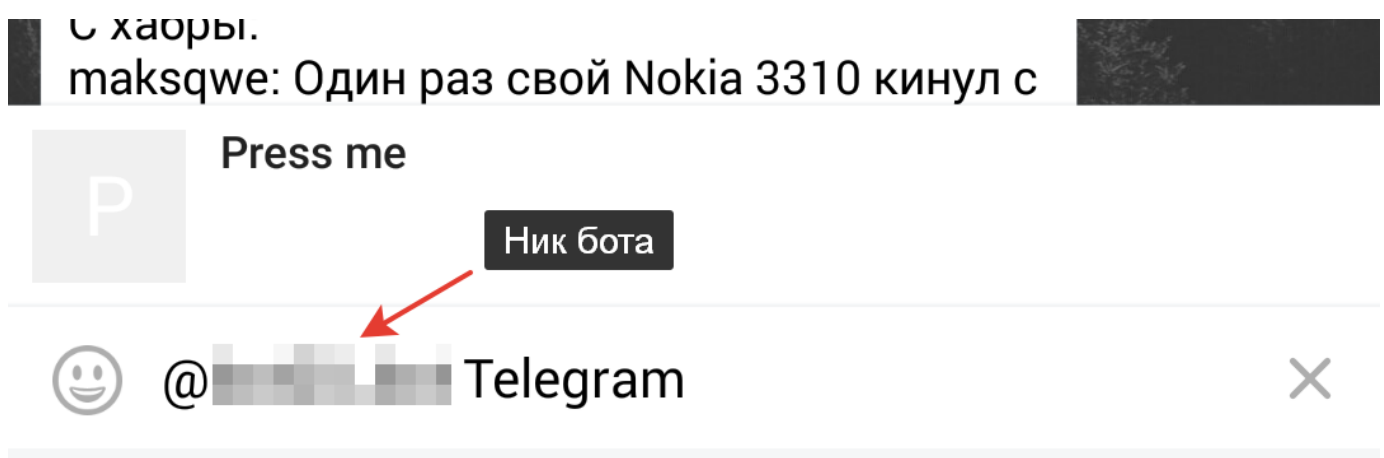
## Switch-кнопки

Наконец, остался последний тип кнопок - **Switch** (переключатель). Они нужны, чаще всего, для обучения пользователей работе с ботом в инлайн-режиме. Чтобы активировать сделать кнопку такого типа, нужно указать аргумент `switch_inline_query` либо пустой, либо с каким-либо текстом. В последнем случае этот текст будет сразу подставлен в поле ввода, например, для показа демонстрации инлайна. Как вообще работает такая кнопка? При нажатии на неё Telegram предложит выбрать чат, после чего подставит в поле ввода ник вашего бота и (если есть), текст, указанный вами в аргументе `switch_inline_query`. Давайте попробуем так сделать. Добавим кнопку, которая будет перенаправлять пользователя в какой-либо

чат и предлагать в инлайн-режиме запрос "Telegram". Код всего хендлера выглядит вот так:

```
@bot.message_handler(content_types=["text"])
def any_msg(message):
    keyboard = types.InlineKeyboardMarkup()
    switch_button = types.InlineKeyboardButton(text="Нажми меня", switch_inline_
    keyboard.add(switch_button)
    bot.send_message(message.chat.id, "Я – сообщение из обычного режима", reply_
```

Теперь, если мы нажмем на кнопку и выберем чат, вот что получится:



### Результат

Итак, в этом уроке мы познакомились с новыми кнопками в Telegram Bot API, научились ~~переписывать историю~~ редактировать сообщения и отправлять небольшие уведомления по нажатию. В следующий раз продолжим изучать новые возможности для ботов. А исходники к этому уроку можно найти в [этом репозитории](#).

[← Урок №7](#)[Урок №9 →](#)