

# Методы ticket

- `ticket.update()`
- `ticket.getData()`
- `ticket.getCustomData()`
- `ticket.issetData()` и `ticket.isDataExist()`
- `ticket.reloadCustomData()`
- `ticket.setAllCustomData()`
- `ticket.getAllCustomData()`
- `ticket.setCustomData()`
- `ticket.issetCustomData()`
- `ticket.isCustomDataExist()`
- `ticket.getTicketTypeName()`
- `ticket.getStatusId()` и `ticket.getAttr('status_id')`
- `ticket.getStatusName()`
- `ticket.getStatusType()`
- `ticket._sanitizeString()`
- `ticket.isInStatus()`
- `ticket.setStatusId()`
- `ticket.setStatusByName()`
- `ticket.reloadTicket()`
- `ticket.getAuthorPersonId()`
- `ticket.getAssignedPersonId()`
- `ticket.getClientPersonId()`
- `ticket.getAuthorPersonFullName()`
- `ticket.getAssignedPersonFullName()`
- `ticket.getClientPersonFullName()`
- `ticket.getAuthorPersonRoleId()`
- `ticket.getAssignedPersonRoleId()`

- `ticket.getClientPersonRoleId()`
- `ticket.getAuthorPersonRoleName()`
- `ticket.getAssignedPersonRoleName()`
- `ticket.getClientPersonRoleName()`
- `ticket.getAuthorPersonRoleTitle()`
- `ticket.getAssignedPersonRoleTitle()`
- `ticket.getClientPersonRoleTitle()`
- `ticket.getAuthorPersonRoleFormattedTitle()`
- `ticket.getAssignedPersonRoleFormattedTitle()`
- `ticket.getClientPersonRoleFormattedTitle()`
- `ticket.getPerson1Id()`
- `ticket.getPerson2Id()`
- `ticket.getPerson3Id()`
- `ticket.setPerson1Id()`
- `ticket.setPerson2Id()`
- `ticket.setPerson3Id()`
- `ticket.assignToNextQueuePerson()`
- `ticket.moveAssignedPersonToEndOfQueue()`
- `ticket.assignToPersonById()`
- `ticket.getAssignedPersonRejectsCount()`
- `ticket.setAssignedPersonRejectsCount()`
- `ticket.incAssignedPersonRejectsCount()`
- `ticket.assignToNextPersonAndMoveToEnd()`
- `ticket.getAllData()`
- `ticket.setData()`
- `ticket.loadCustomDataIfNotLoaded()`
- `ticket.getSubtotalFormatted()`
- `ticket.getTotalFeeFormatted()`
- `ticket.getDeliveryFeeFormatted()`
- `ticket.getTotalFormatted()`
- `ticket.getTicketLines()`

# ticket.update()

## Обновление заявки, через объект заявки

```
ticket.update($data)
```

### Описание:

Обновляет заявку соответствующую указанным данным.

### Атрибуты:

Имя	Тип	Описание
data	json	Данные о заявке в формате json. Возможные поля расписаны ниже
title	string	Заголовок заявки
ticket_type_id	int	Id типа заявки
status_id	int	Id статуса заявки
author_person_id	int	Id автора заявки
client_person_id	int	Id клиента заявки
assigned_person_id	int	Id исполнителя заявки
description	string	Описание заявки

### Возвращает:

**bool** - подтверждение обновления заявки.

### Пример:

```
let ticketData = {  
  'title': 'Создано в боте',  
  'ticket_type_id': 1013,  
  'status_id': 1096  
};  
  
ticketData.custom = {  
  'client_contacts': 'my contacts',  
  'details': 'some details'  
};  
  
let isUpdated = ticket.update(ticketData);  
  
lead.setAttr('is_ticket_updated', isUpdated);
```

Пример обновления одного поля заявки:

```
let ticket = bot.findTickets(['id', ticketId])[0];  
let ticketData = {  
  'client_comment': lead.getAttr('Отзыв')  
};  
  
ticket.update(ticketData);
```

# ticket.getData()

## Получить значение данных заявки

```
ticket.getData(string $key)
```

### Описание:

Получить значение данных заявки по ключу или названию поля БД.

### Атрибуты:

Имя	Тип	Описание
key	string	Ключ наименование поля

В качестве ключа можно указать:

- **id** - id заявки;
- **business\_id** - id бизнеса;
- **bot\_id** - id бота;
- **author\_person\_id** - id автора заявки;
- **client\_person\_id** - id клиента заявки;
- **assigned\_person\_id** - id исполнителя заявки;
- **ticket\_type\_id** - id вида заявки, виды заявки создаются вручную;
- **payment\_type\_id** - id способа оплаты. Доступные значения:
  - **0** - PAYMENT\_TYPE\_NONE;
  - **1** - PAYMENT\_TYPE\_ROBOKASSA;
  - **2** - PAYMENT\_TYPE\_YANDEXKASSA;
  - **3** - PAYMENT\_TYPE\_CASH;
- **status\_id** - id статуса заявки;
- **title** - заголовок заявки;
- **description** - описание заявки;
- **created\_at** - время создания заявки;
- **updated\_at** - время обновления заявки;
- **deleted\_at** - время удаления заявки;
- все системные поля заявки (по английски).

## Возвращает:

**mixed** - если значение найдено

или

**null** - если значение не найдено.

## Пример:

```
let Data = ticket.getData('title');
```

# ticket.getCustomData()

## Считывание произвольных полей заявки

```
ticket.getCustomData(string $key)
```

### Описание:

Находит значение указанного типа произвольного поля заявки.

### Атрибуты:

Имя	Тип	Описание
key	string	Ключ наименование поля

В качестве ключа можно указать:

- **названия всех произвольных полей заявки;**
- **id** - id заявки;
- **business\_id** - id бизнеса;
- **bot\_id** - id бота;
- **author\_person\_id** - id автора заявки;
- **client\_person\_id** - id клиента заявки;
- **assigned\_person\_id** - id исполнителя заявки;
- **ticket\_type\_id** - id вида заявки, виды заявки создаются вручную;
- **payment\_type\_id** - id способа оплаты. Доступные значения:
  - **0** - PAYMENT\_TYPE\_NONE;
  - **1** - PAYMENT\_TYPE\_ROBOKASSA;
  - **2** - PAYMENT\_TYPE\_YANDEXKASSA;
  - **3** - PAYMENT\_TYPE\_CASH;
- **status\_id** - id статуса заявки;
- **title** - заголовок заявки;
- **description** - описание заявки;
- **created\_at** - время создания заявки;
- **updated\_at** - время обновления заявки;
- **deleted\_at** - время удаления заявки;
- все системные поля заявки (по английски).

Можно получить код значения, используя суффикс **".raw"** к имени произвольного поля. Т.е. если в заявке установлено из опций поля с именем age второе значение:

```
({
  "1": "18-25 лет",
  "2": "25-30 лет",
  "3": "30-35 лет",
  "4": "35-40 лет",
  "5": "Больше 40 лет",
  "6": "Не важно"
})
```

то **ticket.getCustomData('age')** вернёт "25-30 лет", **ticket.getCustomData('age.raw')** вернёт "2".

## Возвращает:

**mixed** - если значение найдено

или

**null** - если значение не найдено.

## Пример:

```
let tickets = bot.findTickets();

for (let i = 0; i < tickets.length; i++) {
  lead.setAttr('tickets_' + i + '_tickets_title', tickets[i].getData('title'));
  lead.setAttr('tickets_' + i + '_client_contacts',
tickets[i].getCustomData('client_contacts'));
  lead.setAttr('tickets_' + i + '_details', tickets[i].getCustomData('details'));
  lead.setAttr('tickets_' + i + '_статус_имя', tickets[i].getCustomData('статус.имя'));
}

lead.setAttr('found_tickets_count', tickets.length);
```



# ticket.issetData() и ticket.isDataExist()

## Проверить наличие данных по заявке

```
ticket.issetData(string $key)
```

или

```
ticket.isDataExist(string $key)
```

### Описание:

Проверяет существуют ли данные заявки по заданному ключу.

### Атрибуты:

Имя	Тип	Описание
key	string	Ключ наименование поля

В качестве ключа можно указать:

- **id** - id заявки;
- **business\_id** - id бизнеса;
- **bot\_id** - id бота;
- **author\_person\_id** - id автора заявки;
- **client\_person\_id** - id клиента заявки;
- **assigned\_person\_id** - id исполнителя заявки;
- **ticket\_type\_id** - id вида заявки, виды заявки создаются вручную;
- **payment\_type\_id** - id способа оплаты. Доступные значения:
  - **0** - PAYMENT\_TYPE\_NONE;
  - **1** - PAYMENT\_TYPE\_ROBOKASSA;
  - **2** - PAYMENT\_TYPE\_YANDEXKASSA;
  - **3** - PAYMENT\_TYPE\_CASH;

- **status\_id** - id статуса заявки;
- **title** - заголовок заявки;
- **description** - описание заявки;
- **created\_at** - время создания заявки;
- **updated\_at** - время обновления заявки;
- **deleted\_at** - время удаления заявки;
- все системные поля заявки (по английски).

## Возвращает:

**bool** - подтверждение существования данных.

## Пример:

```
let Data = ticket.issetData('title');
```

или

```
let Data = ticket.isDataExist('title');
```

# ticket.reloadCustomData()

## Перезагрузка кастомных данных

```
ticket.reloadCustomData(bool $isReloadModel = false)
```

### Описание:

Перезагружает кастомные поля заявки.

### Атрибуты:

Имя	Тип	Описание
isReloadModel	bool	Перезагружает и системные поля заявки, вызывая ticket.reloadTicket()

### Возвращает:

**bool** — подтверждение успешной перезагрузки.

# ticket.setAllCustomData()

## Переопределить список кастомных полей

```
ticket.setAllCustomData(array $customData)
```

### Описание:

Не переносится в БД. Будет переопределен массив кастомных полей в памяти, используется для внутренних целей задания значений полей.

### Атрибуты:

Имя	Тип	Описание
customData	array	Массив кастомных полей

### Возвращает:

**self** - тип указанный при вызове метода.

### Пример:

```
ticket.setAllCustomData([ {  
    "место": "Место 01",  
    "время": "12:30"  
}]);
```

# ticket.getAllCustomData()

## Получить список кастомных полей

```
ticket.getAllCustomData()
```

### Описание:

Возвращает значения всех кастомных полей.

### Возвращает:

**array** - массив значений кастмных полей.

# ticket.setCustomData()

## Установить значение кастомного поля

```
ticket.setCustomData(string $key, $value)
```

### Описание:

Устанавливает значение кастомного поля. Не сохраняется в БД.

### Атрибуты:

Имя	Тип	Описание
key	string	Ключ наименование поля
value	self	Значение поля

В качестве ключа можно указать:

- **названия всех произвольных полей заявки;**
- **id** - id заявки;
- **business\_id** - id бизнеса;
- **bot\_id** - id бота;
- **author\_person\_id** - id автора заявки;
- **client\_person\_id** - id клиента заявки;
- **assigned\_person\_id** - id исполнителя заявки;
- **ticket\_type\_id** - id вида заявки, виды заявки создаются вручную;
- **payment\_type\_id** - id способа оплаты. Доступные значения:
  - **0** - PAYMENT\_TYPE\_NONE;
  - **1** - PAYMENT\_TYPE\_ROBOKASSA;
  - **2** - PAYMENT\_TYPE\_YANDEXKASSA;
  - **3** - PAYMENT\_TYPE\_CASH;
- **status\_id** - id статуса заявки;
- **title** - заголовок заявки;
- **description** - описание заявки;
- **created\_at** - время создания заявки;
- **updated\_at** - время обновления заявки;
- **deleted\_at** - время удаления заявки;
- все системные поля заявки (по английски).

## Возвращает:

**self** - тип указанный при вызове метода.

## Пример:

```
ticket.setCustomData("место", "Место1");
```

# ticket.issetCustomData()

## Имеет ли значение кастомное поле

```
ticket.issetCustomData(string $key)
```

### Описание:

Проверяет существует ли ключ и не имеет ли он значение null.

### Атрибуты:

Имя	Тип	Описание
key	string	Ключ наименование поля

В качестве ключа можно указать:

- **названия всех произвольных полей заявки;**
- **id** - id заявки;
- **business\_id** - id бизнеса;
- **bot\_id** - id бота;
- **author\_person\_id** - id автора заявки;
- **client\_person\_id** - id клиента заявки;
- **assigned\_person\_id** - id исполнителя заявки;
- **ticket\_type\_id** - id вида заявки, виды заявки создаются вручную;
- **payment\_type\_id** - id способа оплаты. Доступные значения:
  - **0** - PAYMENT\_TYPE\_NONE;
  - **1** - PAYMENT\_TYPE\_ROBOKASSA;
  - **2** - PAYMENT\_TYPE\_YANDEXKASSA;
  - **3** - PAYMENT\_TYPE\_CASH;
- **status\_id** - id статуса заявки;
- **title** - заголовок заявки;
- **description** - описание заявки;
- **created\_at** - время создания заявки;
- **updated\_at** - время обновления заявки;
- **deleted\_at** - время удаления заявки;
- все системные поля заявки (по английски).



Возвращает:

**bool** - подтверждение существования ключа.

Пример:

```
ticket.issetCustomData("время");
```

# ticket.isCustomDataExist()

## Существует ли кастомное поле

```
ticket.isCustomDataExist(string $key)
```

### Описание:

Проверяет существует ли такой ключ в списке кастомных полей.

### Атрибуты:

Имя	Тип	Описание
key	string	Ключ наименование поля

В качестве ключа можно указать:

- **названия всех произвольных полей заявки;**
- **id** - id заявки;
- **business\_id** - id бизнеса;
- **bot\_id** - id бота;
- **author\_person\_id** - id автора заявки;
- **client\_person\_id** - id клиента заявки;
- **assigned\_person\_id** - id исполнителя заявки;
- **ticket\_type\_id** - id вида заявки, виды заявки создаются вручную;
- **payment\_type\_id** - id способа оплаты. Доступные значения:
  - **0** - PAYMENT\_TYPE\_NONE;
  - **1** - PAYMENT\_TYPE\_ROBOKASSA;
  - **2** - PAYMENT\_TYPE\_YANDEXKASSA;
  - **3** - PAYMENT\_TYPE\_CASH;
- **status\_id** - id статуса заявки;
- **title** - заголовок заявки;
- **description** - описание заявки;
- **created\_at** - время создания заявки;
- **updated\_at** - время обновления заявки;
- **deleted\_at** - время удаления заявки;
- все системные поля заявки (по английски).

Возвращает:

**bool** - подтверждение существования ключа.

Пример:

```
ticket.isCustomDataExist("место");
```

# ticket.getTicketTypeName()

## Получить название типа заявки

```
ticket.getTicketTypeName()
```

### Описание:

Находит название типа заявки.

### Возвращает:

**string** — название типа заявки

или

**null** — если у заявки нет типа.

# ticket.getStatusId() и ticket.getAttr('status\_id')

## Получить ID статуса заявки

```
ticket.getStatusId()
```

или

```
ticket.getAttr('status_id')
```

### Описание:

Находит id статуса заявки.

### Атрибуты:

Имя	Тип	Описание
status_id	string	Ключ id статуса заявки

### Возвращает:

**int** — id статуса заявки

или

**null** — если у заявки нет статуса.

# ticket.getStatusName()

## Получить название статуса заявки

```
ticket.getStatusName()
```

### Описание:

Находит название статуса заявки.

### Возвращает:

**string** — название статуса заявки

или

**null** — если у заявки нет статуса.

# ticket.getStatusType()

## Получить тип статуса заявки

```
ticket.getStatusType()
```

### Описание:

Находит тип статуса заявки.

### Возвращает:

**string** — тип статуса заявки

или

**null** — если у заявки нет статуса или у статуса нет типа.

# ticket.\_sanitizeString()

## Приведение названия статуса к стандартному виду

```
ticket._sanitizeString($string)
```

### Описание:

Вспомогательная функция используемая для приведения названия статуса к нижнему регистру и удаления лишних символов из названия.

### Атрибуты:

Имя	Тип	Описание
statusNme	string	Наименование статуса заявки

### Возвращает:

**string** - обновленное наименование статуса заявки.

В v8 скриптах не используется.



# ticket.isInStatus()

## Проверить статус заявки

```
ticket.isInStatus($statusName)
```

### Описание:

Проверить, что заявка находится в статусе с именем `$statusName`.

### Атрибуты:

Имя	Тип	Описание
statusName	string	Наименование статуса заявки

### Возвращает:

**bool** — подтверждение, что заявка находится в указанном статусе.

# ticket.setStatusId()

## Установить статус заявки по id

```
ticket.setStatusId(int|null $statusId, int|null $personQueueIdForLog = null, string|null $personQueueTableForLog = null)
```

### Описание:

Устанавливает указанный статус для заявки по id.

### Атрибуты:

Имя	Тип	Описание
statusId	int null	id статуса заявки
personQueueIdForLog	int null	id очереди записываемый в лог
personQueueTableForLog	string null	Название таблицы-связки для очередей записываемое в лог

### Возвращает:

**bool** — подтверждение установки нового статуса.

# ticket.setStatusByName()

## Установить статус заявки по имени

```
ticket.setStatusByName(string $statusName, int|null $personQueueIdForLog = null, string|null $personQueueTableForLog = null)
```

### Описание:

Устанавливает указанный статус для заявки по имени.

### Атрибуты:

Имя	Тип	Описание
statusName	string	Наименование статуса заявки
personQueueIdForLog	int null	id очереди записываемый в лог
personQueueTableForLog	string null	Название таблицы-связки для очередей записываемое в лог

### Возвращает:

**bool** — подтверждение установки нового статуса.

# ticket.reloadTicket()

## Загрузить заявку повторно

```
ticket.reloadTicket()
```

### Описание:

Загрузить заявку повторно из БД. Автоматически выполняется, например, в случае изменения статуса заявки, назначении другого исполнителя и т.п.

### Возвращает:

**bool** — подтверждение повторной загрузки заявки.

# ticket.getAuthorPersonId()

## Получить ID автора заявки

```
ticket.getAuthorPersonId( )
```

### Описание:

Возвращает id автора заявки.

### Возвращает:

**int** — id автора заявки в формате числа

или

**null** — если автор не найден.

# ticket.getAssignedPersonId()

## Получить ID исполнителя заявки

```
ticket.getAssignedPersonId()
```

### Описание:

Возвращает id исполнителя заявки.

### Возвращает:

**int** — id исполнителя заявки в формате числа

или

**null** — если исполнитель не найден.

# ticket.getClientPersonId()

## Получить ID клиента заявки

```
ticket.getClientPersonId()
```

### Описание:

Возвращает id клиента заявки.

### Возвращает:

**int** — id клиента заявки в формате числа

или

**null** — если клиент не найден.

# ticket.getAuthorPersonFullName()

## Получить ФИО автора заявки

```
ticket.getAuthorPersonFullName()
```

### Описание:

Возвращает ФИО автора заявки.

### Возвращает:

**string** — ФИО автора заявки.



# ticket.getAssignedPersonFullName()

## Получить ФИО исполнителя заявки

```
ticket.getAssignedPersonFullName()
```

### Описание:

Возвращает ФИО исполнителя заявки.

### Возвращает:

**string** — ФИО исполнителя заявки.

# ticket.getClientPersonFullName()

## Получить ФИО клиента заявки

```
ticket.getClientPersonFullName()
```

### Описание:

Возвращает ФИО клиента заявки.

### Возвращает:

**string** — ФИО клиента заявки.

# ticket.getAuthorPersonRoleId() ( )

## Получить ID роли автора заявки

```
ticket.getAuthorPersonRoleId( )
```

### Описание:

Возвращает ID роли автора заявки.

### Возвращает:

**int** — ID роли автора заявки

или

**null** — если клиент или роль не найдены.

# ticket.getAssignedPersonRoleId()

## Получить ID роли исполнителя заявки

```
ticket.getAssignedPersonRoleId()
```

### Описание:

Возвращает ID роли исполнителя заявки.

### Возвращает:

**int** — ID роли исполнителя заявки

или

**null** — если клиент или роль не найдены.

# ticket.getClientPersonRoleId( )

## Получить ID роли клиента заявки

```
ticket.getClientPersonRoleId( )
```

### Описание:

Возвращает ID роли клиента заявки.

### Возвращает:

**int** — ID роли клиента заявки

или

**null** — если клиент или роль не найдены.

# ticket.getAuthorPersonRoleName()

## Получить имя роли автора заявки

```
ticket.getAuthorPersonRoleName()
```

### Описание:

Возвращает имя роли автора заявки.

### Возвращает:

**string** — имя роли автора заявки.

# ticket.getAssignedPersonRoleName()

## Получить имя роли исполнителя заявки

```
ticket.getAssignedPersonRoleName()
```

### Описание:

Возвращает имя роли исполнителя заявки.

### Возвращает:

**string** — имя роли исполнителя заявки.

# ticket.getClientPersonRoleName()

## Получить имя роли клиента заявки

```
ticket.getClientPersonRoleName()
```

### Описание:

Возвращает имя роли клиента заявки.

### Возвращает:

**string** — имя роли клиента заявки.



# ticket.getAuthorPersonRoleTitle()

## Получить заголовок роли автора заявки

```
ticket.getAuthorPersonRoleTitle()
```

### Описание:

Возвращает заголовок роли автора заявки.

### Возвращает:

**string** — заголовок роли автора заявки.

# ticket.getAssignedPersonRoleTitle()

## Получить заголовок роли исполнителя заявки

```
ticket.getAssignedPersonRoleTitle()
```

### Описание:

Возвращает заголовок роли исполнителя заявки.

### Возвращает:

**string** — заголовок роли исполнителя заявки.

# ticket.getClientPersonRoleTitle()

## Получить заголовок роли клиента заявки

```
ticket.getClientPersonRoleTitle()
```

### Описание:

Возвращает заголовок роли клиента заявки.

### Возвращает:

**string** — заголовок роли клиента заявки.

# ticket.getAuthorPersonRoleFormattedTitle()

Получить форматированное название роли автора заявки

```
ticket.getAuthorPersonRoleFormattedTitle()
```

## Описание:

Возвращает форматированное название роли автора заявки.

## Возвращает:

**string** — форматированное название роли автора заявки.

# ticket.getAssignedPersonRoleFormattedTitle()

Получить форматированное название роли исполнителя заявки

```
ticket.getAssignedPersonRoleFormattedTitle()
```

## Описание:

Возвращает форматированное название роли исполнителя заявки.

## Возвращает:

**string** — форматированное название роли исполнителя заявки.

# ticket.getClientPersonRoleFormattedTitle()

Получить форматированное название роли клиента заявки

```
ticket.getClientPersonRoleFormattedTitle()
```

Описание:

Возвращает форматированное название роли клиента заявки.

Возвращает:

**string** — форматированное название роли клиента заявки.

# ticket.getPerson1Id()

## Получение id первой дополнительной персоны

```
ticket.getPerson1Id()
```

### Описание:

Возвращает id первой дополнительной персоны заявки.

### Возвращает:

**int** — id персоны в формате числа

или

**null** — если персона не найдена.

# ticket.getPerson2Id()

## Получение id второй дополнительной персоны

```
ticket.getPerson2Id()
```

### Описание:

Возвращает id второй дополнительной персоны заявки.

### Возвращает:

**int** — id персоны в формате числа

или

**null** — если персона не найдена.



# ticket.getPerson3Id()

## Получение id третьей дополнительной персоны

```
ticket.getPerson3Id()
```

### Описание:

Возвращает id третьей дополнительной персоны заявки.

### Возвращает:

**int** — id персоны в формате числа

или

**null** — если персона не найдена.

# ticket.setPerson1Id()

## Установка id первой дополнительной персоны

```
ticket.setPerson1Id(int|null $personId)
```

### Описание:

Устанавливает id первой дополнительной персоны заявки.

### Атрибуты:

Имя	Тип	Описание
personId	int null	ID персоны

### Возвращает:

**bool** — подтверждение установки значения.

# ticket.setPerson2Id()

## Установка id второй дополнительной персоны

```
ticket.setPerson2Id(int|null $personId)
```

### Описание:

Устанавливает id второй дополнительной персоны заявки.

### Атрибуты:

Имя	Тип	Описание
personId	int null	ID персоны

### Возвращает:

**bool** — подтверждение установки значения.

# ticket.setPerson3Id()

## Установка id третьей дополнительной персоны

```
ticket.setPerson3Id(int|null $personId)
```

### Описание:

Устанавливает id третьей дополнительной персоны заявки.

### Атрибуты:

Имя	Тип	Описание
personId	int null	ID персоны

### Возвращает:

**bool** — подтверждение установки значения.

# ticket.assignToNextQueuePerson()

## Назначить заявку следующему исполнителю

```
ticket.assignToNextQueuePerson(int $queueMode = 1, int|null $personQueueId = null, string|null $personQueueTable = null, int|null $currentAssignAttempt = null, bool $isIncRejectsCount = true, bool $isMoveToEndIfRejectsExceed = true)
```

### Описание:

Назначить заявку следующему исполнителю. Исполнитель будет определен, как первый в очереди распределения по типу заявки.

Логика работы функции:

1. Ищет первую персону из очереди распределения, которой еще не назначали тикет (заявку).
2. Если персоны нет, то возвращаем false (чтобы остановить триггер распределения очереди).
3. Увеличивает счетчик отказов у найденной персоны.
4. Увеличивается счетчик отказов по найденному исполнителю.
5. Если кол-во отказов  $\geq$  \$maxRejects, то исполнитель переносится в конец очереди (исполнитель текущий, которого нашли, а не первый в очереди).
6. Назначает заявку найденной персоне.

Поиск исполнителя выполняется по следующим критериям:

- Исполнитель есть в очереди по типу заявки;
- Исполнитель активен в очереди распределения (включена соответствующая галка);
- Исполнителю еще не назначалась данная заявка.

Условие (количество отказов по исполнителю rejects\_count < \$maxRejects+1) НЕ НУЖНО, т.к.:

1. Как только лимит превышен персона сразу будет перемещена в конец очереди, а счетчик сброшен.
2. Может возникнуть баг, если лимит превышен, персона никогда не попадет в выборку и ее счетчик не будет сброшен.

## Атрибуты:

Имя	Тип	Описание
queueMode	int	Режим работы очереди. Для работы с очередью на основе видов заявок указывать не нужно, будет использовано значение по умолчанию <b>1</b> ; Для работы с очередью на основе видов очередей указать <b>2</b> .
personQueueId	int null	id очереди
personQueueTable	string null	Название таблицы-связки для очередей
currentAssignAttempt	int null	Номер попытки прогона одного и того же тикета по той же очереди
isIncRejectsCount	bool	Увеличивать или нет счетчик попыток у персоны (в очереди исполнителей)
isMoveToEndIfRejectsExceed	bool	Переносить или нет персону в конец очереди и сбрасывать счетчик попыток, если кол-во попыток превышено (превышает число указанное в настройках бота)

## Возвращает:

**bool** — подтверждение назначения заявки следующему исполнителю.

# ticket.moveAssignedPersonToEndOfQueue()

## Перенести исполнителя в конец очереди

```
ticket.moveAssignedPersonToEndOfQueue(int $queueMode = 1, int|null $personQueueId = null, string|null $personQueueTable = null, bool $isResetRejectsCount = true)
```

### Описание:

Используется для скрипта "Приема заявки" в боте, чтобы переместить исполнителя в конец очереди, после того как он принял заявку.

Переносит персону, которой в данный момент назначен тикет, в конец очереди, при этом исполнитель заявки не меняется.

### Атрибуты:

Имя	Тип	Описание
queueMode	int	Режим работы очереди. Для работы с очередью на основе видов заявок указывать не нужно, будет использовано значение по умолчанию <b>1</b> ; Для работы с очередью на основе видов очередей указать <b>2</b> .
personQueueId	int null	id очереди
personQueueTable	string null	Название таблицы-связки для очередей
isResetRejectsCount	bool	Обновить или нет счетчик попыток у персоны (в очереди исполнителей)

Возвращает:

**bool** — подтверждение переноса исполнителя.



# ticket.assignToPersonById()

## Назначить исполнителя заявки

```
ticket.assignToPersonById(int|null $personId, int|null $personQueueIdForLog = null,  
string|null $personQueueTableForLog = null, int|null $assignAttemptForLog = null)
```

### Описание:

Назначает указанного исполнителя для заявки. Для сброса персоны необходимо указать null в аргументах функции.

### Атрибуты:

Имя	Тип	Описание
personId	int null	id персоны-исполнителя
personQueueIdForLog	int null	id очереди записываемый в лог
personQueueTableForLog	string null	Название таблицы-связки для очередей записываемое в лог
assignAttemptForLog	int null	Номер попытки прогона одного и того же тикета по той же очереди записываемый в лог

### Возвращает:

**bool** — подтверждение назначения исполнителя.

# ticket.getAssignedPersonRejectsCount()

## Получить количество отказов от заявок

```
ticket.getAssignedPersonRejectsCount(int|null $personQueueId = null, string|null $personQueueTable = null): ?int
```

### Описание:

Возвращает количество отказов от заявок для текущего исполнителя.

### Атрибуты:

Имя	Тип	Описание
personQueueId	int null	id очереди
personQueueTable	string null	Название таблицы-связки для очередей

### Возвращает:

**int** — количество отказов от заявок в формате числа.

# ticket.setAssignedPersonRejectsCount()

## Установить количество отказов от заявок

```
ticket.setAssignedPersonRejectsCount(int $rejectsCount, int|null $personQueueId = null, string|null $personQueueTable = null)
```

### Описание:

Установить количество отказов от заявок для текущего исполнителя.

### Атрибуты:

Имя	Тип	Описание
rejectsCount	int	Количество отказов
personQueueId	int null	id очереди
personQueueTable	string null	Название таблицы-связки для очередей

### Возвращает:

**bool** — подтверждение успешной установки отказов.

# ticket.incAssignedPersonRejectsCount()

## Увеличить количество отказов от заявок

```
ticket.incAssignedPersonRejectsCount(int|null $personQueueId = null, string|null  
$personQueueTable = null)
```

### Описание:

Увеличить количество отказов от заявок для текущего исполнителя на 1. Метод не сбрасывает кол-во попыток, если значение будет больше чем указанное в боте максимальное количество отказов.

### Атрибуты:

Имя	Тип	Описание
personQueueId	int null	id очереди
personQueueTable	string null	Название таблицы-связки для очередей

### Возвращает:

**bool** — подтверждение успешного увеличения отказов.

# ticket.assignToNextPersonAndMoveToEnd()

Назначить заявку следующему исполнителю и перенести в конец очереди

```
ticket.assignToNextPersonAndMoveToEnd()
```

## Описание:

Назначает заявку следующему исполнителю и переносит его в конец очереди.

## Возвращает:

**bool** — подтверждение назначения заявки.

# ticket.getAllData()

## Получить значение всех данных заявки

```
ticket.getAllData()
```

### Описание:

Возвращает значение всех данных заявки.

### Возвращает:

**array** - массив данных заявки:

- **id** - id заявки;
- **business\_id** - id бизнеса;
- **bot\_id** - id бота;
- **author\_person\_id** - id автора заявки;
- **client\_person\_id** - id клиента заявки;
- **assigned\_person\_id** - id исполнителя заявки;
- **ticket\_type\_id** - id вида заявки, виды заявки создаются вручную;
- **payment\_type\_id** - id способа оплаты. Доступные значения:
  - **0** - PAYMENT\_TYPE\_NONE;
  - **1** - PAYMENT\_TYPE\_ROBOKASSA;
  - **2** - PAYMENT\_TYPE\_YANDEXKASSA;
  - **3** - PAYMENT\_TYPE\_CASH;
- **status\_id** - id статуса заявки;
- **title** - заголовок заявки;
- **description** - описание заявки;
- **created\_at** - время создания заявки;
- **updated\_at** - время обновления заявки;
- **deleted\_at** - время удаления заявки;
- все системные поля заявки (по английски).

# ticket.setData()

## Установить значение данных заявки по ключу

```
ticket.setData(string $key, $value)
```

### Описание:

Устанавливает значение данных заявки по ключу.

### Атрибуты:

Имя	Тип	Описание
key	string	Ключ наименование поля
value	mixed	Значение данных

В качестве ключа можно указать:

- **id** - id заявки;
- **business\_id** - id бизнеса;
- **bot\_id** - id бота;
- **author\_person\_id** - id автора заявки;
- **client\_person\_id** - id клиента заявки;
- **assigned\_person\_id** - id исполнителя заявки;
- **ticket\_type\_id** - id вида заявки, виды заявки создаются вручную;
- **payment\_type\_id** - id способа оплаты. Доступные значения:
  - **0** - PAYMENT\_TYPE\_NONE;
  - **1** - PAYMENT\_TYPE\_ROBOKASSA;
  - **2** - PAYMENT\_TYPE\_YANDEXKASSA;
  - **3** - PAYMENT\_TYPE\_CASH;
- **status\_id** - id статуса заявки;
- **title** - заголовок заявки;
- **description** - описание заявки;
- **created\_at** - время создания заявки;
- **updated\_at** - время обновления заявки;
- **deleted\_at** - время удаления заявки;

- все системные поля заявки (по английски).

Возвращает:

**self** - установленные данные.



# ticket.loadCustomDataIfNotLoaded()

## Загрузить кастомные данные

```
ticket.loadCustomDataIfNotLoaded(bool $isReloadModel = false)
```

### Описание:

Загружает кастомные данные, если они не загружены.

### Атрибуты:

Имя	Тип	Описание
isReloadModel	bool	

### Возвращает:

**bool** - подтверждение загрузки.

# ticket.getSubtotalFormatted( )

## Узнать форматированную промежуточную стоимость заказа

```
ticket.getSubtotalFormatted()
```

### Описание:

Возвращает форматированную промежуточную стоимость заказа.

### Возвращает:

**string** - промежуточная стоимость заказа.

# ticket.getTotalFeeFormatted() ( )

## Узнать форматированную стоимость заказа

```
ticket.getTotalFeeFormatted( )
```

### Описание:

Возвращает форматированную стоимость заказа.

### Возвращает:

**string** - стоимость заказа.

# ticket.getDeliveryFeeFormatted()

## Узнать форматированную стоимость доставки

```
ticket.getDeliveryFeeFormatted()
```

### Описание:

Возвращает форматированную стоимость доставки.

### Возвращает:

**string** - стоимость доставки.

# ticket.getTotalFormatted()

## Узнать форматированную итоговую стоимость заказа

```
ticket.getTotalFormatted()
```

### Описание:

Возвращает форматированную итоговую стоимость заказа.

### Возвращает:

**string** - итоговая стоимость заказа.

# ticket.getTicketLines()

Получить массив объектов  
ScriptTicketLine

```
ticket.getTicketLines()
```

## Описание:

Возвращает массив объектов ScriptTicketLine.

## Возвращает:

**ScriptTicketLine** - массив объектов.