



---

# СЕРВИС ДЛЯ УПРАВЛЕНИЯ ЛИМИТАМИ ПЛАТЕЖЕЙ

---

Руководство пользователя



2023 ГОД

ООО ТЕСТ

115088, г.Москва, ул.Симоновский Вал, 26А

**ОГЛАВЛЕНИЕ**

1. Введение.....	2
1.1 Область применения.....	2
1.2 Термины и сокращения.....	2
1.3 Краткое описание возможностей.....	2
1.4 Уровень подготовки пользователей.....	2
1.5 Перечень программной документации.....	2
2. Назначение и требования для эксплуатации ПО.....	3
2.1 Функциональное назначение.....	3
2.2 Эксплуатационное назначение.....	3
2.3 Требования к эксплуатации серверной части.....	3
3. Формат запросов.....	4
3.1 HTTP-заголовки.....	4
3.2 X-Data-Hash.....	4
4. Описание операций.....	4
4.1 limits.check_currency_limits.....	4
4.1.1 Формат запроса (пример).....	5
4.1.2 Формат ответа (пример).....	6
4.2 limits.currency_limits.....	6
4.2.1 Формат запроса (пример).....	7
4.2.2 Формат ответа (пример).....	7
5. Аварийные ситуации.....	8
6. Рекомендации по освоению.....	8

## 1. ВВЕДЕНИЕ

Наименование программного обеспечения “Сервис для управления лимитами платежей”.

### 1.1 ОБЛАСТЬ ПРИМЕНЕНИЯ

Настоящий документ (далее – Описание) распространяется на программное обеспечение (далее – ПО) “Сервис для управления лимитами платежей”.

“Сервис для управления лимитами платежей” должен представлять собой техническое решение, самостоятельный программный продукт, который интегрируется с любыми IT-решениями. Продукт предназначен для автоматизации деятельности по контролю лимитов, позволяет получать и проверять установленные лимиты платежей по валюте и типу операции (пополнение, выплата).

### 1.2 ТЕРМИНЫ И СОКРАЩЕНИЯ

<b>ПО</b>	Программное обеспечение – совокупность компьютерных программ и связанных с ними данных, которая содержит инструкции по указанию компьютеру, что и как делать.
<b>Сервис</b>	Независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей. Сервисы обычно выполняются в виде библиотек общего пользования.
<b>Платежный лимит</b>	Установленное ограничение на максимальную сумму денежных операций, которые можно выполнить за определенный период времени.
<b>Пополнение</b>	Процесс внесения дополнительных средств на электронный счет пользователя.
<b>Выплата</b>	Операция выплаты средств на счет пользователя по его реквизитам.
<b>API</b>	Программный интерфейс приложения — это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными.
<b>Очередь</b>	Совокупность объектов, которые поддерживаются в последовательности и могут быть изменены путем добавления объектов на одном конце последовательности и удаления объектов с другого конца последовательности. Операции очереди делают ее структурой данных, которая обеспечивает хранение и передачу двоичных данных между различными участниками системы.
<b>Инстанс</b>	Дубликат объекта, сохраняющий неразрывную связь с оригиналом и полную зависимость от него. Модификация любого образца равносильна модификации оригинала – результаты сказываются как на оригинале объекта, так и на всех образцах.
<b>Мерчант</b>	Партнер, пользователь программного обеспечения.

### 1.3 КРАТКОЕ ОПИСАНИЕ ВОЗМОЖНОСТЕЙ

- Масштабирование благодаря наличию ресурсов для запуска нескольких экземпляров сервиса;
- Штатный функционал позволяет подписать **N инстансов** на одну и ту же очередь, сообщения из которой будут случайным образом приходит в тот или иной инстанс;
- Передача данных в виде запросов в Базу данных и получение этих данных;
- Проверка лимита платежа по валюте (можно ли производить оплату) и типу операции (пополнение, выплата);
- Получение лимита платежа по валюте и направлению (пополнение, выплата).

### 1.4 УРОВЕНЬ ПОДГОТОВКИ ПОЛЬЗОВАТЕЛЕЙ

Для интеграции API пользователь должен иметь квалификацию разработчика не ниже уровня Regular Middle.

### 1.5 ПЕРЕЧЕНЬ ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

- “Сервис для управления лимитами платежей”. Техническое задание (ГОСТ 19.201-78);
- “Сервис для управления лимитами платежей”. Руководство пользователя;
- “Сервис для управления лимитами платежей”. Инструкция по развертыванию экземпляра ПО;
- “Сервис для управления лимитами платежей”. Описание функциональных характеристик программного обеспечения;
- “Сервис для управления лимитами платежей”. Описание жизненного цикла разработки ПО;
- “Сервис для управления лимитами платежей”. Инструкция для проверки в тестовой среде ПО.

## 2. НАЗНАЧЕНИЕ И ТРЕБОВАНИЯ ДЛЯ ЭКСПЛУАТАЦИИ ПО

Программное обеспечение “Сервис для управления лимитами платежей” предназначено для интеграции и дальнейшей возможности использования у любого Заказчика.

### 2.1 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

ПО должно представлять собой техническое решение, самостоятельный программный продукт, который интегрируется с любыми IT-решениями. Продукт предназначен для автоматизации деятельности по контролю лимитов, позволяет получать и проверять установленные лимиты платежей по валюте и типу операции (пополнение, выплата).

### 2.2 ЭКСПЛУАТАЦИОННОЕ НАЗНАЧЕНИЕ

ПО будет интегрироваться и эксплуатироваться на стороне Заказчика и должно предоставлять следующие преимущества при использовании сервиса:

- ✓ Независимость компонентов системы друг от друга. Благодаря использованию очереди, компоненты взаимодействуют через некий общий интерфейс, но ничего не знают о существовании друг друга.
- ✓ Экономия ресурсов достигается вследствие возможности разумно распределять информацию, поступающую в очередь от одних процессов, между другими процессами, осуществляющими ее обработку. Кроме того, благодаря тому, что нет необходимости промежуточного хранения необработанных данных, достигается дополнительная экономия ресурсов.
- ✓ Надежность очередей достигается благодаря возможности накапливать сообщения, амортизируя недостаток вычислительных возможностей системы, а также благодаря независимости компонентов. Помимо этого, очередь может аккомодировать сбои отдельных компонентов, осуществляя доставку «опоздавших» сообщений после восстановления.
- ✓ И, наконец, гарантия последовательной обработки, позволяющая точно контролировать потоки данных в системе и запускать асинхронную обработку там, где это необходимо, не беспокоясь, что одна операция выполнится раньше другой, от результата которой она зависит.

### 2.3 ТРЕБОВАНИЯ К ЭКСПЛУАТАЦИИ СЕРВЕРНОЙ ЧАСТИ

Обеспечение функционирования ПО серверной части “Сервис для управления лимитами платежей” должно быть реализовано на базе серверной операционной системы Linux. Минимальной конфигурацией аппаратной составляющей должны являться:

- Современная ОС: Linux;
- Оперативная память: 32 Гб;
- Свободное дисковое пространство: не менее 50 Gb;

- Количество логических ядер процессора: 4;
- Частота процессора: 3.50 GHz.

Необходимо реализовать возможность разворачивания экземпляра ПО и на других ОС, поддерживающих платформу для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации Docker, например Windows 10 (Профессиональная или Корпоративная).

### 3 ФОРМАТ ЗАПРОСОВ

Оповещения передаются на POST запросов на url в формате json грс 2.0, предоставляемый при подключении платформы. Важно проверять X-Data-Hash запроса.

#### 3.1 HTTP-ЗАГОЛОВКИ

Обязательные к передаче заголовки HTTP запроса:

Content-Type: application/json

X-Data-Hash

#### 3.2 X-DATA-HASH

Используется для подтверждения легитимности и подлинности запроса. Передается в HTTP заголовке X-Data-Hash.

$X\text{-Data-Hash} = \text{sha512}(\%request\_body\% + \%secret\_key\%)$

- `request_body` – объект запроса, сериализованный в виде JSON строки (берем исходную строку).
- `secret_key` – секретный ключ, выдаваемый при подключении приложения.

На выходе получаем ЭЦП к запросу.

Проверка производится следующим образом:

$isValidRequest = X\text{-Data-Hash} == \text{sha512}(\%request\_body\% + \%secret\_key\%)$

- `isValidRequest == true` – запрос легитимный;
- `isValidRequest == false` – запрос следует проигнорировать.

## 4. ОПИСАНИЕ ОПЕРАЦИЙ

### 4.1 LIMITS.CHECK\_CURRENCY\_LIMITS

Проверка лимита суммы валюты по `service_id`

Параметр	Тип	Обязательность	Описание
<code>method</code>	String	+	Метод для для проверка лимита суммы валюты - <code>limits.check_currency_limits</code>

params	Object	+	Параметры для метода
params.payment.service_id	Number	+	Идентификатор платежной системы в провайдере
params.payment.destination	String	+	Направление транзакции (in – пополнение, out – выплата)
params.payment.amount	Number	+	Сумма платежа(в копейках)
params.payment.amount.currency	String	+	Валюта платежа(ISO 4217)
service_id	Number	+	Для данного метода service_id = 0

#### 4.1.1 ФОРМАТ ЗАПРОСА (ПРИМЕР)

```
{
  "method": "limits.check_currency_limits",
  "params": {
    "payment": {
      "service_id": 201,
      "destination": "in",
      "amount": {
        "value": 550,
        "currency": "USD"
      }
    }
  }
},
```

```
"service_id": 0
}
```

#### 4.1.2 ФОРМАТ ОТВЕТА (ПРИМЕР)

```
{
  "success": true,
  "result": {
    "included_in_limits": false
  }
}
```

#### 4.2 LIMITS.CURRENCY\_LIMITS

Получение лимитов валюты по направлению

Параметр	Тип	Обязательность	Описание
method	String	+	Метод для для проверка лимита суммы валюты - limits.currency_limits
params	Object	+	Параметры для метода
params.payment.destination	String	+	Направление транзакции (in - пополнение, out - выплата)
params.payment.amount	Number	+	Сумма платежа(в копейках)
params.payment.amount.currency	String	+	Валюта платежа(ISO 4217)
service_id	Number	+	Для данного метода service_id = 0

#### 4.2.1 ФОРМАТ ЗАПРОСА (ПРИМЕР)

```
{  
  
  "service_id": 0,  
  
  "method": "limits.currency_limit",  
  
  "params": {  
  
    "payment": {  
  
      "service_id": 1301,  
  
      "destination": "in",  
  
      "amount": {  
  
        "currency": "RUB"  
  
      }  
  
    }  
  
  }  
  
}
```

#### 4.2.2 ФОРМАТ ОТВЕТА (ПРИМЕР)

```
{  
  
  "success": true,  
  
  "result": {  
  
    "payment": {  
  
      "amount": {  
  
        "currency": "RUB",  
  
        "min_value": 99,  
  
        "max_value": 1000000000  
  
      },  
  
    }  
  
  }  
  
}
```

```
        "destination": "in",  
        "service_id": 1301  
    }  
}  
}
```

## 5. АВАРИЙНЫЕ СИТУАЦИИ

В случае аварийных ситуаций обращаться в техническую поддержку.

## 6. РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ

Для успешной работы с ПО “Сервис для управления лимитами платежей” необходимо:

- Иметь квалификацию разработчика не ниже уровня Regular Middle;
- Иметь оборудованное рабочее место с подключением к сети Интернет;
- Рабочее место должно соответствовать минимальным требованиям, указанным в разделе “2.3 Требования к эксплуатации серверной части” данного документа;
- Ознакомиться с документом “Руководство пользователя “Сервис для управления лимитами платежей””;
- Ознакомиться с документом “Разворачивание экземпляра ПО “Сервис для управления лимитами платежей””.