

Сведения об эксплуатации экземпляра программы

Система управления платежами Forward PMS

Правообладатель:
ООО «Форвард-Телеком»

Forward PMS (Payment Management System) – Система управления платежами, представляет собой распределенную масштабируемую систему для организации и автоматизации процессов приема и обработки платежей, поддерживающую интеграцию с внешними платёжными системами для приёма транзакций, управления жизненным циклом платежей (создание, изменение, удаление, получение статуса, поиск), а также загрузки и обработки реестров платежей.

Система обеспечивает поддержку работы с рекуррентными платежами и мёрчантами, предоставляет гибкие интерфейсы для настройки бизнес-логики и интеграции с внешними учётными и биллинговыми системами, а также реализует механизмы мониторинга, журналирования и анализа операций по платёжным событиям различной структуры.

Использование данного решения позволяет операторам существенно снизить затраты на интеграцию с различными платёжными провайдерами, автоматизацию обработки транзакций, снижение числа ошибок при ручной обработке данных, а также ускорить запуск новых платёжных сценариев за счёт гибкой настройки бизнес-правил и процессов.

Контактная информация: support@fw-t.ru

1 Общая информация

Forward PMS обеспечивает организацию и автоматизацию процессов:

- Приема платежей;
- Обработки платежей,

и поддерживает интеграцию с внешними платёжными системами для приёма транзакций, управления жизненным циклом платежей (создание, изменение, удаление, получение статуса, поиск), а также загрузки и обработки реестров платежей.

Forward PMS обеспечивает поддержку работы с:

- рекуррентными платежами;
- мёрчантами,

предоставляет гибкие интерфейсы для настройки бизнес-логики и интеграции с внешними учётными и биллинговыми системами, а также реализует механизмы мониторинга, журналирования и анализа операций по платёжным событиям различной структуры.

Примечания:

Параметры приложения публикуются автоматически в документ вида «Finances properties (generated)»

Коды ошибок публикуются автоматически в документ вида «Finances Error Codes (generated)»

Forward PMS состоит из нескольких подсистем, активируемых профилями приложения:

- Payscale – активирует подсистему приёма онлайн платежей. В данном режиме активируются платёжные сервлеты;
- Registers-load – активирует подсистему автоматической загрузки реестров платежей;
- Export-kafka – активирует подсистему выгрузки платежей в Kafka;
- Recurrents – активирует подсистему рекуррентных платежей.

Примечание:

При конфигурации приложения указываются те профили системы, которые должны быть включены. Все профили, доступные для активации, перечислены в документе «Finances properties (generated)» – параметр `spring.profiles.active`.

2 Эксплуатация системы

Основные, активированные при конфигурации, профили:

- Payscale – активирует подсистему приёма онлайн платежей. В данном режиме активируются платёжные сервлеты;
- Registers-load – активирует подсистему автоматической загрузки реестров платежей;
- Export-kafka – активирует подсистему выгрузки платежей в Kafka;
- Recurrents – активирует подсистему рекуррентных платежей,

формируют распределённую масштабируемую систему управления платежами Forward PMS.

Для корректной работы PMS выполняются проверки наличия абонента. Для этого используется модуль Subscriber – информация о создании/удалении платежа отправляется в виде событий в Kafka.

2.1 Профили PMS

2.1.1 Профиль: **payscale**

Данный профиль активирует подсистему приёма онлайн платежей. В этом режиме активируются платёжные сервлеты.

Доступные платёжные системы, поддерживаемые действия:

- проверка возможности платежа;
- платёж;
- возврат;
- удаление платежа.

При активации данного профиля нужно описать настройки поддерживаемых платёжных систем в виде массива JSON: данный конфиг или передается в приложение в виде текста в параметре `app.payscale.json`, или в виде файла, путь к которому указывается в параметре `app.payscale.json-path`. Первая настройка, в виде текста, имеет приоритет.

При вызове платёжной системы предполагается использование уровня прокси (в случае K8 – это может быть Ingress).

При вызове платёжного URL проксирующий слой должен передать заголовок `X-PaymentSystem`, в котором должен быть указан уникальный идентификатор платёжной системы.

Важно! Уникальность платежей из платёжных систем проверяется по идентификатору платежа в рамках платёжной системы (для каждой платёжной системы он свой) + источник платежа (в конфиге json это paymentSourceIdent).

Таким образом, например, возможно настроить две разные платёжные системы, принимающих платежи от разных организаций по одному и тому-же протоколу

Примечание:

Активация новой платёжной системы производится путём модификации json файла активных платёжных систем и последующим рестартом сервиса. При работе под управлением K8, рекомендуется запускать 2 и более экземпляров приложения для обеспечения бесшовного рестарта сервиса.

2.1.2 Профиль: registers-load

Данный профиль активирует подсистему автоматической загрузки реестров платежей.

Аналогично платёжным системам, активные загрузчики онлайн реестров платежей описываются в json формате и передаётся либо в виде текста в параметре app.registers-load.json, или в виде файла, пусть к которому указывается в параметре app.registers-load.json-path. Первая настройка, в виде текста, имеет приоритет.

Для парсера указываются три директории:

- dirIn - директория, в которой собираются входные файлы реестров платежей
- dirError - директория, в которую выгружаются реестры платежей, обработанные с ошибкой во время парсинга
- dirDone - директория, в которую перемещаются успешно обработанные реестры платежей

Алгоритм работы:

- Периодически сканируются входные директории
- Вычисляется hash-сумма каждого нового файла (дедубликация)
- Реестр платежей парсится и сохраняется в таблицу платежей
- Производится работа с модулем Subscriber для определения целевых получателей платежа
- Платежи, для которых был определён получатель, отправляются в Kafka / SubscriberOra

2.1.3 Профиль: export-kafka

Данный профиль активирует подсистему выгрузки платежей в Kafka.

Одновременный запуск нескольких экземпляров процесса не является проблемой – процессы между собой согласовываются (через БД), кто именно будет выполнять

выгрузку в определённый момент времени. Для этого используется библиотека shedlock.

Выгрузка платежей в кафку имеет смысл, если задано значение настройки:

`app.scheduler.payments.exportTo=Kafka`

2.1.4 Профиль: **recurrents**

Данный профиль активирует подсистему рекуррентных платежей.

Конфиг активированных подсистем приёма рекуррентных платежей также задаётся в виде json. Данный конфиг нужно или передать в приложение в виде текста в параметре `app.recurrents.json`, или в виде файла, путь к которому указывается в параметре `app.recurrents.json-path`. Первая настройка, в виде текста, имеет приоритет.

Описание параметров:

- `merchant.name` - Название мерчанта
- `merchant.ident` - Внешний идентификатор мерчанта (уникальный)
- `paymentSourceIdent` - Источник платежа
- `description` - Описание источника платежа
- `currencyCode` - Валюта
- `protocol` - Платежный протокол
- `options` - Набор параметров для мерчанта (пример: логин, пароль для авторизации в платёжную систему)

2.1.4.1 Отправка запроса на списание рекуррентного платежа

2.1.4.1.1 HOST:PORT/recurrent/queue (POST)

При успешном запросе будет возвращён `operationId` (формат - UUID) для каждого запроса на списание рекуррентного платежа.

По данному `operationId` можно будет увидеть в аудите\логах\таблицах каждый пройденный этап.

Параметры:

- `contractIdent` - контракт, у которого будет произведено списание
- `createdAt` - дата в формате 'yyyyMMddHHmmss'
- `sum` - сумма
- `ident` - внешний идентификатор мерчанта
- `parameters` - набор параметров передаваемых в транзакцию (формат JSON)

Пример тела запроса:

[

```

    {
      "contractIdent": "rc-contract1",
      "sum": 10221.1,
      "createdAt": "20180405000000",
      "merchant": {
        "ident": "CloudPaymentsIdent"
      },
      "parameters": {
        "Phone": "89991234567"
      }
    }
  ]

```

2.2 Интеграция с модулем Subscriber

Модуль финансов нуждается в модуле Subscriber для проверки наличия абонента. Информация о создании/удалении платежа отправляется в виде событий в Kafka.

2.2.1 Отправка событий в Kafka

Для отправки в виде события в Kafka необходимо запустить модуль финансов в режиме

`app.scheduler.payments.exportTo=Kafka`

а также активировать профиль `export-kafka`.

В таком случае внешние сервисы получают информацию о создании/удалении платежа из Kafka.

Данный метод будет основным при работе в микросервисной среде.

В кафку при этом будет отправлено сообщение примерно со следующим телом запроса:

```

{
  "operationId": "Fin-39f35702-34d2-4a1e-b859-1d3a83ea3ead",
  "ident": "dd2",
  "status": "A",
  "sum": 120.4561,
  "contractIdentOriginal": "rc-contract2",
  "contractIdent": "rc-contract2",
  "dateReal": "1682599230000",
  "currency": {
    "code": "USD",
    "name": "Доллар США"
  },
  "paymentSource": {
    "ident": "osmp",
    "name": "Приём платежей организации 1 через ОСМП"
  },
}

```

```

    "paymentMethod": {
      "type": "onlinePayment",
      "onlinePayment": {
        "paymentSystemIdent": "osmp-2"
      }
    }
  }
}

```

Топик, в который модуль финансов будет отправлять события о платежах, задаётся в параметре

```
app.scheduler.payments.topic=payments-flow
```

Процедура - обработчик, которой разбирается очередь событий в биллинге:
 PACK_MS_FINANCES.importPayment (из опции Биллинга
 option_microservices_integration).

Приложения

Приложение 1. Стандартные параметры приложения

Опция	Значение по умолчанию	Комментарий
spring.application.name	Finances	Название приложения
server.port	8080	Порт приложения
spring.datasource.hikari.connection-timeout	5000	Таймаут подключения к БД
spring.datasource.hikari.maximum-pool-size	5	Размер пула подключений
spring.jpa.properties.hibernate.jdbc.batch_size	30	Размер пачки выполнения batch операций в БД
grpc.client.subscriber.address	static://localhost:8051	Путь к gRPC сервису сабскрайбинга
spring.profiles.active	paysystems,export-kafka,export-subscriber-ora,registers-load,recurrents	Профили приложения активируют разные аспекты работы модуля: • paysystems - интеграция с платёжными системами (платёжные сервлеты) • export-kafka - выгрузка информации о платежах в Kafka • export-subscriber-ora - выгрузка платежей напрямую в сабскрайбер на Оракле • registers-load - модуль загрузки реестров платежей (сканирование директорий) • recurrents - подсистема приёма рекуррентных платежей
app.default.lockAtMostFor	PT10M	Максимальное время, на которое устанавливается распределённая блокировка shedlock
app.default.currency	RUB	Валюта по умолчанию
app.scheduler.initialDelay	PT5S	Время задержки перед

Опция	Значение по умолчанию	Комментарий
		первым выполнением задачи
app.paysystems.header-real-ip	X-Real-IP	Http header, указывающий на реальный IP
app.paysystems.header-payment-system-ident	X-PaymentSystem	Заголовок с названием платёжной системы, в сторону которой идёт обращение
app.paysystems.audit-requests	true	Признак записи в аудит платёжных систем - таблица payment_system_audit
app.paysystems.skip-save-headers	accept,accept-encoding,connection,cookie,postman-token	Список заголовков http запроса, которые не будут сохранены в аудит
app.paysystems.json		Текст конфигурационного файла активных платёжных систем в формате json
app.paysystems.json-path	classpath:paysystems.json	Путь к конфигурационному файлу активных платёжных систем в формате json
app.scheduler.payments.topic	payments-flow	Название топика Kafka для отправки событий о приёме/удалении платежа
app.scheduler.payments.exportFixedRate	PT5S	Частота запуска шедулера - экспорт платежей в Kafka
app.scheduler.payments.lockAtLeastFor	PT5S	Минимальное время блокировки шедулера - экспорт платежей в Kafka
app.scheduler.payments.exportTo	Kafka,SubscriberOra	Системы для экспорта платежей • Kafka - выгрузка событий в Kafka. Выгрузку осуществляет процесс finances с профилем export-kafka • SubscriberOra - выгрузка событий в Subscriber-Ora. Выгрузку осуществляет процесс finances с профилем export-subscriber-ora

Опция	Значение по умолчанию	Комментарий
app.registers-load.threads	5	Кол-во потоков загрузки реестров платежей
app.registers-load.loadFixedRate	PT5S	Частота запуска шедулера - загрузка реестров платежей
app.registers-load.lockAtLeastFor	PT5S	Минимальное время блокировки шедулера - загрузка реестров платежей
app.registers-load.maximumFileLock	PT5M	Максимальное время обработки файла реестра платежей. Если обработка файла не закончена в указанное время - файл будет взят на обработку другим потоком.
app.registers-load.fillContractRegCreated	PT72H	Заполнять контракт для реестров, созданных в указанный период от текущего момента
app.registers-load.fillContractCheckDelay	PT1H	Задержка повторной проверки контракта
app.registers-load.json		Текст конфигурационного файла активных загрузчиков реестров платежей в формате json
app.registers-load.json-path	classpath:registers.json	Путь к конфигурационному файлу активных загрузчиков реестров платежей в формате json
app.registers-lead.leadFixedRate	PT5S	Частота запуска шедулера - поиск контрактов для платежа реестра
app.registers-lead.lockAtLeastFor	PT5S	Минимальное время блокировки шедулера - поиск контрактов для платежей реестра
app.recurrents.json		Текст конфигурационного файла активных обработчиков рекуррентных платежей в формате json
app.recurrents.json-path	classpath:recurrents.json	Путь к конфигурационному файлу активных

Опция	Значение по умолчанию	Комментарий
		обработчиков рекуррентных платежей в формате json
app.recurrent.max-retry-count	3	Кол-во попыток выполнения рекуррентного платежа при возникновении ошибок
app.cloudpayments.base-url	https://api.cloudpayments.ru	Адрес сервиса cloudpayments
app.paymaster.base-url	https://paymaster.ru/api/v2	Адрес сервиса paymaster
app.yookassa.base-url	https://api.yookassa.ru	Адрес сервиса yookassa
app.recurrent.scheduler.initialDelay	PT5S	Время задержки перед первым выполнением задачи
app.recurrent.recurrentFixedRate	PT5S	Выражение для фиксированного выполнения задачи (обработка очереди рекуррентных платежей)
app.recurrent.lockAtLeastFor	PT5S	Время блокировки обработчика задач (обработка очереди рекуррентных платежей)
app.recurrent.recurrentProcessedRequestsFixedRate	PT5S	Выражение для фиксированного выполнения задачи (обработка рекуррентных платежей с зависшим статусом)
app.recurrent.token.scheduler.initialDelay	PT5S	Время задержки перед первым выполнением задачи (обработка очереди токенов)
app.recurrent.token.recurrentFixedRate	PT5S	Выражение для фиксированного выполнения задачи (обработка очереди токенов)
app.recurrent.token.lockAtLeastFor	PT5S	Время блокировки обработчика задач (обработка очереди токенов)
app.scheduler.paymentTokens.topic	paymentTokens-flow	Название топика, по которому будет создано

Опция	Значение по умолчанию	Комментарий
		событие в кафку о токене
app.scheduler.paymentTokens.exportFixedRate	PT5S	Выражение для фиксированного выполнения задачи (экспорт токенов в кафку)
app.scheduler.paymentTokens.lockAtLeastFor	PT5S	Время блокировки обработчика задач (экспорт токенов в кафку)

Приложение 2. Стандартные коды ошибок

Код ошибки	Текст ошибки	Комментарий	Действие
FINANCES-00001	Invalid app configuration: {0}	Необходимо исправить ошибки конфигурации системы	Исправить ошибки конфигурации системы
FINANCES-00002	Unknown currency '{0}'	Указанная валюта не поддерживается. Проверьте настройки системы.	Завести обращение
FINANCES-00003	Subscriber gRPC call status returned: {0} ({1})	Ошибка выполнения операции на сервисе subscriber по протоколу gRPC	Проверить настройки подключения к сервису subscriber
FINANCES-00004	Requested action requires application profile '{0}' to be activated	Запрошенное действие не может быть выполнено, так как требуемый профиль приложения не активирован	Проверить активные профили приложения - настройка spring.profiles.active
FINANCES-00005	Recurrent notification was before than recurrent payment was created	Нотификация пришла раньше чем был создан рекуррентный платеж, или такого платежа вовсе нет	дождаться следующей нотификации в случае наличия платежа
FINANCES-00006	Contract not found by identifier {0}	Указанный контракт не найден	убедитесь в правильном вводе контракта
FINANCES-00100	Exchange response object cannot be null	Процессор платежной системы вернул пустой ответ	Завести обращение
FINANCES-00101	Internal error - active payment not found for active online payment from payment system '{0}' and identifier '{1}'	В системе числится активный платёж от платёжной системы, но при этом не найден порождённый активный платёж	Завести обращение
FINANCES-00103	Payment system '{0}' working via protocol '{1}' does not support settings	Класс, имплементирующий платёжный протокол, запросил настройки, хотя протокол их не предполагает	Завести обращение

Код ошибки	Текст ошибки	Комментарий	Действие
FINANCES-00104	Unknown operation '{0}' for payment system '{1}'	Запрошена неизвестная операция платёжной системы	Проверить параметры запросе, пришедшего на уровень приложения
FINANCES-00105	Unknown operation code '{0}' for payment system '{1}'	Запрошена операция платёжной системы с неизвестным кодом	Проверить параметры запросе, пришедшего на уровень приложения
FINANCES-00106	Address {0} implements protocol {1} but payment system {2} is configured for protocol {3}	Запрос отправлен на сервлет, работающий по одному протоколу, при этом X-PaymentSystem передан от другого протокола	Проверить адрес, на который поступают запросы
FINANCES-00107	Error loading public key	Ошибка чтения публичного ключа платёжной системы	Проверить настройки платёжной системы
FINANCES-00108	Error loading private key	Ошибка чтения приватного ключа платёжной системы	Проверить настройки платёжной системы
FINANCES-00109	Signature check error: {0}	Ошибка проверки цифровой подписи запроса	Проверить настройки платёжной системы
FINANCES-00110	Error when generating response signature: {0}	Ошибка генерации цифровой подписи ответа	Проверить настройки платёжной системы
FINANCES-00111	Merchant '{0}' working via protocol '{1}' does not support settings	Класс, имплементирующий рекуррентный протокол, запросил настройки, хотя протокол их не предполагает	Завести обращение
FINANCES-00112	Not found payment system definition with ident: '{0}', specified at '{1}' HTTP header	При вызове платёжной системы передан заголовок с незарегистрированной платёжной системой	Проверить настройки системы и параметры вызова
FINANCES-00200	Unable to create {0} register working directories - {1}: {2}	Приложение не смогло создать рабочие директории загрузки реестров	Проверить настройки загрузки реестров и повторить запуск
FINANCES-00300	Not found recurrent definition with merchant ident: '{0}'	Не найдено описание мерчанта для работы с подсистемой рекуррентных платежей	Проверить настройки системы и параметры вызова
FINANCES-00400	Failed to read request: {0}	Невалидный формат входящего запроса, например - нечитаемый json	
FINANCES-00401	Invalid request params: {0}	Один или несколько входных параметров - невалидны	Исправить ошибки и повторить вызов