

Monopoly One API

Версия 190107-1855

Взаимодействие с API

Запросы

Запросы к API строятся следующим образом:

Метод HTTP	GET, POST
URL	https://monopoly-one.com/api/<method_name>

Ответы сервер присылает в формате JSON. Схема ответа:

code	Код ответа: 0 — успешный запрос; иное — ошибка. <i>положительное число</i>
data	Параметры ответа метода API. <i>объект или массив</i>
description	Описание ошибки. <i>строка</i>

Авторизация

Авторизация при выполнении запроса обеспечивается передачей параметра **access_token**.

Чаще всего **access_token** ограничен по времени жизни и **всегда** привязан к IP-адресу, на который он был выдан.

Так же нас имеется система рефреш-токенов (параметр **refresh_token**), при помощи которых вы сможете получить новый **access_token**. Узнать подробнее о процедуре обновления токена вы можете в методе API **auth.refresh**.

Относительно авторизации методы API делятся на три типа:

обязательна	Метод не работает без авторизации.
не требуется	Метод работает и без авторизации, но при использовании авторизации меняет своё поведение (разрешает опускать параметры или возвращает расширенные данные).
не применяется	Метод не использует авторизацию в принципе.

Глобальные коды ошибок

1	Ошибка авторизации.
2	Не передан обязательный параметр или переданные параметры невалидны.
3	Внутренняя ошибка сервера.
5	Вызван неизвестный метод API.
6	Превышено число запрашиваемых объектов. <i>Максимальное число объектов, которые можно передать в один запрос — 50.</i>

7	Превышен лимит запросов.
8	Требуется капча.
10	Сервер не может обработать этот запрос прямо сейчас.
11	Требуется подтверждение. <i>Для обхода этой ошибки отправьте запрос с добавлением параметра force=1.</i>
12	Требуется ввод кода из аутентификатора. <i>Для обхода этой ошибки отправьте запрос с добавлением параметра totp_token и указанием кода из аутентификатора.</i>
99	Неизвестная ошибка.

Методы API: Авторизация

auth.signin

Авторизует пользователя в системе.

Обратите внимание, что в большинстве случаев вы не сможете автоматически пройти авторизацию, поскольку сервер будет отвечать кодом 8 «Требуется капча».

Авторизация	не применяется
-------------	----------------

Параметры запроса

Параметр	Описание
email	Адрес почты пользователя. <i>строка, обязательный параметр</i>
password	Пароль пользователя. <i>строка, обязательный параметр</i>

Параметры ответа

Параметры ответа различаются в зависимости от настроек аккаунта.

Если у пользователя настроена двухфакторная аутентификация, то ответ будет таким:

Параметр	Описание
totp_session_id	Идентификатор TOTP сессии. Сессия действительная в течение 5 минут. <i>строка</i>

Если двухфакторная аутентификация не настроена, то ответ будет таким:

Параметр	Описание
user_id	Идентификатор пользователя. <i>положительное число</i>
access_token	Новый ключ авторизации для дальнейших запросов. <i>строка</i>
expires	Время, через которое access_token станет недействителен. <i>положительное число</i>
refresh_token	Новый ключ обновления авторизации. <i>строка</i>

Коды ошибок

412	Аккаунт глобально заблокирован.
-----	---------------------------------

auth.totpVerify

Завершает авторизацию, прерванную двухфакторной аутентификацией.

Авторизация	не применяется
-------------	----------------

Параметры запроса

Параметр	Описание
totp_session_id	Идентификатор TOTP сессии. <i>строка, обязательный параметр</i>
token	Код двухфакторной аутентификации. <i>строка, обязательный параметр</i>

Параметры ответа

Параметр	Описание
user_id	Идентификатор пользователя. <i>положительное число</i>
access_token	Новый ключ авторизации для дальнейших запросов. <i>строка</i>
expires	Время, через которое access_token станет недействителен. <i>положительное число</i>
refresh_token	Новый ключ обновления авторизации. <i>строка</i>

Коды ошибок

412	Аккаунт глобально заблокирован.
413	Неверный код.
414	Слишком частая TOTP авторизация.

auth.refresh

Обменивает `refresh_token` на новый `access_token`.

Авторизация	не применяется
-------------	----------------

Параметры запроса

Параметр	Описание
<code>refresh_token</code>	Ключ обновления авторизации. <i>строка, обязательный параметр</i>

Параметры ответа

Параметр	Описание
<code>user_id</code>	Идентификатор пользователя. <i>положительное число</i>
<code>access_token</code>	Новый ключ авторизации для дальнейших запросов. <i>строка</i>
<code>expires</code>	Время, через которое <code>access_token</code> станет недействителен. <i>положительное число</i>
<code>refresh_token</code>	Новый ключ обновления авторизации. <i>строка</i>

bots.getToken

Создаёт новый токен для бота. При этом отзываются все предыдущие токены, выданные этому боту.

Авторизация	обязательна
-------------	-------------

Параметры запроса

Параметр	Описание
user_id	Идентификатор пользователя-бота. <i>положительное число, обязательный параметр</i>
ip	IPv4, к которому будет привязан токен. <i>строка, по умолчанию — текущий IP</i>

Параметры ответа

Параметр	Описание
user_id	Идентификатор пользователя-бота. <i>положительное число</i>
access_token	Новый ключ авторизации для дальнейших запросов. <i>строка</i>
expires	Время, через которое access_token станет недействителен. Всегда равен 0 (вечный). <i>положительное число</i>

Коды ошибок

1801	Бота не существует.
------	---------------------

Примечания

Метод вызывается с токеном пользователя, который владеет ботом.

Методы API: Пользователи

users.get

Возвращает информацию о пользователях.

Авторизация	не требуется
-------------	--------------

Параметры запроса

Параметр	Описание
user_id	Идентификатор (или домен) пользователя. <i>число или строка</i>
user_ids	Идентификаторы (или домены) пользователей через запятую. <i>числа или строки через запятую</i>
type	Тип информации о пользователе: short — короткая другое значение — полная <i>строка</i>

Параметры ответа

Не применяются. API возвращает в параметре **data** массив с данными о пользователях (объекты **user**).

Примечания

Если параметры **user_id** и **user_ids** не переданы, то в случае передачи **access_token** будет возвращена информация о текущем пользователе.

Методы API: Общий чат

gchat.get

Возвращает сообщения из общего чата.

Авторизация	не требуется
-------------	--------------

Параметры ответа

Параметр	Описание
messages	Сообщения в общем чате в порядке их появления. <i>массив объектов <code>gchat_message</code></i>
users_info	Данные обо всех пользователях, упомянутых в ответе API в коротком формате. <i>массив объектов <code>user</code></i>

gchat.send

Отправляет сообщение в общий чат.

Авторизация	обязательна
-------------	-------------

Параметры запроса

Параметр	Описание
message	Текст сообщения. <i>строка, обязательный параметр</i>
is_public	1 — сообщение будет публичным. Данный параметр доступен только ботам. <i>положительное число, по умолчанию — 0</i>

Параметры ответа

Параметр	Описание
msg_id	Идентификатор нового сообщения в чате. <i>строка</i>

Коды ошибок

104	На пользователя наложен мут. В поле ответа <code>\$.data.muted_until</code> будет указана метка времени, когда мут будет снят.
1203	Сообщение не отправлено из-за ограничений в чате. В поле ответа <code>\$.data.type</code> может быть указана причина отказа: DISABLED — чат закрыт для приёма сообщений; STRICT — чат в строгом режиме, и сообщение не прошло фильтрацию; EMOTES_ONLY — чат в режиме «только смайлы»; BOT_LIMITATIONS — бот превысил какой-либо лимит (чаще всего это лимит публичных сообщений).
1204	Команда принята, но сообщение не создано.

Методы API: Инвентарь

inventory.get

Возвращает список предметов в инвентаре пользователя.

Авторизация	не требуется
-------------	--------------

Параметры запроса

Параметр	Описание
user_id	Идентификатор или домен пользователя. <i>положительное число или строка</i> <i>обязательный параметр (если нет авторизации)</i>
include_stock	1 — добавить в ответ предметы стокового класса. <i>положительное число, по умолчанию 0</i>
add_user_info	1 — добавить информацию о пользователе. <i>положительное число, по умолчанию 0</i>
order	Сортировка предметов: time — по времени получения; quality — по классу. <i>строка, по умолчанию time</i>
offset	Отступ от начала списка. <i>положительное число, по умолчанию 0</i>
count	Количество предметов, которые надо получить. <i>положительное число, по умолчанию 10</i>

Параметры ответа

Параметр	Описание
count	Количество предметов в инвентаре пользователя. <i>положительное число</i>
things	Предметы в инвентаре. <i>массив объектов item</i>
thing_types	Описания типов предметов из параметра things . <i>массив объектов item_type</i>
qualities	Описания классов предметов из параметра things . <i>массив объектов item_quality</i>
collections	Описания коллекций предметов из параметра things . <i>массив объектов item_collection</i>
user_info	Данные о пользователе. <i>объект user</i>

Коды ошибок

101	Пользователя не существует.
------------	-----------------------------

Методы API: Обмены

trades.getIncome

Возвращает список предложений обмена, отправленных текущему пользователю.

Авторизация	обязательна
-------------	-------------

Параметры запроса

Параметр	Описание
offset	Отступ от начала списка. <i>положительное число, по умолчанию 0</i>
count	Количество предложений обмена, которые надо получить. <i>положительное число, по умолчанию 10</i>

Параметры ответа

Параметр	Описание
trades	Предложения обмена, направленные пользователю. <i>массив объектов trade</i>
users_data	Данные обо всех пользователях, упомянутых в ответе API. <i>массив объектов user</i>
thing_types	Описания типов предметов, упомянутых в возвращённых предложениях обмена. <i>массив объектов item_type</i>
qualities	Описания классов предметов, упомянутых в возвращённых предложениях обмена. <i>массив объектов item_quality</i>
collections	Описания коллекций предметов, упомянутых в возвращённых предложениях обмена. <i>массив объектов item_collection</i>

trades.getById

Возвращает информацию о предложении обмена, направленном пользователю.

Авторизация	обязательна
-------------	-------------

Параметры запроса

Параметр	Описание
trade_id	Идентификатор предложения обмена <i>положительное число</i>

Параметры ответа

Параметр	Описание
trade	Предложение обмена. <i>объект trade</i>
users_data	Данные обо всех пользователях, упомянутых в ответе API. <i>массив объектов user</i>
thing_types	Описания типов предметов, упомянутых в возвращённом предложении обмена. <i>массив объектов item_type</i>
qualities	Описания классов предметов, упомянутых в возвращённом предложении обмена. <i>массив объектов item_quality</i>
collections	Описания коллекций предметов, упомянутых в возвращённом предложении обмена. <i>массив объектов item_collection</i>

Коды ошибок

611	Предложения обмена не существует.
-----	-----------------------------------

trades.getOutbound

Возвращает список предложений обмена, отправленных текущим пользователем.

Авторизация	обязательна
-------------	-------------

Параметры запроса

Параметр	Описание
offset	Отступ от начала списка. <i>положительное число, по умолчанию 0</i>
count	Количество предложений обмена, которые надо получить. <i>положительное число, по умолчанию 10</i>

Параметры ответа

Параметр	Описание
trades	Предложения обмена, отправленные пользователем. <i>массив объектов trade</i>
users_data	Данные обо всех пользователях, упомянутых в ответе API. <i>массив объектов user</i>
thing_types	Описания типов предметов, упомянутых в возвращённых предложениях обмена. <i>массив объектов item_type</i>
qualities	Описания классов предметов, упомянутых в возвращённых предложениях обмена. <i>массив объектов item_quality</i>
collections	Описания коллекций предметов, упомянутых в возвращённых предложениях обмена. <i>массив объектов item_collection</i>

trades.history

Возвращает историю предложений обмена текущего пользователя.

Авторизация	обязательна
-------------	-------------

Параметры запроса

Параметр	Описание
offset	Отступ от начала списка. <i>положительное число, по умолчанию 0</i>
count	Количество предложений обмена, которые надо получить. <i>положительное число, по умолчанию 10</i>

Параметры ответа

Параметр	Описание
trades	Предложения обмена. <i>массив объектов trade</i>
users_data	Данные обо всех пользователях, упомянутых в ответе API. <i>массив объектов user</i>
thing_types	Описания типов предметов, упомянутых в возвращённых предложениях обмена. <i>массив объектов item_type</i>
qualities	Описания классов предметов, упомянутых в возвращённых предложениях обмена. <i>массив объектов item_quality</i>
collections	Описания коллекций предметов, упомянутых в возвращённых предложениях обмена. <i>массив объектов item_collection</i>

trades.historyWith

Возвращает историю предложений обмена между текущим пользователем и указанным пользователем.

Авторизация	обязательна
-------------	-------------

Параметры запроса

Параметр	Описание
user_id	Идентификатор пользователя, историю обменов с которым требуется получить. <i>положительное число</i>
offset	Отступ от начала списка. <i>положительное число, по умолчанию 0</i>
count	Количество предложений обмена, которые надо получить. <i>положительное число, по умолчанию 10</i>

Параметры ответа

Параметр	Описание
trades	Предложения обмена. <i>массив объектов trade</i>
users_data	Данные обо всех пользователях, упомянутых в ответе API. <i>массив объектов user</i>
thing_types	Описания типов предметов, упомянутых в возвращённых предложениях обмена. <i>массив объектов item_type</i>
qualities	Описания классов предметов, упомянутых в возвращённых предложениях обмена. <i>массив объектов item_quality</i>
collections	Описания коллекций предметов, упомянутых в возвращённых предложениях обмена. <i>массив объектов item_collection</i>

trades.create

Создаёт новое предложение обмена.

Авторизация	обязательна
-------------	-------------

Параметры запроса

Параметр	Описание
user_id	Идентификатор пользователя, которому отправляется предложение обмена. <i>положительное число, обязательный параметр</i>
thing_ids_from	Идентификаторы предметов, которые текущий пользователь отдаёт. <i>положительные числа через запятую максимум 35 элементов</i>
thing_ids_to	Идентификаторы предметов, которые текущий пользователь запрашивает. <i>положительные числа через запятую максимум 35 элементов</i>

Параметры ответа

Параметр	Описание
trade_id	Идентификатор созданного обмена. <i>положительное число</i>

Коды ошибок

107	Текущий пользователь находится в чёрном списке у пользователя, которому отправляется обмен.
108	Текущий пользователь не подходит под настройки приватности пользователя, которому отправляется обмен.
191	У текущего пользователя не подтверждена электронная почта.
601	Один или несколько предметов не могут быть отправлены в обмен по каким-либо причинам.
612	Превышен лимит предметов в обмене.
617	Внутренняя ошибка.
619	У текущего пользователя имеется активная блокировка обменов.
620	У пользователя, которому отправляется обмен, имеется активная блокировка обменов.
703	Ошибка безопасности.

trades.accept

Принимает предложение обмена.

Авторизация	обязательна
-------------	-------------

Параметры запроса

Параметр	Описание
trade_id	Идентификатор обмена. <i>положительное число</i>

Коды ошибок

191	У текущего пользователя не подтверждена электронная почта.
611	Предложения обмена не существует.
617	Внутренняя ошибка.
703	Ошибка безопасности.

trades.decline

Отклоняет предложение обмена.

Авторизация	обязательна
-------------	-------------

Параметры запроса

Параметр	Описание
trade_id	Идентификатор обмена. <i>положительное число</i>

Коды ошибок

611	Предложения обмена не существует.
-----	-----------------------------------

trades.cancel

Отменяет отправленное предложение обмена.

Авторизация	обязательна
-------------	-------------

Параметры запроса

Параметр	Описание
trade_id	Идентификатор обмена. <i>положительное число</i>

Коды ошибок

611	Предложения обмена не существует.
-----	-----------------------------------

Методы API: Данные

data.getThingPrototypes

Возвращает прототипы предметов.

Авторизация	не применяется
-------------	----------------

Параметры запроса

Параметр	Описание
thing_prototype_ids	Идентификаторы прототипов предметов. <i>положительные числа через запятую</i>

Параметры ответа

Параметр	Описание
thing_prototypes	Прототипы предметов. <i>массив объектов item</i>
thing_types	Описания типов предметов из параметра thing_prototypes . <i>массив объектов item_type</i>
qualities	Описания классов предметов из параметра thing_prototypes . <i>массив объектов item_quality</i>
collections	Описания коллекций предметов из параметра thing_prototypes . <i>массив объектов item_collection</i>

Коды ошибок

611	Предложения обмена не существует.
-----	-----------------------------------

Стандартные объекты

Стандартные объекты — это определённый формат, который стандартизирован между методами API.

Объект `user`

Параметры, которые входят в тип `short`:

Параметр	Описание
<code>user_id</code>	Идентификатор пользователя. <i>положительное число</i>
<code>domain</code>	Домен пользователя. <i>строка</i>
<code>approved</code>	1 — страница пользователя подтверждена (галочка в профиле). <i>положительное число</i>
<code>nick</code>	Никнейм пользователя. <i>строка</i>
<code>gender</code>	Пол игрока: 0 — мужской, 1 — женский. <i>положительное число</i>
<code>avatar</code>	Ссылка на аватар пользователя. <i>строка</i>
<code>online</code>	1 — пользователь онлайн. <i>положительное число</i>
<code>current_game</code>	Текущая игра пользователя. <i>объект <code>current_game</code></i>
<code>rank</code>	Звание игрока. <i>объект <code>rank</code></i>
<code>vip</code>	1 — пользователь имеет VIP статус. <i>положительное число</i>
<code>bot</code>	1 — пользователь является ботом. <i>положительное число</i>
<code>bot_owner</code>	Идентификатор пользователя-владельца бота. <i>положительное число</i>
<code>moderator</code>	1 — пользователь является модератором. <i>положительное число</i>

Параметры, которые входят в обычную информацию:

Параметр	Описание
<code>nicks_old</code>	Прошлые никнеймы пользователя. <i>массив строк</i>
<code>profile_cover</code>	Ссылка на фон профиля. <i>строка</i>
<code>social_vk</code>	Идентификатор пользователя во Вконтакте. <i>положительное число</i>
<code>games</code>	Количество сыгранных матчей. <i>положительное число</i>

Параметр	Описание
games_wins	Количество побед в матчах. <i>положительное число</i>
xp	Количество опыта пользователя. <i>положительное число</i>
xp_level	Уровень пользователя. <i>положительное число</i>
badge	Значок в профиле пользователя. <i>объект item</i>
friendship	Статус отношений (дружбы или чёрного списка) между текущим пользователем и запрашиваемым пользователем: 0 — дружбы нет; 1 — дружба активна; 2 — текущий пользователь отправил запрос дружбы запрашиваемому пользователю (исходящий запрос); 3 — запрашиваемый пользователь отправил запрос дружбы текущему пользователю (входящий запрос); 4 — текущий пользователь добавил запрашиваемого пользователя в чёрный список; 5 — запрашиваемый пользователь добавил текущего пользователя в чёрный список; 6 — оба пользователя добавили друг друга в чёрный список. <i>положительное число</i>
muted	1 — пользователь имеет мут. <i>положительное число</i>
mfp_ban_history	История блокировок MFP. <i>объект</i>

Параметры, которые возвращаются только при запросе текущего пользователя:

Параметр	Описание
muted_until	Время, когда мут закончится. <i>unixtime</i>

Объект `gchat_message`

Параметр	Описание
<code>msg_id</code>	Идентификатор сообщения. <i>положительное число</i>
<code>ts</code>	Время создания сообщения. <i>unixtime</i>
<code>type</code>	Тип сообщения: 1 — обычное сообщение; 2 — уведомление о выпавшем предмете. <i>положительное число</i>
<code>user_id</code>	Идентификатор пользователя-автора сообщения. <i>положительное число</i>

Следующие параметры представлены в объекте только если `type` равен **1** (текстовое сообщение):

Параметр	Описание
<code>is_public</code>	1 — сообщение является публичным. <i>положительное число</i>
<code>text</code>	Текст сообщения. <i>строка</i>
<code>users_mentions</code>	Список пользователей, упомянутых в сообщении. <i>массив положительных чисел</i>

Следующие параметры представлены в объекте только если `type` равен **2** (уведомление о выпавшем предмете):

Параметр	Описание
<code>case</code>	Информация о кейсе. <i>объект</i>
<code>drop</code>	Информация о выпавшем предмете. <i>объект</i>
<code>info</code>	Дополнительная информация о кейсе и предмете. <i>объект</i>

Объект `item`

Строго говоря, данные каждого предмета состоят из двух частей:

1. параметров **прототипа** (`thing_prototype`), которые равны для всех вещей одного прототипа (например, все кубики «Дракон» имеют одинаковое название и эксклюзивный класс);
2. параметров **предмета** (`thing`), уникальных именно для него (владелец, время получения или автограф).

Параметры прототипа (возвращаются, если никакой конкретно предмет не имеется в виду):

Параметр	Описание
<code>thing_prototype_id</code>	Идентификатор прототипа предмета. <i>положительное число</i>
<code>thing_prototype_status</code>	Статус прототипа предмета: 0 — обычный предмет; 1 — недоступен для продажи и передачи. <i>положительное число</i>
<code>thing_type</code>	Тип предмета: 0 — карточка; 1 — подарочный набор; 2 — кейс; 3 — ключ; 4 — генератор чисел; 5 — значок; 6 — пропуск (одноразовый предмет); 7 — подписка (многократный предмет); 8 — насмешка; 9 — стикер. <i>положительное число</i>
<code>image</code>	Ссылка на картинку прототипа предмета. <i>строка</i>
<code>title</code>	Название. <i>строка</i>
<code>description</code>	Описание предмета <i>текст</i>
<code>group</code>	Монополия, к которой относится карточка. <i>положительное число; существует только при <code>thing_type=0</code></i>
<code>quality</code>	Класс прототипа предмета: 0 — стоковый (серый); 1 — обыкновенный (голубой); 2 — стандартный (синий); 3 — особенный (фиолетовый); 4 — высочайший (красный); 5 — эксклюзивный (золотой). <i>положительное число</i>
<code>collection</code>	Идентификатор коллекции, к которой относится прототип предмета либо -1 , если предмет не относится ни к какой коллекции. <i>число</i>
<code>twin_thing_prototype_id</code>	Идентификатор прототипов предметов, которые нельзя установить (<code>equip</code>) вместе с текущим прототипом (например, обыкновенную карту и карту с копилкой одного бренда нельзя вместе поставить на поле). <i>положительное число или массив положительных чисел</i>

Параметр	Описание
delete_price	Цена продажи предмета в систему в копейках. <i>положительное число</i>
can_be_upgraded	Можно ли предмет отправить в крафт. <i>положительное число</i>
buy_cost	Цена покупки предмета у системы в копейках. <i>положительное число</i>
key	Идентификатор прототипа ключа, которым можно открыть данный кейс. <i>положительное число; существует только при thing_type=2</i>
cases	Идентификаторы прототипов кейсов, которые можно открыть этим ключом. <i>массив положительных чисел; существует только при thing_type=3</i>
drop	Список прототипов предметов, которые падают из этого набора или кейса. <i>массив объектов; существует только при thing_type=2 или thing_type=3</i>

Параметры самого предмета:

Параметр	Описание
thing_id	Идентификатор предмета. <i>положительное число</i>
user_id	Идентификатор пользователя — владельца предмета. <i>положительное число</i>
replacement	Идентификатор слота, в который установлен (equip) предмет: 0..99 — слот на поле для карточек; 100 — для кубиков; 200 — для значка профиля; 300 — для насмешки. <i>положительное число</i>
owned_time	Время получения предмета. <i>unixtime</i>
can_give	Время, с которого предмет можно будет передавать в предложения обмена либо -1 , если предмет нельзя передавать бессрочно. <i>unixtime либо -1</i>
can_sell	Время, с которого предмет можно будет продавать (в систему или на маркете) либо -1 , если предмет нельзя продавать бессрочно. <i>unixtime либо -1</i>
souvenir	Сувенирная надпись на предмете. <i>строка</i>
autograph	Автограф. Содержит поля: user_id — идентификатор пользователя — автора автографа; text — текст автографа. <i>объект</i>
moneybox	Данные о копилке на предмете. Если это карточка, то содержит поля: transactions — количество попаданий соперников на карточку; money_inside — сумма собранной аренды. Если это насмешка, то содержит поля: count — количество обанкроченных соперников. <i>объект</i>

Параметр	Описание
uses_left	Количество раз, которые предмет можно использовать. <i>положительное число, существует только при thing_type=7</i>
uses_origin	Количество раз, которые предмет можно было использовать изначально. <i>положительное число, существует только при thing_type=7</i>

Объект `trade`

Параметр	Описание
<code>trade_id</code>	Идентификатор предложения обмена. <i>положительное число</i>
<code>status</code>	Статус предложения обмена: 0 — отправлен; 1 — принят; 2 — отклонён; 3 — отозван; 4 — отменён системой. <i>положительное число</i>
<code>user_id</code>	Идентификатор пользователя, отправившего предложение обмена. <i>положительное число</i>
<code>thing_ids_from</code>	Предметы, которые автор предложение обмена предложил. <i>массив объектов <code>item</code></i>
<code>thing_ids_to</code>	Предметы, которые автор предложение обмена запросил. <i>массив объектов <code>item</code></i>
<code>create_time</code>	Время создания предложения обмена. <i>unixtime</i>
<code>reaction_time</code>	Время ответа на предложения обмена. <i>unixtime</i>

Боты

Боты — особые пользователи, которые управляются автоматически. Для них предусмотрена более удобная авторизация и значок в общем чате. Пример профиля бота можно посмотреть [здесь](#).

Ограничения методов API

У ботов есть ограничения на вызовы методов API. Кроме открытых методов (где авторизация не требуется), боты имеют право вызывать следующие методы API:

friends.add	friends.delete	friends.getRequests	gchat.send
inventory.exchange	trades.accept	trades.cancel	trades.create
trades.decline	trades.getIncome	trades.getOutbound	trades.history

Ограничения в общем чате

Боты могут писать в общий чат два типа сообщений:

1. «персональные»: такие сообщения увидят только те пользователи, которые упомянуты в сообщении. Кроме этого, такие сообщения не сохраняются в логе общего чата, а только рассылаются через вебсокеты. Обратите внимание, что «персональные» сообщения *в общем чате* никак не закрыты приватностью и доступны для чтения всем желающим через API или консоль браузера. **Не передавайте таким образом никакую приватную информацию!**
2. «публичные»: такие сообщения сохраняются в логе общего чата, и их видят все желающие. Бот может отправить только пять «публичных» сообщений за час, и при этом все такие сообщения должны быть разными.

При вызове метода `gchat.send` сервер API считает сообщение «публичным», когда значение параметра `is_public` установлено в `1` либо в сообщении нет ни одного упоминания пользователей.

Обратите внимание, что сообщения пользователей, начинающиеся с `!`, не сохраняются в логе общего чата, как и «персональные» сообщения от бота. Информацию о таких сообщениях можно получить только через вебсокет.

Создание бота

Создать профиль бота автоматически сейчас невозможно. Получить бота сейчас можно следующим образом:

1. зарегистрируйте обычный аккаунт, чтобы использовать его как бота;
2. создайте функционал бота, авторизуясь при помощи метода `auth.signin`;
3. напишите в [личные сообщения Monopoly One](#) или [на почту](#) с просьбой выдать аккаунту вашего бота значок (сменить тип аккаунта с «человеческого» на «бота»).

Если ваш бот имеет хороший и интересный функционал, то вашему боту будет выдан значок.

Авторизация бота с «человеческим» аккаунтом на вашем сервере

Пока ваш бот не имеет значка, вы не можете использовать метод `bots.getToken`. Однако скорее всего, вы не сможете авторизовать бота и при помощи метода `auth.signIn`, так как сервер API может отвечать кодом 8 «Требуется капча». В этом случае вы можете использовать следующий порядок действий:

1. авторизоваться в аккаунте бота через браузер при помощи логина и пароля;
2. пройти при этом капчу;
3. взять из браузера рефреш-токен (в консоли браузера `localStorage.refresh_token`) и перенести его на сервер;
4. вызвать на сервере метод `auth.refresh` и получить аксес-токен, привязанный к айпи вашего сервера.

После этого ваш бот сможет слать запросы с сервера. **Не забудьте** вставить в свой код функцию, которая будет обновлять аксес-токен вызовом метода `auth.refresh` каждый раз, когда сервер API вернёт код 1 «Требуется авторизация».