



**Обслуживание
программного обеспечения
«Процессинговый центр Pay-logic»
Руководство администратора**

АННОТАЦИЯ

Описывает процедуры, выполняемые при обслуживании программного обеспечения «Процессинговый центр Pay-logic» и программного обеспечения, необходимого для его функционирования

Версия руководства: 1.7

Руководство актуально для ПО «Процессинговый Центр Pay-logic» версий 5.0.x

2008–2025 ООО «Софт-Лоджик», г. Барнаул, Россия

Данный документ входит в комплект поставки программных продуктов.

Права использования данного документа предусмотрены соответствующим лицензионным договором.

ООО «Софт-Лоджик»

656006, г. Барнаул, Малахова ул., дом 146в

Тел: (3852) 72-27-27

© *Soft-logic*

Web: <https://soft-logic.ru/>

Mail: info@soft-logic.ru

ОГЛАВЛЕНИЕ

ИСТОРИЯ ИЗМЕНЕНИЙ.....	5
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.....	5
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.....	5
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.2.....	5
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.3.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.4.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.5.....	7
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.6.....	7
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.7.....	7
1 ОСНОВНЫЕ ТЕРМИНЫ И СОКРАЩЕНИЯ.....	8
2 ВВЕДЕНИЕ.....	9
3 ПРЕДПОЛАГАЕМАЯ АУДИТОРИЯ.....	10
4 НАСТРОЙКА СИСТЕМЫ ШЛЮЗОВ.....	11
4.1 ОБЩИЕ СВЕДЕНИЯ ПО АРХИТЕКТУРЕ СИСТЕМЫ ШЛЮЗОВ.....	11
4.2 ДОБАВЛЕНИЕ НОВЫХ ШЛЮЗОВ.....	14
4.3 НАСТРОЙКА СИСТЕМЫ ОПОВЕЩЕНИЙ.....	16
4.3.1 ОБЩИЕ СВЕДЕНИЯ.....	16
4.3.2 SMS-РАССЫЛКА.....	19
4.3.3 EMAIL-РАССЫЛКА.....	22
4.3.4 TELEGRAM.....	23
4.3.5 TELEGRAMVOT.....	24
4.4 НАСТРОЙКА ОТПРАВКИ РЕЕСТРОВ.....	25
4.5 ПОРЯДОК СМЕНЫ РЕКВИЗИТОВ АВТОРИЗАЦИИ НА ВНЕШНИХ СИСТЕМАХ.....	26
4.6 НАСТРОЙКА ШЛЮЗА, ВЫПОЛНЯЮЩЕГО ОТЛОЖЕННЫЕ ЗАДАНИЯ ОБНОВЛЕНИЯ СПРАВОЧНИКОВ.....	27
4.7 ПОРЯДОК ЗАПУСКА МОДУЛЯ ПРЕДВАРИТЕЛЬНОЙ ПРОВЕРКИ.....	29
4.8 РАСЧЕТ ВОЗНАГРАЖДЕНИЙ В ВАЛЮТЕ ПОСТАВЩИКА.....	30

5 СИСТЕМА ЖУРНАЛИРОВАНИЯ.....	32
5.1 СИСТЕМА ЖУРНАЛИРОВАНИЯ ШЛЮЗОВ.....	32
5.2 СИСТЕМА ЖУРНАЛИРОВАНИЯ WEB-ПРИЛОЖЕНИЙ.....	34
5.3 СИСТЕМА ЖУРНАЛИРОВАНИЯ ДРУГИХ КОМПОНЕНТОВ ПРОЦЕССИНГА.....	35
5.3.1 СТРУКТУРА ЛОГОВ ПРОЦЕССИНГА.....	35
5.3.2 ОБЩЕСИСТЕМНЫЕ ЛОГИ.....	36
5.3.3 ЛОГИ ПРИЛОЖЕНИЙ, РАБОТАЮЩИХ НА GF.....	37
5.3.4 ЛОГИ ПОДСИСТЕМЫ ШЛЮЗОВ.....	40
6 ОБНОВЛЕНИЕ ПРОГРАММНОГО КОМПЛЕКСА.....	45
6.1 ОБЩИЕ СВЕДЕНИЯ.....	45
6.2 ПОДГОТОВКА К ОБНОВЛЕНИЮ.....	46
6.3 ОБНОВЛЕНИЕ СТРУКТУРЫ БД.....	49
6.3.1 ОШИБКИ ПРИ ОБНОВЛЕНИИ БАЗЫ ДАННЫХ.....	50
6.4 ОБНОВЛЕНИЕ СИСТЕМЫ ШЛЮЗОВ.....	51
6.5 ОБНОВЛЕНИЕ WEB-ПРИЛОЖЕНИЙ.....	52
6.6 ОБНОВЛЕНИЕ СТАТИКИ WEB-СЕРВЕРА.....	53
7 ПРОВЕРКА СИСТЕМЫ.....	54
7.1 ПРОВЕРКА СОСТОЯНИЯ КОМПОНЕНТОВ СИСТЕМЫ.....	54
8 РЕКОМЕНДАЦИИ ПО ВЫХОДУ ИЗ АВАРИЙНЫХ СИТУАЦИЙ.....	56
8.1 ОТКАЗ В РАБОТЕ СЕРВЕРА ПРИЛОЖЕНИЙ GLASSFISH ИЛИ HTTPD.....	56
8.2 ОТКАЗ В РАБОТЕ СУБД.....	60
8.3 ОТКАЗ В РАБОТЕ СИСТЕМЫ ШЛЮЗОВ.....	62

ИСТОРИЯ ИЗМЕНЕНИЙ

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0

Дата публикации: 28.04.2017.

Изменение	Раздел
Общие улучшения:	
Документ создан	-

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1

Дата публикации: 04.08.2017.

Изменение	Раздел
Новый функционал в версии 4.5.0:	
Добавлен механизм отложенного выполнения заданий	4.6

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.2

Дата публикации: 13.07.2018.

Изменение	Раздел
Улучшения в версии 4.6.4:	
Добавлена возможность разделения логгирования по точкам	4.2

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.3

Дата публикации: 22.08.2018.

Изменение	Раздел
Дополнения в документации к ранее имевшемуся функционалу:	
Добавлено описание системы журналирования компонентов процессинга	5.3

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.4

Дата публикации: 26.10.2018.

Изменение	Раздел
Улучшения в версии 4.6.8	
Добавлена возможность расчета вознаграждений в валюте поставщика	4.8
Дополнения в документации к ранее имевшемуся функционалу:	
Добавлено описание запуска модуля предварительной проверки	4.7

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.5

Дата публикации: 22.02.2019.

Изменение	Раздел
Исправленные в документации несоответствия и ошибки:	
Исправлено значение параметра <code>param.transport-id</code> для Email-рассылки	4.3.3

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.6

Дата публикации: 23.10.2019.

Изменение	Раздел
Общие улучшения в документации:	
Дополнено описание процедуры обновления программного комплекса	6

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.7

Дата публикации: 05.11.2025.

Изменение	Раздел
Улучшения в версии 5.8.6:	
Реализован метод проверки работоспособности системы <code>healthcheck</code>	7.1

1 ОСНОВНЫЕ ТЕРМИНЫ И СОКРАЩЕНИЯ

БД — база данных.

ОС — операционная система.

Платежный шлюз (шлюз) — это программный модуль, обеспечивающий взаимодействие с поставщиком согласно протоколу.

ПО — программное обеспечение.

Платежная система (ПС) — юридическое лицо, самостоятельно предоставляющее возможность оплачивать свои услуги, либо юридическое лицо, предоставляющее возможность оплачивать услуги других компаний.

СУБД — система управления базами данных.

Точка приема платежей — программно-аппаратное устройство, с помощью которого (через которое) осуществляется прием платежа.

2 ВВЕДЕНИЕ

Данное руководство описывает процедуры, выполняемые при обслуживании ПО «Процессинговый центр Pay-logic» версии 4.x.x и программного обеспечения, необходимого для его функционирования.

К таким процедурам относятся:

1. Добавление новых шлюзов в систему и их настройка (раздел [4](#)).
2. Настройка системы оповещений и отправки реестров (раздел [4.3](#)).
3. Смена реквизитов авторизации на внешних системах (раздел [4.5](#)).
4. Анализ журналов событий системы (раздел [5](#)).
5. Обновление программного комплекса (раздел [6](#)).
6. Решение аварийных ситуаций (раздел [8](#)).

3 ПРЕДПОЛАГАЕМАЯ АУДИТОРИЯ

Предполагается, что обслуживание будет осуществляться человеком, знакомым со следующими областями:

1. JEE deployment (на уровне знания базовых единиц развертывания JEE, настройки DataSource, connection pools и т. п.).
2. Базовое знакомство с Glassfish Application Server V3.
3. PostgreSQL administration (на уровне заведения пользователей, знания базовых настроек postgresql.conf и pg_hba.conf).
4. Знакомство с настройкой Apache Httpd сервер.

Помимо этого, для обслуживания ПО, развернутого на ОС, отличных от Windows (Solaris, Linux, etc.), необходимы знания и навыки работы с shell-интерпретатором и знание базовых Unix-команд.

4 НАСТРОЙКА СИСТЕМЫ ШЛЮЗОВ

4.1 ОБЩИЕ СВЕДЕНИЯ ПО АРХИТЕКТУРЕ СИСТЕМЫ ШЛЮЗОВ

Система шлюзов находится в папке `/home/gates` и имеет следующую структуру:

1. `./configs` — папка с конфигурационными файлами шлюзов.
2. `./ext` — папка с библиотеками отдельных шлюзов.
3. `./lib` — папка с системными библиотеками системы шлюзов.
4. `./log` — папка с журналами системы шлюзов.
5. `./Gates.jar` — ядро системы шлюзов.
6. `./config.xml` — основной конфигурационный файл системы шлюзов.
7. `./log4j.properties` — файл настроек системы журналирования.

Основной конфигурационный файл имеет следующую структуру:

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <!-- Параметры подключения к СУБД (с3р0) -->
  <!-- Для задания параметров пула воспользуйтесь средствами с3р0 -->
  <!-- (xml файл, property файл) -->
  <data-source
    driver-class="org.postgresql.Driver"
    user="postgres"
    password="123"
    dsn="jdbc:postgresql://192.168.0.10/work"
  />

  <!-- Шлюзы -->
  <gates>
```

```
<gate enable="true"
      factory="ru.softlogic.processing.gates.universal.Factory"
      id-provider="1690" config="/home/gates/configs/gate/gate.xml"/>
</gates>

<!-- Конфигурация шлюзов поддержки -->
<support config="/home/gates/configs/support.xml"/>

<!-- Таймаут обработки оффлайн платежей -->
<offline timeout="20"/>
</config>
```

В первом разделе указываются параметры доступа к СУБД PostgreSQL, в частности указывается:

1. **user** — имя пользователя (postgres)
2. **password** — пароль (если нет, задается любой)
3. **dsn** — URL для доступа к БД (задается IP, порт и название базы данных).

Далее следует раздел шлюзов, в котором каждому шлюзу соответствует элемент `<gate>`:

1. **enable** — флаг активности шлюза.
2. **factory** — названия класса фабрики шлюза (предоставляется разработчиками для каждого шлюза отдельно).
3. **id-provider** — ID поставщика услуг в системе (раздел в кабинете «Провайдеры — Провайдеры»).
4. **config** — путь к конфигурационному файлу шлюза (предоставляется разработчиками).

Секция `<support>` служит для указания файла настройки системы оповещений (см. раздел [4.3](#)).

Секция `<offline>` доступна с версии 4.4.2 и служит для указания таймаута обработки оффлайн-платежей. Допустимые значения находятся в диапазоне от 5 до 3600 сек. Если секция не указана, то таймаут составляет 300 сек.

4.2 ДОБАВЛЕНИЕ НОВЫХ ШЛЮЗОВ

При добавлении в систему новых шлюзов, разработчиками компании Soft-logic выдаются следующие файлы (если шлюз не основан на универсальном):

1. Модуль шлюза — jar-файл, например *beeline.jar*.
2. Пример конфигурационного файл шлюза, например, *beeline.xml*.
3. Название фабрики шлюза (factory), а также название логгера шлюза.
4. В случае необходимости, дополнительные инструкции.

Модуль шлюза необходимо скопировать в папку */home/gates/ext/*.
Конфигурационный файл в папку */home/gates/configs/<название шлюза>/*.

Следующим шагом необходимо добавить ссылку на шлюз в основной конфигурационный файл (раздел *<gates>*).

Последним шагом настраивается журналирование работы шлюза. Для этого в файл *log4j.properties* необходимо добавить секцию следующего содержания:

```
log4j.logger.beeline=debug, beeline
log4j.appender.beeline =org.apache.log4j.DailyRollingFileAppender
log4j.appender.beeline.DatePattern='.'yyyy-MM-dd
log4j.appender.beeline.File=${app.dir}/log/gate/beeline/beeline .log
log4j.appender.beeline.layout=org.apache.log4j.PatternLayout
log4j.appender.beeline.layout.conversionPattern=%d{ABSOLUTE} %5p %t
%c{1}:%M:%L - %m%n
```

При добавлении имя *beeline* необходимо заменить на название логгера шлюза.

В версии 4.6.4 добавлена возможность разделения логирования по точкам.

Для настройки:

1. Задайте логгер в `log4j.properties` в домене glassfish:

- 1) имя логгера должно быть в формате `p<id-точки>`. Например, `id` точки — 1853, имя логгера — `p1583`;
- 2) если конфигурация логгера не заполнена, то информация попадёт в общий лог;
- 3) настройки логгера точки следует указывать в файле `log4j.properties`, который расположен в конфигурации домена, в котором запущен `external paylogic2`.

2. Для применения конфигурационного файла перезагрузите glassfish.

Пример логгера:

```
log4j.logger.p243=info, p243
log4j.appender.p243=org.apache.log4j.DailyRollingFileAppender
log4j.appender.p243.DatePattern='.'yyyy-MM-dd
log4j.appender.p243.File=../paylogic2/points/243.log
```

4.3 НАСТРОЙКА СИСТЕМЫ ОПОВЕЩЕНИЙ

4.3.1 ОБЩИЕ СВЕДЕНИЯ

Система оповещений предназначена для рассылки служебных (технических, финансовых, и т.п.) сообщений пользователям по различным каналам: SMS, ICQ, XMPP, Email.

Конфигурационный файл называется *support.xml* и находится в том же каталоге, что и основной конфигурационный файл шлюза *config.xml* — */home/gates/*.

Общий вид конфигурационного файла:

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <!--      Конфигурация E-Mail-рассыльщика
  Рассыльщик работает на базе почтового сервиса google.
  Преимущества: Безопасная доставка почты (SSL)
                 Высокая надежность google-сервисов
  Если нужны другие сервера - необходимо связаться с разработчиками
  для написания соответствующего ПО      -->
  <email enable="false">
    <!-- Конфигурация ком-порта -->
    <auth user="" password=""/>
    <params>
      <!-- Id типа доставки из соответствующей таблицы -->
      <param name="transport-id" value="3"/>
      <!-- Id сервиса технического сопровождения клиента,
           по которому пойдет списание -->
      <param name="service-id" value="2"/>
    </params>
  </email>
  <!--      Конфигурация SMS-рассыльщика      -->
  <sms-gate enable="false">
```

```
        api="ru.softlogic.processing.gates.synapse.SynapseApi" >
<!-- Конфигурация ком-порта -->
<auth url="http://synapse.vectramind.com/API/SYNAPSEWS.ASMX?WSDL"
      login="Test" password="pay6141"
      source="synapse.vectramind.com"/>
<params>
  <param name="host" value="synapse.vectramind.com"/>
  <param name="port" value="80"/>
  <param name="url" param="/API/SYNAPSEWS.ASMX?WSDL"/>
  <param name="wsdlurl"
        param="http://synapse.vectramind.com/API/
        SYNAPSEWS.ASMX?WSDL"/>
  <param name="transport-id" value="1"/>
  <param name="service-id" value="2"/>
  <param name="senderId" value="Test"/>
  <param name="send-with-status" value="true"/>
</params>
</sms-gate>
<!--      Конфигурация SMS-рассыльщика для отправки рекламы      -->
<sms-gate-adv enable="false"
              api="ru.softlogic.processing.gates.synapse.SynapseApi">
<!-- Конфигурация ком-порта -->
<auth url="http://synapse.vectramind.com/API/SYNAPSEWS.ASMX?WSDL"
      login="Test" password="pay6141"/>
<params>
  <param name="host" value="synapse.vectramind.com"/>
  <param name="port" value="80"/>
  <param name="url" param="/API/SYNAPSEWS.ASMX?WSDL"/>
  <param name="wsdlurl"
        param="http://synapse.vectramind.com/API/
        SYNAPSEWS.ASMX?WSDL"/>
  <param name="transport-id" value="1"/>
  <param name="service-id" value="2"/>
  <param name="senderId" value="Test"/>
  <param name="send-with-status" value="true"/>
</params>
</sms-gate-adv>
<!--      Конфигурация Telegram-рассыльщика      -->
<telegram enable="true">
<!-- Стандартные параметры авторизации Telegram-клиента-->
<auth host="127.0.0.1" port="11555"/>
<params>
  <!-- Id типа доставки из соответствующей таблицы -->
```

```
<param name="transport-id" value="5"/>
<!-- Id сервиса технического сопровождения клиента, по
которому пойдет списание -->
<param name="service-id" value="2"/>
</params>
</telegram>
<!--      Конфигурация TelegramBot-рассыльщика      -->
<telegram-bot enable="true">
<!-- Стандартные параметры авторизации TelegramBot -->
<auth host="api.telegram.org" port="443" />
<params>
<!-- Id типа доставки из соответствующей таблицы -->
<param name="transport-id" value="6"/>
<!-- Путь до файла с текстовками,
отображаемыми TelegramBot -->
<param name="messages"
value="/home/gates_ng/bot_messages.properties"/>
</params>
</telegram-bot>
</config>
```

4.3.2 SMS-РАССЫЛКА

За настройку SMS-рассылки отвечает секция `<sms-gate>` — пример конфигурации приведен в разделе [4.3.1](#). SMS-рассылка осуществляется через SMS-шлюзы, расположенные в интернете. На данный момент реализовано несколько протоколов SMS-шлюзов, включая SMPP.

Для включения SMS-рассыльщика установите флаг **`enable=true`**. Параметры рассыльщика:

1. **`api`** — название класса API рассыльщика (уточняется у специалистов компании Soft-logic).
2. **`auth.url`** — относительный url для отправки SMS.
3. **`auth.login`** — имя пользователя для авторизации при отправке.
4. **`auth.password`** — пароль пользователя для авторизации при отправке.
5. **`auth.source`** — название обратного адреса для отправки.
6. **`param.host`** — IP-адрес для отправки.
7. **`param.transport-id`** — всегда значение 1, что соответствует типу доставки сообщений SMS.
8. **`param.service-id`** — идентификатор сервиса технического сопровождения клиента, по которому пойдет списание.
9. Возможны дополнительные параметры в зависимости от реализации протокола того или иного шлюза.

Как правило, `<sms-gate>` отвечает за отправку системных сообщений, таких как пароли, события мониторинга и т. д. При отсутствии в конфигурационном файле секции `<sms-gate-adv>`, отвечающей за отправку рекламных сообщений, доставкой рекламы так же занимается рассыльщик, сконфигурированный в секции `<sms-gate>`.

В конфигурационном файле возможно указать одну или несколько секций, отвечающих за отправку рекламных SMS. Для каждой из секций возможно указать имя в атрибуте **name**:

```
<sms-gate-adv name="best"...>  
    ....  
</sms-gate-adv>
```

В таком случае в параметре **«Название API для SMS рассылки, если не задано, то API по умолчанию»** при добавлении **«Рекламы по SMS»** в рекламной кампании возможно указать имя рассыльщика, который будет отправлять сообщения данной рекламной кампании — рисунок 4.3.2.1.

В случае подобной настройки отправлять рекламные SMS в соответствии с данной рекламной кампанией будет рассыльщик с **api="ru.softlogic.processing.gates.synapse.SynapseApi"**. Таким образом возможно распределить нагрузку отправки рекламных сообщений.

Возможность настройки рассыльщика для отправки рекламных сообщений предусмотрена в связи с тем, что для отправки рекламы обычно применяются более дешевые тарифные пакеты, как правило, не обеспечивающие гарантию доставки.

Для отправки критичной информации, такой как события мониторинга, одноразовые пароли, значение имеет не только гарантия доставки сообщения, но и время доставки, так как, например, для одноразовых паролей может быть установлено время жизни.

Добавить Реклама по SMS

ID

Название * Интернет

Активна

Вес (чем выше, тем приоритетней) * 1

Кол-во отправлять всего (если не задано, без ограничений)

Кол-во отправлять ежедневно

Время с (ежедневно) 00:00:00

Время по (ежедневно) 23:59:59

Только в будние дни

Сообщение * Домашний интернет и цифровое телевидение, а также мобильное приложение «Билайн ТВ» будут стоить всего 1 рубль в месяц при подключении одного из тарифов линейки «ВСЕ!».

Длина сообщения: 0

Сервис Билайн 10%

Агент "Soft-logic"

Дополнительное SQL условие на платежи для отправки

Название API для SMS рассылки, если не задано, то API по умолчанию ru.softlogic.processing.gates.syna

Отправлять на платежи * Успешные

* Обязательные поля

Рисунок 4.3.2.1 — Добавление рекламы по SMS в рекламную кампанию

4.3.3 EMAIL-РАССЫЛКА

За настройку Email-рассылки отвечает секция `<email>`. Email-рассылка осуществляется через почтовый сервис Google.

Для включения Email-рассылщика установите флаг **`enable=true`**. Параметры рассылщика:

1. **`auth.user`** — имя пользователя для авторизации при отправке.
2. **`auth.password`** — пароль пользователя для авторизации при отправке.
3. **`param.transport-id`** — всегда значение 3, что соответствует типу доставки сообщений Email.

4.3.4 TELEGRAM

За настройку Telegram-рассылки отвечает секция `<telegram>`. Для включения Telegram-рассыльщика установите флаг **`enable=true`**. Параметры рассыльщика:

1. **`auth.host`** — хост для отправки.
2. **`auth.port`** — порт для отправки.
3. **`param.transport-id`** — всегда значение 5, что соответствует типу доставки сообщений Telegram.
4. **`param.service-id`** — идентификатор сервиса технического сопровождения клиента, по которому пойдет списание.

Помимо конфигурации рассыльщика, для работы типа доставки требуется установить модуль «Paylogic-Telegram». Подробнее в документе [«Кабинет агента. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).

4.3.5 TELEGRAMBOT

Тип доставки TelegramBot используется только для рассылки системных сообщений, позволяет избежать блокировки учетной записи Telegram из-за отправки большого количества сообщений.

За настройку TelegramBot-рассылки отвечает секция `<telegram-bot>`. Для включения TelegramBot-рассыльщика установите флаг **`enable=true`**. Параметры рассыльщика:

1. **`auth.host`** — хост для отправки.
2. **`auth.port`** — порт для отправки.
3. **`param.transport-id`** — всегда значение 6, что соответствует типу доставки сообщений TelegramBot.
4. **`param.messages`** — путь до файла с текстовками, отображаемыми TelegramBot.

4.4 НАСТРОЙКА ОТПРАВКИ РЕЕСТРОВ

Для отправки реестров заданных в системе используется служба почтовой рассылки Google, для указания имени и пароля пользователя для отправки необходимо редактировать скрипт запуска системы шлюзов `/etc/init.d/paylogic`, а именно строку запуска шлюзов.

Найдите и измените следующие параметры:

```
-Dreport.mail.username=user — имя пользователя  
-Dreport.mail.password=password — пароль пользователя
```

В версии 4.3.4 добавлен параметр **calc-point-profit=true**, обеспечивающий запуск процедуры перерасчета прибыли для всех точек по понедельникам в 6.00 утра.

4.5 ПОРЯДОК СМЕНЫ РЕКВИЗИТОВ АВТОРИЗАЦИИ НА ВНЕШНИХ СИСТЕМАХ

При использовании шлюзов для подключения к системам провайдеров (внешним платежным системам) может потребоваться изменить авторизационные данные (ключи и сертификаты, идентификаторы точек, авторизационные данные пользователей). В этом случае, необходимо:

1. Обеспечить полное отсутствие платежей в очереди по указанному провайдеру. Этого возможно добиться следующими способами:

1) временно заблокировать провайдера — установив флаг **«Блокирован»** на вкладке «Юридическое лицо провайдера — Основные свойства» в меню редактирования провайдера («Провайдеры — Провайдеры») в кабинете агента процессинга (бэк-офисе);

2) временно массово перевести все сервисы на другого провайдера, используя функционал массового изменения направлений раздела «Диспетчерская — Направления проведения платежей» кабинета агента (бэк-офиса).

2. После того, как прием платежей по поставщику будет остановлен, убедитесь, что платежи в очереди отсутствуют. Для поиска платежей в очереди используйте раздел «Диспетчерская — Очередь платежей» кабинета агента (бэк-офиса) — показывает платежи, проводящиеся шлюзом в текущий момент времени.

3. Если платежи в очереди отсутствуют, то осуществите смену авторизационных данных в следующем порядке:

- 1) замените ключи в кабинете внешней ПС;
- 2) замените данные в настройках шлюза;
- 3) перезапустите шлюзы.

**Предупреждение!**

При нарушении регламента возможно дублирование платежей: проведение как по старым, так и по новым реквизитам.

4.6 НАСТРОЙКА ШЛЮЗА, ВЫПОЛНЯЮЩЕГО ОТЛОЖЕННЫЕ ЗАДАНИЯ ОБНОВЛЕНИЯ СПРАВОЧНИКОВ

Подробно механизм отложенного выполнения заданий описан в руководстве [«Программное обеспечение «Процессинговый центр Pay-logic». Руководство администратора»](#).

Обработку отложенных заданий обновления справочников выполняет шлюз *ref-queue-table-handler*.

Для запуска шлюза:

1. Скопируйте файл *ref-queue-table-handler-1.0.1.jar* в каталог */home/gates/ext*.
2. В каталоге */home/gates/configs* создайте папку *ref-queue-table-handler*, скопируйте в созданную папку */home/gates/configs/ref-queue-table-handler* файл *ref-queue-table-handler.xml*.
3. В файл */home/gates/config.xml* добавьте строку для шлюза с уникальным отрицательным значением *id-provider*. Отрицательное значение *id-provider* обозначает служебного провайдера. Служебные провайдеры не добавляются в кабинете процессинга, настраиваются исключительно в конфигурационных файлах шлюзов:

```
<gate id-provider="-123" enable="true"  
factory="ru.softlogic.processing.gates.directoryupdate.Factory"  
config="/home/gates/configs/ref-queue-table-handler/ref-queue-table-  
handler.xml"/>
```

4. Выполните на БД скрипт `date.sql` (добавляет колонку `queue_date` в таблицу `reference_queue`).

5. Перезапустите шлюзы.

В конфигурационном файле шлюза `ref-queue-table-handler.xml` в параметре **`time-to-sleep`** указывается периодичность обработки в секундах.

После запуска шлюза установите значение параметра `get_reference_queue_batch`, выполнив запрос:

```
CREATE OR REPLACE FUNCTION public.get_reference_queue_batch()  
 RETURNS smallint AS  
 $BODY$BEGIN  
 RETURN 100 ;  
 END;$BODY$  
 LANGUAGE plpgsql VOLATILE  
 COST 100;  
 ALTER FUNCTION public.get_reference_queue_batch()  
 OWNER TO postgres;
```

Вместо «100» подставьте необходимое значение.

4.7 ПОРЯДОК ЗАПУСКА МОДУЛЯ ПРЕДВАРИТЕЛЬНОЙ ПРОВЕРКИ

Для запуска модуля предварительной проверки:

1. Выложите *jar-файл* в `/home/gates/ext`.
2. Настройте модуль в кабинете в справочнике «Справочники — Сервисы — Модули предварительной проверки».
3. Перезапустите шлюзы командой **service paylogic restart**.
4. Выполните в кабинете настройку модуля предпроверки для нужных сервисов.

4.8 РАСЧЕТ ВОЗНАГРАЖДЕНИЙ В ВАЛЮТЕ ПОСТАВЩИКА

С версии 4.6.8 доступно получение консолидированных за период (день, неделя, период) данных о вознаграждении/комиссии поставщика в валюте этого поставщика, то есть в валюте `sum_prov`. Возможность актуальна для тех пользователей, которые ведут расчеты с контрагентами в валюте, отличающейся от валюты процессинга.

Доступность создания перерасчетов для типа «Пересчет вознаграждения в валюте поставщика» в кабинете определяется параметром `Glassfish.ru.softlogic.recalcFeeCurrencyEnabled`. Задания на перерасчет создаются в разделе «Справочники — Задания расчета», тип задания: «Пересчет вознаграждения в валюте поставщика».

Для отображения отчета «Расчет прибыли в валюте поставщика» в меню необходимо установить правило доступа «Меню — Провайдеры — Расчет прибыли в валюте поставщика».

В шлюзах реализован ежедневный пересчет вознаграждений поставщиков в валюте.

Ежедневный пересчет вознаграждений поставщиков в валюте выполняется по cron-расписанию "1 40 3 * * ?".

Ежедневный пересчет вознаграждений поставщиков в валюте в шлюзах включается параметром конфигурационного файла `/home/gates/config.xml` `<recalc-fee-currency enable="true"/>`.

Задания пересчета обрабатываются без учета этого параметра, т. к., если в кабинете отключено создание задач такого типа, то в базу данных они не добавляются.

При включении параметра **recalc-fee-currency** нужно также настроить логгер **foreign-money-transfer** в файле `/home/gates_ng/log4j.properties`:

```
log4j.logger.foreign-money-transfer=debug, foreign-money-transfer
log4j.appender.foreign-money-
transfer=org.apache.log4j.DailyRollingFileAppender
log4j.appender.foreign-money-transfer.DatePattern='.'yyyy-MM-dd
log4j.appender.foreign-money-transfer.File=${app.dir}/log/consolidate/
foreign-money-transfer.log
log4j.appender.foreign-money-
transfer.layout=org.apache.log4j.PatternLayout
log4j.appender.foreign-money-transfer.layout.conversionPattern=
%d{ABSOLUTE} %5p %t %c{1}:%M:%L - %m%n
```

5 СИСТЕМА ЖУРНАЛИРОВАНИЯ

5.1 СИСТЕМА ЖУРНАЛИРОВАНИЯ ШЛЮЗОВ

Все журналы системы шлюзов создаются в папке `/home/gates/logs`, которая имеет следующую структуру:

- 1) `./consolidate` — журналы по расчету агрегированных данных:
 - a) `accounts.log` — подсчет оборотов лицевых счетов;
 - b) `fee.log` — подсчет внутреннего вознаграждения;
 - c) `overdraft.log` — подсчет процентов по овердрафтам;
 - d) `stat.log` — подсчет данных статистики.
- 2) `./report` — журналы по отправке реестров (ежедневных, ежемесячных и т. п.).
- 3) `./service` — служебные журналы по обработке платежей:
 - a) `begin.log` — журнал начала обработки платежей;
 - b) `collector-in.log` — журнал по обработке очереди `collector-in` (отслеживание уникальности платежей);
 - c) `collector-out.log` — журнал финальной обработки платежей;
 - d) `daycash.log` — журнал подсчета ежедневных остатков;
 - e) `enrolment.log` — журнал возврата денежных средств по неуспешным платежам;

-
- f) *money-collection.log* — журнал обработки инкассаций;
 - g) *partition.log* — журнал переноса платежей в партиции по месяцам;
 - h) *reprocess.log* — журнал повторной автоматической обработки платежей;
 - i) *update-task.log* — журнал работы автоматической системы обновлений;
 - j) *writeoff.log* — журнал работы системы списания денежных средств с агентов.
- 4) ***./support*** — журналы работы системы оповещений:
 - a) *email.log* — оповещения по email;
 - b) *sms.log* — оповещения по SMS;
 - c) *icq.log* — оповещения по ICQ;
 - d) *xmpp.log* — оповещения по XMPP.
 - 5) ***./gate*** — журналы работы отдельных шлюзов.
 - 6) ***./start.log*** — журнал запуска и остановки системы шлюзов.
 - 7) ***./system.log*** — журнал системных сообщений.
 - 8) ***./verify.log*** — журнал онлайн запросов с клиентского ПО (онлайн проверка номеров и т. п.).

Все журналы разбиваются и хранятся по суткам.

5.2 СИСТЕМА ЖУРНАЛИРОВАНИЯ WEB-ПРИЛОЖЕНИЙ

Все журналы Web-приложений создаются в каталоге `/srv/glassfish/domains/domain1/logs/`, который имеет следующую структуру:

- 1) `./paylogic` — журналы приложения по работе с терминалами самообслуживания:
 - a) `common.log` — сетевой обмен с терминалами;
 - b) `device.log` — обработка пакетов по устройствам терминалов;
 - c) `event.log` — обработка пакетов с событиями;
 - d) `inkass.log` — обработка поступающих инкассаций;
 - e) `payment.log` — обработка платежей;
 - f) `reference.log` — обработка запросов справочников.
- 2) `./rma` — журналы приложения по работе с рабочим местом кассира.
- 3) `./external` — журналы приложения по работе с внешними платежными системами.
- 4) `./auth` — журналы попыток авторизации в системе пользователе личного кабинета.
- 5) `./mobile` — журналы обмена с мобильными приложениями.
- 6) `./server.log` — системные сообщения и критические ошибки.

Все журналы разбиваются и хранятся по суткам.

5.3 СИСТЕМА ЖУРНАЛИРОВАНИЯ ДРУГИХ КОМПОНЕНТОВ ПРОЦЕССИНГА

5.3.1 СТРУКТУРА ЛОГОВ ПРОЦЕССИНГА

Процессинг архитектурно состоит из компонент:

5. Работающих на сервере приложений:

- 1) кабинет (бэк-офис);
- 2) коннекторы:
 - a) коннектор терминала (ws);
 - b) коннектор РМА;
 - c) внешние коннекторы (Paylogic2, система подписок, обработчик коллбэк-вызовов платежных инструментов);

6. Работающих как самостоятельная служба:

- 1) шлюзовая подсистема. Может быть запущено несколько экземпляров, при этом один из экземпляров является главным и только на нем работают системные процессы:
 - a) шлюзы к поставщикам услуг;
 - b) процессы, обслуживающие жизненный цикл операции;
 - c) процессы, обслуживающие рассылку отчетов и сообщений;
 - d) процессы, обслуживающие статистику;
 - e) шлюзы к поставщикам-банковским источникам (платежные инструменты).

В процессе работы компоненты ведут большое количество журналов событий (логфайлов). Используя лог-файлы, достаточно легко установить причины возникновения проблем в работе процессинга. Лог-файлы за текущий день именуется следующим образом: <имя файла>.log. Лог-файлы за прошлые дни именуется следующим образом: <имя файла>.log.ГГГГ-ММ-ДД.

5.3.2 ОБЩЕСИСТЕМНЫЕ ЛОГИ

Общесистемные логи включают:

1. `/srv/apache-activemq/data/` — логи службы activeMQ.
2. `/srv/glassfish3/glassfish/domains/domain1/config/velocity.log` — лог velocity-процессора (как правило, малоинформативен).
3. `/srv/glassfish3/glassfish/domains/domain1/config/derby.log` — лог встроенной СУБД Apache Derby.
4. `/srv/glassfish3/glassfish/domains/domain1/logs/` — общий каталог логов сервера приложений. В подкаталогах находятся логи приложений, запущенных на сервере GlassFish.
5. `/srv/glassfish3/glassfish/domains/domain1/logs/server.log` — сохраняются неперехваченные исключения и общие логи по всем приложениям, запущенным в домене GlassFish; информация при старте/остановке сервера приложений.

5.3.3 ЛОГИ ПРИЛОЖЕНИЙ, РАБОТАЮЩИХ НА GF

При работе приложений на GlassFish сохраняются следующие лог-файлы:

1. `/srv/glassfish3/glassfish/domains/domain1/logs/rma/` — логи коннектора кассового ПО.
2. `/srv/glassfish3/glassfish/domains/domain1/logs/mobile/` — логи коннектора мобильного ПО.
3. `/srv/glassfish3/glassfish/domains/domain1/logs/paylogic/` — логи коннектора терминального ПО. Разделены на несколько поддиректорий для удобства:
 - 1) `/srv/glassfish3/glassfish/domains/domain1/logs/paylogic/payment.log` — информация о принятых с ТПО платежах, онлайн-проверках;
 - 2) `/srv/glassfish3/glassfish/domains/domain1/logs/paylogic/inkass.log` — информация о принятых инкассациях;
 - 3) `/srv/glassfish3/glassfish/domains/domain1/logs/paylogic/reference.log` — информация об обмене справочниками между ТПО и процессингом;
 - 4) `/srv/glassfish3/glassfish/domains/domain1/logs/paylogic/event.log` — информация о принятых с ТПО событиях;
 - 5) `/srv/glassfish3/glassfish/domains/domain1/logs/paylogic/device.log` — информация о принятой конфигурации оборудования;
 - 6) `/srv/glassfish3/glassfish/domains/domain1/logs/paylogic/serviceMenu.log` — информация о выполненных в сервисном меню действиях (как правило, малоинформативна);

4. `/srv/glassfish3/glassfish/domains/domain1/logs/paylogic/common.log` — все остальные события, связанные с работой приложения терминального коннектора.

5. `/srv/glassfish3/glassfish/domains/domain1/logs/paylogic/seam.log` — лог бэкофиса кабинета.

6. Логи внешних коннекторов. Группа каталогов, каждый из которых отвечает за определенный тип протокола:

1) `/srv/glassfish3/glassfish/domains/domain1/logs/external-cyber/external-cyber.log` — логи старого (не модульного) коннектора «Входящий киберплат», который должен был быть выведен из эксплуатации;

2) `/srv/glassfish3/glassfish/domains/domain1/logs/external/external.log` — логи старого (не модульного) коннектора «Paylogic2», который должен был быть выведен из эксплуатации;

3) `/glassfish3/glassfish/domains/domain1/logs/external2/` — логи подсистемы модульного внешнего коннектора. В его составе несколько модулей, согласно применяемым протоколам:

a) `/glassfish3/glassfish/domains/domain1/logs/external2/paylogic2-external/paylogic2-external.log` — лог обработчика протокола Paylogic;

b) `/srv/glassfish3/glassfish/domains/domain1/logs/external2/external2.log` — лог ядра модульного коннектора;

c) `/glassfish3/glassfish/domains/domain1/logs/external2/paylogic2-external/paylogic2-external.log` — лог обработчика протокола Paylogic2;

d) `/glassfish3/glassfish/domains/domain1/logs/external2/paylogic2-external-forms/paylogic2-external-forms.log` — логи обработчика форм;

e) `/glassfish3/glassfish/domains/domain1/logs/external2/subscription-external/subscription-external.log` — логи запросов подсистемы подписок в Paylogic2;

f) `/glassfish3/glassfish/domains/domain1/logs/external2/subscription-external/subscription-external.log` — логи запросов подсистемы мониторинга текущей версии коннекторов Paylogic2 и работающих на системе протоколов;

g) `/glassfish3/glassfish/domains/domain1/logs/external2/subscription-external/subscription-external.log` — логи запросов подсистемы мониторинга текущей версии коннекторов Paylogic2 и работающих на системе протоколов;

h) `/glassfish3/glassfish/domains/domain1/logs/external2/payment-tools-callback-external/payment-tools-callback-external.log` — лог обработчика коллбэк-оповещений в платежных инструментах Paylogic2. Является частью подсистемы платежных инструментов и необходим для приема и обработки коллбэк-оповещений от системы эквайринга;

4) `/srv/glassfish3/glassfish/domains/domain1/logs/autopayment` — принимающие модули для автоплатежей по порогу баланса. Подкаталоги для каждого провайдера, работающего по автоплатежам:

a) `/srv/glassfish3/glassfish/domains/domain1/logs/autopayment/rapida.log` — принимающий запросы по платежам по порогу баланса обработчик для киберплата.

5.3.4 ЛОГИ ПОДСИСТЕМЫ ШЛЮЗОВ

Логи сохраняются в каталог, назначенный экземпляру службы шлюзов. Может быть несколько экземпляров службы шлюзов (например, `gates` и `gates2`), но системным может быть только один экземпляр (`gates`). Приведенному примеру экземпляров службы шлюзов соответствуют каталоги с экземплярами `/srv/gates/` и `/srv/gates2`, все подкаталоги в каталоге `/log`. Структура логов экземпляра следующая:

1. `<gates_home>/log/gate/` — логи шлюзов поставщиков услуг/реализаций платежных инструментов:

1) `<gates_home>/log/gate/<название поставщика услуги/платежного инструмента>/` — лог шлюза конкретного поставщика услуги или платежного инструмента согласно конфигурации шлюзов;

2) `<gates_home>/log/gate/direction/direction.log` — лог системной службы определения направления проведения. В нем фиксируются запросы определения провайдера. Позволяет установить причину, по которой для какой-то услуги провайдер не определяется;

3) `<gates_home>/log/gate/verify/verify.log` — лог онлайн-проверок. Фиксируются все онлайн-проверки до того, как будет выполнен запрос к поставщику услуги.

2. Системные логи шлюзовой службы:

1) `<gates_home>/log/c3p0.log` — лог компонента пула соединений с БД. Как правило, малоинформативен;

2) `<gates_home>/log/jdbc/jdbc.log` — лог jdbc-драйвера. Может быть полезен для диагностики непонятных проблем с выполнением запросов в шлюзах;

3) `<gates_home>/velocity.log` — лог службы velocity-процессора. Как правило, не содержит полезной на практике информации;

4) `<gates_home>/quartz.log` — лог службы планировщика кварц (отвечает за выполнение задач по расписанию CRON (используется для шлюзов, в конфигурационных файлах которых указываются cron-выражения). Полезен разработчикам;

5) `<gates_home>/log/start.log` — лог старта-останова службы шлюзов. Лог нужно анализировать, если какой-то шлюз не запускается, не останавливается или не записывает события в свой лог;

3. `<gates_home>/log/service` — логи процессов шлюзовой службы, отвечающей за жизненный цикл операции (смену статусов, проведение, списание и т.д.) и сервисных операций. Доступен только для основного экземпляра шлюзов:

1) `<gates_home>/log/service/` — логи процессов шлюзовой службы, отвечающей за жизненный цикл операции (смену статусов, проведение, списание и т.д.):

a) `<gates_home>/log/service/begin.log` — лог начала обработки, офлайн-платежи, предпроверка;

b) `<gates_home>/log/service/provider.log` — лог стадии жизненного цикла «Определение провайдера»;

c) `<gates_home>/log/service/reprocess.log` — лог стадии жизненного цикла «Повторное проведение операции (нефинальные)»;

d) `<gates_home>/log/service/money-collection.log` — лог офлайн-обработки инкассаций (подсчет сумм по операциям, расхождений с операциями);

e) `<gates_home>/log/service/collector-in.log` — лог обработки входящего коллектора (перенос из него в очередь обработки);

f) `<gates_home>/log/service/collector-out.log` — лог переноса операций в финальном статусе в исходящий коллектор;

g) `<gates_home>/log/service/writeoff.log` — лог выполнения списаний со счетов агентов;

h) `<gates_home>/log/service/daycash.log` — подсчет накопленных итогов по точкам;

i) `<gates_home>/log/service/enrolment.log` — лог работы службы регистрации точек у поставщиков услуг.

4. `<gates_home>/log/payment-tools/` — логи процессов шлюзовой службы, отвечающей за жизненный цикл операции на платежном инструменте. Доступен только на основном экземпляре шлюзов. Группа согласно жизненному циклу:

- 1) `<gates_home>/log/payment-tools/bank-authorization;`
- 2) `<gates_home>/log/payment-tools/bank-authorization-begin;`
- 3) `<gates_home>/log/payment-tools/bank-authorization-end;`
- 4) `<gates_home>/log/payment-tools/bank-authorization-process;`
- 5) `<gates_home>/log/payment-tools/bank-callback-timeout;`
- 6) `<gates_home>/log/payment-tools/bank-capture-process;`
- 7) `<gates_home>/log/payment-tools/bank-overall-timeout;`
- 8) `<gates_home>/log/payment-tools/bank-reverse-process;`
- 9) `<gates_home>/log/payment-tools/bank-reverse-retry;`
- 10) `<gates_home>/log/payment-tools/bank-status;`

5. `<gates_home>/log/support/` — логи процессов шлюзовой службы, отвечающей за рассылку сообщений по разным каналам. Подкаталоги соответствуют настроенным каналам рассылки в `/gates/config/support.xml`:

- a) `<gates_home>/log/support/main.log` — общий лог работы с очередью сообщений;
- b) `<gates_home>/log/support/sms.log` — лог работы с очередью сообщений SMS;
- c) `<gates_home>/log/support/email.log` — лог работы с очередью сообщений Email;
- d) `<gates_home>/log/support/xmpp.log` — лог работы с очередью сообщений XMPP/Jabber;
- e) `<gates_home>/log/support/icq.log` — лог работы с очередью сообщений icq;
- f) `<gates_home>/log/support/telegram.log` — лог работы с очередью сообщений telegram;

6. `<gates_home>/log/consolidate/` — логи процессов расчета статистики. Только на основном экземпляре шлюзов:

- 1) `<gates_home>/log/consolidate/stat.log` — логи расчета данных статистики;
- 2) `<gates_home>/log/consolidate/accounts.log` — логи расчета оборотов по лицевым счетам агентов;
- 3) `<gates_home>/log/consolidate/fee.log` — логи расчета премий (вознаграждений);
- 4) `<gates_home>/log/consolidate/certificate.log` — процесс, проверяющий сроки действия клиентских сертификатов, использованных в шлюзах. Генерирует

события по окончании или скором окончании, меняет статус в мониторинге, системный процесс.

6 ОБНОВЛЕНИЕ ПРОГРАММНОГО КОМПЛЕКСА

6.1 ОБЩИЕ СВЕДЕНИЯ

Во время выхода новой версии серверного ПО комплекса «Pay-logic» формируется пакет обновления, который передается в виде архива клиенту. Архив имеет следующую структуру:

1. `./bin` — каталог содержит бинарные обновления модулей системы.
2. `./sql/update_utf8.sql` — скрипт обновления структуры и данных в БД.
3. `/gates-libs` — каталог, содержащий jar-файлы (библиотеки).
4. `/html` — в каталоге расположены файлы статике для бэк-офиса.
5. `changelog` — содержит список изменений, выполненных в текущей версии.
6. `readme` — инструкция по обновлению программного комплекса.

Для того, чтобы верно составить план обновления серверного ПО, прежде всего ознакомьтесь с документами `changelog`, `readme` и с содержимым пакета обновления.

Следующие разделы содержат последовательность действий, которые необходимо выполнить для обновления программного комплекса. При описании предполагается, что обновление компонент соответствует очередности, приведенной в данном документе.

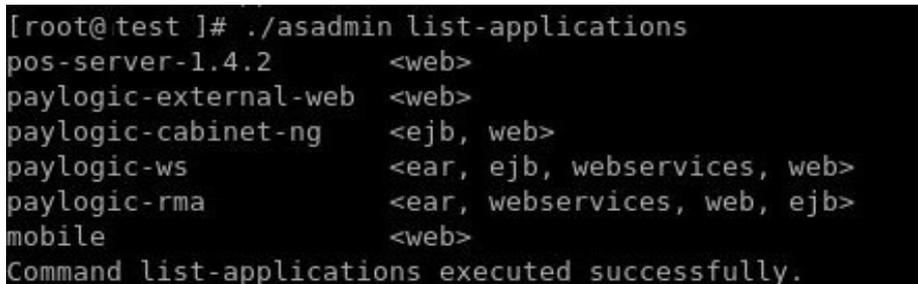
6.2 ПОДГОТОВКА К ОБНОВЛЕНИЮ

Зайдите в директорию, содержащую файлы сервера приложений GlassFish:

```
/srv/glassfish3/bin/
```

Затем выполните команду, которая отобразит список приложений, развернутых на сервере (рисунок 6.2.1):

```
./asadmin list-applications
```



```
[root@test]# ./asadmin list-applications
pos-server-1.4.2      <web>
paylogic-external-web <web>
paylogic-cabinet-ng  <ejb, web>
paylogic-ws          <ear, ejb, webservices, web>
paylogic-rma         <ear, webservices, web, ejb>
mobile               <web>
Command list-applications executed successfully.
```

Рисунок 6.2.1 — Отображение списка запущенных приложений

После этого выгрузите каждый модуль, содержащийся в списке (пример списка показан на рисунке 6.2.1), с сервера приложений с помощью команды:

```
./asadmin undeploy <название файла модуля>
```

Например, для первого модуля в списке, представленном на рисунке 6.2.1, команда будет иметь следующий вид:

```
./asadmin undeploy pos-server-1.4.2
```

Перезапустите сервер приложений GlassFish командой:

```
service glassfish restart
```

Вышеперечисленные действия необходимо повторить для каждого сервера, на котором установлен сервер приложений GlassFish.

Затем необходимо остановить систему шлюзов командой:

```
service paylogic stop
```

Также следует проверить, запущен ли процесс `rmiregistry` командой (рисунок 6.2.2):

```
ps ax | grep rmiregistry
```



```
[root@test]# ps ax | grep rmiregistry
1039 ?        Sl      46:08 /usr/java/latest/bin/rmiregistry
6455 pts/2    S+     0:00 grep --color=auto rmiregistry
```

Рисунок 6.2.2 — Запущенный процесс `rmiregistry`

Если данный процесс запущен, это означает, что скрипт остановки шлюзов отработал некорректно. Поэтому процесс `rmiregistry` необходимо принудительно завершить командой:

```
kill -9 id_процесса
```

Для примера, представленного на рисунке 6.2.2, данная команда будет иметь вид:

```
kill -9 1039
```

Затем следует убедиться, что шлюзы были остановлены. Для этого откройте файл логов системы шлюзов `start.log` и проверьте, что в конце файла выведена строка:



```
08:41:06,512 INFO NetworkCommandListener start - Шлюзы остановлены
```

Рисунок 6.2.3 — Строка, означающая успешную остановку шлюзов

В том случае, если данной строки нет, найдите процесс `gates` с помощью команды:

```
ps ax | grep gates
```

Пример выполнения данной команды представлен на рисунке 6.2.4.

```
[root@test]# ps ax | grep gates
14085 ?        Sl      16:24 /usr/java/latest/bin/java -Djava.rmi.server.codebase=file:///home/gates/lib
/remote.jar -Dhttps.protocols=TLSv1,TLSv1.1,TLSv1.2 -Dapp.dir=/home/gates -Dscar.lib=/home/gates/lib/
J0scarLib.jar -Dsun.rmi.transport.tcp.readTimeout=60000 -Djava.rmi.server.hostname=192.168.50.6 -Dlog4
j.configuration=file:/home/gates/log4j.properties -Dreport.mail.username=daily@unipay.ru -Dreport.mail
.password=ejzZX-MN_hi -Dreport.mail.properties=/home/gates/configs/email.properties gates.Main
18126 pts/2    S+     0:00 grep gates
```

Рисунок 6.2.4 — Запущенный процесс gates

После этого принудительно завершите систему шлюзов командой:

```
kill -9 id_процесса
```

Для примера, представленного на рисунке 6.2.4, данная команда будет иметь вид:

```
kill -9 14085
```

Если в инструкции по обновлению программного комплекса (файл *readme*), полученной в пакете обновления, присутствуют указания произвести дополнительные действия с файлами шлюзов либо сервера приложений, то необходимо их выполнить.

Таковыми действиями могут быть:

1. Копирование новых библиотек;
2. Обновление существующих библиотек;
3. Внесение правок в файлы конфигурации;
4. Добавление конфигурации логирования;
5. Очистка кэш-памяти сервера приложений.

6.3 ОБНОВЛЕНИЕ СТРУКТУРЫ БД

Перед обновлением структуры БД необходимо остановить систему шлюзов и сервер приложений GlassFish. Если остановка системы шлюзов не была произведена при подготовке к обновлению (раздел 6.2), то воспользуйтесь командой:

```
service paylogic stop
```

Остановите сервер приложений GlassFish с помощью команды:

```
service glassfish stop
```

Для того, чтобы обновить структуру БД, выполните скрипт обновления:

```
psql -U postgres -f update_utf8.sql work
```

Данный скрипт необходимо выполнять на сервере, где установлена СУБД.

Признаком успешного выполнения транзакции является запись **"COMMIT"**, в случае возникновения ошибок вся транзакция будет отменена и последней записью будет **"ROLLBACK"**.

6.3.1 ОШИБКИ ПРИ ОБНОВЛЕНИИ БАЗЫ ДАННЫХ

В том случае, если возникли ошибки (была выведена запись "ROLLBACK"), следует произвести анализ файла журнала работы базы данных `/sql/update.log` и выяснить, какие именно команды приводят к ошибкам. Затем необходимо передать полученную информацию сотрудникам компании Soft-logic.

Если программный комплекс обновляется сразу на несколько версий, то возникшие проблемы возможно решить только силами сотрудников компании Soft-logic, поэтому следует остановить процедуру обновления.

Если ПО обновляется на одну версию, то выполните следующие действия:

1. Запустите систему шлюзов командой:

```
service paylogic start
```

2. Запустите сервер приложений и загрузите в него ранее используемые приложения командами:

```
service glassfish start  
./asadmin deploy paylogic-cabinet-ng.war  
./asadmin deploy paylogic-ws.ear
```

6.4 ОБНОВЛЕНИЕ СИСТЕМЫ ШЛЮЗОВ

В том случае, если в папке **bin** пакета обновления присутствует файл *Gates.jar*, необходимо обновление ядра системы шлюзов. Для того, чтобы обновить ядро системы шлюзов, выполните следующие шаги:

1. Если система шлюзов не была остановлена при подготовке к обновлению (раздел 6.2), то остановите ее командой:

```
service paylogic stop
```

2. Скопируйте файл *Gates.jar* из архива обновления в директорию **/home/gates**.
3. Скопируйте файлы с расширением **.jar* из директории **gates-libs/** архива обновления в директорию **/home/gates/lib/**.
4. Удалите файлы старых версий из директории **/home/gates/lib/**.
5. Запустите систему шлюзов командой:

```
service paylogic start
```

6.5 ОБНОВЛЕНИЕ WEB-ПРИЛОЖЕНИЙ

В случае наличия одного из следующих файлов в папке **bin** пакета установки:

1. `paylogic-cabinet.war` — модуль кабинета агента и управления системой.
2. `paylogic-ws.ear` — модуль обмена с терминалами.
3. `paylogic-rma.ear` — модуль обмена с Рабочим местом кассира.
4. `paylogic-external.war` — модуль обмена с внешними платежными системами.

Необходимо обновление соответствующих модулей.

Если сервер приложений GlassFish не был остановлен при обновлении структуры БД (раздел 6.3), то остановите его командой:

```
service glassfish stop
```

Чтобы установить модули, последовательно для каждого из них выполните команду:

```
/srv/glassfish3/bin/asadmin deploy -force=true <название файла модуля>
```

Например:

```
/srv/glassfish3/bin/asadmin deploy -force=true paylogic-cabinet.war
```

Затем произведите запуск и загрузку сервера приложений из директории `/srv/glassfish3/bin/`, выполнив команды:

```
service glassfish start
./asadmin deploy paylogic-cabinet-ng.war
./asadmin deploy paylogic-ws.ear
```

6.6 ОБНОВЛЕНИЕ СТАТИКИ ВЕБ-СЕРВЕРА

Для того, чтобы выполнить обновление статических файлов веб-сервера, извлеките содержимое архива *html.zip* в директорию `/var/www/html/server`.

7 ПРОВЕРКА СИСТЕМЫ

7.1 ПРОВЕРКА СОСТОЯНИЯ КОМПОНЕНТОВ СИСТЕМЫ

Для мониторинга работы основных компонентов системы и их взаимосвязи между собой в системе реализуется метод **healthcheck**. Данный метод поддерживается коннекторами *paylogic-external-api*, *paylogic2-external*, *paylogic-api*, *gates*, *dynamic-input*. При обращении на любой из указанных коннекторов можно указать параметр проверки — **mode**, который является не обязательным, и может принимать следующие значения:

- **STATIC** — проверка состояния компонентов в статическом режиме работы;
- **DYNAMIC** — проверка состояния компонентов при использовании модуля динамической маршрутизации;
- **ALL** — проверка состояния компонентов всех вариантов работы (**STATIC** и **DYNAMIC**, если модуль динамической маршрутизации используется в системе).

Коннектор примет указанный запрос и проверит свою работоспособность и доступ к базе данных. Затем, если в системе используется модуль динамической маршрутизации, коннектор передаст запрос к нему и модуль проверит свою работоспособность. Далее коннектор передаст запрос к системе шлюзов, и система шлюзов проверит свое состояние и доступ к базе данных. Если модуль динамики не используется, то коннектор сразу передаст запрос к системе шлюзов.

Если проверка всех компонентов прошла успешно, то в ответ на запрос **healthcheck** вернется «OK», в ином случае – «ERROR».

Таким образом можно отследить взаимосвязь компонентов и их взаимосвязь друг с другом. Выполнение метода **healthcheck** является ключевым показателем возможности создания и обработки операции в системе

Примечание:

Данный метод для работы требует авторизацию по сертификату точки. Рекомендуется создать в системе новую точку с типом «Шлюз по сертификату», сгенерировать для нее сертификат и использовать единую точку для мониторинга работоспособности системы. Но также система позволяет использовать разные точки для проверок по методу **healthcheck**. Создание точки и генерация сертификата описаны в документе [«Кабинет агента. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя.»](#).

Пример вызова метода для протокола *paylogic-external-api* с авторизацией по логину и паролю:

```
% curl --location 'https://<адрес сервера>/paylogic/external/api/healthcheck?mode=STATIC ' \
--header 'Pay-logic-Login: admin' \
--header 'Pay-logic-Password: admin' -k
```

Пример ответа на запрос для протокола *paylogic-external-api*:

```
{
  "systemState": "OK"
}
```

Пример вызова метода для протокола *paylogic2-external* с авторизацией по сертификату:

```
<request point="2498">
<healthcheck mode="ALL"/>
</request>
```

Пример ответа на запрос для протокола *paylogic2-external*:

```
<response>
<result system-state="ERROR"/>
</response>
```

8 РЕКОМЕНДАЦИИ ПО ВЫХОДУ ИЗ АВАРИЙНЫХ СИТУАЦИЙ

8.1 ОТКАЗ В РАБОТЕ СЕРВЕРА ПРИЛОЖЕНИЙ GLASSFISH ИЛИ HTTPD

В случае отказа в работе сервера Glassfish или одного из установленных в него приложений, возможны следующие ситуации и методы выхода из них:

1. Невозможно установить соединение с сервером Apache Httpd

В случае невозможности установить соединение с Apache проверьте, запущен ли он на сервере командой:

```
ps ax | grep http
```

Если Apache не запущен, запустите его командой:

```
service httpd start
```

Если Apache прекратил свою работу нештатно, проанализируйте его журналы, которые располагаются по пути `/var/log/httpd`.

В том случае, если Apache запущен, находится в памяти, но соединение не устанавливается, перезапустите его командой:

```
service httpd restart
```

И проанализируйте логи, которые находятся по пути `/var/log/httpd`.

2. Сервер Apache отвечает ошибкой Internal Error 500

В данном случае сервер Apache не может соединиться с сервером приложений Glassfish. Для обнаружения причины невозможности соединения изучите журналы сервера приложений:

```
/srv/glassfish3/glassfish/domains/domain1/logs/server.log
```

А также логи модуля MOD_JK, который обеспечивает связку с сервером приложений:

```
/var/log/httpd/mod_jk.log
```

Возможно, проблема в высокой загрузке сервера приложений (п. 5 данного раздела). Вне зависимости от причины, из-за которой сервер Apache не может соединиться с сервером приложений Glassfish, перезапустите сервер приложений командой:

```
service glassfish restart
```

3. Сервер медленно отвечает на запросы

В случае если сервер медленно отвечает на запросы или долго открываются страницы бэк-офиса, проверьте нагрузку на сервер, согласно пункту 5 данного раздела.

Если нагрузка не велика, проанализируйте сетевую нагрузку на сервер Apache. В случае если установлен модуль Server Status для Apache, перейдите на страницу <https://<адрес:порт>/server-status> и проанализируйте текущие запросы/ответы с точек приема платежей, свободные и занятые потоки Apache. Возможно, существуют проблемы на стороне интернет-провайдера, на площадке которого, расположен сервер.

4. Не отдаются справочники на точки приема платежей (терминалы)

Если справочная информация не отдается длительное время на точки приема платежей, даже в случае их неоднократного сетевого соединения с сервером (не пропадает оранжевый кружок в мониторинге для точек), значит, что вышел из строя поток обновления версий справочников для терминального коннектора. Выполните deploy приложения терминального коннектора:

```
./asadmin deploy -force=true paylogic-ws.ear
```

5. Большая нагрузка на сервере приложений

Проверить загрузку серверов системы с помощью команды top:

```
top - 09:06:59 up 116 days, 8:45, 1 user, load average: 0.33, 0.35, 0.29
Tasks: 275 total, 1 running, 274 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.1%us, 0.2%sy, 0.0%ni, 98.4%id, 0.2%wa, 0.0%hi, 0.1%si, 0.0%st
Mem: 12320008k total, 10392484k used, 1927524k free, 87244k buffers
Swap: 8630264k total, 178860k used, 8451404k free, 7662848k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
  9131 root        20   0 1344m 293m 4432  S   3.3   2.4   36:30.96 java
 16732 root        20   0 3193m 991m 4016  S   2.7   8.2  134:01.83 java
 18708 postgres    20   0 3316m  79m  72m  S   2.7   0.7    0:00.98 postmaster
 18748 postgres    20   0 3318m  64m  57m  S   2.3   0.5    0:00.23 postmaster
 18784 postgres    20   0 3318m  78m  70m  S   2.3   0.6    0:00.24 postmaster
 16275 postgres    20   0 3310m 3452 2232  S   1.7   0.0   25:23.71 postmaster
 18739 postgres    20   0 3318m  98m  90m  S   1.7   0.8    0:00.43 postmaster
```

Возможны различные ситуации:

1. Один из процессов Java занимает не менее 100% процессорного времени и нагрузка не спадает в течение длительного времени. В данном случае перегрузите сервер приложений Glassfish командой `service glassfish restart`. В том случае, если Glassfish долго не останавливается, принудительно завершите процесс командой `kill`.
2. Большой показатель Load average (свыше 5) при незначительной нагрузке на процессор. В данном случае проанализируйте нагрузку на дисковую систему командой `iostat`.
3. Один из процессов PostgreSQL занимает свыше 60% процессорного времени в течение длительного периода. Возможно, данный процесс строит какой-то сложный отчет. Для того, чтобы понять какой именно, войдите в консоль PostgreSQL: `psql -U postgres work`, и выполните запрос:

```
SELECT query_start,datname,procpid,current_query FROM pg_stat_activity
where procpid=19523;
```

Данный запрос покажет сколько время начала выполнения запроса, текст запроса и название БД, на которой он выполняется:

```
      query_start          | datname |
current_query
-----+-----
+-----+-----
--
2013-11-06 09:21:29.211283+06 | work    | autovacuum: VACUUM ANALYZE
operations.queue
2013-11-06 09:24:04.998334+06 | work    | select
co.id_operation,co.state,co.substate,co.id_processing_error from
operations.master co where co.id_point=4961 and co.id_operation=226464654
```

После получения текста запроса, возможно приступить к анализу, почему он вызывает такую нагрузку.

8.2 ОТКАЗ В РАБОТЕ СУБД

1. Не запускается СУБД

В данном случае проанализируйте логи СУБД, которые находятся в каталоге `/srv/pgsql/data/pg_log`.

Возможно, что на диске, отведенном под СУБД, закончилось место. Свободное место возможно проверить командой `df`:

Файловая система	1К-блоков	Исп	Доступно	Исп%	смонтирована на
/dev/cciss/c0d0p3	9920624	7569108	1839448	81%	/
/dev/cciss/c0d0p7	233856228	165385988	56399404	75%	/srv
/dev/cciss/c0d0p5	9920592	3509628	5898896	38%	/var
/dev/cciss/c0d0p2	19840924	7257968	11558816	39%	/home
/dev/cciss/c0d0p1	101086	17037	78830	18%	/boot
tmpfs	2023256	0	2023256	0%	/dev/shm

В том случае, если места недостаточно, освободите нужное пространство для работы СУБД, а затем перезапустите её командой:

```
service postgresql restart
```

2. Блокировки в СУБД

Иногда при работе СУБД могут возникать блокировки, которые приводят к нестабильной работе системы, вызывают задержки при работе в кабинете и при изменении тех или иных сущностей. Признаком блокировок может быть невозможность изменения каких-либо параметров в кабинете, когда, например, при изменении сервисов или точек, кабинет «зависает» на несколько минут и в итоге показывает ошибку.

Для поиска долгих блокировок в СУБД выполните запрос:

```
select t.relname,l.locktype,page,virtualtransaction,pid,mode,granted from  
pg_locks l, pg_stat_all_tables t where l.relation=t.relid order by  
relation asc;
```

Результатом выполнения будет таблица, в которой будут показаны все блокировки, сущности, на которые они распространяются, и каким процессом (PID) порождены, например:

relname	pid	mode	granted
pg_class	14130	AccessShareLock	t
pg_index	14130	AccessShareLock	t
pg_namespace	14130	AccessShareLock	t
fee_history	14253	AccessShareLock	t
legals	14253	AccessShareLock	t
points	14253	AccessShareLock	t
processing_directions	14253	AccessShareLock	t
services	14253	AccessShareLock	t
queue	14253	RowExclusiveLock	t
queue	14331	ShareUpdateExclusiveLock	t
queue	14253	AccessShareLock	t

Наиболее интересны блокировки RowExclusiveLock, и в том случае, если они висят долго, необходимо смотреть запросы процесса который их порождает:

```
SELECT query_start,datname,procpid,current_query FROM pg_stat_activity  
where procpid=14253;
```

Если в выбранном процессе не выполняются запросы, связанные с обновлением данных, или он находится в состоянии IDLE и блокировка не снимается длительное время, возможно принять решение об остановке данного процесса:

```
kill 14253
```

8.3 ОТКАЗ В РАБОТЕ СИСТЕМЫ ШЛЮЗОВ

В случае отказа в работе системы шлюзов либо конкретного шлюза к поставщику услуг, возможны следующие ситуации:

1. Произошла непредвиденная остановка одного из шлюзов к поставщикам услуг

В данном случае проанализируйте последние журналы данного шлюза на наличие исключительных ситуаций, а также журнал *system.log*, на наличие не перехваченных исключений. В случае обнаружения подозрительных записей в вышеупомянутых журналах отправьте их разработчикам компании «Soft-logic».

Для возобновления работы шлюза перезапустите систему шлюзов командой:

```
service paylogic restart
```

2. Не обрабатывается очередь платежей

В данном случае проверьте, что система шлюзов запущена, командой:

```
ps ax | grep gates
```

Если системы шлюзов нет в памяти, запустите её командой:

```
service paylogic start
```

Если система шлюзов не была остановлена сотрудниками компании, проанализируйте журнал *system.log* и отправьте его сотрудникам компании Soft-logic для его анализа.

3. Не проходят онлайн-проверки с точек приема платежей

В данном случае проанализируйте журнал *verify.log* системы шлюзов. В том случае, если туда не добавляются новые записи, то возможно система шлюзов не запущена (см. п. 2) и ее необходимо запустить.

Если система шлюзов запущена, проверьте наличие в памяти процесса *rmiregistry* командой:

```
ps ax | grep rmiregistry
```

Если данного процесса нет в памяти, то перезапустите систему шлюзов. Это необходимо потому, что процесс *rmiregistry* запускается при старте системы шлюзов, а также при старте в него прописываются служебные компоненты.

4. Не проводятся платежи к одному из поставщиков услуг, не проходят онлайн-платежи

В данном случае необходимо анализировать логи конкретного шлюза к поставщику услуг.

В том случае, если в логах появляются исключения *IOException*, *ConnectionTimeoutException* и т.п., означает, что сервер поставщика недоступен, и необходимо обратиться в его поддержку.

В том случае, если в логах на запросы возвращаются конкретные ответы, согласно протоколу, проанализируйте характер кодов ответов, и обратитесь в службу поддержки поставщика услуг.

5. Зависание потоков при остановке шлюза

Основная информация о сбоях в работе шлюзов доступна в следующих журналах событий системы шлюзов:

1. *./start.log* — информация о запуске и остановке шлюзов, в том числе сведения об ошибках конфигурации.

2. `./system.log` — если шлюз запустился, но в процессе работы возникли какие-либо ошибки, то информация о них запишется в `system.log`.

Возможно возникновение ситуации, когда при остановке шлюза один или несколько потоков не могут быть завершены. В этом случае попытки остановить поток будут повторяться до тех пор, пока не завершатся успехом или не будут остановлены вручную.

Фрагмент `start.log`, свидетельствующий о многократных попытках остановить поток:

```
14:26:39,629 INFO NetworkCommandListener start - Поток еще не
остановлен: UniversalParentWorker-&207
14:26:40,629 INFO NetworkCommandListener start - Завершаю поток:
UniversalParentWorker-&207
14:26:40,729 INFO NetworkCommandListener start - Поток еще не
остановлен: UniversalParentWorker-&207
14:26:41,729 INFO NetworkCommandListener start - Завершаю поток:
UniversalParentWorker-&207
14:26:41,830 INFO NetworkCommandListener start - Поток еще не
остановлен: UniversalParentWorker-&207
14:26:42,830 INFO NetworkCommandListener start - Завершаю поток:
UniversalParentWorker-&207
14:26:42,930 INFO NetworkCommandListener start - Поток еще не
остановлен: UniversalParentWorker-&207
14:26:43,930 INFO NetworkCommandListener start - Завершаю поток:
UniversalParentWorker-&207
14:26:44,031 INFO NetworkCommandListener start - Поток еще не
остановлен: UniversalParentWorker-&207
14:26:45,031 INFO NetworkCommandListener start - Завершаю поток:
UniversalParentWorker-&207
14:26:45,131 INFO NetworkCommandListener start - Поток еще не
остановлен: UniversalParentWorker-&207
14:26:46,131 INFO NetworkCommandListener start - Завершаю поток:
UniversalParentWorker-&207
14:26:46,232 INFO NetworkCommandListener start - Поток еще не
остановлен: UniversalParentWorker-&207
14:26:47,232 INFO NetworkCommandListener start - Завершаю поток:
UniversalParentWorker-&207
```

```
14:26:47,332 INFO NetworkCommandListener start - Поток еще не
остановлен: UniversalParentWorker-&207
14:26:48,332 INFO NetworkCommandListener start - Завершаю поток:
UniversalParentWorker-&207
14:26:48,432 INFO NetworkCommandListener start - Поток еще не
остановлен: UniversalParentWorker-&207
14:26:49,432 INFO NetworkCommandListener start - Завершаю поток:
UniversalParentWorker-&207
14:26:49,533 INFO NetworkCommandListener start - Поток еще не
остановлен: UniversalParentWorker-&207
14:26:50,533 INFO NetworkCommandListener start - Завершаю поток:
UniversalParentWorker-&207
14:26:50,633 INFO NetworkCommandListener start - Поток еще не
остановлен: UniversalParentWorker-&207
14:26:51,633 INFO NetworkCommandListener start - Завершаю поток:
UniversalParentWorker-&207
14:26:51,733 INFO NetworkCommandListener start - Поток еще не
остановлен: UniversalParentWorker-&207
```

Для того, чтобы найти шлюз, с зависшим потоком, выполните команду:

```
ps ax | grep gates.Main
```

В результате будет выведена информация по зависшим потокам: идентификатор процесса и подробная информация о нем. Например,

```
2536 pts/0    Sl      0:33 /usr/java/jre1.7.0_79/bin/java -
Djava.rmi.server.codebase=file:///home/gates/lib/remote.jar -
Dsun.rmi.transport.tcp.readTimeout=60000 -
Djava.rmi.server.hostname=192.168.10.1 -Dapp.dir=/home/gates -
Doscar.lib=/home/gates/lib/JOscarLib.jar
-Dlog4j.configuration=file:/home/gates/log4j.properties -
Dreport.mail.username=* -Dreport.mail.password=* gates.Main
```

Для того, чтобы остановить шлюз, выполните команду:

```
kill [pid]
```

Где [pid] — идентификатор процесса. Например,

```
kill 2536
```

Для запуска шлюзов выполните команду:

```
service paylogic start
```