

2023 год

Сервис для расчета комиссий платежа

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

ОГЛАВЛЕНИЕ

1. ВВЕДЕНИЕ	2
1.1 НАИМЕНОВАНИЕ ПРОГРАММЫ	2
1.2 ОБЛАСТЬ ПРИМЕНЕНИЯ	2
1.3 ГЛОССАРИЙ	2
1.4 ФУНКЦИИ И ВОЗМОЖНОСТИ СЕРВИСА	3
1.5 ОРГАНИЗАЦИЯ ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ	3
1.6 УРОВЕНЬ ПОДГОТОВКИ ПОЛЬЗОВАТЕЛЕЙ	3
1.7 ПЕРЕЧЕНЬ ПРОГРАММНОЙ ДОКУМЕНТАЦИИ	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. ПОДГОТОВКА К РАБОТЕ	5
3.1 ФОРМАТ ЗАПРОСОВ	5
3.1.1 HTTP-ЗАГОЛОВКИ	5
4. ОПИСАНИЕ ОПЕРАЦИЙ	6
4.1 PAYMENT.FEES	6
4.1.2 ФОРМАТ ЗАПРОСА (ПРИМЕР)	6
4.1.3 ФОРМАТ ОТВЕТА (ПРИМЕР)	7
4.2 PAYMENT.STATUS	7
4.2.2 ФОРМАТ ЗАПРОСА (ПРИМЕР)	7
4.2.3 ФОРМАТ ОТВЕТА (ПРИМЕР)	7
5. АВАРИЙНЫЕ СИТУАЦИИ	8
6. РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ	8

1. Введение

1.1 Наименование программы

Наименование программного обеспечения “Сервис для расчета комиссий платежа”.

1.2 Область применения

Программное обеспечение “Сервис для расчета комиссий платежа” предназначено для интеграции и дальнейшей возможности использования у любого Заказчика.

ПО должно представлять собой техническое решение, которое в свою очередь позволит производить расчет комиссии платежей для конечного пользователя.

1.3 Глоссарий

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 - совокупность компьютерных программ и связанных с ними данных, которая содержит инструкции по указанию компьютеру, что и как делать.

БД - база данных - набор структурированных данных, хранящихся в виде таблицы.

API - программный интерфейс приложения – это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными.

Очередь - 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.

Метод - это функция или процедура, принадлежащая какому-то классу или объекту.

Мерчант - партнер, пользователь программного обеспечения «Сервис для расчета комиссий платежа».

Инстанс - экземпляр класса (объекта) в объектно-ориентированном программировании.

Идентификатор платежа - это числовой код, который позволяет идентифицировать конкретный платёж среди других аналогичных платежей.

1.4 Функции и возможности сервиса

1. Расчет комиссии платежа;
2. Получение платежа и его комиссии;
3. Реализация возможности безопасного хранения данных;
4. Реализация возможности масштабирования благодаря наличию ресурсов для запуска нескольких экземпляров сервиса;
5. Разработка штатного функционала, который в свою очередь позволит подписывать несколько инстансов в одну очередь, сообщения из которой будут случайным образом приходить в один из инстансов.
6. Передача данных в виде запросов в Базу данных и получение этих данных;
7. В связи с большим объемом и разнообразием информации, база данных систематизирует необходимые сведения в одном месте в едином формате.

1.5 Организация входных и выходных данных

1. Валидация полученной информации о размере комиссии;
2. Передача данных в унифицированном формате в БД с дальнейшим получением ответа;
1. Проверка подлинности для подтверждения легитимности и подлинности запроса;
2. Сервис должен осуществлять передачу полученных данных из БД в соответствующий внешний сервис;
3. Необходимо реализовать возможность получения ответа от внешнего сервиса;
4. ПО должно модифицировать полученные данные от внешнего сервиса;
5. На выходе получаем ЭЦП к запросу.

1.6 Уровень подготовки пользователей

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

1.7 Перечень программной документации

- “Сервис для расчета комиссий платежа”. Техническое задание (ГОСТ 19.201-78);

- “Сервис для расчета комиссий платежа”. Руководство пользователя;
- “Сервис для расчета комиссий платежа”. Инструкция по развертыванию экземпляра;
- “Сервис для расчета комиссий платежа”. Описание функциональных характеристик;
- “Сервис для расчета комиссий платежа”. Описание жизненного цикла разработки;
- “Сервис для расчета комиссий платежа”. Инструкция для проверки в тестовой среде.

2. Назначение и требования для эксплуатации ПО

Для корректной работы программного обеспечения необходимо оборудованное рабочее место Пользователя с подключением к сети Интернет.

2.1 Функциональное назначение

“Сервис для расчета комиссий платежа” позволит производить расчет комиссии платежей для любого типа транзакций, для операций проводимых на стороне мерчанта (Заказчика) реализованных через платежную систему.

2.2 Эксплуатационное назначение

ПО интегрируется и эксплуатируется на стороне Заказчика.

2.3 Требования к эксплуатации серверной части

Обеспечение функционирования ПО серверной части “Сервис для расчета комиссий платежа” 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. Подготовка к работе

3.1 Формат запросов

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

3.1.1 HTTP-заголовки

Обязательные к передаче заголовки HTTP запроса:

```
Content-Type: application/json
```

```
X-Data-Hash
```

3.1.2 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 (берем исходную строку).
- `secret_key` - секретный ключ, выдаваемый при подключении приложения.

На выходе получаем ЭЦП к запросу.

Проверка производится следующим образом:

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

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

4. Описание операций

4.1 payment.fees

Данный метод необходим для расчета комиссии платежа.

Параметр	Тип	Обязательность	Описание
method	String	+	Метод для для расчета комиссии платежа - payment.fees
params	Object	+	Параметры для метода
params.payments.c_d	Number	+	Идентификатор платежа на стороне мерчанта
params.payments.p_d	String	-	Идентификатор платежа на стороне платежной системы
params.operation.fees[].value	Number	+	Сумма комиссии (в копейках)
params.operation.fees[].currency	String	+	Валюта комиссии (ISO 4217)
service_id	Number	+	Для данного метода service_id можно использовать 2501 (зависит от платежной системы)

4.1.2 Формат запроса (пример)

```
{
  "service_id": 2501,
  "method": "payment.fees",
  "params": {
    "payment": {
      "identifiers": {
        "c_id": 3988515532472,
        "p_id": 12345
      }
    },
    "operation": {
      "fees": [
        {
          "value": 100,
          "currency": "RUB"
        }
      ]
    }
  }
}
```

4.1.3 Формат ответа (пример)

```
{"success":true,"next":null,"request_id":"-wPv5yQwerlx32ODuU6xX","processing_time":155}
```

4.2 payment.status

Данный метод необходим для получения платежа и его комиссии.

Параметр	Тип	Обязательность	Описание
method	String	+	Метод для для расчета комиссии платежа - payment.fees
params	Object	+	Параметры для метода
params.payments.c_d	Number	+	Идентификатор платежа на стороне мерчанта
service_id	Number	+	Для данного метода service_id 0

4.2.2 Формат запроса (пример)

```
{"service_id":0,"method":"payment.status","params":{"payment":{"identifiers":{"c_id":1}}}}
```

4.2.3 Формат ответа (пример)

```
{"success":true,"result":{"payment":{"amount":{"value":99992,"currency":"RUB"},"description":null,"identifiers":{"c_id":9999,"h_id":38,"p_id":null},"status":{"status":"success","final":true,"success":true,"error":null,"history":[{"status":"created","final":false,"success":null,"created":"2019-06-06T13:49:47.470Z","reason":null,"amount":"99992"}],"status":"success","final":true,"success":true,"created":"2019-06-06T13:49:47.516Z","reason":{"code":2000,"message":"Wrote..."},"amount":"99992"}}, "timestamps":{"created":"2019-06-06T13:49:47.470Z","updated":"2019-06-06T13:49:47.516Z","finished":"2019-06-06T13:49:47.516Z"},"destination":"in","operations":[{"id":"20","operation_type":"payme
```

```
nt.in", "amount": {"value": 1000, "currency": "RUB"}, "timestamps": {"created": "2019-06-06T13:49:47.554Z", "updated": "2019-06-06T13:49:47.554Z", "finished": null}, "status": {"status": "success", "final": true, "success": true, "error": null}, "fees": [{"type": "FROM_CUSTOMER_TO_USER", "currency": "RUB", "value": 50}], "next": null, "request_id": "gIPml9LmvJh2mTemS4MeB", "processing_time": 45}
```

5. Аварийные ситуации

В случае аварийных ситуаций обращаться в техническую поддержку.

6. Рекомендации по освоению

Для успешной работы с ПО “Сервис для расчета комиссий платежа” 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 расчета комиссий платежа””;

Ознакомиться с документом “Разворачивание экземпляра ПО “Сервис для расчета комиссий платежа””.