

Актуальная версия документации

Актуальную версию данной документации всегда можно найти по следующей ссылке:

<https://mostbetshop.com/api-documentation.pdf>

FAQ по использованию API для агентов Mostbet Cash

Как получить доступ к API и API-ключи (Key и Secret)?

Чтобы начать работать с API Mostbet Cash, необходимо получить уникальные ключи доступа – **API Key** и **Secret**. Порядок действий зависит от того, зарегистрированы ли вы уже как кассир:

Если вы ещё не зарегистрированы как кассир:

1. Выберите способ регистрации:

Вы можете пройти регистрацию одним из следующих способов:

- a. Через мобильное приложение кассира Mostbet Cash.
- b. Через сайт, заполнив форму на: <https://mostbetshop.com>
- c. В веб-версии приложения на <https://mostbetshops.com>

2. Пройдите верификацию своего аккаунта:

Независимо от способа регистрации, необходимо пройти верификацию — это обязательное условие для работы через API.

Без подтверждённой учётной записи доступ к API невозможен.

3. Если возникли вопросы:

Ознакомьтесь с разделом **FAQ** на сайте проекта — <https://mostbetshop.com/en/faq/>
Там подробно описан процесс регистрации, загрузки документов и активации.

Также вы всегда можете обратиться в поддержку: info@mostbetshop.com

Если вы уже зарегистрированы и верифицированы:

Перейдите к созданию ключей доступа к API:

1. **Создание API-ключа:** После верификации зайдите в раздел «Мои кассы» в личном кабинете. Выберите нужную кассу, для которой хотите использовать

API, и в дополнительном меню найдите опцию «Создать API-ключ». Нажмите эту кнопку.

2. **Подтверждение и получение ключей:** На открывшейся странице подтвердите создание API-ключа. **Важно:** ключи (API Key и Secret) будут показаны **только один раз** – обязательно сразу сохраните их. Повторно посмотреть или получить эти значения позже будет невозможно.
3. **Сохранение API Key и Secret:** Скопируйте оба значения. **API Key** представляет собой уникальный идентификатор кассы (например: `api-key:3d83ac24-7fd2-498d-84b4-f2a7e80401fb`) и передаётся в заголовке каждого запроса для идентификации кассы. **Secret** – секретный ключ, используемый **только** для формирования цифровой подписи запросов; **Secret не отправляется в самих запросах.**

После получения ключей вы готовы вызывать методы API по базовому URL:

<https://apimb.com/mbc/gateway/v1/api/cashpoint/>, подставляя идентификатор вашей кассы (Cashpoint ID) и необходимый метод (см. ниже). Все запросы должны выполняться по HTTPS и подписываться с помощью HMAC SHA3-256 для безопасности.

Как сформировать подпись запроса (HMAC SHA3-256)?

Для каждого запроса к API необходимо вычислять **цифровую подпись** (передаётся в заголовке **X-Signature**). Подпись рассчитывается алгоритмом HMAC SHA3-256 с использованием вашего Secret. Алгоритм формирования подписи такой:

1. **Составление строки для подписи:** Конкатенируйте (соедините) следующие элементы в одну строку **без разделителей** в строгом порядке:
<API_KEY><PATH><REQUEST_BODY><TIMESTAMP>.
 - **<API_KEY>** – ваш API-ключ в том же виде, как он используется в заголовке (например, `api-key:3d83ac24-7fd2-498d-84b4-f2a7e80401fb`).
 - **<PATH>** – путь запроса (URI) начиная с `/mbc/...`, как в документации. Например: `/mbc/gateway/v1/api/cashpoint/48436/player/deposit`.
 - **<REQUEST_BODY>** – тело запроса в формате JSON (без пробелов и переводов строк) **или пустая строка**, если тело отсутствует (например, для GET-запросов). Пример тела для депозита:
`{"brandId":1,"playerId":"774","amount":200,"currency":"RUB"}`

- **<TIMESTAMP>** – значение метки времени, используемое в заголовке **X-Timestamp** (формат: **YYYY-MM-DD hh:mm:ss**, пример: **2025-05-13 10:19:38**). Дату и время необходимо передавать для часового пояса UTC+0.

Пример: для запроса депозита (пополнения) с API-ключом **api-key:3d83ac24-7fd2-498d-84b4-f2a7e80401fb**, путем **/mbc/gateway/v1/api/cashpoint/48436/player/deposit**, телом **{"brandId":1,"playerId":"774","amount":200,"currency":"RUB"}** и меткой времени **2025-05-13 10:19:38** сформированная строка будет:

None

```
api-key:3d83ac24-7fd2-498d-84b4-f2a7e80401fb/mbc/gateway/v1/api/cashpoint/48436/player/deposit{"brandId":1,"playerId":"774","amount":200,"currency":"RUB"}2025-05-13 10:19:38
```

2. **Вычисление HMAC:** На полученную строку примените криптографическую функцию HMAC SHA3-256 с использованием вашего Secret в качестве ключа. В результате получится 256-битная (64-хзначная в шестнадцатеричной форме) подпись. Например, используя Secret **baa104d1-73a6-4914-866a-ddbbe0aae11a** и строку из примера выше, получится подпись:

None

```
da7b80330f5a7610239643a154c8d2e65b27008d71e38c1f717c7e1bbfe5b115
```

3. **Добавление подписи в заголовки:** Полученную строку подписи включайте в каждый запрос в заголовок **X-Signature**. Кроме того, каждый запрос должен содержать заголовок **X-API-Key** (ваш API Key) и **X-Timestamp** (метка времени, которую вы использовали при формировании подписи). Эти три заголовка обязательны для всех методов API. В случае POST-запросов требуется также заголовок **X-Project** (о нём ниже) и тело запроса (JSON) с нужными параметрами.

Важно: Часы клиента должны быть синхронизированы, так как сервер может отклонить запрос, если **X-Timestamp** значительно отличается от текущего времени. Все взаимодействие с API происходит по защищённому протоколу HTTPS, а подпись HMAC обеспечивает целостность и аутентичность запроса.

Какие HTTP-заголовки и структура тела необходимы для запросов к API?

Обязательные заголовки: Каждый запрос к API должен содержать следующие HTTP-заголовки для аутентификации и идентификации:

- **X-API-Key:** ваш API-ключ кассы. Указывается вместе с префиксом **api-key:** перед самим UUID ключа (пример значения: **api-key:3d83ac24-7fd2-498d-84b4-f2a7e80401fb**). Этот заголовок идентифицирует кассу, от имени которой выполняется запрос.
- **X-Timestamp:** временная метка запроса в формате **YYYY-MM-DD HH:MM:SS** (например, **2025-05-12 09:14:38**), необходимо передавать значение по UTC+0. Метка времени включается в процесс формирования цифровой подписи и передается в этом заголовке.
- **X-Signature:** цифровая подпись запроса, рассчитанная по алгоритму HMAC SHA3-256 с использованием Secret (как описано выше). Сервер с помощью этого значения проверяет целостность и подлинность запроса.
- **X-Project:** наименование проекта, от имени которого выполняется запрос. Требуется для некоторых методов, в основном для операций с игровыми счетами, таких как пополнение или вывод. **Для всех операций в рамках проекта Mostbet Cash значение должно быть строго MBC.** В **GET**-запросах, которые просто получают информацию (например, проверка баланса), **X-Project** не нужен; в запросах на вывод списка заявок он считается необязательным.

Помимо этих заголовков, в **POST** запросах необходимо указывать **тело запроса** в формате JSON, содержащее параметры, необходимые для выполнения операции. Структура тела зависит от метода API (подробнее в описаниях конкретных методов ниже). Например, для пополнения счета требуется указать сумму, идентификаторы бренда и игрока, валюту и т.д.. В **GET** запросах тело запроса обычно не используется (должно быть пустым), но могут требоваться определенные **query-параметры** в URL (например, для фильтрации результатов). Ниже в описаниях методов указано, какие параметры передаются через URL.

Идентификатор кассы (cashpointId): Обратите внимание, что в URL каждого метода присутствует `{cashpointId}` – это идентификатор вашей кассы, полученный при регистрации кассы в системе. Его нужно подставлять в путь запроса вручную. Например, если ID кассы = 48436, то вызов баланса будет `.../cashpoint/48436/balance`.

Формат обмена данными: API ожидает и возвращает данные в формате JSON. В заголовке `Accept` в примерах используется `*/*` – это означает, что вы запрашиваете ответ в JSON (по умолчанию API возвращает JSON).

Ниже приведены подробные сведения по каждому доступному методу API, включая требуемые параметры, примеры запросов и форматы ответов.

Как проверить баланс кассы через API?

Метод: `GET /cashpoint/{cashpointId}/balance` – позволяет узнать текущий баланс выбранной кассы (точки выплат) Mostbet Cash. Здесь `{cashpointId}` – идентификатор вашей кассы.

Обязательные заголовки: Для вызова баланса необходимо указать:

- `X-API-Key` – ваш API-ключ кассы (с префиксом `api-key:`),
- `X-Timestamp` – текущее время запроса,
- `X-Signature` – подпись HMAC SHA3-256 для данного запроса.

Пример запроса на проверку баланса (curl):

```
Shell
curl --location
'https://apimb.com/mbc/gateway/v1/api/cashpoint/48436/balance' \
--header 'X-Timestamp: 2025-05-12 09:30:02' \
--header 'X-Signature:
e23d94be4e334a83a3f429bad28b5af5f50b45cd0bf59176c719bac9cfafcb6a'
\
--header 'X-API-Key:
api-key:3d83ac24-7fd2-498d-84b4-f2a7e80401fb'
```

Успешный ответ (200 OK): В случае успеха сервер вернёт JSON с полями:

JSON

```
{
  "balance": 4234358929299.00,
  "currency": "RUB"
}
```

где **balance** – текущий баланс кассы (число с плавающей запятой), **currency** – код валюты баланса (строка, обычно буквенный код по ISO 4217, например "RUB").

Ошибка (Unauthorized): Если аутентификация не удалась (например, неверная подпись или ключ), вернётся ошибка **401 Unauthorized** с телом:

JSON

```
{
  "code": "<код_ошибки>",
  "message": "<описание_ошибки>",
  "additional_data": {}
}
```

Поле **code** содержит символьный код ошибки, **message** – человекочитаемое описание причины, **additional_data** – дополнительная техническая информация (если есть). Например, при неверном API-ключе или подписи будет указан соответствующий код и сообщение об ошибке.

Как пополнить счет игрока через API?

Метод: **POST /cashpoint/{cashpointId}/player/deposit** – создаёт транзакцию пополнения средств на счёт игрока (депозит через кассу). Используется, когда агент (кассир) принимает наличные от игрока и зачисляет их на игровой счёт игрока.

Заголовки: Для выполнения пополнения требуется передать следующие заголовки:

- **X-Api-Key** – API-ключ вашей кассы,
- **X-Timestamp** – метка времени запроса,
- **X-Signature** – HMAC SHA3-256 подпись запроса,

- **X-Project** – проект, от имени которого выполняется пополнение. Значение: **MBC** (указывается обязательно).

Тело запроса (JSON): В теле POST-запроса нужно передать JSON с параметрами транзакции:

```
JSON
{
  "brandId": <number>,
  "playerId": "<string>",
  "amount": <number>,
  "currency": "<string>"
}
```

Где:

- **brandId** – ID бренда, в рамках которого зарегистрирован счёт игрока. **При работе в рамках проекта Mostbet Cash (X-Project: MBC) допустимо использовать только значение brandId: 1**, что соответствует бренду **Mostbet**. Указание других значений приведёт к ошибке или игнорированию запроса.
- **playerId** – идентификатор игрока в системе (строка или число, соответствующее аккаунту игрока, который пополняется). Этот ID игрок узнаёт у себя в профиле или предоставляет вам.
- **amount** – сумма пополнения (число).
- **currency** – код валюты в формате ISO 4217 (три буквы, например **"RUB"** для российских рублей).

Успешный ответ (200 OK): Если запрос принят, API вернёт JSON с информацией о созданной транзакции пополнения:

```
JSON
{
  "transactionId": 11839895,
  "status": "NEW"
}
```

```
}
```

- **transactionId** – уникальный идентификатор созданной транзакции пополнения (его можно использовать для отслеживания).
- **status** – статус новой транзакции. При успешном создании пополнения статус будет **"NEW"** (новая транзакция).

После создания транзакции она будет обработана системой. Статус может изменяться по мере выполнения. Возможные значения статуса транзакции (актуально для **любых** транзакций, включая пополнение и вывод) следующие:

- **NEW** – транзакция успешно создана (новая).
- **NEW_ERROR** – ошибка при создании транзакции (например, транзакция не была создана из-за недостатка баланса на кассе или технической ошибки).
- **ACCEPTED** – транзакция принята в обработку (начата).
- **CANCELED** – транзакция отменена.
- **PROCESSING** – транзакция в процессе выполнения.
- **PROCESSING_ERROR** – ошибка во время обработки транзакции.
- **EXPIRED** – транзакция просрочена (например, если пополнение/вывод не были подтверждены пользователем в отведённое время).
- **COMPLETED** – транзакция успешно завершена (средства зачислены на счет или выданы).

Ошибка: При проблемах с запросом возможен ответ об ошибке. Например, если подпись неверна или не хватает прав, вернётся **401 Unauthorized** (либо **400** в случае некорректного запроса) с описанием ошибки в JSON-формате, аналогичном приведённому выше (**code**, **message**, **additional_data**).

Пример запроса на пополнение (curl):

Shell

```
curl --location
'https://apimb.com/mbc/gateway/v1/api/cashpoint/48436/player/deposit' \
--header 'X-Timestamp: 2025-05-12 09:38:52' \
--header 'X-Signature:
9b2da4e4647521ede22ce2b95f025f919d089bb60b8400d6226b7431f3a312c4' \
--header 'X-API-Key:
api-key:3d83ac24-7fd2-498d-84b4-f2a7e80401fb' \
--header 'X-Project: MBC' \
--header 'Content-Type: application/json' \
--data-raw '{
    "brandId": 1,
    "playerId": "774",
    "amount": 200,
    "currency": "RUB"
}'
```

В примере выше выполняется пополнение игрока с ID **774** на 200 RUB для бренда Mostbet (brandId 1). Заголовок **X-Project: MBC** указывает, что операция проводится в проекте Mostbet Cash.

Пример успешного ответа:

JSON

```
{
  "transactionId": 11839895,
  "status": "COMPLETED"
}
```

Статус стал **COMPLETED**, что означает успешное завершение транзакции (средства зачислены игроку). Если бы статус был иным (например, **PROCESSING**), это означало бы, что транзакция ещё выполняется и её итог нужно проверить позже через историю транзакций.

Как получить информацию о заявках на вывод средств игрока?

Метод: `GET /cashpoint/{cashpointId}/player/cashout/list/page` – возвращает список запросов игроков на вывод средств через данную кассу. Этот метод позволяет агенту увидеть **актуальные заявки** на выплату, созданные игроками, и их параметры.

Заголовки: Требуется передать стандартные заголовки аутентификации: `X-Api-Key`, `X-Timestamp`, `X-Signature`. Заголовок `X-Project` **необязателен** для этого метода – если не указать, используется контекст по умолчанию (MBC).

Параметры запроса (Query): Метод поддерживает пагинацию и поиск:

- **page** – номер страницы (начиная с 0 или 1, в документации пример со страницы 2).
- **size** – количество записей на странице. Например, `page=2&size=10` вернёт десятую страницу по 10 заявок (при нумерации страниц с 0 – это третья страница результатов). Эти параметры обязательны для вызова.
- **searchString** – необязательный строковый фильтр поиска. Позволяет искать заявки по частичному совпадению: это может быть часть ID транзакции, сумма или ID игрока. Если указан, будут возвращены только те заявки, чьи параметры совпадают с этой строкой.

Успешный ответ (200 OK): Возвращается JSON с двумя основными полями:

- **items** – список заявок (массив объектов),
- **totalCount** – общее количество заявок, подходящих под критерии (размер выборки).

Каждая заявка в **items** содержит информацию о конкретном запросе на вывод:

```
JSON
{
  "items": [
    {
      "transactionId": 11839894,
      "playerId": "774",
```

```
        "amount": 20000.00,  
        "brandId": 1,  
        "currency": "RUB"  
    }  
],  
    "totalCount": 1  
}
```

В полях заявки:

- **transactionId** – идентификатор транзакции вывода (заявки). По нему можно, например, затем подтвердить выдачу (см. ниже) или свериться в истории.
- **playerId** – ID игрока, который запросил вывод.
- **amount** – запрошенная сумма вывода.
- **brandId** – ID бренда игрока. Используется то же значение, что и при пополнении (Mostbet = 1).
- **currency** – валюта запрошенного вывода, код ISO 4217.
- **totalCount** (вне массива) – общее число заявок на вывод, доступных по заданным критериям (например, всего 1, как в примере выше).

Если список пуст (нет активных/соответствующих заявок), **items** будет пустым массивом, а **totalCount** = 0.

Ошибка: При проблемах с авторизацией или запросом вернётся ошибка 401/400 с описанием в JSON (**code**, **message**, **additional_data**), аналогично другим методам.

Пример запроса (curl) получения списка заявок: (параметры **page** и **size** указаны для примера)

Shell

```
curl --location
'https://apimb.com/mbc/gateway/v1/api/cashpoint/48436/player/cash
out/list/page?page=2&size=10' \
--header 'X-Timestamp: 2025-05-13 10:18:10' \
--header 'X-Signature:
15554c8bef430db90e9ac083094c4f6ebe66213dbb7b86c082ad88767ef9ec1d'
\
--header 'X-API-Key:
api-key:3d83ac24-7fd2-498d-84b4-f2a7e80401fb'
```

Данный запрос получит 10 заявок со страницы №2. В ответ придёт JSON с массивом `items` (в примере выше он содержал одну заявку) и полем `totalCount`. Агент может просматривать эти заявки и выбирать нужную для подтверждения выплаты (см. следующий вопрос).

Как подтвердить (выполнить) вывод средств игрока через API?

Метод: `POST /cashpoint/{cashpointId}/player/cashout/confirmation` – подтверждает и проводит выплату средств игроку по ранее созданной заявке на вывод. Этот метод используется после того, как игрок запросил вывод, и кассир получил от игрока уникальный код подтверждения (отправленный игроку по SMS).

Заголовки: Для подтверждения выплаты требуются заголовки: `X-API-Key`, `X-Timestamp`, `X-Signature` (аналогично другим методам) и `X-Project` (обязательно, указывает проект MBC).

Тело запроса (JSON): Для подтверждения вывода необходимо передать JSON с двумя полями:

JSON

```
{
  "code": "<string>",
  "transactionId": <number>
}
```

- **code** – строковый **код подтверждения**, который был выслан игроку для верификации вывода. Кассир должен запросить этот код у игрока и передать в запросе. Без верного кода операция вывода не будет выполнена.
- **transactionId** – ID транзакции (заявки на вывод), которую нужно подтвердить. Этот идентификатор получается из списка заявок на вывод (см. предыдущий раздел) или другим способом от бекенда. Он однозначно идентифицирует запрос на вывод, ожидающий подтверждения.

Успешный ответ (200 OK): В случае успешного подтверждения API вернёт JSON с информацией о транзакции вывода:

```
JSON
{
  "transactionId": 11839894,
  "status": "NEW"
}
```

Где:

- **transactionId** – идентификатор выполненной транзакции (совпадает с ID переданным в запросе, подтверждающим что именно эта заявка теперь обработана).
- **status** – статус транзакции. На момент создания подтверждения обычно возвращается **"NEW"** (транзакция создана и ожидает обработки). Далее возможны изменения статуса по мере выполнения, аналогично статусам, описанным для пополнения (NEW, ACCEPTED, PROCESSING, COMPLETED и т.д.). Расшифровки статусов такие же, как приведено ранее. Обратите внимание на статус **NEW_ERROR** – при подтверждении вывода он может означать, что транзакция не создана, например, из-за недостаточного баланса кассы для выдачи запрошенной суммы.

Когда статус сменится на **COMPLETED**, это означает, что выплата успешно проведена и средства перечислены с игрового счёт на счёт кассира. Статус **CANCELED** означает, что заявка была отменена (например, игрок не подтвердил операцию вовремя, код просрочен или неверный).

Ошибка: Если указаны неверные данные (неправильный код подтверждения, просроченный код и т.п.), API вернёт ошибку (например, **400/401 Unauthorized** с описанием в полях `code`, `message`).

Пример запроса на подтверждение вывода (curl):

Shell

```
curl --location
'https://apimb.com/mbc/gateway/v1/api/cashpoint/48436/player/cash
out/confirmation' \
--header 'X-Timestamp: 2025-05-12 09:43:13' \
--header 'X-Signature:
ad723b01b018b333f515acb33e7d118d3f5610be427d4c93429c9b0d9308c012'
\
--header 'X-API-Key:
api-key:3d83ac24-7fd2-498d-84b4-f2a7e80401fb' \
--header 'X-Project: MBC' \
--header 'Content-Type: application/json' \
--data-raw '{
    "code": "679ZP7FK",
    "transactionId": 11839894
}'
```

В примере подтверждается заявка `transactionId: 11839894` с кодом подтверждения `679ZP7FK` (код получен игроком, проект – MBC).

Пример успешного ответа:

JSON

```
{
  "transactionId": 11839894,
  "status": "COMPLETED"
}
```

Статус **COMPLETED** свидетельствует, что операция вывода завершена успешно: заявка обработана, и указанная сумма выдана игроку. Кассир может выдать наличные и закрыть заявку. Если бы был возвращён статус **PROCESSING**, это значило бы, что операция ещё

выполняется (например, ожидается подтверждение на бекенде), и нужно проверить статус позже через метод истории транзакций.

Как получить историю транзакций кассы через API?

Метод: `GET /cashpoint/{cashpointId}/transactions/list/all` – возвращает историю всех транзакций по кассе (как пополнений, так и выплат) за указанный период. С помощью этого метода можно отследить все операции, проведённые через кассу, и их текущие статусы.

Заголовки: Передаются стандартные заголовки: `X-API-Key`, `X-Timestamp`, `X-Signature` (заголовок `X-Project` здесь не используется).

Параметры запроса (Query): Необходимо указать диапазон дат, за который запрашивается история:

- **dateStart** – начальная дата периода (в формате `YYYY-MM-DD`),
- **dateEnd** – конечная дата периода (в формате `YYYY-MM-DD`).

Будут возвращены транзакции, совершенные с `dateStart` по `dateEnd` включительно (вероятно, учитывается время 00:00 начала дня и 23:59 конца дня). Эти параметры обязательны.

Дополнительно можно указать фильтры для сужения результатов (необязательно):

- **brandId** – ID бренда, чтобы получить транзакции только определённого бренда (в случае с Mostbet Cash это Mostbet = 1). Если не указан, вернутся транзакции всех брендов по кассе.
- **playerId** – ID игрока, чтобы отобрать только транзакции конкретного игрока.
- **transactionId** – ID конкретной транзакции, если нужно найти информацию только по ней.

Успешный ответ (200 OK): Результатом является JSON с массивом транзакций и их деталями:

```
JSON
{
```

```
"items": [  
  {  
    "transactionId": 11839911,  
    "type": "CASHOUT",  
    "status": "COMPLETED",  
    "subject": "CLIENT",  
    "brandId": 1,  
    "playerId": "774",  
    "date": "2025-05-12 14:29:42",  
    "amount": 20000.00,  
    "currency": "RUB"  
  },  
  {  
    "transactionId": 11839895,  
    "type": "DEPOSIT",  
    "status": "COMPLETED",  
    "subject": "CLIENT",  
    "brandId": 1,  
    "playerId": "774",  
    "date": "2025-05-12 12:39:02",  
    "amount": 200.00,  
    "currency": "RUB"  
  },  
  {  
    "transactionId": 11839894,  
    "type": "CASHOUT",  
    "status": "COMPLETED",  
    "subject": "CLIENT",  
    "brandId": 1,  
    "playerId": "774",  
    "date": "2025-05-12 11:53:02",  
    "amount": 20000.00,  
    "currency": "RUB"  
  }  
]  
}
```


Поле **items** содержит список транзакций (каждая в виде объекта). Краткое описание полей транзакции:

- **transactionId** – уникальный идентификатор транзакции (совпадает с тем, что возвращался при создании пополнения/вывода).
- **type** – тип транзакции: **"DEPOSIT"** (пополнение счета игрока) или **"CASHOUT"** (вывод средств игроку).
- **status** – статус транзакции на момент запроса. Возможные статусы уже перечислялись ранее: NEW, NEW_ERROR, ACCEPTED, CANCELED, PROCESSING, PROCESSING_ERROR, EXPIRED, COMPLETED. В приведённом примере все транзакции имеют статус COMPLETED (завершены). Если транзакция ещё в процессе, здесь может быть, например, PROCESSING.
- **subject** – субъект (категория) транзакции. Это поле поясняет, чем вызвана транзакция:
 - **BACKOFFICE** – ручная операция с кассой (например, корректировка баланса кассы оператором через бэк-офис, начисление бонуса или компенсации).
 - **CLIENT** – операции, связанные с клиентами (т.е. пополнение или выплата игроку через кассу). В примере выше **subject** у транзакций = "CLIENT", так как это пополнение/вывод игроку.
 - **EXTERNAL** – внешнее пополнение кассы (например, заведение средств на кассу извне, не от игрока).
 - **FEE_REWARD** – начисление комиссионного вознаграждения.
- **brandId** – ID бренда, к которому относится транзакция.
- **playerId** – ID игрока, участвующего в транзакции (для операций **CLIENT**). Для транзакций типа BACKOFFICE или EXTERNAL может отсутствовать или не использоваться.
- **date** – дата и время совершения транзакции (в ISO-формате, часовой пояс не уточнён, вероятно UTC или локальное время кассы).
- **amount** – сумма транзакции.
- **currency** – валюта транзакции, код ISO 4217.

Если за указанный период транзакций нет, массив `items` будет пустым.

Ошибка: При некорректных параметрах (например, отсутствует `dateStart/dateEnd`) или проблемах с авторизацией возвращается ошибка 400/401. Формат ошибки – JSON с полями `code`, `message`, `additional_data`.

Пример запроса истории транзакций (curl): за период с 12.04.2025 по 17.04.2025 для кассы 48436:

Shell

```
curl --location
'https://apimb.com/mbc/gateway/v1/api/cashpoint/48436/transactions/list/all?dateStart=2025-04-12&dateEnd=2025-04-17' \
--header 'X-Timestamp: 2025-05-12 09:35:56' \
--header 'X-Signature:
c2c6a8e3e3532d41b5aab92986f0193b2d9765244e2b98a73906afd3f4eed6d7'
\
--header 'X-API-Key:
api-key:3d83ac24-7fd2-498d-84b4-f2a7e80401fb'
```

В ответ придёт JSON со списком транзакций за указанный период (как показано выше). В приведённом примере были возвращены три транзакции: две выплаты (**CASHOUT**) и одно пополнение (**DEPOSIT**) для игрока `774`, все успешно завершены. Анализируя поле `status`, можно убедиться, какие операции ещё ожидают исполнения, а какие выполнены.

Какие ограничения (лимиты) существуют на запросы к API?

Для API предусмотрены ограничения на частоту вызовов. В частности, установлено ограничение: **не более 200 запросов в минуту** с одного API-ключа (кассы). Если этот лимит превышен, сервер временно будет возвращать ошибку **429 Too Many Requests**, что сигнализирует о слишком большом количестве запросов за короткий промежуток времени.

При получении ответа 429 следует уменьшить частоту запросов. Возможной стратегией является реализация паузы или очереди запросов на стороне клиента. Лимит в 200 запросов в минуту достаточно высок для большинства случаев; однако, при интеграции следует учитывать его, чтобы не попасть под ограничение.

Также стоит помнить, что слишком частые ошибки авторизации (401) или другие могут привести к временному блокированию ключа. Поэтому убедитесь, что правильно рассчитываете подпись и используете корректные ключи.

Что означает параметр X-Project?

В API Mostbet Cash параметр **X-Project** используется для указания контекста, в котором выполняется операция.

Для всех запросов в рамках проекта Mostbet Cash значение этого параметра должно быть строго MBC.

- Заголовок **X-Project**: **MBC** обязателен при выполнении финансовых операций — например, при пополнении счёта игрока (**player/deposit**) или подтверждении вывода средств (**player/cashout/confirmation**).
- Для информационных запросов (**GET**-методы — например, проверка баланса кассы, получение списка заявок на вывод или просмотр истории транзакций) этот заголовок не требуется или считается необязательным.

Если заголовок **X-Project** будет отсутствовать в запросе, для которого он обязателен, система вернёт ошибку авторизации.

Какой бренд поддерживается в API Mostbet Cash?

Во всех API-запросах, где требуется указание бренда (например, при пополнении счёта игрока, выводе средств или просмотре истории транзакций), используется параметр **brandId**.

В рамках проекта Mostbet Cash допустимо использовать только один бренд — **brandId: 1, соответствующий бренду Mostbet.**

Указание любого другого значения приведёт к ошибке или будет проигнорировано. Всегда проверяйте корректность **brandId** перед выполнением операции, чтобы избежать сбоев или отклонения транзакции системой.

Что делать, если остались вопросы по API Mostbet Cash?

Если после прочтения этого FAQ у вас остались вопросы, возникли сложности с реализацией или требуется техническая помощь — смело обращайтесь в поддержку проекта Mostbet Cash.

Вы можете написать на почту: **info@mostbetshop.com** — специалисты помогут вам разобраться с настройками, ключами, форматами запросов и любыми другими вопросами по API.

Этот FAQ охватывает все основные аспекты: от получения доступа до примеров использования методов, включая параметры, подписи и формат ответов. Следуя ему, вы сможете корректно интегрироваться с системой и выполнять операции через API безопасно и эффективно.