

# Подробное описание сложных методов

Некоторые методы обладают сложной структурой и требуют более детального объяснения. В этом разделе вы найдете отдельные страницы, посвященные подробному описанию параметров и особенностей работы таких методов.

- [where](#) для `bot.findLeads()`
- [orderBy](#) для `bot.findLeads()`

# where для bot.findLeads()

## Базовая структура:

Каждое условие внутри массива `where` имеет следующую структуру:

- 1. **Простое условие:** `[' поле' , ' значение' ]`  
Проверяет, равно ли поле заданному значению.
- 2. **Условие с оператором:** `[' поле' , ' оператор' , ' значение' ]`  
Проверяет поле с использованием указанного оператора.

```
let where = [  
  [' поле' , ' значение' ],  
  [' поле' , ' оператор' , ' значение' ]  
]  
  
bot.findLeads([], where, [])
```

## Доступные категории полей:

- 1. Поля лидов
- 2. Каналы (channels)
- 3. Статусы лидов (leadstatuses)
- 4. Атрибуты лидов (lead\_attributes)

### 1. Поля лидов

Это основные поля, доступные для каждого лида.

Поле	Описание
id	ID лида
name	Имя лида
identification	ID в мессенджере (например, Telegram ID)
person_id	ID персоны
manager_id	ID менеджера
bot_id	ID бота

Поле	Описание
channel_id	ID канала
lang_code	Код языка
status_id	ID статуса
is_mute	Включен ли бот для лида (0 или 1)
is_forwarded	Передан ли оператору (0 или 1)
last_activity	Дата последней активности
last_response	Timestamp последнего ответа
messenger	ID мессенджера
bot_debug_level	Уровень отладки бота (0 - обычный, 11 - админ)
created_at	Дата создания
updated_at	Дата обновления
dialog_id	ID диалога
unread_count	Количество непрочитанных сообщений
forwarded_at	Дата передачи оператору

Пример использования:

```
where = [  
  ['id', 145492],  
  ['name', 'like', '%Иван%'],  
  ['created_at', '>=', '2023-01-01']  
]
```

## 2. Каналы (channels)

Для поиска по каналам используются поля `channels.code` и `channels.id`.

Код	ID	Описание
unknown	0	Неизвестный канал
avito	1	Avito
facebook	2	Facebook
instagram	3	Instagram
livechatWidget	4	LiveChat Widget
skype	5	Skype

Код	ID	Описание
telegram	6	Telegram
twitter	7	Twitter
viber	8	Viber
vk	9	Vkontakte
whatsapp	10	WhatsApp
youla	11	Youla
wechat	12	WeChat
metabotWidget	13	MetaBot Widget
webim	14	Webim
jivoWidget	15	Jivo Widget
odnoklassniki	16	Odnoklassniki
apple_business_chat	17	Apple Business Chat

Канал текущего лида можно получить с помощью метода `Lead.getMessengerCode()`.

Пример использования:

```
// Ищет только telegram
let where = [
  ['channels.code', 'telegram']
]

// Ищет telegram и whatsapp
let where = [
  ['channels.id', 'IN', [6, 10]]
]
```

### 3. Статусы лидов (leadstatuses)

Для поиска по статусам используются поля `leadstatuses.name` и `leadstatuses.id`.

Пример использования:

```
let where = [
  ['leadstatuses.name', 'Новый'],
  ['leadstatuses.id', 1]
]
```

Статусы можно получить на странице <https://app.metabot24.com/status>

## Рекомендации:

Используйте вместо `leadstatuses.id` - `status_id`

А вместо `channels.id` - `channel_id`

Если вы выполняете большие операции это сократит время вычисления

## 4. Атрибуты лидов (lead\_attributes)

Для поиска по атрибутам используются следующие поля:

Поле	Описание
<code>lead_attributes.attribute_type</code>	Тип атрибута (tag, variable, system, json, context)
<code>lead_attributes.attribute_key</code>	Ключ атрибута
<code>lead_attributes.attribute_value</code>	Значение атрибута. Если type = tag, то можно не использовать

Пример использования:

```
let where = [  
  ['lead_attributes.attribute_type', 'variable'],  
  ['lead_attributes.attribute_key', 'is_pay'],  
  ['lead_attributes.attribute_value', '1']  
]
```

## Операторы

В условиях можно использовать различные операторы:

- Сравнения: `=`, `<`, `>`, `<=`, `>=`, `<>`, `!=`
- Строковые: `like`, `not like`
- Множества: `IN`, `NOT IN`

Оператор	Описание	Тип переменной	Пример
<code>=</code>	Равно	string, int	<code>['id', '=', 100]</code>
<code>&lt;</code>	Меньше	int, float	<code>['price', '&lt;', 1000]</code>
<code>&gt;</code>	Больше	int, float	<code>['age', '&gt;', 18]</code>

Оператор	Описание	Тип переменной	Пример
<=	Меньше или равно	int, float	['quantity', '<= ', 50]
>=	Больше или равно	int, float	['rating', '>= ', 4.5]
<> или !=	Не равно	string, int	['status', '<> ', 'closed']
like	Соответствует шаблону	string	['name', 'like', '%John%']
not like	Не соответствует шаблону	string	['email', 'not like', '%@spam.com']
ilike	Соответствует шаблону (без учета регистра)	string	['title', 'ilike', '%book%']
IN	Содержится в списке	array	['id', 'IN', [1, 2, 3]]
NOT IN	Не содержится в списке	array	['status', 'NOT IN', ['closed', 'deleted']]
is	Равно (для null)	null	['manager_id', 'is', null]
is not	Не равно (для null)	null	['end_date', 'is not', null]
regex	Соответствует регулярному выражению	string	['phone', 'regex', '^\\+7']
not regex	Не соответствует регулярному выражению	string	['username', 'not regex', '[0-9]']

Примечание: Некоторые операторы могут быть недоступны или работать иначе.

Пример использования операторов:

```
let where = [  
  ['id', '>', 1000],  
  ['name', 'like', '%Иван%'],  
  ['channels.code', 'IN', ['telegram', 'whatsapp']]  
]
```

# Полный пример использования

```
let where = [  
  ['name', 'like', '%Иван%'],  
  ['channels.code', 'telegram'],  
  ['leadstatuses.name', 'Новый'],  
]
```

```
['lead_attributes.attribute_type', 'tag'],  
['lead_attributes.attribute_key', 'важный_клиент']  
]
```

```
let leads = bot.findLeads([], where, [])  
debug( leads)
```

# orderBy для bot.findLeads()

## Базовая структура:

Параметр `orderBy` используется для сортировки результатов поиска лидов по заданным полям. Он представляет собой массив условий, где каждое условие задаёт поле для сортировки и порядок сортировки (по возрастанию или по убыванию).

Каждое условие внутри массива `orderBy` имеет следующую структуру:

```
let orderBy = [
  [ 'поле', 'порядок' ]
]

bot.findLeads([], [], orderBy)
```

1. **Поле:** Название поля, по которому будет производиться сортировка (например, `created_at`, `updated_at`, `id`).
2. **Порядок:** Определяет, как будет сортироваться поле. Возможны два значения:
  - `ASC` — по возрастанию.
  - `DESC` — по убыванию.

### Доступные категории полей:

1. [Поля лидов](#)
2. [Статусы лидов \(leadstatuses\)](#)

## Примеры использования

### 1. Сортировка по одному полю

```
let orderBy = [
  [ 'created_at', 'ASC' ]
]

let leads = bot.findLeads([], [], orderBy)
```

Этот пример сортирует лидов по дате создания в порядке возрастания (от самого раннего к самому позднему).

## 2. Сортировка по нескольким полям

```
let orderBy = [  
  ['updated_at', 'DESC'],  
  ['id', 'ASC']  
]  
  
let leads = bot.findLeads([], [], orderBy)
```

Этот пример сначала сортирует лидов по дате обновления в порядке убывания (от самого последнего к самому раннему), а затем — по `id` в порядке возрастания.

## Пример с `where` и `orderBy`

```
let where = [  
  ['name', 'like', '%Иван%'],  
  ['channels.code', 'telegram']  
]  
  
let orderBy = [  
  ['created_at', 'DESC']  
]  
  
let leads = bot.findLeads([], where, orderBy)
```