



СЕРВИС ПРОВЕДЕНИЯ КАСКАДНЫХ ОПЕРАЦИЙ

Руководство пользователя

Оглавление

1. Введение	2
1.1 Общие сведения.....	2
1.2 Назначение программы.....	2
1.3 Краткое описание возможностей.....	2
1.4 Уровень подготовки пользователей	3
1.4 Перечень программной документации	3
1.5 Глоссарий.....	3
2. Назначение и требования для эксплуатации ПО	4
2.1 Функциональное назначение	4
2.2 Эксплуатационное назначение	4
2.3 Требования к эксплуатации серверной части.....	4
3. Формат запросов	5
3.1 X-Data-Application-ID	5
3.2 X-Data-Hash.....	5
4. Описание операций.....	5
4.1 Каскадные операции пополнения.....	5
4.1.1 Основные параметры	6
4.1.2 Формат запроса (пример).....	8
4.1.3 Формат ответа (пример).....	8
4.2 Каскадные операции выплат	9
4.2.1 Основные параметры	9
4.2.2 Формат запроса (пример).....	12
4.2.3 Формат ответа (пример).....	12
5. Аварийные ситуации	13
6. Рекомендации по освоению	13

1. Введение

Настоящий документ (далее – Описание) распространяется на программное обеспечение (далее – ПО) “Сервис проведения каскадных операций” разработанное компанией **ООО «Фарос Медиа»**.

Контакты:

Сайт: <https://faros-media.ru/>

Телефон: [+7 967 101-46-43](tel:+79671014643)

Почта: CEO@FAROS-MEDIA.RU

1.1 Общие сведения

“Сервис проведения каскадных операций” – это техническое решение, предоставляющее возможность дополнительных попыток проведения платежа без изменения параметров запроса в случае прерывания по различным причинам. Дополнительные последовательные попытки происходят через резервных провайдеров автоматически на стороне сервиса по заранее заданной конфигурации.

1.2 Назначение программы

Программа предназначена для случаев, когда проведение платежа прерывается по различным причинам. Функционал каскадного проведения операций включает в себя последовательные дополнительные попытки проведения платежа через резервных провайдеров без изменения платежного метода.

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, подтверждение легитимности и подлинности запроса;
- Инициализация операции каскадного пополнения;
- Инициализация операции каскадных выплат;
- В случае сбоя или иных причин, автоматическое повторение операции (проведение платежа) по заранее заданной конфигурации;
- Выполнение проверки запросов на параметры.

1.4 Уровень подготовки пользователей

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

1.4 Перечень программной документации

“Сервис проведения каскадных операций”. Техническое задание (ГОСТ 19.201-78);

“Сервис проведения каскадных операций”. Руководство пользователя;

“Сервис проведения каскадных операций”. Инструкция по развертыванию экземпляра ПО;

“Сервис проведения каскадных операций”. Описание функциональных характеристик программного обеспечения;

“Сервис проведения каскадных операций”. Описание жизненного цикла разработки ПО;

“Сервис проведения каскадных операций”. Инструкция для проверки в тестовой среде ПО.

1.5 Глоссарий

Термин	Значение
API	Описание способов взаимодействия одной компьютерной программы с другими.
БД	База данных – набор структурированных данных, хранящихся в виде таблицы.
Инстанс	Экземпляр класса (объекта) в объектно-ориентированном программировании.
Мерчант	Партнер, пользователь программного обеспечения “Сервис проведения каскадных операций”.
Очередь	Некая структура данных, которая обеспечивает хранение и передачу двоичных данных между различными участниками системы.
Сервис	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.
ПО	Программное обеспечение.
Каскадные операции	Дополнительные попытки проведения платежа без изменения параметров запроса в случае прерывания по различным причинам.
Конфигурация	Набор настроек, методов и параметров, направленных на систематизацию поведения программного обеспечения.

2. Назначение и требования для эксплуатации ПО

Программное обеспечение “Сервис проведения каскадных операций” предназначено для использования на стороне мерчанта.

2.1 Функциональное назначение

“Сервис проведения каскадных операций” – это техническое решение, предоставляющее возможность дополнительных попыток проведения платежа без изменения параметров запроса в случае прерывания по различным причинам.

2.2 Эксплуатационное назначение

ПО интегрируется и эксплуатируется на стороне Заказчика, также предоставляет 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 проведения каскадных операций” реализовано на базе серверной операционной системы Linux. Минимальной конфигурацией аппаратной составляющей являются:

- Современная ОС: Linux;
- Оперативная память: 2 ГБ;
- Свободное дисковое пространство: не менее 20 Gb;
- Количество логических ядер процессора: 2;

- Частота процессора: 3.50 GHz.

Возможно разворачивание 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

Взаимодействие происходит посредством отправки запросов и ответов на них.

Запросы отправляются по протоколу HTTPS POST-запросами по умолчанию, если для конкретного метода не указано иное.

Запросы должны содержать обязательные к передаче заголовки HTTP запроса:

`Content-Type: application/json, X-Data-Application-Id, X-Data-Hash`

3.1 X-Data-Application-ID

Статичный параметр, выдаваемый при подключении приложения. Целое число.

3.2 X-Data-Hash

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

`X-Data-Hash = sha512(%request_body% + %secret_key%)`

- `request_body` - объект запроса, сериализованный в виде JSON строки.
- `secret_key` - секретный ключ, выдаваемый при подключении приложения.
- `sha512` - алгоритм однонаправленного хэширования. На выходе получаем ЭЦП к запросу.

Ответы содержат заголовок `Content-Type: "application/json"` и объекты в JSON формате.

4. Описание операций

4.1 Каскадные операции пополнения

Инициализация операции каскадного пополнения.

4.1.1 Основные параметры

Параметр	Тип	Обязательный	Описание
method	string	+	Название метода
params.payment	Payment	+	Информация о транзакции
params.payment.service_id	integer	+	Идентификатор платежной системы
params.payment.amount	Amount	+	Объект суммы и валюты платежа
params.payment.amount.value	integer	+	Сумма платежа
params.payment.amount.currency	string(3)	+	Валюта платежа
params.payment.identifiers	Identifier	+	Объект с идентификаторами платежа
params.payment.identifiers.c_id	integer	+	Идентификатор платежа на стороне Мерчанта
params.payment.payer	Payer	-	Объект с платежными данными пользователя
params.payment.payer.card	Card	-	Карточные данные при платеже
params.payment.payer.card.id	string	-	Карточный токен при платеже
params.payment.payer.card.mask	string	-	Маска карты при платеже
params.payment.payer.card.holder	string	-	Имя Фамилия владельца карты при платеже

<code>params.payment.payer.payment_system_account</code>	Account	-	Объект с аккаунтом пользователя платежной системы при платеже
<code>params.payment.payer.payment_system_account.id</code>	string	+	Идентификатор аккаунта пользователя платежной системы при платеже
<code>params.payment.payer.person</code>	Person	-	Объект с персональными данными
<code>params.payment.payer.person.firts_name</code>	string	-	Имя пользователя
<code>params.payment.payer.person.last_name</code>	string	-	Фамилия пользователя
<code>params.payment.payer.person.birth_date</code>	string	-	Дата рождения пользователя (DD-ММ-YYYY)
<code>params.payment.payer.person.documents</code>	Array	-	Список документов пользователя
<code>params.payment.payer.person.documents[0].type</code>	string	-	Тип документа
<code>params.payment.payer.person.documents[0].number</code>	string	-	Номер документа
<code>params.payment.payer.customer_account</code>	object	-	Данные по пользователю у Мерчанта
<code>params.payment.payer.customer_account.id</code>	string	+	Идентификатор привязанного пользователя
<code>params.payment.redirect</code>	Redirect	-	Объект с информацией о переадресациях пользователя

params.payment.redirect.on_success	string	-	Ссылка для редиректа в случае успеха
params.payment.redirect.on_fail	string	-	Ссылка для редиректа в случае ошибки/отказа/
params.params.payment.redirect.on_return	string	-	Ссылка для редиректа в случае возврата со страницы
params.payment.client	object	-	Объект с данными устройства пользователя
params.payment.client.ip	string	-	IP адрес пользователя
params.payment.client.browser	string	-	Браузер пользователя
params.payment.client.country	string	-	Страна пользователя
params.payment.client.language	string	-	Язык пользователя

4.1.2 Формат запроса (пример)

```
{ "service_id": 10010, "method": "payment.in", "params": { "payment": { "identifiers": { "c_id": 123 }, "description": "комментарий", "amount": { "value": 10000, "currency": "BRL" }, "payer": { "email": "test@test.com", "person": { "first_name": "Pablo", "last_name": "FSF", "birth_date": "12-12-1990", "documents": [ { "type": "BR_CPF", "number": "12345678946" } ] }, "customer_account": { "id": "test@gmail.com" }, "payment_system_account": { "id": "23" } }, "client": { "language": "RU" } } } }
```

4.1.3 Формат ответа (пример)

```
{ "success": true, "result": { "payment": { "amount": { "value": 10000, "currency": "BRL" }, "description": "комментарий", "identifiers": { "c_id": 123, "h_id": 22, "p_id": "f34r" }, "redirect": { "to":
```

```
"https://cashier.paywallk.com/payment?payment_amount=10000&payment_currency=RUB&payment_id=50312&language_code=EN&payment_description=%D0%BA%D0%BE%D0%BC%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%B8%D0%B9&force_payment_method=neteller&project_id=1960&signature=4btW WpdcT6OYOSmJDLmHg9dNBCTG%2BE36vIVw7BYrvEq5mec6gSyhJBDmpoF4eCg O7C%2FO9tISO2b6EzrH4BmTBQ%3D%3D" }, "status": { "status": "await_redirect", "final": false, "success": null, "error": null, "history": [ { "status": "created", "final": false, "success": null, "created": "2020-02-26T14:20:09.127Z", "reason": null, "amount": 10000 }, { "status": "await_redirect", "final": false, "success": null, "created": "2020-02-26T14:20:09.545Z", "reason": null, "amount": 10000 } ] }, "timestamps": { "created": "2020-02-26T14:20:09.127Z", "updated": "2020-02-26T14:20:09.476Z", "finished": null }, "destination": "in", "operations": [ { "id": "617", "operation_type": "payment.in", "amount": { "value": 10000, "currency": "BRL" }, "timestamps": { "created": "2020-02-26T14:20:09.200Z", "updated": "2020-02-26T14:20:09.571Z", "finished": null }, "status": { "status": "await_user_action", "final": false, "success": null, "error": null } } ], "operation": { "id": "617", "operation_type": "payment.in", "amount": { "value": 10000, "currency": "RUB" }, "timestamps": { "created": "2020-02-26T14:20:09.200Z", "updated": "2020-02-26T14:20:09.571Z", "finished": null }, "status": { "status": "await_user_action", "final": false, "success": null, "error": null } }, "next": null, "request_id": "gMUy61ExaRfzYjHIRY1dl", "processing_time": 1135 }
```

4.2 Каскадные операции выплат

Инициализация операции каскадных выплат.

4.2.1 Основные параметры

Параметр	Тип	Обязательный	Описание
method	string	+	Название метода
params.payment	Payment	+	Информация о транзакции
params.payment.service_id	integer	+	Идентификатор платежной системы
params.payment.amount	Amount	+	Объект суммы и валюты платежа
params.payment.amount.value	integer	+	Сумма платежа

params.payment.amount.currency	string(3)	+	Валюта платежа
params.payment.identifiers	Identifier	+	Объект с идентификаторами платежа
params.payment.identifiers.c_id	integer	+	Идентификатор платежа на стороне Мерчанта
params.payment.receiver	Receiver	-	Объект с данными пользователя при выплате
params.payment.receiver.card	Card	-	Карточные данные при выплате
params.payment.receiver.card.id	string	-	Карточный токен при выплате
params.payment.receiver.card.mask	string	-	Маска карты при выплате
params.payment.receiver.card.holder	string	-	Имя Фамилия владельца карты при выплате
params.payment.receiver.payment_system_account	string	-	Объект с аккаунтом пользователя платежной системы при выплате
params.payment.receiver.payment_system_account.id	string	+	Идентификатор аккаунта пользователя платежной системы при выплате
params.payment.receiver.person	Person	-	Объект с персональными данными
params.payment.receiver.person.firts_name	string	-	Имя пользователя

params.payment.receiver.person.last_name	string	-	Фамилия пользователя
params.payment.receiver.person.birth_date	string	-	Дата рождения пользователя (DD-ММ-YYYY)
params.payment.receiver.person.documents	Array	-	Список документов пользователя
params.payment.receiver.person.documents[0].type	string	-	Тип документа
params.payment.receiver.person.documents[0].number	string	-	Номер документа
params.payment.receiver.customer_account	object	-	Данные по пользователю у Мерчанта
params.payment.receiver.customer_account.id	string	+	Идентификатор привязанного пользователя
params.receiver.redirect	Redirect	-	Объект с информацией о переадресациях пользователя
params.receiver.redirect.on_success	string	-	Ссылка для редиректа в случае успеха
params.receiver.redirect.on_fail	string	-	Ссылка для редиректа в случае ошибки/отказа/
params.receiver.redirect.on_return	string	-	Ссылка для редиректа в случае возврата со страницы
params.payment.client	object	-	Объект с данными устройства пользователя

params.payment.client.ip	string	-	IP адрес пользователя
params.payment.client.browser	string	-	Браузер пользователя
params.payment.client.country	string	-	Страна пользователя
params.payment.client.language	string	-	Язык пользователя

4.2.2 Формат запроса (пример)

```
{ "service_id": 10010, "method": "payment.out", "params": { "payment": {
  "description": "Some description", "identifiers": { "c_id": 123 }, "receiver": { "person":
  { "last_name": "Dis", "first_name": "Dis", "birth_date": "21-01-2021", "documents": [ {
  "type": "BR_CPF", "number": "12345678946" } ] }, "email": "test@test.com",
  "customer_account": { "id": "123" }, "payment_system_account": { "id": "23" } },
  "amount": { "value": 20000, "currency": "BRL" }, "client": { "ip": "192.168.1.1" } } } }
```

4.2.3 Формат ответа (пример)

```
{ "success": true, "result": { "payment": { "amount": { "value": 15000, "currency":
  "BRL" }, "description": "комментарий", "identifiers": { "c_id": 123, "h_id": 123, "p_id":
  "123" }, "status": { "status": "processing", "final": false, "success": null, "error": null,
  "history": [ { "status": "created", "final": false, "success": null, "created": "2020-02-
  26T13:17:03.608Z", "reason": null, "amount": 15000 }, { "status": "processing", "final":
  false, "success": null, "created": "2020-02-26T13:17:05.271Z", "reason": null,
  "amount": 15000 } ] }, "timestamps": { "created": "2020-02-26T13:17:03.608Z",
  "updated": "2020-02-26T13:17:05.222Z", "finished": null }, "destination": "out",
  "operations": [ { "id": "616", "operation_type": "payment.out", "amount": { "value":
  15000, "currency": "BRL" }, "timestamps": { "created": "2020-02-26T13:17:03.681Z",
  "updated": "2020-02-26T13:17:05.291Z", "finished": null }, "status": { "status":
  "processing", "final": false, "success": null, "error": null } } ] }, "operation": { "id": "616",
  "operation_type": "payment.out", "amount": { "value": 15000, "currency": "BRL" },
  "timestamps": { "created": "2020-02-26T13:17:03.681Z", "updated": "2020-02-
  26T13:17:05.291Z", "finished": null }, "status": { "status": "processing", "final": false,
  "success": null, "error": null } } }, "next": null, "request_id": "bx2murjp45nO2kfKrNTJx",
  "processing_time": 2345 }
```

5. Аварийные ситуации

В случае аварийных ситуаций обращаться в техническую поддержку.

6. Рекомендации по освоению

Для успешной работы с ПО *“Сервис проведения каскадных операций”* 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 проведения каскадных операций”*;
- Ознакомиться с документом *“Разворачивание экземпляра ПО “Сервис проведения каскадных операций”*.