



**Протокол взаимодействия  
между внешней платежной системой и  
Процессинговым центром Pay-logic**  
Руководство разработчика

---

**АННОТАЦИЯ**

Содержит описание и техническую спецификацию протокола взаимодействия с внешними платежными системами

Версия руководства: 6.11

*Руководство актуально для программного обеспечения «Процессинговый центр Pay-logic» версий 5.9.x*

2008–2026 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://www.pay-logic.ru/>

Mail: [info@soft-logic.ru](mailto:info@soft-logic.ru)

## ОГЛАВЛЕНИЕ

<b>ИСТОРИЯ ИЗМЕНЕНИЙ.....</b>	<b>6</b>
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.4.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.5.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.6.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.7.....	7
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.8.....	7
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.9.....	8
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.10.....	8
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.11.....	8
<b>1 ОСНОВНЫЕ ТЕРМИНЫ.....</b>	<b>9</b>
<b>2 ОСНОВНЫЕ СОКРАЩЕНИЯ.....</b>	<b>11</b>
<b>3 ПРЕДПОЛАГАЕМАЯ АУДИТОРИЯ.....</b>	<b>12</b>
<b>4 ВВЕДЕНИЕ.....</b>	<b>13</b>
<b>5 ОБЩИЕ ПОЛОЖЕНИЯ.....</b>	<b>14</b>
5.1 ЦЕЛЬ И НАЗНАЧЕНИЕ.....	14
5.2 ТРЕБОВАНИЯ К АС ВПС И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ.....	15
<b>6 ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ.....</b>	<b>16</b>
6.1 СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ.....	16
6.2 ЭЛЕКТРОННАЯ ПОДПИСЬ.....	17
6.2.1 ЭЛЕКТРОННАЯ ПОДПИСЬ В ФОРМАТЕ RSA.....	17
6.2.2 ЭЛЕКТРОННАЯ ПОДПИСЬ В ФОРМАТЕ ГОСТ Р 34.10-2012.....	17
6.3 ПЕРСОНАЛЬНЫЙ СЕРТИФИКАТ.....	20
6.4 ЛОГИН И ПАРОЛЬ.....	20
<b>7 СОЗДАНИЕ ТОЧКИ ТИПА ШЛЮЗ.....</b>	<b>22</b>

---

<b>8 ПРОВЕДЕНИЕ ПЛАТЕЖЕЙ</b> .....	<b>27</b>
<b>8.1 ЖИЗНЕННЫЙ ЦИКЛ ОПЕРАЦИЙ</b> .....	<b>27</b>
<b>8.2 ОПИСАНИЕ ФОРМАТОВ ДАННЫХ</b> .....	<b>32</b>
<b>8.3 СТРУКТУРНЫЕ ЭЛЕМЕНТЫ ПАКЕТА</b> .....	<b>33</b>
8.3.1 ОБЩИЕ СВЕДЕНИЯ.....	33
8.3.2 КОРНЕВОЙ ЭЛЕМЕНТ — ТЕГ <REQUEST>.....	36
8.3.3 БАЛАНС ЛИЦЕВОГО СЧЕТА — ТЕГ <BALANCE>.....	37
8.3.4 СПРАВОЧНИК ПЛАТЕЖНЫХ ИНСТРУМЕНТОВ — ТЕГ <TOOLS>.....	38
8.3.5 ОФЛАЙНОВОЕ ПРОВЕДЕНИЕ ОПЕРАЦИИ — ТЕГ <PAYMENT>.....	40
8.3.6 ПРОВЕДЕНИЕ ОПЕРАЦИИ В РЕЖИМЕ ОНЛАЙН — ТЕГ <OPAYMENT>.....	49
8.3.7 ПАРАМЕТРЫ ПРОВЕРКИ НОМЕРА АБОНЕНТА — ТЕГ <VERIFY>.....	50
8.3.8 РАСЧЕТ ВНЕШНЕЙ КОМИССИИ — ТЕГ <COMMISSION>.....	54
8.3.9 ПАРАМЕТРЫ СЛОЖНОЙ ПРОВЕРКИ РЕКВИЗИТОВ — ТЕГ <ADVANCED>.....	55
8.3.10 ПАРАМЕТРЫ ПРОВЕРКИ СТАТУСА ПЛАТЕЖА — ТЕГ <STATUS>.....	58
8.3.11 ПОДТВЕРЖДЕНИЕ ОТЛОЖЕННОЙ ОПЕРАЦИИ — ТЕГ <CONFIRM>.....	60
8.3.12 ОБРАБОТКА ОПЕРАЦИЙ С РАЗРЫВОМ В ПРОВЕДЕНИИ — ТЕГ <PAYMENT-GAP> .....	61
8.3.13 ПАРАМЕТРЫ РЕЗУЛЬТАТА — ТЕГ <RESULT>.....	62
8.3.14 ВОЗВРАЩАЕМЫЕ ПАРАМЕТРЫ ПРИ ADVANCED-ЗАПРОСЕ — ТЕГ <INPUT>.....	65
<b>8.4 СЛОЖНАЯ ПРОВЕРКА РЕКВИЗИТОВ ADVANCED-ЗАПРОСА</b> .....	<b>66</b>
<b>8.5 ПОПЛАТЕЖНАЯ СВЕРКА</b> .....	<b>70</b>
<b>8.6 РЕГИСТРАЦИЯ ТОЧЕК У ПОСТАВЩИКА</b> .....	<b>73</b>
<b>8.7 ОТМЕНА ТРАНЗАКЦИЙ</b> .....	<b>76</b>
<b>8.8 ЗАПРОС КАТАЛОГА УСЛУГ (&lt;MENU&gt;)</b> .....	<b>78</b>
8.8.1 ЗАПРОС ВЕРСИИ КАТАЛОГА УСЛУГ.....	90
<b>8.9 ЗАПРОС КАТЕГОРИЙ (ТИПОВ) СЕРВИСОВ</b> .....	<b>91</b>
<b>8.10 ЗАПРОС СПИСКА ФОРМ</b> .....	<b>93</b>
<b>8.11 ЗАПРОС СЦЕНАРИЕВ</b> .....	<b>95</b>
<b>8.12 ЗАПРОС ПАРАМЕТРОВ КОМИССИИ ПРОВАЙДЕРА</b> .....	<b>97</b>
<b>8.13 ОБРАБОТКА НЕПРАВИЛЬНЫХ ПАКЕТОВ</b> .....	<b>99</b>
<b>8.14 БЛОКИРОВКА ЗАПРОСОВ ОТ ПАРТНЕРА ПРИ НЕДОСТАТОЧНОМ БАЛАНСЕ</b> .....	<b>100</b>
<b>8.15 БЛОКИРОВКА ПОВТОРНЫХ ВХОДЯЩИХ ЗАПРОСОВ</b> .....	<b>101</b>
<b>8.16 ОТПРАВКА CALLBACK ПО ФАКТУ ФИНАЛИЗАЦИИ СТАТУСА ПЛАТЕЖА</b> .....	<b>104</b>
8.16.1 НАЗНАЧЕНИЕ.....	104
8.16.2 НАСТРОЙКА.....	104
8.16.3 ФОРМАТ CALLBACK.....	106
<b>9 МОДУЛЬ УПРАВЛЕНИЯ ПОДПИСКАМИ</b> .....	<b>107</b>

---

---

<b>9.1 НАЗНАЧЕНИЕ МОДУЛЯ УПРАВЛЕНИЯ ПОДПИСКАМИ.....</b>	<b>107</b>
<b>9.2 СОВЕРШЕНИЕ АВТОПЛАТЕЖЕЙ И ПОЛУЧЕНИЕ НАЧИСЛЕНИЙ.....</b>	<b>108</b>
<b>9.3 СУЩНОСТИ ПОДПИСКИ.....</b>	<b>109</b>
<b>9.4 ТИПЫ ЗАПРОСОВ.....</b>	<b>111</b>
<b>9.5 ПОСЛЕДОВАТЕЛЬНОСТЬ ВЗАИМОДЕЙСТВИЙ.....</b>	<b>112</b>
<b>9.6 ТИП РЕЗУЛЬТАТА.....</b>	<b>114</b>
<b>9.7 СПЕЦИФИКАЦИЯ ЗАПРОСОВ.....</b>	<b>115</b>
9.7.1 ПРОВЕРКА ДАННЫХ АБОНЕНТА ПЕРЕД ОФОРМЛЕНИЕМ ПОДПИСКИ.....	115
9.7.2 ПОДКЛЮЧЕНИЕ АБОНЕНТА НА УСЛУГУ ПОЛУЧЕНИЯ НАЧИСЛЕНИЙ.....	117
9.7.3 ОТКЛЮЧЕНИЕ АБОНЕНТА ОТ УСЛУГИ ПОЛУЧЕНИЯ НАЧИСЛЕНИЙ.....	119
9.7.4 ИЗМЕНЕНИЕ ПАРАМЕТРОВ ПОДПИСКИ.....	121
9.7.5 ПОЛУЧЕНИЕ СПИСКА ДЕЙСТВУЮЩИХ ПОДПИСОК.....	124
9.7.6 ПОЛУЧЕНИЕ СПИСКА НАЧИСЛЕНИЙ.....	126
9.7.7 РЕЗУЛЬТАТ СПИСАНИЯ.....	130
9.7.8 ПОЛУЧЕНИЕ СТАТУСА ПЛАТЕЖА.....	132
<b>ПРИЛОЖЕНИЕ А. РЕЗУЛЬТАТ ДАННЫХ, ВОЗВРАЩАЕМЫХ ПЦ.....</b>	<b>133</b>
A.1 КОДЫ СТАТУСОВ И СУБСТАТУСОВ ПЛАТЕЖА.....	133
A.2 НОМЕРА ОШИБОК.....	138
A.3 НОМЕРА ОШИБОК АТРИБУТА SERVICE ПРИ ADVANCED-ЗАПРОСЕ.....	140
A.4 КОДЫ СТАТУСОВ ОБРАБОТКИ ЗАЯВОК НА ОТМЕНУ.....	142
A.5 КОДЫ ОТВЕТОВ МОДУЛЯ УПРАВЛЕНИЯ ПОДПИСКАМИ.....	143
A.6 ТИПЫ СОБЫТИЙ.....	144
A.7 КОДЫ ОШИБОК ПРОВАЙДЕРА ПЛАТЕЖНОГО ИНСТРУМЕНТА.....	154
<b>ПРИЛОЖЕНИЕ В. ГЕНЕРАЦИЯ И ПРОВЕРКА ЭП.....</b>	<b>155</b>
V.1 ПРИМЕРЫ ГЕНЕРАЦИИ И ПРОВЕРКИ ЭП НА JAVA.....	155
V.2 ПРИМЕР ГЕНЕРАЦИИ И ПРОВЕРКИ ЭП НА PHP.....	158
<b>ПРИЛОЖЕНИЕ С. ПАРАМЕТРЫ КОННЕКТОРА.....</b>	<b>159</b>

## ИСТОРИЯ ИЗМЕНЕНИЙ

### ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.4

Дата публикации: 22.07.2022.

Изменение	Раздел
<b>Общие улучшения:</b>	
Для advanced-запроса добавлен параметр <b>id</b> , при наличии которого на стороне процессинга будет вызван метод резервирования ID платежа	8.3.9

### ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.5

Дата публикации: 30.09.2022.

Изменение	Раздел
<b>Новое в версии 5.3.4:</b>	
Реализована возможность отправки callback по факту финализации статуса платежа в формате paylogic2	8.16

### ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.6

Дата публикации: 23.03.2023.

---

Изменение	Раздел
<b>Улучшения в версии 5.4.3:</b>	
В запрос <code>menu</code> добавлено новое поле <code>hotkeyPosition</code>	8.8

#### ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.7

Дата публикации: 15.09.2023.

Изменение	Раздел
<b>Дополнения в документации:</b>	
Дополнен раздел, описывающий обработку запроса с <code>terminal-vps-id</code>	8.3.5
Добавлено описание статусов 20 (5), 80 (10, 11, 12)	A.1

#### ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.8

Дата публикации: 26.09.2023.

Изменение	Раздел
<b>Дополнения в документации:</b>	
Исключен раздел 9. Раздел 10 перенумерован в раздел 9	9

**ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.9**

Дата публикации: 09.08.2024.

Изменение	Раздел
<b>Дополнения в документации:</b>	
Актуализирован раздел «Баланс лицевого счета - тег <BALANCE>»	8.3.3

**ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.10**

Дата публикации: 26.11.2024.

Изменение	Раздел
<b>Дополнения в документации:</b>	
Добавлено описание об авторизации с помощью электронной подписи в формате ГОСТ.	6.2.2

**ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 6.11**

Дата публикации: 10.02.2026.

Изменение	Раздел
<b>Улучшения в документации:</b>	
Добавлено описание способа авторизации по логину и паролю	6

## 1 ОСНОВНЫЕ ТЕРМИНЫ

**Абонент** — физическое или юридическое лицо, заключившее с Поставщиком услуг Абонентский договор.

**Агент** — банк, платежная система или другой финансовый институт, осуществляющий прием платежей.

**Л/с агента** — здесь: аналитический счет в учетной системе Платежной системы, на котором учитываются операции, связанные с приемом платежей агентом и взаиморасчетами между агентом и платежной системой в соответствии с договором, который заключили агент и платежная система.

**Номер абонента** — индивидуальный телефонный номер абонента, идентифицирующий его в системе Поставщика услуг.

**Платеж** — здесь: платеж, совершенный с помощью следующих платежных инструментов — наличные или кредитная карта — в пользу абонента.

**Платежная система** — юридическое лицо, предоставляющее услуги процессинга, с которым заключают договоры агенты.

**Платежный инструмент** — здесь: средство, при помощи которого может быть совершен платеж. Примеры платежных средств:

1. Наличные деньги.
2. Кредитная карта.
3. Электронные платежные инструменты и т.п.

**Плательщик** — здесь: конкретное физическое лицо, совершившее платеж.

**Поставщик услуг** — юридическое лицо, предоставляющее услуги, с которым заключают договоры абоненты.

**Сеть приема платежей** — комплекс аппаратно-технических средств, используемый для информационного взаимодействия Участников.

**Таймаут** — здесь: время ожидания.

**Терминал** — здесь: программно-аппаратное устройство, с помощью которого (через который) осуществляется прием платежа.

**Терминальная сеть** — здесь: терминалы, объединенные в централизованно-управляемый программно-аппаратный комплекс. Терминальная сеть управляется (эксплуатируется) организацией, уполномоченной на это платежной системой.

**Участник** — юридическое лицо, подписавшее «Договор участия в системе приема платежей».

## 2 ОСНОВНЫЕ СОКРАЩЕНИЯ

**АС** — автоматизированная система.

**ВПС** — здесь: внешняя платежная система.

**ОТТ** — общие технические требования. Данный документ.

**ПЦ** — здесь: процессинговый центр.

**СПП** — сеть приема платежей.

**ЭП** — электронная подпись.

### 3 ПРЕДПОЛАГАЕМАЯ АУДИТОРИЯ

Данное руководство предназначено для использования в качестве спецификации для разработки протокола взаимодействия со стороны агента внешней платежной системы.

Документ адресован IT-специалистам (программистам и администраторам), обслуживающим информационную систему поставщика услуг и знакомым с технологиями разработки ПО, языками программирования, основами работы с XML и сетевым взаимодействием по HTTP/HTTPS, владеющим основами криптографии.

Необходимо знание языков программирования и соответствующих библиотек для Java или PHP.

## 4 ВВЕДЕНИЕ

Документ предназначен для ВПС и разработчиков АС ВПС:

1. Для проведения доработки (разработки) программного обеспечения для АС ВПС, разработки технических заданий на доработку (разработку) соответствующих систем, обеспечивающих возможность подключения к СПП.
2. Для оценки качества проведенных доработок и соответствия доработанной системы требованиям ОТТ во время комплексных испытаний. Реализация требований данного документа позволит подключить ВПС к СПП и обеспечить прием платежей абонентов через терминальную сеть ВПС.

## 5 ОБЩИЕ ПОЛОЖЕНИЯ

### 5.1 ЦЕЛЬ И НАЗНАЧЕНИЕ

Система предназначена для обеспечения приема платежей через терминалы ВПС и проведения платежей в режиме, приближенном к режиму реального времени.

Бизнес-цель создания системы — расширить сеть пунктов приема платежей и перечень платежных инструментов, с помощью которых абонент может совершить платеж.

## 5.2 ТРЕБОВАНИЯ К АС ВПС И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Сервер АС ВПС должен быть подключен к интернету. Обращение к серверу ПЦ осуществляется через интернет по шифрованному соединению. Уровень защиты и способ шифрования потока данных на усмотрение ПЦ.

Общие требования к программному обеспечению:

1. Вся передача данных происходит по шифрованному соединению.
2. Каждый платеж снабжается уникальным идентификатором, который обеспечивает отсутствие повторных платежей на текущий номер.
3. АС ВПС взаимодействует с системой, отправляя POST-запросы на сервер ПЦ.

## 6 ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

### 6.1 СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ

Сетевое взаимодействие ВПС и ПЦ должно происходить в рамках протокола HTTPS. Для взаимодействия с ВПС, ПЦ предоставляет адрес, на который будут отправляться запросы. Запросы отправляются методом POST.

При ответе на полученный пакет ПЦ возвращает пакет с подписью, в котором содержатся результаты обработки данных пакета. Данные должны помещаться в теле страницы, генерируемой в ответ на обращение со стороны АС ВПС.

Сервер ПЦ должен иметь возможность обрабатывать параллельно несколько запросов.

Авторизация на сервере ПЦ может происходить тремя способами:

1. С использованием электронной подписи.
2. С использованием персонального сертификата.
3. С использование логина и пароля.

Выполнить расшифровку возвращаемых данных пакета можно по таблицам, представленным в приложении [«А. Результат данных, возвращаемых ПЦ»](#).

## 6.2 ЭЛЕКТРОННАЯ ПОДПИСЬ

В случае использования авторизации с помощью ЭП, для гарантии подлинности отправителя и получателя, используется электронная подпись запроса и ответа. Доступны варианты использования подписей в формате RSA и ГОСТ Р 34.10-2012.

### 6.2.1 ЭЛЕКТРОННАЯ ПОДПИСЬ В ФОРМАТЕ RSA

Для авторизации запроса ЭП передается в формате Base64 в HTTP-заголовке запроса PayLogic-Signature. Для авторизации ответа ЭП передается в формате Base64 в HTTP-заголовке ответа PayLogic-Signature.

Ключи генерируются на сервере ПЦ при создании точки с типом «Шлюз» для ВПС.

Примеры создания и проверки подписи представлены в приложении [«В. Генерация и проверка ЭП»](#).

### 6.2.2 ЭЛЕКТРОННАЯ ПОДПИСЬ В ФОРМАТЕ ГОСТ Р 34.10-2012

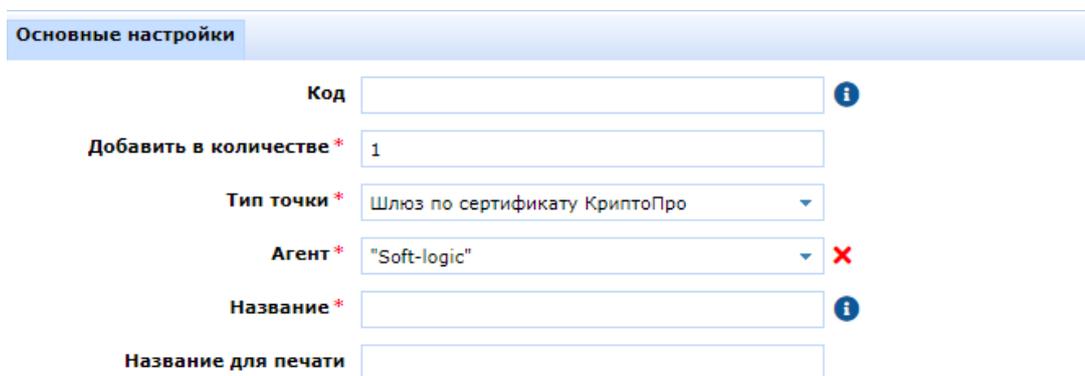
Для авторизации с ГОСТ-подписью в бек-офисе должен быть создан новый тип точки, в настройках которого будет указано **«Авторизация по КриптоПро ключам» = ДА**.

Создание нового типа точки описано в документе [«Программное обеспечение «Процессинговый центр Pay-logic». Руководство администратора»](#).

С помощью криптопровайдера КриптоПро на стороне агента должны быть сгенерированы ключи в одном из форматов:

1. CryptoProSignature;
2. CryptoProSignature\_2012\_256;
3. CryptoProSignature\_2012\_512.

Далее в кабинете процессинга необходимо создать новую точку в разделе «Точки – Точки», указав **«Тип точки»** с авторизацией по КриптоПро, в примере используется тип «Шлюз по сертификату КриптоПро» (рисунок 6.2.2.1).



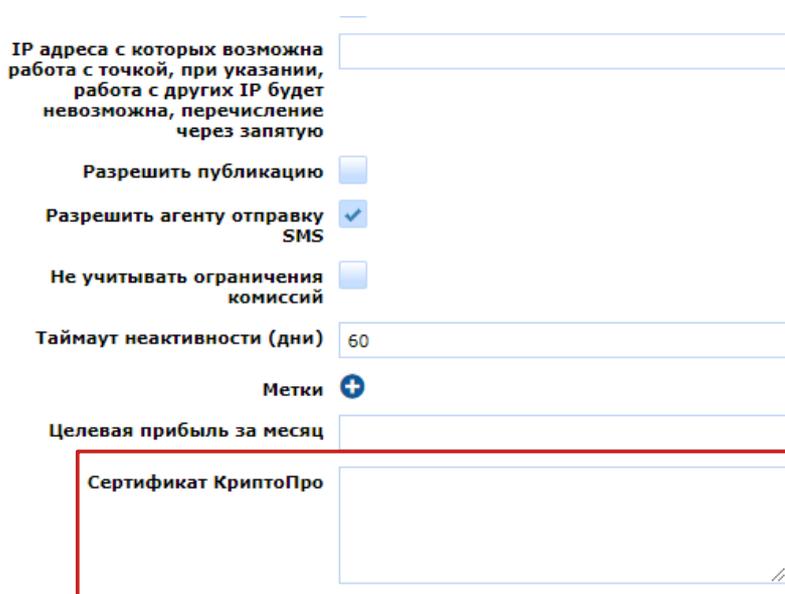
Основные настройки	
Код	<input type="text"/>
Добавить в количестве *	<input type="text" value="1"/>
Тип точки *	<input type="text" value="Шлюз по сертификату КриптоПро"/>
Агент *	<input type="text" value="Soft-logic"/>
Название *	<input type="text"/>
Название для печати	<input type="text"/>

Рисунок 6.2.2.1 — Создание точки с типом авторизации по КриптоПро

После того как точка создана, нужно открыть точку с помощью опции «Изменить». В настройках точки доступно поле **«Сертификат КриптоПро»** (рисунок 6.2.2.2), в которое необходимо прописать открытую часть сгенерированного ключа в формате Base64.

Получить его в необходимом виде можно либо закодировав файл в формате DER в Base64, либо удалив из файла в формате PEM заголовки

-----BEGIN CERTIFICATE----- и -----END CERTIFICATE-----.



IP адреса с которых возможна работа с точкой, при указании, работа с других IP будет невозможна, перечисление через запятую

Разрешить публикацию

Разрешить агенту отправку SMS

Не учитывать ограничения комиссий

Таймаут неактивности (дни) 60

Метки

Целевая прибыль за месяц

**Сертификат КристоПро**

Рисунок 6.2.2.2 — Поле «Сертификат КристоПро» в настройках точки

Далее, с помощью закрытой части сгенерированного ключа, требуется сформировать подпись. Подпись формируется по телу подписываемого сообщения.

Пример создания и проверки подписи приведен в приложении «В. Генерация и проверка ЭП».

Сформированная подпись передается в формате Base64 в HTTP-заголовке запроса PayLogic-Signature-Gost.

Ответ на запрос от сервера будет сгенерирован так же с подписью в формате ГОСТ, подпись формируется по телу ответа с помощью закрытой части ключа, расположенной на сервере ПЦ. Подпись ответа передается в HTTP-заголовке ответа PayLogic-Signature-Gost. Сертификат для проверки подлинности подписи необходимо запросить у сотрудников ПЦ.

### 6.3 ПЕРСОНАЛЬНЫЙ СЕРТИФИКАТ

В случае использования персонального сертификата для целей авторизации ВПС на сервере ПЦ каждый запрос на сервер ПЦ должен происходить с использованием персонального сертификата, полученного при создании точки с типом «Шлюз по сертификату».

### 6.4 ЛОГИН И ПАРОЛЬ

Для авторизации по логину и паролю необходимо создать в бек-офисе новую точку с типом «Шлюз» или «Шлюз по сертификату» (рисунок 6.4.1). Создание нового типа точки описано в документе [«Программное обеспечение «Процессинговый центр Pay-logic». Руководство администратора»](#).

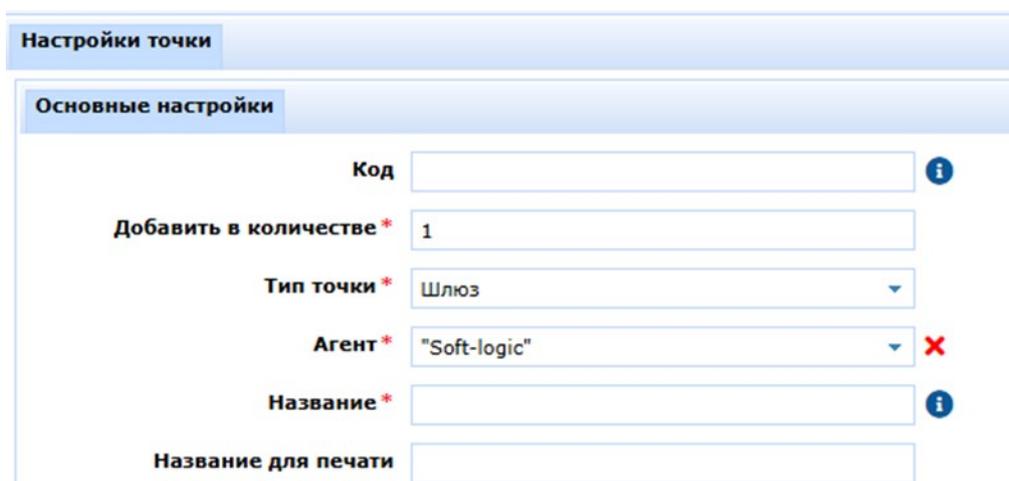


Рисунок 6.4.1 — Создание точки типа «Шлюз»

Следующим шагом для авторизации необходимо привязать пользователя кабинета процессинга к точке. Для этого перейдите в раздел «Агент — Пользователи — Пользователи» и в форме редактирования пользователя, который будет работать с

точкой, выберите созданную точку в параметре «Точка для проведения платежей» (рисунок 6.4.2).

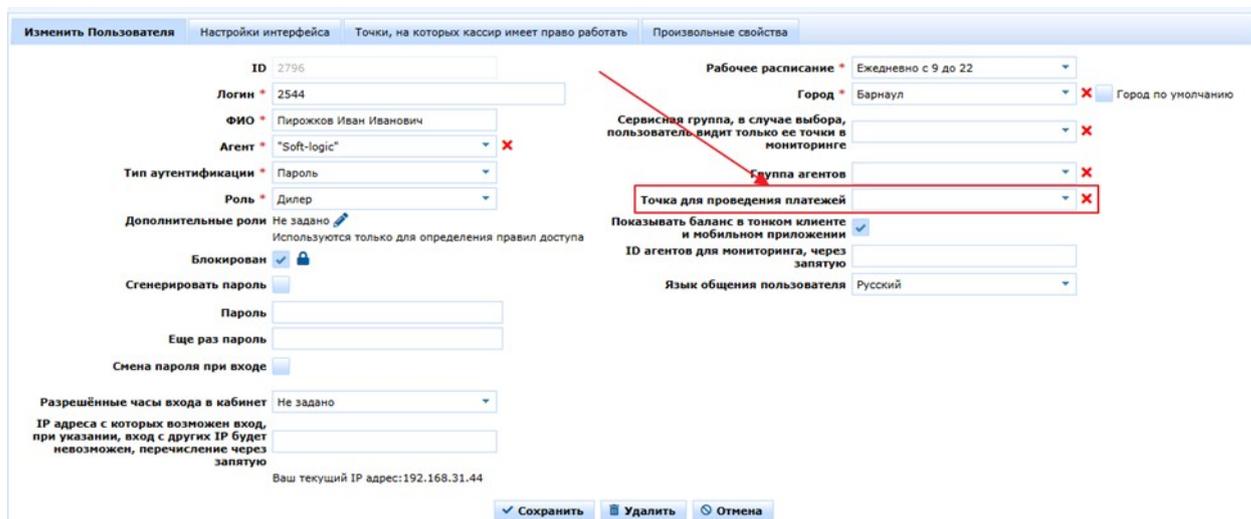


Рисунок 6.4.2 — Редактирование пользователя, заполнение параметра «Точка для проведения платежей»

Для прохождения авторизации в запросах необходимо передавать 2 HTTP-заголовка:

- **Pay-logic-Login** — логин пользователя, к которому привязана точка для проведения платежей;
- **Pay-logic-Password** — пароль пользователя, к которому привязана точка для проведения платежей.

## 7 СОЗДАНИЕ ТОЧКИ ТИПА ШЛЮЗ

В разделе меню «Точки — Точки» создание точек типа «Шлюз» и «Шлюз по сертификату» происходит однотипно, пример создания точки представлен на рисунках 7.1, 7.2.

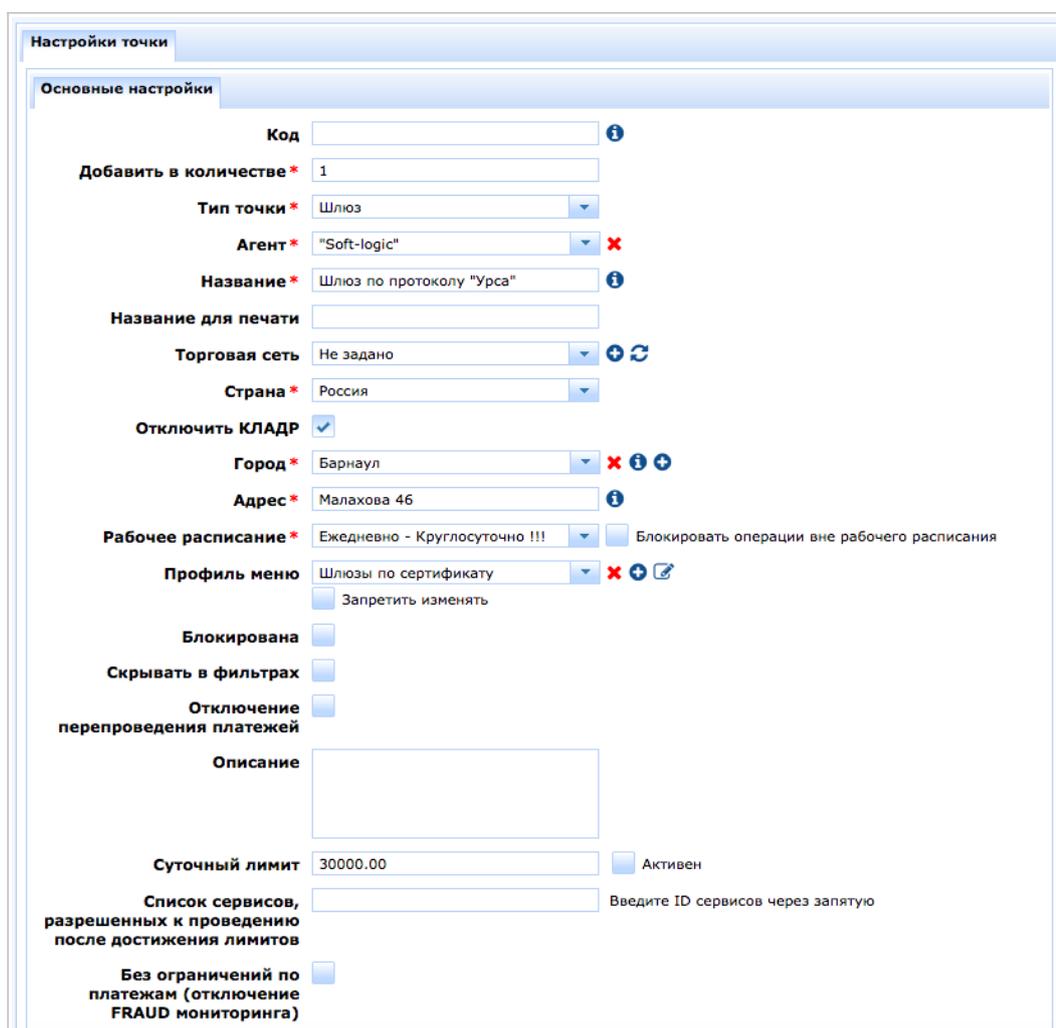
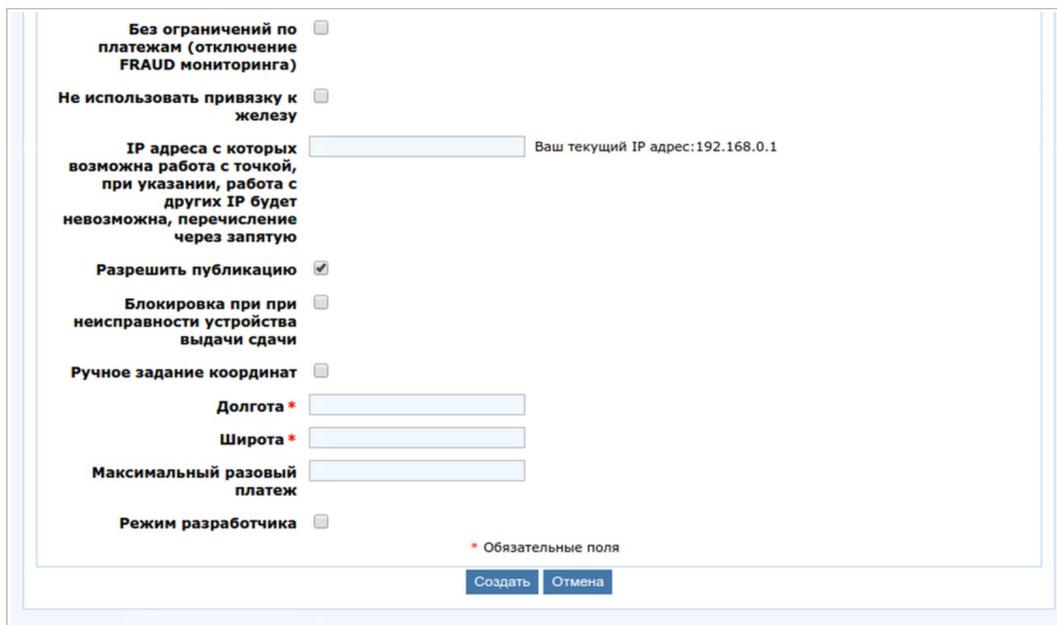


Рисунок 7.1 — Создание точки типа «Шлюз»



Form fields and options for creating a Gateway point:

- Без ограничений по платежам (отключение FRAUD мониторинга)
- Не использовать привязку к железу
- IP адреса с которых возможна работа с точкой, при указании, работа с других IP будет невозможна, перечисление через запятую:  Ваш текущий IP адрес: 192.168.0.1
- Разрешить публикацию
- Блокировка при неисправности устройства выдачи сдачи
- Ручное задание координат
  - Долгота \*
  - Широта \*
  - Максимальный разовый платеж
- Режим разработчика

\* Обязательные поля

Buttons:

Рисунок 7.2 — Создание точки типа «Шлюз». Продолжение

Для точки типа «Шлюз» после сохранения ее в системе необходимо сгенерировать ключи (рисунок 7.3) и сохранить их (рисунок 7.3), после чего сохранить изменения.

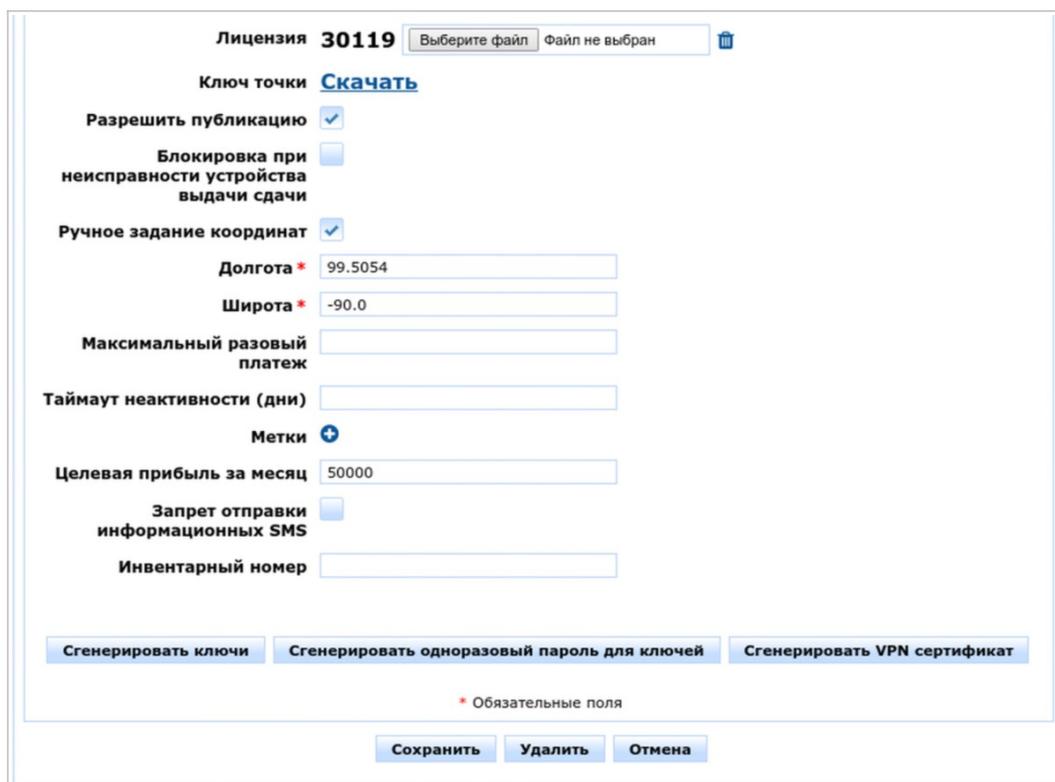


Рисунок 7.3 — Генерация ключей точки шлюз и сохранение ключей точки типа «Шлюз»

Номер точки и ключ передаются платежной системе, которая будет присылать запросы, для настройки шлюза. URL запроса в таком случае:

`/external/extended`

В случае точки типа «Шлюз по сертификату» необходимо сгенерировать сертификат (рисунок 7.4), сохранить его (рисунок 7.5) и сохранить пароль для сертификата (рисунок 7.6). После этого сохранить изменения.

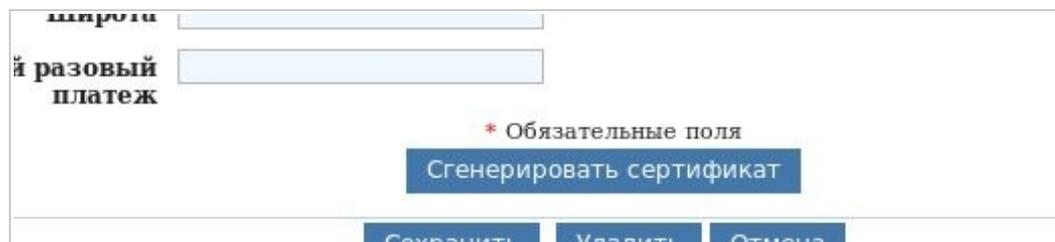


Рисунок 7.4 — Генерация сертификата для точки типа «Шлюз по сертификату»

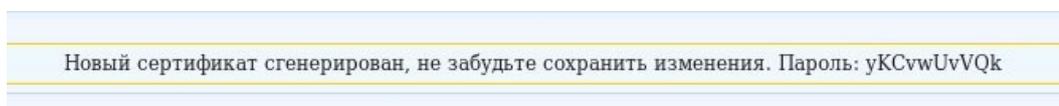


Не использовать привязку к железу

Сертификат точки [Скачать](#)

IP адреса с которых возможна работа с точкой,  Ваш текущий IP

Рисунок 7.5 — Сохранение сертификата точки типа «Шлюз по сертификату»



Новый сертификат сгенерирован, не забудьте сохранить изменения. Пароль: yKCvwUvVQk

Рисунок 7.6 — Получение пароля сертификата для точки типа «Шлюз по сертификату»

Номер точки, сертификат и пароль от сертификата передаются платежной системе, которая будет присылать запросы, для настройки шлюза. URL запроса в таком случае:

```
/external/extended-cert
```

Для разделения логирования по точкам выполните следующие 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
```

## 8 ПРОВЕДЕНИЕ ПЛАТЕЖЕЙ

### 8.1 ЖИЗНЕННЫЙ ЦИКЛ ОПЕРАЦИЙ

Протокол обмена предполагает работу в соответствии с жизненным циклом проведения операций. Жизненный цикл состоит из нескольких этапов, последовательно сменяющих друг друга. Каждому этапу жизненного цикла соответствует определенный запрос, с помощью которого выполняются необходимые действия.

Жизненный цикл операции состоит из двух стадий:

- 1. Первая стадия, или стадия ввода данных** — предполагает подготовку и проверку допустимости (корректности) введенных данных. На этой стадии выполняются запросы `verify` и `advanced`. Проверки на данном этапе являются опциональными и могут не выполняться. Если на стадию проведения будет сформирована операция с неверными или неполными атрибутами, она будет отклонена (получит в итоге ошибочный статус).
- 2. Вторая стадия, или стадия проведения** — на этой стадии выполняется регистрация транзакции в процессинге «Pay-logic», а на стороне агента — дальнейшие действия вплоть до получения финального результата проведения платежа (успех или ошибка). На этой стадии выполняются запросы `pay`, `status` и `confirm`.

Стадия проведения и жизненный цикл операции различаются для случаев проведения операции с онлайн-овым и офлайн-овым проведением.

Схемы действий, выполняемых на этапе проверки и проведения данных приведены рисунках 8.1.1, 8.1.2, 8.1.3.

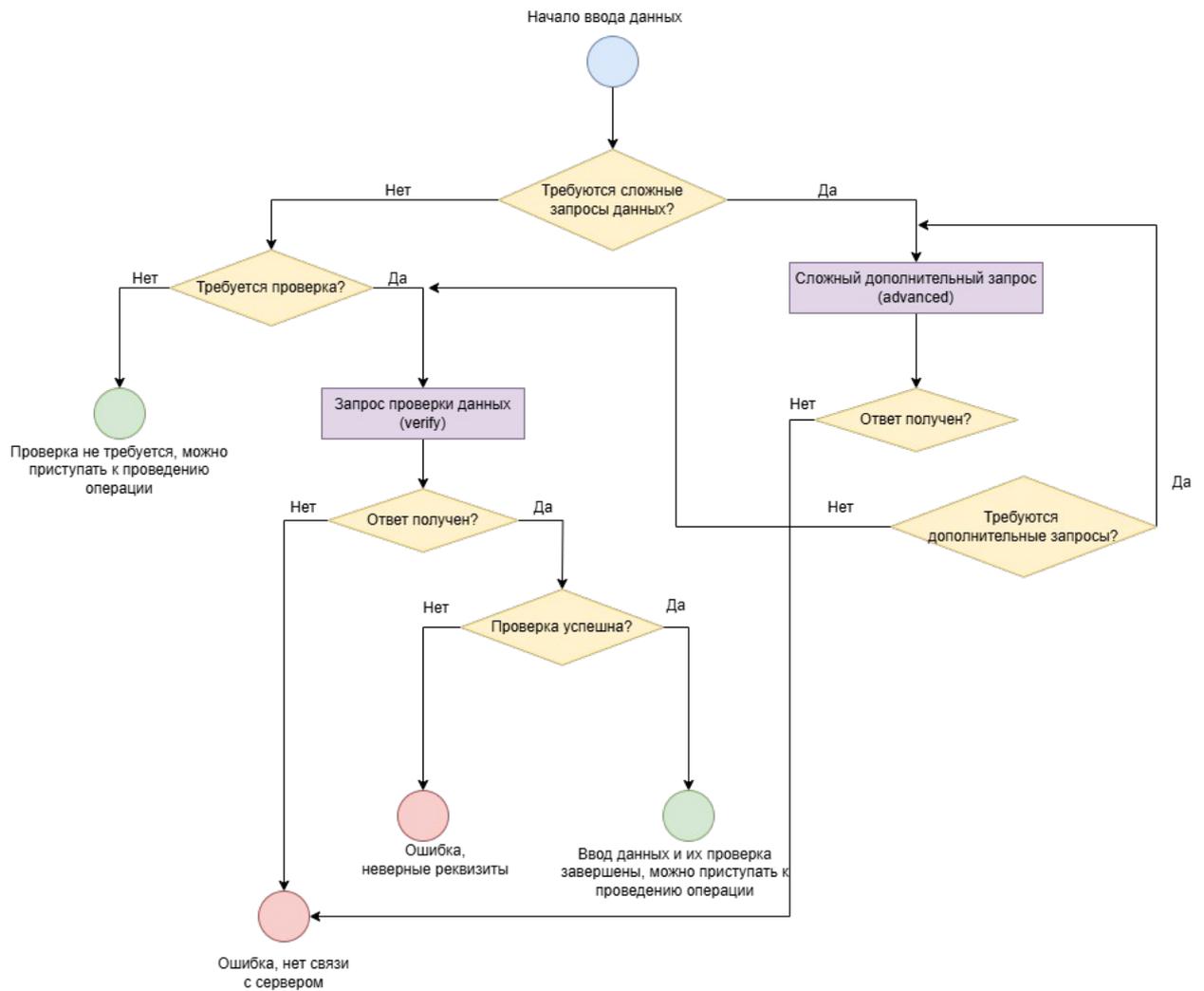


Рисунок 8.1.1 — Проверка данных

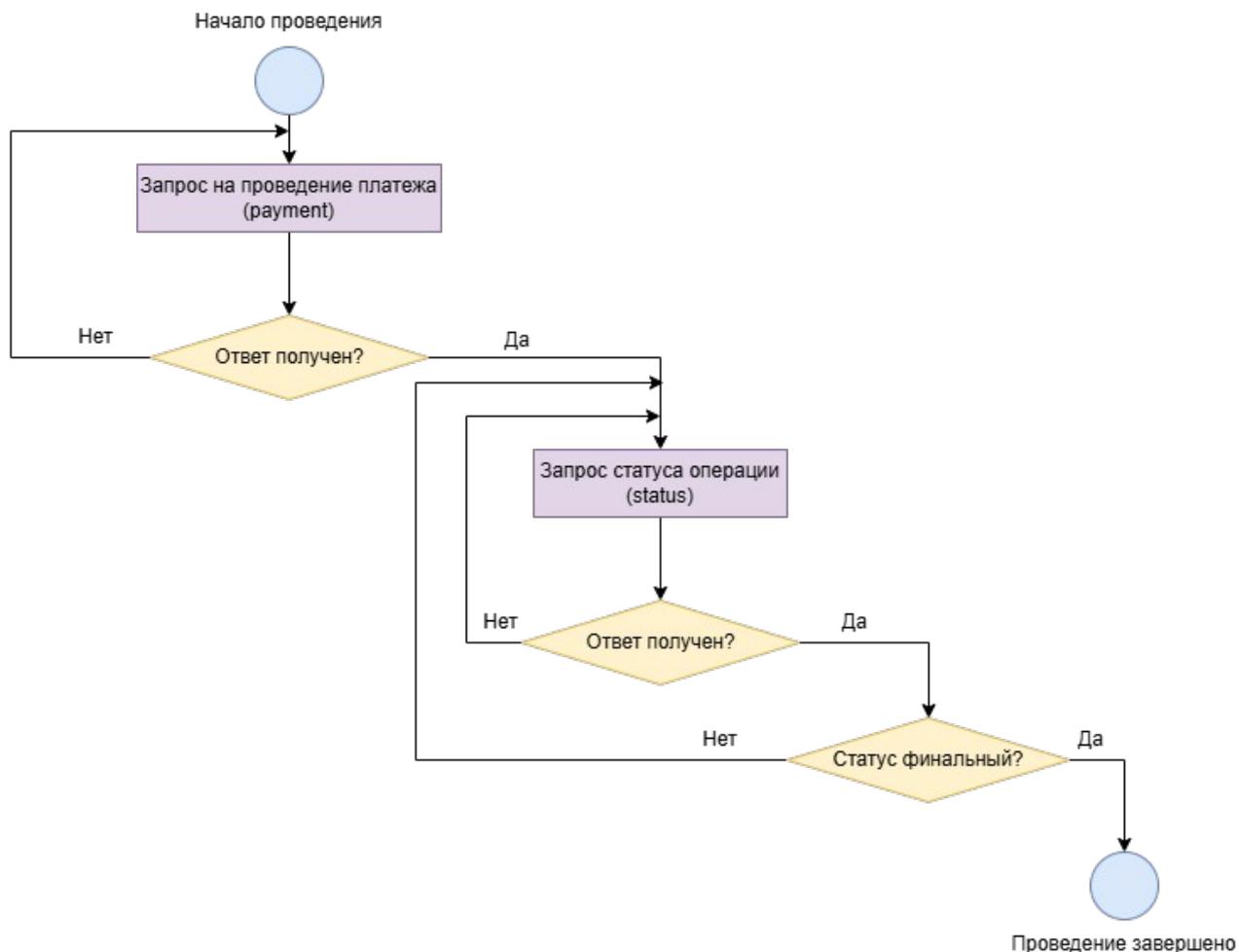


Рисунок 8.1.2 — Проведение операции, не требующей подтверждения

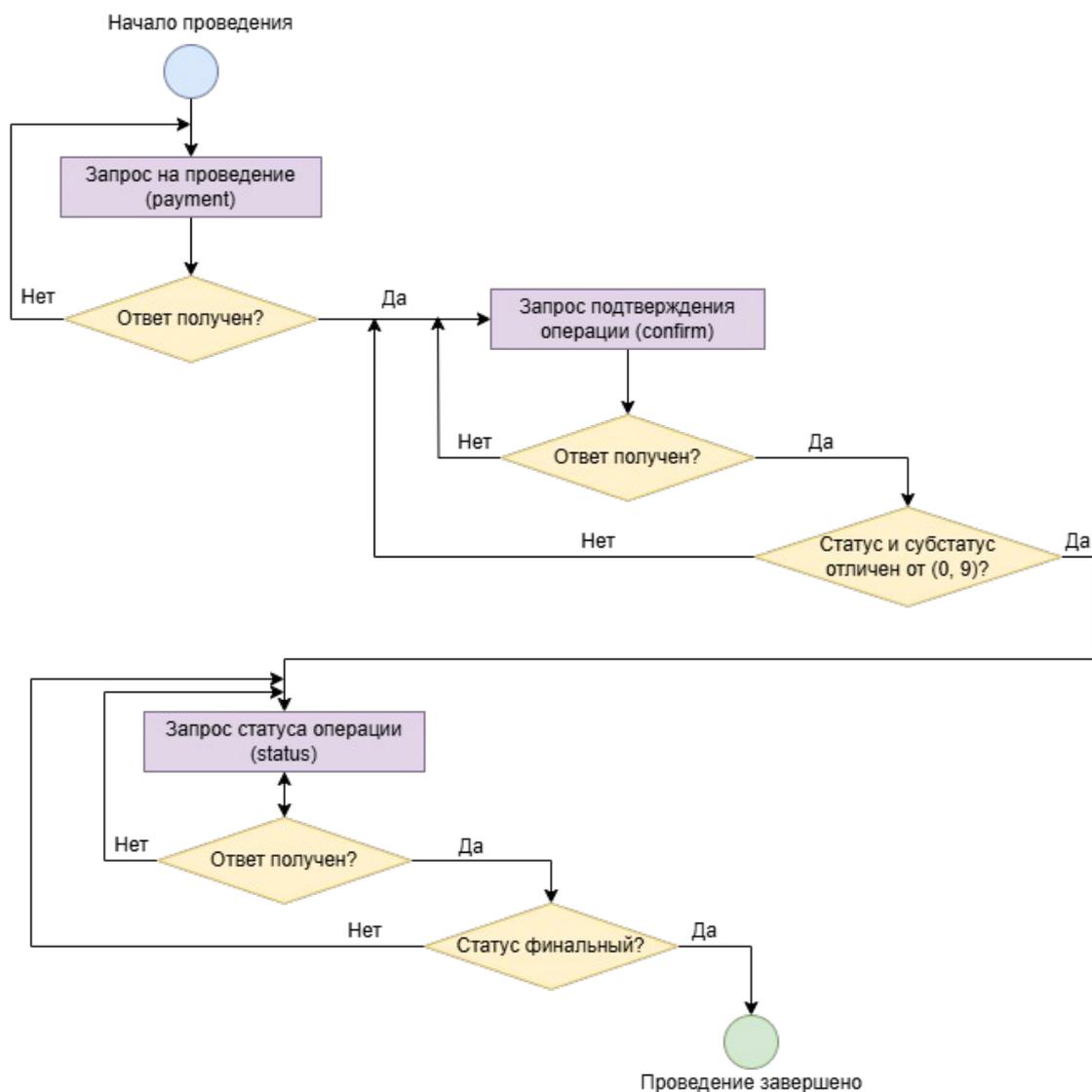


Рисунок 8.1.3 — Проведение отложенной операции (требующей подтверждения)

При реализации протокола следует строго придерживаться приведенных выше схем. Соответствующие запросы, а также статусы, которые возможны в качестве ответов на запросы проверки, подтверждения и запроса статуса приведены далее в спецификации запросов протокола и таблице статусов операций (Приложение [«А. Результат данных, возвращаемых ПЦ»](#)).

**Предварительная верификация** — механизм проверки реквизитов платежа на предмет того, допустимо ли сейчас его проведение. Предварительная верификация позволяет настроить один или более последовательно выполняющих проверки обработчиков — верификаторов, каждый из которых реализует определенного вида проверку, например, проверку превышения оборота за определенное время (в разрезе атрибутов) или проверку реквизитов перевода (фамилии, паспортных данных) в федеральных списках или иных стоп-листах. Фактически, проверка может быть любого рода и содержания.

Для обработки этих проверок в жизненный цикл **введен дополнительный статус 30**, статус и все его субстатусы не являются финальными. Субстатусы соответствуют этапам выполнения проверки по цепочке верификаторов. **Добавлен код ошибки 50** — НЕ пройдена верификация. Для случая онлайн-ового проведения верификаторы не применяются.

## 8.2 ОПИСАНИЕ ФОРМАТОВ ДАННЫХ

Используемые форматы данных приведены в таблице 8.2.1.

Таблица 8.2.1 — Используемые форматы данных

Формат	Описание
intN	Целое N-байтовое число
numeric(N,M)	Дробное, N — количество знаков, M — количество знаков дробной части. Разделитель дробной и целой части — точка

Формат	Описание
varchar(N)	Строковое, не превышающее длиной N байт. Все строки передаются в кодировке UTF-8
boolean	Логическое, допустимые значения — true, false
time	Тип данных, хранящий дату и время в формате YYYY-MM-DDThh:mm:ss[+ -]HHMMI, где: <ol style="list-style-type: none"><li>1. YYYY — год, записанный 4-ми цифрами.</li><li>2. MM — месяц с лидирующим нулем.</li><li>3. DD — день месяца.</li><li>4. T — разделитель.</li><li>5. hh — количество часов в 24-часовом формате с лидирующим нулем.</li><li>6. mm — количество минут с лидирующим нулем.</li><li>7. ss — количество секунд с лидирующим нулем.</li><li>8. [+ -]HHMMI — часовой пояс.</li></ol>

## 8.3 СТРУКТУРНЫЕ ЭЛЕМЕНТЫ ПАКЕТА

### 8.3.1 ОБЩИЕ СВЕДЕНИЯ

Корневым элементом запроса к серверу поставщика является элемент <request>. Корневым элементом ответа является элемент response.

---

Запрос может включать следующие элементы:

1. Запрос баланса (<balance>). В одном пакете может быть один такой элемент.
2. Проверка номера абонента (<verify>). В одном пакете может быть один такой элемент.
3. Сложный запрос проверки реквизитов и вызова произвольных функций по поставщику (<advanced>). В одном пакете может быть один такой элемент.
4. Платеж (<payment>). В одном пакете может быть несколько элементов, но не более 100.
5. Атрибуты платежа (<attribute>). Внутри каждого элемента <payment> может быть произвольное количество атрибутов.
6. Запрос статуса платежа (<status>). В одном пакете может быть несколько элементов, но не более 100.
7. Запрос подтверждения платежа (<confirm>). В одном пакете может быть несколько элементов, но не более 100.
8. Запрос на регистрацию точки приема платежей у поставщика (<point>). В одном пакете может быть несколько элементов, но не более 100.
9. Запрос на общую или поплатежную сверку операций (<reconciliation>). В одном пакете может быть один такой элемент.
10. Запрос на заявку отмены операции (<cancel>). В одном пакете может быть один такой элемент.
11. Запрос на расчет внешней комиссии с клиента (<commission>). В одном пакете может быть один такой элемент.
12. Запрос каталога услуг (<menu>). В одном пакете может быть один такой элемент.
13. Запрос списка форм (<forms>). В одном пакете может быть один такой элемент.

14. Запрос сценария (<scenario>). Сценарии усовершенствованного обработчика запрашиваются по каждому отдельному сервису.

15. Запрос платежных инструментов (<tools>).

16. Запрос, обрабатывающий операции в разрывном статусе (<payment-gap>). В одном пакете может быть несколько элементов, но не более 100.

Ответ поставщика может включать следующие элементы:

1. Баланс лицевого счета агента.

2. Результат операции (<result>). В одном пакете может быть несколько результатов.

### 8.3.2 КОРНЕВОЙ ЭЛЕМЕНТ — ТЕГ <REQUEST>

Запрос **<request>** включает один обязательный атрибут (таблица 8.3.2.1) и любой из запросов, приведенных в разделе [8.3.1](#).

Таблица 8.3.2.1 — Атрибуты элемента <request>

Название	Тип	Формат	Обяз.	Описание
<b>point</b>	Атрибут	int4	Да	Номер точки приема платежей

ПРИМЕР:

```
<request point="12345">  
    ...  
</request>
```

Любой запрос из запросов, приведенных в разделе [8.3.1](#), должен быть вложен в элемент <request>.

### 8.3.3 БАЛАНС ЛИЦЕВОГО СЧЕТА — ТЕГ <BALANCE>

Для запроса баланса лицевого счета агента необходимо поместить в пакет пустой элемент <balance>. Структура элемента <balance> приведена в таблице 8.3.3.1.

Таблица 8.3.3.1 — Структура элемента <balance>

Название	Тип	Формат	Обяз.	Описание
<b>balance</b>	Атрибут	int8	Нет	Остаток денежных средств минус сумма зарезервированных денежных средств на счете платежной системы в копейках
<b>overdraft</b>	Атрибут	int8	Нет	Сумма овердрафта в копейках
<b>reserved</b>	Атрибут	int8	Нет	Сумма зарезервированных денежных средств на счете платежной системы в копейках
<b>realbalance</b>	Атрибут	int8	Нет	Общий остаток денежных средств на счете платежной системы в копейках

ПРИМЕР ЗАПРОСА БАЛАНСА:

```
<request point="17235">  
  <balance/>  
</request>
```

В ответе должен быть элемент <balance> с остатком лицевого счета ВПС.

ОТВЕТ ПЦ:

```
<response>  
  <balance balance="9000" overdraft="2000" reserved="1000"  
  realbalance="10000"/>  
</response>
```

### 8.3.4 СПРАВОЧНИК ПЛАТЕЖНЫХ ИНСТРУМЕНТОВ — ТЕГ <TOOLS>

Запрос справочника платежных инструментов возвращает список платежных инструментов, доступных агенту для проведения платежей. Доступность платежного инструмента определяется доступностью к проведению для агента сервисов, настроенных на вариант оплаты этим платежным инструментом.

Доступные коды для каждой конкретной системы находятся в справочнике «Источники оплаты» (таблица `payment_sources`) в кабинете «Справочники — Финансы — Источники оплаты».

Для запроса справочников платежных инструментов агент должен поместить в пакет пустой элемент `<tools>`. Опционально в запросе может быть указан идентификатор сервиса, по которому необходимо получить допустимые платежные инструменты. Структура запроса `<tools>` приведена в таблице 8.3.4.1.

Таблица 8.3.4.1 — Структура запроса `<tools>`

Название	Тип	Формат	Обяз.	Описание
<code>service</code>	Атрибут	int2	Нет	Идентификатор сервиса. Опциональный. Если не задан — возвращаются доступные платежные инструменты для всех сервисов

Ответ представляет собой список платежных инструментов `<tools>`. Структура элемента ответа `<tools>` приведена в таблице 8.3.4.2.

Таблица 8.3.4.2 — Структура элемента ответа <tools>

Название	Тип	Формат	Обяз.	Описание
tool	Элемент	-	Да	Описание одного платежного инструмента

Таблица 8.3.4.3 — Структура элемента ответа <tool>

Название	Тип	Формат	Обяз.	Описание
id	Атрибут	int2	Да	Идентификатор платежного инструмента
code	Атрибут	varchar20	Да	Код платежного инструмента
name	Атрибут	varchar100	Да	Название платежного инструмента

ПРИМЕР ЗАПРОСА:

```
<tools />
```

ПРИМЕР ЗАПРОСА:

```
<tools service = 12/>
```

ПРИМЕР ОТВЕТА:

```
<tools>  
  <tool id="0" code="CASH" name="Наличные"/>  
  <tool id="1" code="CBC" name="Карта сдачи"/>  
  <tool id="2" code="BANKCARD" name="Банковская карта"/>  
  <tool id="3" code="KEEPER" name="Кошелек"/>  
  <tool id="4" code="BANKCARD_OTHER" name="Банковская карта другого  
банка"/>  
</tools>
```

### 8.3.5 ОФЛАЙНОВОЕ ПРОВЕДЕНИЕ ОПЕРАЦИИ — ТЕГ <PAYMENT>

Запрос офлайнного проведения предназначен для проведения офлайновых операций, которые могут быть в том числе отложенными. Запрос является пакетным, т. е. в теге <request> может присутствовать более одного тега <payment>.

Результатом выполнения запроса <payment> является создание офлайновых операций. В ответе вернется пакет с тегами <result> для каждого <payment>, причем статус операции в результате нефинальный. Для получения актуального статуса необходимо выполнять запрос <status> (раздел [8.3.10](#)).

Структура элемента <payment> приведена в таблице 8.3.5.1.

Таблица 8.3.5.1 — Структура элемента <payment>

Название	Тип	Формат	Обяз.	Описание
<b>id</b>	Атрибут	int8	Да	Идентификатор операции агента. ПЦ должен гарантировать отсутствие у себя в базе повторных операций с одинаковым <b>id</b> . Если поступает операция, которая уже есть в базе, в ответ необходимо вернуть текущий статус операции
<b>sum</b>	Атрибут	int4	Да	Сумма пополнения лицевого счета в копейках
<b>sum-in</b>	Атрибут	int4	Нет	Сумма вложенная, в копейках. Если сумма передана, шлюзом рассчитывается сумма комиссии как разница между

Название	Тип	Формат	Обяз.	Описание
				<b>sum-in</b> и <b>sum</b>
<b>check</b>	Атрибут	int2	Да	Номер чека, выданного клиенту. Переданное значение сохраняется, если попадает в диапазон от 0 до 32767. В противном случае сохраняется 0
<b>service</b>	Атрибут	int4	Да	Номер услуги
<b>source</b>	Атрибут	String	Нет	Код платежного инструмента. Если не указан — считается, что платеж наличный. Список кодов, доступных в системе, приведен на странице «Справочники — Финансы — Источники оплаты». Если указан любой другой источник оплаты, операция считается банковской
<b>account</b>	Атрибут	varchar100	Да	Номер абонента в системе поставщика услуг
<b>date</b>	Атрибут	time	Да	Дата поступления платежа в АС ВПС. Используется для сверки
<b>attribute</b>	Элемент	-	Нет	Используется для указания дополнительных атрибутов операции (кроме банковских). Может быть несколько. Название атрибутов определяет ПЦ. Структура элемента приведена в таблице 8.3.5.2

Название	Тип	Формат	Обяз.	Описание
<b>derivation</b>	Элемент	String	Нет	Используется для указания дополнительных атрибутов банковской операции. Структура элемента совпадает со структурой элемента <code>&lt;attribute&gt;</code>
<b>delayed</b>	Атрибут	int1	Нет	Признак отложенной операции. Если он принимает значение 1, то платеж ставится в статус «Ожидает подтверждения» и будет находиться в этом статусе до выполнения запроса <code>&lt;confirm&gt;</code> для данной id-транзакции
<b>terminal-vps-id</b>	Атрибут	varchar100	Нет	Идентификатор точки во внешней платежной системе
<b>bill-id</b>	Атрибут	int4	Нет	Идентификатор начисления, по которому создается платеж. Начисления получаются через систему начислений
<b>send-sms-check</b>	Атрибут	varchar10	Нет	Признак необходимости отправки SMS-оповещения о проведении операции (SMS-чека). Допустимые значения: <b>true, false</b> . Отправка чека выполняется при условии, что шаблон SMS-чека для сервиса задан в настройках кабинета процессинга, и для Партнера по шлюзу отправка SMS-

Название	Тип	Формат	Обяз.	Описание
				оповещений разрешена
<b>online-verify</b>	Атрибут	boolean	Нет	Используется, чтобы произвести онлайн-проверку перед созданием платежа. Проверка осуществляется в шлюзе, проводящем платеж. Если проверка прошла успешно — платеж будет создан. В ином случае вернется ошибка. Возможные значения: true, false

Таблица 8.3.5.2 — Структура элемента <attribute>

Название	Тип	Формат	Обяз.	Описание
<b>name</b>	Атрибут	varchar50	Да	Название атрибута
<b>value</b>	Атрибут	varchar100	Да	Значение атрибута
<b>key-title</b>	Атрибут	varchar	Нет	Альтернативное название атрибута для отображения в кабинете. Если он отсутствует в запросе или имеет пустое значение, или его значение состоит только из пробелов, то вместо него в кабинете будет отображаться значение <b>name</b>
<b>value-title</b>	Атрибут	varchar	Нет	Альтернативное значение атрибута, предназначенное для отображения в кабинете. Если <b>value-title</b> отсутствует в запросе или имеет пустое значение,

---

Название	Тип	Формат	Обяз.	Описание
				или его значение состоит только из пробелов, то вместо него будет использовано значение <b>value</b>

**ПРИМЕР СТАНДАРТНОГО ЗАПРОСА:**

```
<request point="17235">
  <payment id="14546"
    sum="1000"
    check="17235"
    service="1"
    account="9132345678"
    date="2007-10-12T12:00:00+0300"/>
</payment>
</request>
```

**ОТВЕТ ПЦ:**

```
<response>
  <result id="14546" state="60"
    substate="0" code="0"
    final="1" trans="123"/>
</response>
```

**ПРИМЕР ЗАПРОСА С ДОПОЛНИТЕЛЬНЫМИ АТТРИБУТАМИ (НАЛИЧНЫЙ ПЛАТЕЖ):**

```
<request point="17235">
  <payment id="14547"
    sum="1000"
    check="17235"
    service="2"
    account="12345"
    date="2007-10-12T12:00:00+0300">
    <attribute name="email"
      value="info@rol.ru"/>
  </payment>
</request>
```

## ПРИМЕР ЗАПРОСА С АТТРИБУТАМИ KEY-TITLE, VALUE-TITLE

```
<request point="1145">
  <payment id="383828" service="4341" sum="10000" check="1"
account="000 000 000 000 000 000" date="2021-03-25T11:45:43+0700">
  <!-- <attribute name="keeper.login" value="9039585986"/> -->
  <attribute name="att1" key-title="Attribute 1" value="123"
value-title="Value 123"/>
  <attribute name="dg" value="29661" key-title="Номер
договора"/>
  <attribute name="bik" value="040349602" key-title="БИК Банка
получателя"/>
  <...>
</request>
```

ПРИМЕР ЗАПРОСА С ДОПОЛНИТЕЛЬНЫМИ АТТРИБУТАМИ (БАНКОВСКАЯ  
ОПЕРАЦИЯ):

```
<request point="1878">
  <payment account="1400064171"
check="1"
date="`${now}`"
id="`${seq-number}`"
service="15577"
sum="900"
sum-in="1000"
source="card">
  <attribute name="transaction-id"
value="50515220200317114502848198914"/>
  <attribute name="invoice-id"
value="OQ==_MTQwMDA2NDE3MQ==_MTExMTMwMDA="/>
  <attribute name="invoice-type" value="SIMPLE"/>
  <attribute name="payment-receiver" value="980110"/>
  <attribute name="fee-calculation-method" value="GPP_FEE"/>
  <attribute name="id-subtype" value="AVIS"/>
  <attribute name="service-code" value="11113000"/>
  <attribute name="invoice-code" value="3138549"/>
  <attribute name="invoice-date" value="2020-03-29"/>
  <attribute name="invoice-id-subtype" value="DOC2"/>
  <attribute name="client-pin" value="123"/>
  <derivation name="rrn" value="12345498735216543652573"/>
  </payment>
</request>
```

Для того, чтобы в ответе ПЦ возвращалось описание кода ошибки, в конфигурационном файле коннектора `/srv/glassfish3/config.xml` необходимо задать значение **true** для следующего параметра:

```
<paylogic-message-in-response>true</paylogic-message-in-response>
```

После этого в значении параметра **message** будет возвращено описание ошибки платежа. Параметр является необязательным. По умолчанию задано значение **false**.

ОТВЕТ ПЦ С ОПИСАНИЕМ КОДА ОШИБКИ:

```
<response>
  <result id="513" state="80" substate="0" code="3"
    message="Сумма вне допустимого диапазона" final="1"
    trans="0">
  </result>
</response>
```

Если в запросе указан платежный инструмент, помимо основной операции создаются операции списания, и в таком случае проведение выполняется согласно жизненному циклу проведения операции с использованием платежного инструмента.

ПРИМЕР ЗАПРОСА С ПЛАТЕЖНЫМ ИНСТРУМЕНТОМ:

```
<payment id="14546"
  sum="1000" check="17235"
  service="2" source="BANKCARD"
  account="12345" date="2007-10-12T12:00:00+0300">
  <attribute name="email" value="info@rol.ru"/>
  <derivation name="pan" value="749522312321330012"/>
  <derivation name="expiry" value="1912"/>
</payment>
```

При выполнении платежа с внешним платёжным инструментом могут вернуться:

1. Атрибуты операции списания средств в элементе `<derivation>` (аналогично `<attribute>`). Пример:

```
<response>
  <result id="123" state="60" substate="0" code="0" ps_code="0"
    final="1" trans="123" server_time="2016-12-01T15:00:21+0300">
```

```
<derivation name="auth-code" value="123456"/>
<derivation name="bank-trans" value="a5e9021789e84f499e2a"/>
</result>
</response>
```

2. Код ошибки платёжного инструмента в атрибуте **ps\_code** (список кодов приведен в таблице А.7.1). Атрибут **code** (таблица А.2.1) в этом случае равен -2. Пример:

```
<result id="123" state="80" substate="7" code="-2" ps_code="4" final="1"
trans="123"/>
```

3. Атрибуты с указанием кода и текста ошибки провайдера (через которого проводится платёж) или ошибки провайдера платёжного инструмента (через которого выполняется списание средств). Пример:

```
<attribute name="provider-error-code" value="..."/>
<attribute name="provider-error-text" value="..."/>
```

## ОСОБЕННОСТИ ОБРАБОТКИ ЗАПРОСА С ПОЛЕМ TERMINAL-VPS-ID

В поле **terminal-vps-id** агент должен передать код своей точки.

Например, если **id=123** и **код=444**, то **terminal-vps-id="444"**. Если **id=123** и **код=123**, то **terminal-vps-id="123"**.

## ЗАПРОС С АТТРИБУТОМ TERMINAL-VPS-ID:

```
<payment id="14546"
sum="1000"
check="17235"
service="2"
account="12345"
date="2007-10-12T12:00:00+0300"
terminal-vps-id="27">
```

ПЦ, обрабатывая такой запрос на транзакцию, выполняет следующее:

1. Находит для агента и переданного идентификатора **terminal-vps-id** запись в справочнике зарегистрированных точек внешних платёжных систем.

2. Формирует соответствие нашему элементу справочника внешних платежных систем. Операция в процессинге «Pay-Logic» получает ссылку на внутренний идентификатор внешней точки. Никакие атрибуты в атрибуты платежа при этом не добавляются. Если точка внешней платежной системы по коду не найдена для агента — в операции идентификатор внешней точки проставлен не будет.

Полученное соответствие при проведении платежа используется для передачи поставщику информации о фактически зарегистрированной точке внешнего агента согласно протоколу поставщика услуги.

**Внимание!**

В шлюзе поставщика услуги должна быть реализована поддержка работы с внешними точками.

---

При дублировании атрибутов сервер ответит ошибкой (- 100).

ОТВЕТ ПЦ:

```
<result id="14546"  
state="60"  
substate="0"  
code="-100"  
final="1"  
trans="123"/>
```

### 8.3.6 ПРОВЕДЕНИЕ ОПЕРАЦИИ В РЕЖИМЕ ОНЛАЙН — ТЕГ <OPAYMENT>

Запрос онлайн-проведения необходим в случае когда требуется получение результата проведения транзакции в онлайн.

Запрос является одиночным, т. е. в теге <request> может присутствовать не более одного тега <opayment>. Ответ на онлайнное проведение возвращается сразу по факту проведения операции либо по истечении максимального таймаута ожидания ответа сервером от подсистемы шлюзов и поставщика (60 секунд).

Ответом на тег <opayment> служит тег <result> (раздел [8.3.13](#)). В случае таймаута ожидания для операции возвращается неизвестный статус и внешней системе необходимо позднее запросить статус операции.

Структура тега идентична структуре тега <payment> за исключением того, что для онлайнного платежа отсутствует возможность совершить отложенный платеж.

При дублировании атрибутов сервер ответит ошибкой (- 100).

#### ПРИМЕР

```
<result id="14546"
state="60"
substate="0"
code="-100"
final="1"
trans="123"/>
```

### 8.3.7 ПАРАМЕТРЫ ПРОВЕРКИ НОМЕРА АБОНЕНТА — ТЕГ <VERIFY>

Запрос проверки номера счета абонента производится в случае онлайн-проверки введенных абонентом данных. Запрос должен включать один элемент <verify>:

```
<request point="17235">  
  <verify service="1" account="12345"/>  
</request>
```

В ответ сервер должен вернуть результат проверки номера, а также атрибуты от поставщика, в случае их наличия. Состав и возможные значения возвращаемых дополнительных атрибутов определяются особенностями проведения конкретного поставщика, и предоставляются сотрудниками платежной системы:

```
<response>  
  <result code="0">  
    <attribute name="balance" value="100.00"/>  
    <attribute name="fio" value="Пупкин И.В."/>  
  </result>  
</response>
```

Структура элемента <verify> приведена в таблице 8.3.7.1.

Таблица 8.3.7.1 — Структура элемента <verify>

Название	Тип	Формат	Обяз.	Описание
<b>service</b>	Атрибут	int4	Да	Номер услуги
<b>account</b>	Атрибут	varchar100	Да	Номер абонента в системе поставщика услуг
<b>attribute</b>	Элемент	-	Нет	Используется для указания дополнительных атрибутов. Может быть несколько. Структура элемента приведена в таблице 8.3.7.2. Описание и состав дополнительных атрибутов для

Название	Тип	Формат	Обяз.	Описание
				каждого поставщика предоставляется сотрудниками Платежной системы
<b>code</b>	Атрибут	int2	Да	Код ошибки платежа или ошибка проверки реквизитов
<b>error-detail</b>	Атрибут	-	Нет	Используется для возврата дополнительных сведений об ошибке в результате неуспешной проверки

Структура элемента `<attribute>` приведена в таблице 8.3.7.2

Таблица 8.3.7.2 — Структура элемента `<attribute>`

Название	Тип	Формат
<b>name</b>	Атрибут	varchar50
<b>value</b>	Атрибут	varchar100
<b>key-title</b>	Атрибут	varchar
<b>value-title</b>	Атрибут	varchar

Структура элемента `<error-detail>` идентична структуре `<attribute>`.  
Структура элемента `<attribute>` приведена выше.

С версии кабинета 5.0.6 добавлена возможность отправлять в запросе атрибуты **key-title** и **value-title**. Таким образом, структура элемента `<attribute>` в ответе и запросе совпадает. Если в запросе атрибут **key-title** не задан, имеет пустое значение или его значение состоит только из символов пробела, то в качестве значения атрибута **key-title** используется значение атрибута **name**. Если в запросе атрибут **value-title** отсутствует, имеет пустое значение или его значение состоит только из символов пробела, то используется значение **value**.

---

**ПРИМЕР ЗАПРОСА С ДОПОЛНИТЕЛЬНЫМИ АТТРИБУТАМИ:**

```
<verify service="2" account="12345">  
  <attribute name="email" value="info@rol.ru"/>  
</verify>
```

**ПРИМЕР ОТВЕТА НА ПРОВЕРКУ С ДОПОЛНИТЕЛЬНЫМИ АТТРИБУТАМИ:****Успех:**

```
<response>  
  <result code="0">  
    <attribute name="balance" value="23450"/>  
    <attribute name="fio" value="Иванов Иван Иванович"/>  
  </result>  
</response>
```

**Ошибка:**

```
<response>  
  <result code="1000">  
    <error-detail name="description" value="Клиент не участвует в  
проекте"/>  
  </result>  
</response>
```

**ПРИМЕР ОТВЕТА НА ПРОВЕРКУ, ЕСЛИ В ПАРАМЕТРЕ PAYLOGIC-VERIFY-SEPARATE-KEY-TITLE ЗАДАНО ЗНАЧЕНИЕ FALSE:****Успех:**

```
<response>  
  <result code="0">  
    <attribute name="debt" value="23450"/>  
    <attribute name="Договор" value="№123456 от 01.01.2001"/>  
    <attribute name="ФИО" value="Иванов Иван Иванович"/>  
  </result>  
</response>
```

**Ошибка:**

```
<response>
  <result code="1000">
    <error-detail name="description" value="Неправильно набран номер"/>
  </result>
</response>
```

ПРИМЕР ОТВЕТА НА ПРОВЕРКУ, ЕСЛИ В ПАРАМЕТРЕ PAYLOGIC-VERIFY-SEPARATE-KEY-TITLE ЗАДАНО ЗНАЧЕНИЕ TRUE:

**Успех:**

```
<response>
  <result code="0">
    <attribute name="debt" key-title="debt" value="23450"
value-title="23450"/>
    <attribute name="contract" key-title="Договор" value="123456"
value-title="№123456 от 01.01.2001"/>
    <attribute name="fio" key-title="ФИО" value="Иванов Иван Иванович"
value-title="Иванов Иван Иванович"/>
  </result>
</response>
```

**Ошибка:**

```
<response>
  <result code="1000">
    <error-detail name="description" key-title="description"
value="Неправильно набран номер" value-title="Неправильно набран номер"/>
  </result>
</response>
```

### 8.3.8 РАСЧЕТ ВНЕШНЕЙ КОМИССИИ — ТЕГ <COMMISSION>

Запрос расчета внешней комиссии производится для предварительного отображения размера комиссии клиенту. Запрос должен включать один элемент <commission>. Структура элемента <commission> приведена в таблице 8.3.8.1.

Таблица 8.3.8.1 — Структура элемента <commission>

Название	Тип	Формат	Обяз.	Описание
<b>service</b>	Атрибут	int4	Да	Номер услуги
<b>sum</b>	Атрибут	int4	Да	Сумма платежа в копейках
<b>source</b>	Атрибут	String	Нет	Код платежного инструмента

#### ЗАПРОС РАСЧЕТА:

```
<request point="17235">  
  <commission service="1" sum="10000"/>  
</request>
```

#### ОТВЕТ ПЦ С РЕЗУЛЬТАТОМ РАСЧЕТА КОМИССИИ:

```
<response>  
  <result code="0" commission="500">  
  </result>  
</response>
```

Если сумма комиссии отличается в зависимости от источника оплаты, в элементе <commission> возможно указать код платежного инструмента:

```
<commission service="2" sum="1000" source="BANKCARD"/>
```

### 8.3.9 ПАРАМЕТРЫ СЛОЖНОЙ ПРОВЕРКИ РЕКВИЗИТОВ — ТЕГ <ADVANCED>

Структура элемента <advanced> приведена в таблице 8.3.9.1.

Таблица 8.3.9.1 — Структура элемента <advanced>

Название	Тип	Формат	Обяз.	Описание
<b>service</b>	Атрибут	int4	Да	Номер услуги
<b>function</b>	Атрибут	varchar100	Да	Название функции сложной проверки. Названия доступных функций для каждого поставщика, а также требования к дополнительным атрибутам предоставляется сотрудниками Платежной системы
<b>provider</b>	Атрибут	int2	Нет	Указывается идентификатор провайдера в процессинге. Позволяет указать, через какого провайдера (какой шлюз) обрабатывать запрос. Коннектор обрабатывает атрибут, если у точки, приславшей запрос, установлен флаг «Доверенная внешняя точка». Если флаг не установлен, то атрибут не обрабатывается, даже если он задан
<b>attribute</b>	Элемент	-	Нет	Используется для указания дополнительных атрибутов. Может быть несколько. Структура такая же, как и в <payment> (таблица 8.3.5.2)
<b>id</b>	Атрибут		Нет	Идентификатор операции на стороне агента. Если агент передает данный атрибут в запросе, то перед отправкой запроса провайдеру будет выполнен метод

Название	Тип	Формат	Обяз.	Описание
				<p>резервирования ID операции на сервере процессинга. Если резервирование прошло успешно, ID платежа процессинга будет сохранен в одном из следующих атрибутов:</p> <ul style="list-style-type: none"><li>• Если в произвольных свойствах сервиса задано свойство <b>reserve-id.external-responce-param-name</b>, то название параметра будет взято из его значения;</li><li>• Если произвольное свойство не задано, то ID сохранится в параметре <b>operation-id</b>.</li></ul> <p>Пример запроса от агента:</p> <pre>&lt;request point="1878"&gt;   &lt;advanced function="test"     service="4341" id="129"&gt;   &lt;/advanced&gt; &lt;/request&gt;</pre> <p>Пример ответа агенту при успехе:</p> <pre>&lt;result code="0" service="4341"&gt;   &lt;data&gt;     &lt;input key="operation- id" value="123"/&gt;   &lt;/data&gt; &lt;/result&gt;</pre> <p>В случае повторного advanced-запроса с указанным в нем ID платежа сначала будет выполнена проверка, не проводилось ли резервирование ранее. Если проводилось, то данные будут взяты из его результатов</p>

## ПРИМЕР:

```
<advanced service="1" function="simple" provider="10"/>
```

## ПРИМЕР С ДОПОЛНИТЕЛЬНЫМИ АТРИБУТАМИ:

```
<advanced service="2" function="check">  
  <attribute name="email" value="info@rol.ru"/>  
  <attribute name="account" value="123456"/>  
</advanced>
```

### 8.3.10 ПАРАМЕТРЫ ПРОВЕРКИ СТАТУСА ПЛАТЕЖА — ТЕГ <STATUS>

В одном запросе может содержаться произвольное количество тегов <status>. Результат выполнения запроса — тег <result> (раздел [8.3.13](#)).

Структура элемента <status> приведена в таблице 8.3.10.1.

Таблица 8.3.10.1 — Структура элемента <status>

Название	Тип	Формат	Обяз.	Описание
id	Атрибут	int8	Да	Идентификатор операции агента

#### ЗАПРОС СТАТУСА:

Запрос должен включать один или несколько элементов <status>:

```
<request point="17235">  
  <status id="123"/>  
  <status id="125"/>  
</request>
```

#### ОТВЕТ ПЦ:

В ответ сервер должен вернуть результат для каждого элемента <status>:

```
<response>  
  <result id="123" state="60" substate="0" code="0"  
    final="1" trans="123"/>  
  <result id="125" state="40" substate="4" code="0"  
    final="0" trans="312"/>  
</response>
```

---

В случае, если поставщик возвращает атрибуты, то они будут отображены в ответе ПЦ.

#### ЗАПРОС СТАТУСА:

```
<request point="17235">
  <status id="14546"/>
</request>
```

#### ОТВЕТ ПЦ:

```
<response>
  <result id="14546" state="40" substate="7" code="7" final="0"
    trans="3945914" server_time="2016-12-01T15:00:21+0300">
    <attribute name="provider-error-text" value="Ошибка SQL-сервера"/>
  </result>
</response>
```

Для того, чтобы получить описание кода возникшей ошибки, в конфигурационном файле коннектора `/srv/glassfish3/config.xml` необходимо задать значение **true** для следующего параметра:

```
<paylogic-message-in-response>true</paylogic-message-in-response>
```

После этого в значении параметра **message** будет возвращено описание ошибки платежа. Параметр является необязательным. По умолчанию задано значение **false**.

#### ПРИМЕР ОТВЕТА ПЦ:

```
<response>
  <result id="513" state="80" substate="0" code="3"
    message="Сумма вне допустимого диапазона" final="1"
    trans="0">
  </result>
</response>
```

### 8.3.11 ПОДТВЕРЖДЕНИЕ ОТЛОЖЕННОЙ ОПЕРАЦИИ — ТЕГ <CONFIRM>

Структура элемента <confirm> приведена в таблице 8.3.11.1.

Таблица 8.3.11.1 — Структура запроса <confirm>

Название	Тип	Формат	Обяз.	Описание
id	Атрибут	int8	Да	Идентификатор операции агента

В одном запросе может содержаться произвольное количество тегов <confirm>. Результат выполнения запроса — тег <result>.

ПРИМЕР:

```
<confirm id="12345"/>
```

### 8.3.12 ОБРАБОТКА ОПЕРАЦИЙ С РАЗРЫВОМ В ПРОВЕДЕНИИ — ТЕГ <PAYMENT-GAP>

Запрос является пакетным, т. е. в теге <request> может присутствовать более одного тега <payment-gap>.

Таблица 8.3.12.1 — Структура элемента <payment-gap>

Название	Тип	Формат	Обяз.	Описание
<b>id</b>	Атрибут	int8	Да	Идентификатор операции агента
<b>attribute</b>	Элемент	-	Нет	Используется для указания атрибутов. Может быть несколько. Структура такая же, как и в <payment> — таблица 8.3.5.2. Описание и состав атрибутов для каждого поставщика предоставляется сотрудниками Платежной системы

#### ПРИМЕР ЗАПРОСА:

```
<payment-gap id="14546"  
  <attribute name="card" value="2202000043431701"/>  
  <attribute name="ccy" value="RUB"/>  
</payment-gap>
```

В ответ на запрос возвращаются статус текущей операции и ее атрибуты.

#### ПРИМЕР ОТВЕТА:

```
<result id="14546"  
  state="60"  
  substate="0"  
  code="0"
```

```
final="1"  
trans="123456789"/>
```

### 8.3.13 ПАРАМЕТРЫ РЕЗУЛЬТАТА — ТЕГ <RESULT>

Структура элемента <result> приведена в таблице 8.3.13.1.

Таблица 8.3.13.1 — Структура элемента <result>

Название	Тип	Формат	Обяз.	Описание
<b>id</b>	Атрибут	int8	Нет	Идентификатор операции агента. Отсутствует в случае проверки номера абонента
<b>state</b>	Атрибут	int2	Нет	Статус платежа в системе, возвращается на запрос платежа и статуса
<b>substate</b>	Атрибут	int2	Нет	Субстатус платежа в системе, возвращается на запрос платежа и статуса
<b>code</b>	Атрибут	int2	Да	Код ошибки платежа или ошибка проверки реквизитов. При ответе на запрос статуса платежа помимо зарезервированных кодов (первые 50 кодов, таблица А.2.1) администратор системы может добавить в разделе «Справочники — Шлюзы — Ошибки проведения» свои коды (свыше 50 кода) и определить поведение для них

Название	Тип	Формат	Обяз.	Описание
<b>final</b>	Атрибут	int2	Нет	Признак финальности статуса запроса (1 — финальный)
<b>trans</b>	Атрибут	int4	Нет	Номер транзакции ПЦ
<b>service</b>	Атрибут	int4	Нет	Результат проверки реквизитов у поставщика при вызове advanced-запросов. Актуален в случае возврата в атрибуте <b>code</b> значения 0
<b>sum</b>	Атрибут	int4	Нет	Сумма платежа в копейках, возвращается при вызове метода сверки платежей
<b>commission</b>	Атрибут	int4	Нет	Сумма комиссии в копейках, возвращается при вызове метода расчета внешней комиссии
<b>sum_prov</b>	Атрибут	int4	Нет	Сумма транзакции в валюте поставщика
<b>server_time</b>	Атрибут	time	Нет	Время создания операции на сервере ПЦ, может использоваться для сверки с ПЦ
<b>ps_code</b>	Атрибут	int2	Нет	Код ошибки процессинга. При успешном списании средств с платёжного инструмента возвращается <b>ps_code="0"</b> . Если возвращается <b>ps_code</b> , отличный от 0, то в <b>code</b> возвращается -2 — ошибка платёжного инструмента (таблица А.2.1)

Название	Тип	Формат	Обяз.	Описание
<code>title</code>	Атрибут	<code>varchar</code>	Нет	Позволяет отдавать имя атрибута и его значение, а не наименование и значение, отображаемое на экране. Определяется параметром коннектора <code>&lt;paylogic-verify-separate-key-title&gt;true&lt;/paylogic-verify-separate-key-title&gt;</code> .

## ПРИМЕР:

```
<result id="14546"  
state="60"  
substate="0"  
code="0"  
final="1"  
trans="123456789"/>
```

### 8.3.14 ВОЗВРАЩАЕМЫЕ ПАРАМЕТРЫ ПРИ ADVANCED-ЗАПРОСЕ — ТЕГ <INPUT>

Структура элемента <input> приведена в таблице 8.3.14.1.

Таблица 8.3.14.1 — Структура элемента <input>

Название	Тип	Формат	Обяз.	Описание
<b>key</b>	Атрибут	varchar	Да	Идентификатор возвращаемого параметра
<b>keyTitle</b>	Атрибут	varchar	Да	Расшифровка названия возвращаемого параметра
<b>value</b>	Атрибут	varchar	Да	Значение возвращаемого параметра
<b>valueTitle</b>	Атрибут	varchar	Да	Расшифровка значения возвращаемого параметра

ПРИМЕР:

```
<result code="0" service="0">
  <data>
    <input key="fee" keyTitle="Комиссия"
      value="157.80" valueTitle="157.80"/>
    <input key="trans" keyTitle="Транзакция"
      value="560132" valueTitle="560132"/>
    <input key="sum" keyTitle="sum"
      value="7890.00" valueTitle="7890.00"/>
  </data>
</result>
```

## 8.4 СЛОЖНАЯ ПРОВЕРКА РЕКВИЗИТОВ ADVANCED-ЗАПРОСА

### ЗАПРОС ПРОВЕРКИ:

Запрос сложной проверки реквизитов счета абонента производится в случае онлайн-проверки введенных абонентом данных, если для данного поставщика недоступен метод <verify>.

Запрос должен включать один элемент <advanced>:

```
<request point="17243">
  <advanced function="verify"
    service="811">
    <attribute name="s-scname"
      value="ТТЪС"/>
    <attribute name="s-name"
      value="СИИМ"/>
    <attribute name="s-sname"
      value="РИБ"/>
    <attribute name="summ-out"
      value="7890"/>
    <attribute name="idl"
      value="7-968-7542312"/>
    <attribute name="b-name"
      value="ТЪБ"/>
    <attribute name="pp-code"
      value="AKIC"/>
    <attribute name="b-scname"
      value="ИМТИ"/>
    <attribute name="b-sname"
      value="РММЬ"/>
  </advanced>
</request>
```

## ОТВЕТ ПЦ:

В ответ сервер должен вернуть результат проверки номера, а также атрибуты от поставщика, в случае их наличия. Состав и возможные значения возвращаемых дополнительных атрибутов определяются особенностями проведения конкретного поставщика, и предоставляются сотрудниками платежной системы.

```
<response>
  <result code="0" service="0">
    <data>
      <input key="fee"
        keyTitle="Комиссия"
        value="157.80"
        valueTitle="157.80"/>
      <input key="trans"
        keyTitle="Транзакция"
        value="560132"
        valueTitle="560132"/>
      <input key="sum"
        keyTitle="sum"
        value="7890.00"
        valueTitle="7890.00"/>
    </data>
  </result>
</response>
```

Ответ ПЦ в отдельных случаях (определяется реализацией поставщика) возвращает вложенные объекты *nested*, содержащие списки вложенных элементов. Как следует из названия — это объекты, которые содержат внутри себя вложенные элементы. Их назначение — передача сложных структур данных для последующего разбора на стороне Партнера и обработки. Типичный пример использования таких объектов — списки, содержащие несколько значений на одну строку (табличные данные). В примере ниже в ответе возвращается атрибут **address** (обычное поле) и список услуг в объекте с именем `<services>`. Каждая строка вложенного объекта содержит по две колонки: **title** и **sum**.

Вложенность такого рода объектов может быть многоуровневая. Технически ограничений на вложенность не существует. В конечном итоге на нижнем уровне всегда содержатся объекты data со списком input-элементов.

#### ПРИМЕР ОТВЕТА С NESTED-ОБЪЕКТАМИ:

```
<data>
  <input key="address" keyTitle="address" value="street" flags="0"/>
  <nested id="services">
    <data>
      <input key="title" keyTitle="title" value="service 0"
        valueTitle="service 0" flags="0"/>
      <input key="sum" keyTitle="sum" value="1"
        valueTitle="353" flags="0"/>
    </data>
    <data>
      <input key="title" keyTitle="title" value="service 1"
        valueTitle="service 1" flags="0"/>
      <input key="sum" keyTitle="sum" value="778"
        valueTitle="778" flags="0"/>
    </data>
    <data>
      <input key="title" keyTitle="title" value="service 8"
        valueTitle="service 8" flags="0"/>
      <input key="sum" keyTitle="sum" value="6217"
        valueTitle="6217" flags="0"/>
    </data>
    <data>
      <input key="title" keyTitle="title" value="service 9"
        valueTitle="service 9" flags="0"/>
      <input key="sum" keyTitle="sum" value="6994"
        valueTitle="6994" flags="0"/>
    </data>
  </nested>
</data>
```

В следующем примере приведен список услуг со счетчиками, на каждую услугу по два счетчика (дневной и ночной) с текущими показаниями по его тарификационной зоне. Самих услуг также две.

## ПРИМЕР ОТВЕТА С NESTED-ОБЪЕКТАМИ МНОГОУРОВНЕВОЙ ВЛОЖЕННОСТИ:

```
<data>
  <input key="address" keyTitle="address" value="street" flags="0"/>
  <nested id="services">
    <data>
      <input key="title" value="Электричество" flags="0"/>
      <input key="service-id" value="6">
      <nested id="servicecounters">
        <data>
          <input key="CounterZone" value="day"
            valueTitle="day" flags="0"/>
          <input key="CounterValue" keyTitle="Current value"
            value="77000" flags="0"/>
        </data>
        <data>
          <input key="CounterZone" value="night"
            valueTitle="night" flags="0"/>
          <input key="CounterValue" keyTitle="Current value"
            value="3320" flags="0"/>
        </data>
      </nested>
    </data>
  </data>
  <data>
    <input key="title" value="Холодная вода" flags="0"/>
    <input key="service-id" value="7">
    <nested id="servicecounters">
      <data>
        <input key="CounterZone" value="day"
          valueTitle="day" flags="0"/>
        <input key="CounterValue" keyTitle="Current value"
          value="77000" flags="0"/>
      </data>
      <data>
        <input key="CounterZone" value="night"
          valueTitle="night" flags="0"/>
        <input key="CounterValue" keyTitle="Current value"
          value="3320" flags="0"/>
      </data>
    </nested>
  </data>
</nested>
</data>
```

## 8.5 ПОПЛАТЕЖНАЯ СВЕРКА

Запрос сверки предназначен для проведения сверок агента и платежной системы. Сверка выполняется за некоторый период времени и позволяет выявить расхождения в учете платежей либо их статусе по данным агента и данным платежной системы, либо убедиться в отсутствии такого расхождения. Запрос позволяет получить информацию о платежах либо агрегированно, либо поплатежно. Для формирования запроса на сверку используется тег `<reconciliation>`. Структура элемента `<reconciliation>` приведена в таблице 8.5.1.

Таблица 8.5.1 — Структура элемента `<reconciliation>`

Название	Тип	Формат	Обяз.	Описание
<b>begin</b>	Атрибут	time	Да	Дата и время начала периода
<b>end</b>	Атрибут	time	Да	Дата и время окончания периода
<b>payments</b>	Атрибут	int1	Нет	0 или 1 (нужно или нет загружать список платежей). Опционально, если отсутствует то 0
<b>offset</b>	Атрибут	int4	Нет	Начальная позиция в случае загрузки списка платежей. Опционально, если отсутствует то 0

В зависимости от параметра **payments** запрос возвращает либо не возвращает список платежей, суммарные показатели возвращаются в любом случае.

Список платежей возвращается внутри тега `<result>` в виде вложенных тегов `<payment>`. Если запрошена выгрузка платежей, то в каждом ответе может быть не

более 1000 платежей. Все необходимо получать за несколько запросов, используя одинаковые даты и смещая **offset**.

Результатом будет тег `<result>` с атрибутами в соответствии с таблицей 8.5.2.

Таблица 8.5.2 — Атрибуты тега `<result>`

Название	Тип	Формат	Обяз.	Описание
<b>code</b>	Атрибут	int2	Да	Код ответа (0 — успех, 1 — ошибка)
<b>total</b>	Атрибут	int8	Да	Общее количество платежей в ответе
<b>sum</b>	Атрибут	int8	да	Общая сумма платежей за период в копейках
<b>count</b>	Атрибут	int4	Нет	Количество платежей в результате, если идет выгрузка платежей (в запросе <b>payments="1"</b> )
<b>offset</b>	Атрибут	int4	Нет	Начальная позиция в случае загрузки списка платежей. Опционально, если отсутствует то 0

В случае выгрузки списка платежей, внутри тега `<result>` будут элементы `<payment>` для каждого платежа с атрибутами в соответствии с таблицей 8.5.3.

Таблица 8.5.3 — Атрибуты платежа

Название	Тип	Формат	Обяз.	Описание
<b>id</b>	Атрибут	Int8	Да	Номер транзакции агента
<b>date</b>	Атрибут	time	Да	Дата и время поступления платежа
<b>state</b>	Атрибут	int2	Да	Статус платежа в системе

Название	Тип	Формат	Обяз.	Описание
<b>substate</b>	Атрибут	int2	Да	Субстатус платежа в системе
<b>code</b>	Атрибут	int2	Да	Код ошибка платежа
<b>trans</b>	Атрибут	int2	Нет	Номер транзакции в ПЦ
<b>sum</b>	Атрибут	int4	Да	Сумма платежа
<b>service</b>	Атрибут	int4	Да	Код услуги
<b>final</b>	Атрибут	int2	Нет	Признак финальности статуса запроса (1 — финальный)

## ЗАПРОС СВЕРКИ:

```
<request point="12345">  
  <reconciliation begin="2007-10-12T12:00:00+0300"  
    end="2007-10-13T12:00:00+0300"  
    payments="1" offset="1000"/>  
</request>
```

## ОТВЕТ ПЦ:

```
<result code="0"  
  total="250"  
  sum="7456895407"  
  count="1250"  
  offset="1000">  
  <payment id="14547"  
    date="2007-10-12T12:00:00+0300"/sum="1000"  
    state="40"  
    service="4"  
    sum="20000"  
    substate="0"  
    trans="856354300894"  
    final="1"/>  
  ...  
  ...  
  ...  
</result>
```

## 8.6 РЕГИСТРАЦИЯ ТОЧЕК У ПОСТАВЩИКА

Запрос регистрации точек у поставщика предназначен для передачи информации поставщику о точке приема платежей. Отдельные поставщики требуют обязательной регистрации точек до начала приема платежей через них. При регистрации указываются адрес точки, с детальной информацией об адресе согласно классификатора. В качестве классификатора адресов применяется КЛАДР.

Для формирования запроса на регистрацию точки используется тег `<point>`. В одном запросе `<request>` может быть 1 или более тегов `<point>`.

Структура элемента `<point>` приведена в таблице 8.6.1.

Таблица 8.6.1 — Структура элемента `<point>`

Название	Тип	Формат	Обяз.	Описание
<b>id</b>	Атрибут	int4	Да	Код точки во внешней системе
<b>area</b>	Атрибут	varchar	Нет	Регион (область, край)
<b>city</b>	Атрибут	varchar	Нет	Город (Населенный пункт)
<b>cityType</b>	Атрибут	varchar	Нет	Тип населенного пункта (г,с,п — согласно КЛАДР)
<b>kladr</b>	Атрибут	varchar	Нет	Код КЛАДР (опционально: либо область с городом, либо код КЛАДР, второе предпочтительней)
<b>address</b>	Атрибут	varchar	Да	Адрес точки в пределах населенного пункта (полностью)

Название	Тип	Формат	Обяз.	Описание
<b>agent</b>	Атрибут	varchar	Да	Название агента точки (опционально, если нет будет браться из агента ВПС)
<b>inn</b>	Атрибут	varchar	Нет	ИНН юридического лица-владельца точки
<b>support</b>	Атрибут	varchar	Нет	Телефон поддержки агента

Результат выполнения запроса на регистрацию возвращается в виде тега `<result>` — по одному на каждый элемент `<point>` в исходном запросе. Атрибуты тега `<result>` приведены в таблице 8.6.2.

Таблица 8.6.2 — Атрибуты тега `<result>`

Название	Тип	Формат	Обяз.	Описание
<b>id</b>	Атрибут	int4	Да	Код точки во внешней системе
<b>state</b>	Атрибут	int2	Да	Код ответа: 0 — успех, 1 — ошибка формата, 2 — ошибка БД/обработки

Если необходимо изменить данные точки (изменился адрес, перемещена, и т. д.) — необходимо повторно отправить запрос на регистрацию точки с тем же `id`, но с актуальными (измененными) данными.

## ЗАПРОС РЕГИСТРАЦИИ ТОЧКИ:

```
<request point="17235">
  <point id="7442"
        kladr="22000000000000"
        city="Барнаул"
        address="Центральная ул, д.223" id="7442"/>
  <status id="125"/>
</request>
```

## ОТВЕТ ПЦ:

```
<response>
  <result id="7442" state="0"/>
</response>
```

## 8.7 ОТМЕНА ТРАНЗАКЦИЙ

Для осуществления отмены платежей используются заявки на отмену. Механизм заявок на отмену подразумевает, что при необходимости отменить платеж в системе формируется заявка на отмену операции, которая в дальнейшем обрабатывается в процессинге (автоматически по шлюзу, или вручную). Результатом обработки заявки на отмену является либо успешная обработка, либо ошибка (невозможность отмены). В случае успешной обработки в процессинге создается сторнирующая операция, отменяющая исходный платеж. Заявка на отмену получает статус «Обработана».

Информация о состоянии заявки на отмену возвращается в ответе на запрос заявки.

Для формирования заявки на отмену используется тег `<cancel>`. В одном запросе `<request>` может быть только один тег `<cancel>`.

Структура элемента `<cancel>` приведена в таблице 8.7.1.

Таблица 8.7.1 — Структура элемента `<cancel>`

Название	Тип	Формат	Обяз.	Описание
<code>id</code>	Атрибут	int8	Да	Идентификатор отменяемой транзакции
<code>person</code>	Атрибут	int4	Нет	Идентификатор пользователя, инициировавшего заявку на отмену
<code>comment</code>	Атрибут	varchar	Нет	Комментарий к заявке на отмену
<code>sum</code>	Атрибут	int4	Нет	Сумма отмены в копейках — параметр опционален, и если не

---

Название	Тип	Формат	Обяз.	Описание
				передается, то отменен будет весь платеж

Результат возвращается в виде одного тега <result>. Атрибуты тега <result> приведены в таблице 8.7.2.

Таблица 8.7.2 — Атрибуты тега <result>

Название	Тип	Формат	Обяз.	Описание
<b>id</b>	Атрибут	int8	Да	Идентификатор платежа
<b>state</b>	Атрибут	int2	Да	Статус заявки/результат ответа

Расшифровка кодов статусов заявок на отмену приведена в приложении [«А. Результат данных, возвращаемых ПЦ»](#), таблица А.4.1.

Следует учитывать, что отмена на стороне ПЦ может занимать длительное время. Для того, чтобы запросить статус заявки на отмену, ранее созданной, необходимо отправить запрос <cancel> с тем же ID операции повторно. Ответом будет тег <status> с текущим статусом обработки заявки.

#### ЗАПРОС ОТМЕНЫ ОПЕРАЦИИ:

```
<request point="17235">
  <cancel id="193452"
    person="27"
    comment="Отмена по заявлению От 12.12.2014, рег. № 711"/>
</request>
```

#### ОТВЕТ ПЦ:

```
<response>
  <result id="193452" state="1"/>
</response>
```

## 8.8 ЗАПРОС КАТАЛОГА УСЛУГ (<MENU>)

Для формирования запроса каталога услуг используется тег `<menu>`. ПЦ формирует по запросу XML-ответ со структурированным списком получателей платежей, доступных для агента. Список содержит структуру меню, которое установлено в шлюзе агента, лимиты по услугам, информацию о внешней комиссии и т.д.

Справочник содержит в себе набор «папок» меню (элемент `<folder>`), которые могут быть вложены друг в друга с неограниченным уровнем вложенности. Каждый элемент ответа `<folder>` может содержать как переменное количество вложенных папок, так и переменное количество услуг (элемент ответа `<service>`).

Запрос `<menu>` может принимать на вход опциональный параметр — например, тип сервиса или параметр вывода каталога с иерархией. Параметры не взаимоисключающие, но при выводе с фильтрацией по типу сервиса и с учетом иерархии пустые папки не выводятся. Пустыми считаются такие папки, в которых (и в подпапках которых) в итоге нет ни одного сервиса.

Структура тега `<menu>` приведена в таблице ниже.

Таблица 8.8.1 — Структура элемента запроса `<menu>`

Название параметра	Обяз.	Описание
<code>locale="&lt;en&gt;"</code>	Нет	Позволяет получить меню в нужной локали: ru, en, uk, kz, by и т. д. Если параметр не прописан, берется локаль по умолчанию
<code>include-properties="&lt;1&gt;"</code>	Нет	Позволяет получить произвольные свойства сервиса.

Название параметра	Обяз.	Описание
		<p>При значении параметра:</p> <ul style="list-style-type: none"> <li>• <b>1</b> — активирован;</li> <li>• <b>любое другое значение</b> — деактивирован</li> </ul>
<code>include-providers="&lt;1&gt;"</code>	Нет	<p>Возвращает сведения о провайдере в соответствии с текущим направлением проведения для сервиса, а также произвольные свойства провайдера и сервиса. Для работы параметра в кабинете в свойствах шлюзовой точки на вкладке «Параметры шлюзовых точек» должен быть установлен флаг <b>«Отдавать информацию о провайдере в составе меню»</b>.</p> <p>При значении параметра:</p> <ul style="list-style-type: none"> <li>• <b>1</b> — активирован;</li> <li>• <b>любое другое значение</b> — деактивирован</li> </ul>
<code>include-additionals="&lt;1&gt;"</code>	Нет	<p>Возвращает элементы, содержащие сведения о провайдере, указанные в свойствах сервиса:</p> <ul style="list-style-type: none"> <li>• <b>&lt;INN&gt;</b>, <b>&lt;INFO&gt;</b> (код причины постановки на учет, КПП) — с вкладки «Основное»;</li> <li>• <b>&lt;BIC&gt;</b>, <b>&lt;Bank&gt;</b> (название банка), <b>&lt;Pay_Acc&gt;</b> (номер расчетного счета), <b>&lt;KS&gt;</b> (номер корреспондентского счета) — с вкладки «Особенности оплаты».</li> </ul> <p>В значении параметра необходимо указать <b>1</b>. При любом другом значении сведения не вернутся. Если в значении указано <b>1</b>, но какой-то из элементов не вернулся, это означает, что его значение не задано в свойствах сервиса.</p>

Название параметра	Обяз.	Описание
<code>with-hierarchy="&lt;1&gt;"</code>	Нет	Возвращает меню с иерархией (вложенностью) папок. Если параметр выключен, то выводится плоский список, где есть только сервисы, без папок. Если в атрибуте указано: <ul style="list-style-type: none"> <li>любое значение, кроме 0 — параметр активирован;</li> <li>0 — деактивирован</li> </ul> По умолчанию параметр включен
<code>service-type="&lt;id_типа_сервиса&gt;"</code>	Нет	Фильтрует меню по типу сервиса. В атрибуте указывается id типа сервиса. Запрос типов сервисов и получение их id описан в разделе 8.9

Структура элемента ответа `<folder>` приведена в таблице 8.8.2.

Таблица 8.8.2 — Структура элемента ответа `<folder>`

Название	Тип	Формат	Обяз.	Описание
<code>id</code>	Атрибут	int4	Да	Уникальный идентификатор папки
<code>name</code>	Атрибут	varchar	Да	Название папки для отображения клиенту
<code>img</code>	Атрибут	varchar	Нет	Название иконки для папки (например, internet.png)
<code>order</code>	Атрибут	int4	Да	Порядок сортировки папки
<code>service</code>	Элемент	-	Нет	Произвольное количество вложенных элементов услуг
<code>folder</code>	Элемент	-	Нет	Произвольное количество вложенных элементов папок

Структура элемента ответа `<service>` приведена в таблице 8.8.3.

Таблица 8.8.3 — Структура элемента ответа <service>

Название	Тип	Формат	Обяз.	Описание
<b>id</b>	Атрибут	int4	Да	Уникальный идентификатор услуги, который необходимо передавать в платеже
<b>code</b>	Атрибут	int4	Да	Номер сценария услуги
<b>name</b>	Атрибут	varchar	Да	Наименование услуги, для отображения клиенту
<b>img</b>	Атрибут	varchar	Нет	Название иконки для услуги (например, mts.png)
<b>order</b>	Атрибут	int4	Да	Порядок сортировки услуги внутри папки
<b>commission</b>	Атрибут	varchar	Да	Название внешней комиссии для отображения информации клиенту (для расчета комиссии используется метод <commission>)
<b>maxsum</b>	Атрибут	int4	Да	Максимальная сумма оплаты в копейках
<b>minsum</b>	Атрибут	int4	Да	Минимальная сумма оплаты в копейках
<b>hotkey</b>	Атрибут	int2	Нет	Признак «горячей клавиши» (1 или 0). Отображение в TOP услуг. Если отсутствует значит 0
<b>hotkeyPosition</b>	Атрибут	int2	Нет	Позиция (номер) горячей клавиши. Добавлен в версии 5.4.3.
<b>providerName</b>	Атрибут	varchar	Нет	Юридическое лицо поставщика услуг
<b>providerInn</b>	Атрибут	varchar	Нет	ИНН поставщика услуг
<b>providerSupport</b>	Атрибут	varchar	Нет	Телефон поддержки поставщика

Название	Тип	Формат	Обяз.	Описание
				услуг
<b>handlerType</b>	Атрибут	varchar	Да	Тип обработчика для сервиса
<b>service-info</b>	Атрибут	varchar	Нет	Информация о сервисе (опциональная)
<b>receipt</b>	Атрибут	varchar	Да	Имя шаблона чека, используемого для сервиса. Если шаблона чека нет, вернется пустая строка: <code>receipt=""</code>
<commission- details>	Элемент	-	Да	Элемент, описывающий структуру применяемой комиссии в детальном и пригодном для применения в расчете виде (таблица 8.8.4)
<sources>	Элемент	-	Да	Элемент, описывающий перечень доступных платежных инструментов. Содержит вложенные элементы <source> (таблица 8.8.6). Может не содержать ни одного элемента
<regions>	Элемент	-	Да	Элемент, описывающий перечень регионов для сервиса. Содержит вложенные элементы <region> (таблица 8.8.7). Может не содержать ни одного элемента
<cities>	Элемент	-	Да	Элемент, описывающий перечень городов в которых доступен сервис. Содержит вложенные элементы <city> (таблица 8.8.8). Может не содержать ни одного элемента
<properties>	Элемент	-	Да	Элемент, описывающий перечень

Название	Тип	Формат	Обяз.	Описание
				произвольных атрибутов, ассоциированных с сервисом (список произвольных свойств). Содержит вложенные элементы <code>&lt;property&gt;</code> (таблица 8.8.9). Может не содержать ни одного элемента
<code>&lt;subscription&gt;</code>	Элемент	-	Нет	Элемент содержит информацию о некоторых флагах, установленных для сервиса в кабинете процессинга. Подробнее в таблице 8.8.10

Таблица 8.8.4 — Структура элемента `<commission-details>`

Название	Тип	Формат	Обяз.	Описание
<code>mode</code>	Атрибут	varchar15	Да	Способ расчета комиссии. значения: <ul style="list-style-type: none"> <li>DIRECT — прямой расчет, комиссия считается от суммы транзакции (сумы к зачислению);</li> <li>REVERSE — выкрутка</li> </ul>
<code>percent</code>	Атрибут	numeric (15,2)	Да	Процент комиссии
<code>fixed</code>	Атрибут	numeric (5,2)	Нет	Фиксированное значение комиссии
<code>minamount</code>	Атрибут	numeric (15,2)	Нет	Минимальное значение взимаемой комиссии (не менее)
<code>maxamount</code>	Атрибут	numeric (15,2)	Нет	максимальное значение взимаемой комиссии (не более)
<code>&lt;commission-steps&gt;</code>	Элемент	Ступени комиссии	Да	Содержит вложенные элементы <code>&lt;commission-step&gt;</code> (таблица 8.8.5).

Название	Тип	Формат	Обяз.	Описание
				Может не содержать ни одного элемента (если ступеней комиссии нет)

Таблица 8.8.5 – Структура элемента &lt;commission-step&gt;

Название	Тип	Формат	Обяз.	Описание
<b>source</b>	Атрибут	String	Нет	Код платежного инструмента, для которого применяется комиссия. Если не указан — считается, что платеж наличный. Список кодов, доступных в системе, приведен на странице «Справочники — Финансы — Источники оплаты»
<b>amount-from</b>	Атрибут	numeric (15,2)	Да	Пороговое значение суммы платежа, от которой (включительно) применяется ставка
<b>percent</b>	Атрибут	numeric (5,2)	Да	Значение комиссии: процентная ставка для ступени
<b>fixed</b>	Атрибут	numeric (15,2)	Нет	Значение комиссии, фиксированное значение для ступени
<b>minamount</b>	Атрибут	numeric (15,2)	Нет	Минимальное значение взимаемой комиссии (не менее) для ступени
<b>maxamount</b>	Атрибут	numeric (15,2)	Нет	максимальное значение взимаемой комиссии (не более) для ступени

Таблица 8.8.6 — Структура элемента ответа `<source>`

Название	Тип	Формат	Обяз.	Описание
<code>code</code>	Атрибут	<code>varchar20</code>	Да	Код платежного инструмента
<code>name</code>	Атрибут	<code>varchar100</code>	Да	Название платежного инструмента

Таблица 8.8.7 — Структура элемента `<region>`

Название	Тип	Формат	Обяз.	Описание
<code>region-id</code>	Атрибут	<code>int2</code>	Да	Код региона
<code>name</code>	Атрибут	<code>varchar(100)</code>	Да	Наименование региона
<code>kladr</code>	Атрибут	<code>varchar(30)</code>	Да	Код КЛАДР для региона

Таблица 8.8.8 — Структура элемента `<city>`

Название	Тип	Формат	Обяз.	Описание
<code>city-id</code>	Атрибут	<code>int2</code>	Да	Код города
<code>name</code>	Атрибут	<code>varchar(100)</code>	Да	Наименование города
<code>kladr</code>	Атрибут	<code>varchar(30)</code>	Да	Код КЛАДР для города
<code>tz</code>	Атрибут	<code>varchar(30)</code>	Да	Временная зона для города

Таблица 8.8.9 — Структура элемента `<property>`

Название	Тип	Формат	Обяз.	Описание
<code>code</code>	Атрибут	<code>varchar(30)</code>	Да	Код атрибута (свойства)
<code>name</code>	Атрибут	<code>varchar(100)</code>	Да	Наименование свойства
<code>value</code>	Атрибут	<code>varchar(1000)</code>	Да	Значение свойства

Таблица 8.8.10 – Структура элемента &lt;subscription&gt;

Название	Тип	Формат	Обяз.	Описание
<capability>	Элемент	-	Да	Содержит атрибуты, обозначающие наличие различных флагов по сервису. Подробнее в таблице 8.8.11

Таблица 8.8.11 – Структура элемента &lt;capability&gt;

Название	Тип	Формат	Обяз.	Описание
<b>name</b>	Атрибут	varchar(30)	Да	Возможные значения: 1) TEMPLATES — соответствует наличию флага «Может быть шаблоном» 2) AUTOPAYMENTS — соответствует наличию флага «Может быть автоплатежом» 3) SUBSCRIPTIONS — соответствует наличию флага «Поддерживает автоматическое выставление счетов»

Тип обработчика для сервиса (handlerType) может принимать значения:

1. «Mobile» — форма или сценарий не применяются, предполагается ввод единственного параметра: номера телефона.
2. «UniversalProvider» — для сервиса предполагается работа с формой.
3. «AdvancedProvider» — для сервиса предполагается работа со сценарием.

#### ЗАПРОС КАТАЛОГА УСЛУГ:

```
<request point="17235">
  <menu/>
</request>
```

#### ОТВЕТ ПЦ:

```
<response>
  <result code="0">
    <folder id="1" name="Сотовая связь" img="cell.png" order="0">
      <folder id="3" name="Международные операторы" order="10">
        <service id="103" code="1" name="Билайн Таджикистан"
          img="beeline_tj.png" order="1"
          commission="1% минимум 10 рублей"
          maxsum="1500000" minsum="100" hotkey="0">
          <commission-details mode="DIRECT" percent="10.00"
            fixed="0.00" minamount="1.35"
            maxamount="50">
            <commission-steps>
              <commission-step amount-from="100.00"
                percent="8.00" fixed="1.00"
                minamount="1.35" maxamount="50" />
              <commission-step amount-from="1000.00"
                percent="5.00"/>
              <commission-step amount-from="2000.00"
                percent="3.00"/>
              <commission-step source="BANKCARD" amount-
                from="0.01"
                percent="3.00" fixed="1.00"
                minamount="1.35" maxamount="50" />
              <commission-step source="BANKCARD" amount-
                from="1000.00"
                percent="1.00"/>
            </commission-steps>
          </commission-details>
        </service>
      </folder>
    </folder>
  </result>
</response>
```

```
<commission-step      source="BANKCARD" amount-  
                        from="2000.00"  
                        percent="0.01"/>  
</commission-steps>  
</commission-details>  
<regions>  
  <region id="1" name="Алтай"   kladr="1616161600" />  
  <region id="2" name="Бурятия" kladr="2222000000" />  
  <region id="3" name="Москва"  kladr="1000000000" />  
</regions>  
<cities>  
  <city id="1" name="Барнаул"  kladr="16161616161"  
    tz="Etc/Gmt+6"/>  
  <city id="1" name="Новоалтайск" kladr="16161616161"  
    tz="Etc/Gmt+6"/>  
</cities>  
<properties>  
  <property      code="acc_pu"  
    name="Счет учета какого то расчета"  
    value="40303345349937"/>  
</properties>  
</service>  
</folder>  
<service id="100" code="2" name="МТС" img="mts.png" order="1"  
  commission="0%" maxsum="1500000" minsum="100"  
  hotkey="1" hotkeyPosition="1" providerName="ЗАО МТС"  
  providerInn="234242342" providerSupport="0890">  
<commission-details  mode="DIRECT" percent="7.00"  
  fixed="0.00" >  
</commission-details>  
<sources/>  
<subscription>  
  <capability name="SUBSCRIPTIONS"/>  
  <capability name="AUTOPAYMENTS"/>  
  <capability name="TEMPLATES"/>  
</subscription>  
</service>  
<service id="101" code="3" name="Билайн" img="beeline.png"  
  order="2" commission="0%" maxsum="1500000" minsum="100"  
  hotkey="1">  
<commission-details  mode="DIRECT" percent="7.00"  
  fixed="0.00" >  
</commission-details>  
  <property      code="acc_pu"  
    name="Счет учета какого-то расчета"  
    value="40303345349937"/>
```

```
        </properties>  
    </service>  
  </folder>  
</result>  
</response>
```

Ниже приведен пример запроса каталога услуг с дополнительными параметрами:

**ЗАПРОС КАТАЛОГА УСЛУГ С ДОПОЛНИТЕЛЬНЫМИ ПАРАМЕТРАМИ:**

```
<request point="1145">  
  <menu locale="en" include-properties="1" include-providers="1"/>  
</request>
```

### 8.8.1 ЗАПРОС ВЕРСИИ КАТАЛОГА УСЛУГ

Для запроса версии каталога услуг используется тег `<menu-version>`. Структура тега `<menu-version>` повторяет структуру тега `<menu>`, которая приведена в разделе 8.8.

В ответе возвращается высшая версия каталога услуг.

#### ЗАПРОС ПЦ:

```
<request point="код точки">  
  <menu-version/>  
</request>
```

#### ОТВЕТ ПЦ:

```
<response>  
  <result code="0" menu-version="номер версии">  
  </result>  
</response>
```

## 8.9 ЗАПРОС КАТЕГОРИЙ (ТИПОВ) СЕРВИСОВ

Для запроса категорий (типов) сервисов используется тег `<service-categories>`. В ответе возвращается идентификатор типа сервиса (**id**), название типа (**name**) и приоритет типа сервиса при выводе кнопок на терминале (**order**). Чем меньше значение, тем выше приоритет.

### ЗАПРОС КАТЕГОРИЙ СЕРВИСОВ:

```
<request point="1878">
  <service-categories/>
</request>
```

### ОТВЕТ ПЦ:

```
<response>
  <result code="0">
    <service-categories>
      <service-category id="1" name="Сотовая связь" order="1"/>
      <service-category id="2" name="Интернет" order="2"/>
      <service-category id="3" name="Электронные деньги" order="9"/>
      <service-category id="4" name="ЖКХ" order="0"/>
      <service-category id="5" name="Телевидение" order="3"/>
      <service-category id="6" name="Разное" order="8"/>
      <service-category id="8" name="Госпошлины, штрафы" order="6"/>
      <service-category id="9" name="Погашение кредитов" order="7"/>
      <service-category id="10" name="Фиксированная связь" order="5"/>
      <service-category id="11" name="#Скрытая" order="10"/>
      <service-category id="16" name="aaa" order="0"/>
      <service-category id="21" name="bbb" order="2"/>
      <service-category id="22" name="ccc" order="1"/>
      <service-category id="25" name="ИнэтСетьСвязь" order="1"/>
      <service-category id="26" name="Кошелек" order="0"/>
      <service-category id="27" name="Сотовая связь +" order="0"/>
      <service-category id="28" name="TV++" order="95"/>
      <service-category id="29" name="HD" order="0"/>
      <service-category id="30" name="цифровое" order="0"/>
      <service-category id="31" name="Платежи+" order="0"/>
      <service-category id="32" name="Киевстар " order="0"/>
```

```
<service-category id="33" name="New tipe" order="0"/>
<service-category id="34" name="16/07/2014" order="1"/>
<service-category id="35" name="aPlata []*?:/\ 1" order="0"/>
<service-category id="36" name="Выпуск карт" order="0"/>
<service-category id="37" name="деньги электронные" order="0"/>
<service-category id="38" name="электроденьги" order="0"/>
<service-category id="39" name="TV+" order="95"/>
<service-category id="40" name="Услуги" order="0"/>
<service-category id="41" name="Ksu Мерчанты" order="0"/>
<service-category id="42" name="Тип сервиса hisob" order="0"/>
<service-category id="76" name="Новые сервисы" order="0"/>
<service-category id="84" name="Банковские операции" order="0"/>
<service-category id="86" name="ntcn" order="0"/>
<service-category id="87" name="тест" order="0"/>
<service-category id="90" name="Xsolla" order="0"/>
<service-category id="92" name="#Скрытая сотовая связь"
order="0"/>
<service-category id="94" name="123" order="0"/>
<service-category id="96" name="1231" order="0"/>
<service-category id="97" name="Почтомат" order="0"/>
<service-category id="98" name="Фитнес" order="0"/>
<service-category id="99" name="Сохранение средств" order="0"/>
<service-category id="100" name="ЕРИП" order="0"/>
<service-category id="101" name="Банковские услуги" order="0"/>
<service-category id="102" name="ИП Цептер" order="0"/>
<service-category id="103" name="Социальное обслуживание,
Здравоохранение" order="0"/>
<service-category id="105" name="name" order="0"/>
<service-category id="107" name="name1" order="0"/>
<service-category id="108" name="RMA Payout Cash" order="0"/>
<service-category id="109" name="Тестирование" order="0"/>
<service-category id="110" name="Fiscal services" order="0"/>
<service-category id="111" name="Тип сервиса МИП" order="0"/>
<service-category id="112" name="Мобильная связь" order="0"/>
<service-category id="113" name="Тестирование комиссии РМА"
order="0"/>
<service-category id="114" name="Тесты комиссии." order="0"/>
<service-category id="115" name="Донат" order="95"/>
</service-categories>
</result>
</response>
```

## 8.10 ЗАПРОС СПИСКА ФОРМ

ПЦ по запросу формирует XML-ответ, включающий формы для услуг, доступных для агента. Список содержит переменное количество форм, подробное описание которых приведено в документе [«Формы оплаты для универсального модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство администратора»](#).

Запрос списка услуг должен содержать один элемент `<forms>`, включающий атрибуты **offset** (смещение относительно начала списка) и **limit** (количество элементов в результате, максимальное значение 100).

Результирующий элемент `<result>`, помимо обязательного элемента **code** (если не 0, то ошибка), включает атрибуты **count** (количество форм в текущем результате) и **total** (общее количество сценариев).

### ЗАПРОС СПИСКА ФОРМ:

```
<request point="17235">
  <forms offset="100" limit="10"/>
</request>
```

### ИЛИ

```
<request point="17235">
  <forms service="10"/>
</request>
```

### ОТВЕТ ПЦ:

```
<response>
  <result code="0" count="10" total="600">
    <form service-id="35">
      <text-field name="id1"
        keyboard="Digital">
```

```
        max-len="12"
        message="Введите номер лицевого счета"
        title="Номер лицевого счета">
    <verify>
        <rule regex="^\d{6,12}$"/>
    </verify>
    <split default="***-***">
        <rule regex="^[0-9]{7,12}$" value="***-***-***-***"/>
    </split>
</text-field>
</form>
<form service-id="312">
    <text-field name="id1"
        keyboard="any:[ru,symb]:upper:true"
        max-len="11"
        example="66ПК12345"
        message="Введите номер договора"
        title="Номер протокола">
        <verify>
            <rule regex="^[0-9]{2}[А-Яа-яёЁ0-9]{2,3}[0-9]{6,8}$"/>
        </verify>
    </text-field>
    <text-field name="accountdate" keyboard="Digital"
        max-len="8"
        example="01042010"
        message="Введите дату договора: ДД.ММ.ГГГГ"
        title="Дата">
        <verify>
            <rule regex="^(0[1-9]|[12][\d]|3[01])\.(0[1-9]|1[0-2])\.(19|20)[\d]{2}$"/>
        </verify>
    </text-field>
    <numeric-field name="id2"
        show-on-confirm="true"
        message="Введите сумму"
        title="Сумма"
        allow-decimal="false"
        min="50"
        max="14999"
        change-sumpurchase="true"/>
</form>
...
</result>
</response>
```

## 8.11 ЗАПРОС СЦЕНАРИЕВ

Сценарии усовершенствованного обработчика запрашиваются по каждому отдельному сервису. Возможности запросить все сценарии одним запросом не предусмотрено. ПЦ возвращает ответ, содержащий XML-код сценария по запрошенному сервису. Подробное описание синтаксиса сценариев приведено в документе [«Сценарии оплаты для усовершенствованного модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).

Запрос списка услуг должен содержать один элемент `<scenario>`, содержащий атрибут `service=<идентификатор сервиса>`. Для того, чтобы учесть возможность использования одного сценария в нескольких сервисах — следует передавать значение не идентификатора, а кода сервиса.

Результирующий элемент `<result>`, помимо обязательного элемента `code` (если не 0, то ошибка), включает исходный код сценария.

### ЗАПРОС СЦЕНАРИЯ:

```
<request point="243">
  <scenario service="252"/>
</request>
```

### ОТВЕТ ПЦ:

```
<response>
  <result code="0">
    <scenario service-id="252">
      ...
    </scenario>
  </result>
</response>
```

Для получения локализованного сценария в запросе необходимо прописать параметр `locale="<код_языка>"`.

#### ЗАПРОС СЦЕНАРИЯ:

```
<request point="243">  
  <scenario service="252" locale="kg"/>  
</request>
```

В результате осуществится поиск файла с указанной локалью в названии, например: 252\_kg.xml. Если нужная локаль не найдена, вернется файл сценария по умолчанию, без указания локализации в названии.

## 8.12 ЗАПРОС ПАРАМЕТРОВ КОМИССИИ ПРОВАЙДЕРА

Процессинг «Pay-logic» позволяет оперировать при приеме платежа комиссией, определяемой на стороне платежного провайдера. Такая комиссия представляет собой верхнюю (взимаемую с плательщика) комиссию, значение которой устанавливает провайдер (или реализация в платежном шлюзе в соответствии с техническими условиями и требованиями провайдера). Для работы с такой комиссией Партнер должен применять следующий алгоритм:

1. Запросить значение комиссии провайдера до выполнения операции путем выполнения advanced-запроса по сервису.
2. Использовать значение комиссии, полученной в ответ, при приеме платежа от клиента.
3. Передать объект комиссии в операции наряду с атрибутами.

Запрос на транспортном уровне представлен advanced-запросом, описанным в пункте 8.3.9 — параметры сложной проверки реквизитов. Конкретный запрос определяется платежным сценарием по сервису поставщика. Как правило, это advanced-запрос задолженности или запрос проверки реквизитов. Ответ на такой запрос для поставщика, который устанавливает такую комиссию, включает в себя объект `<provider-fee>`.

### ПРИМЕР ОТВЕТА С ИНФОРМАЦИЕЙ О КОМИССИИ:

```
<result code="0" service="0">
  <data>
    <provider-fee fee="10" name="10%" fee-percent="10.0"/>
    <input key="trans" keyTitle="Транзакция" value="560132"/>
    <input key="sum" keyTitle="sum" value="7890.00"/>
  </data>
</result>
```

В указанном примере один из вложенных объектов тега `<data>` — это информация о комиссии провайдера (объект `<provider-fee>`). Структура элемента `<provider-fee>` приведена в таблице 8.12.1.

Таблица 8.12.1 — Структура элемента `<provider-fee>`

Название	Тип	Формат	Обяз.	Описание
<b>fee</b>	Атрибут	int4	Да	Значение комиссии в копейках
<b>name</b>	Атрибут	varchar	Да	Наименование комиссии (сгенерированное для отображения клиенту)
<b>fee-percent</b>	Атрибут	numeric(3,2)	Нет	Процентная ставка, при условии что значение комиссии выражаемо простым процентом

При формировании запроса по такому сервису необходимо в параметры запроса включать объект комиссии провайдера.

#### ПРИМЕР ЗАПРОСА НА ПЛАТЕЖ:

```
<payment id="14546" sum="1000" check="17235"  
  service="2" account="12345" date="2007-10-12T12:00:00+0300">  
  
  <attribute name="email" value="info@rol.ru"/>  
  
  <provider-fee fee="10" name="10%" percent="10.0" />  
</payment>
```

Объект комиссии следует передавать без изменения. Сервер контролирует переданные данные, и при передаче сумм операции, не соответствующих комиссии, или комиссии, измененной по отношению к той, что вернул провайдер (равно как и при ее не передаче) проведение платежа по такому провайдеру завершится с ошибкой.

---

## 8.13 ОБРАБОТКА НЕПРАВИЛЬНЫХ ПАКЕТОВ

Если по каким-то причинам xml-пакет не может быть разобран, в заголовке пакета указаны неверные данные и т.д. — ПЦ возвращает результат, корневым элементом которого является элемент `<error>`:

```
<error>Package error</error>
```

Или:

```
<error>Signature verify error</error>
```

Или:

```
<error>Database error</error>
```

## 8.14 БЛОКИРОВКА ЗАПРОСОВ ОТ ПАРТНЕРА ПРИ НЕДОСТАТОЧНОМ БАЛАНСЕ

При снижении доступного остатка средств ниже порога блокировки (лимит блокировки задается в свойствах агента) — поведение шлюзовой точки определяется параметрами:

1. Системный параметр «Ограничивать операции шлюзовым точкам при достижении балансом лимита блокировки».
2. Свойство «Ограничивать операции при достижении балансом лимита блокировки» на странице редактирования шлюзовых точек.
3. Отображается администратору и пользователем с правом «Точки — Ограничивать операции при достижении балансом лимита блокировки».
4. При создании шлюзовой точки по умолчанию подставляется значение из системных параметров.

## 8.15 БЛОКИРОВКА ПОВТОРНЫХ ВХОДЯЩИХ ЗАПРОСОВ

Избыточные одинаковые запросы создают нагрузку на процессинг, которую возможно ограничить с помощью блокировки повторных входящих запросов.

Для включения блокировки входящих запросов:

1. В конфигурационном файле коннектора `/srv/glassfish3/config.xml` укажите параметр `<duplication-check>...</duplication-check>` со списком запросов, для которых необходимо выполнять проверку и блокировку повторных входящих запросов. В текущей версии поддерживается проверка повторов для запросов, приведенных в таблице 8.15.1. В параметре необходимо указывать не имя запроса, а код из второго столбца таблицы 8.15.1.

Таблица 8.15.1 — Список поддерживаемых запросов с указанием кода для параметра `<duplication-check>`

Название запроса	Код для параметра <code>&lt;duplication-check&gt;</code>
Запрос каталога услуг <code>&lt;menu&gt;</code>	PL_MENU
Запрос справочника платежных инструментов <code>&lt;tools&gt;</code>	PL_TOOLS
Запрос типов сервисов <code>&lt;service-categories&gt;</code>	PL_SERVICE_CATEGORIES
Запрос списка форм <code>&lt;forms&gt;</code>	PL_FORMS
Запрос сценария <code>&lt;scenario&gt;</code>	PL_SCENARIO
Запрос расчета внешней комиссии <code>&lt;commission&gt;</code>	PL_COMMISSION
Запрос баланса <code>&lt;balance&gt;</code>	PL_BALANCE

---

Запрос статуса платежа <status>	PL_STATUS
Проверка номера абонента <verify>	PL_VERIFY
Сложный запрос проверки реквизитов и вызова произвольных функций по поставщику <advanced>	PL_ADVANCED

Если параметр в конфигурационном файле отсутствует, проверка на дубли не выполняется. Недопустимые значения игнорируются. Пример параметра:

```
<duplication-check>PL_STATUS,PL_VERIFY,PL_ADVANCED,PL_FORMS,PL_BALANCE</duplication-check>
```

Список распознанных значений пишется в лог при старте коннектора — пример:

```
15:33:38,145 INFO ExternalInitializer external2:init:149 -  
duplicationCheckRequestTypes: [PL_FORMS, PL_BALANCE, PL_STATUS,  
PL_VERIFY, PL_ADVANCED]
```

2. В параметрах шлюзовой точки задайте значение параметра «Таймаут ограничения повторных запросов, мсек». Возможные значения от 0 до 300000 миллисекунд (до 5 минут). Если задано значение 0 или не указано, то повторные запросы не учитываются и не блокируются.

Входящие запросы будут обрабатываться по следующему алгоритму:

1. Если не задан параметр в конфигурационном файле, то все запросы обрабатываются.
2. Если для точки не задан таймаут, то все запросы обрабатываются.
3. Если для точки задан таймаут, то:
  - 1) Для каждого входящего запроса рассчитывается хеш.

2) Для этой точки и сервиса ищется запрос с таким же хешем в течение заданного таймаута:

а) если время последнего сохраненного запроса и время текущего запроса отличаются менее, чем заданный для точки таймаут — запрос считается повторных, сервер возвращает ошибку: `<error>Duplicate request</error>`;

б) если время последнего сохраненного запроса и время текущего запроса отличаются более, чем заданный для точки таймаут — то сохраняется время текущего запроса;

с) если запрос такой не найден — то для точки и сервиса сохраняются его хеш и текущее время.

## 8.16 ОТПРАВКА CALLBACK ПО ФАКТУ ФИНАЛИЗАЦИИ СТАТУСА ПЛАТЕЖА

### 8.16.1 НАЗНАЧЕНИЕ

Позволяет отправить во внешнюю систему информацию о том, что операция в ПЦ Paylogic перешла в финальный статус («Успех» или «Ошибка»). После перехода операции в конечный статус обработчик помещает операцию в очередь, из которой постобработчик отправляет callback во внешнюю систему.

### 8.16.2 НАСТРОЙКА

Настройка шлюза:

1. Добавьте **jar** шлюза в директорию **ext**.
2. В конфигурационный файл шлюза `/paylogic/gates/config.xml` в секцию **modules** добавьте параметры обработчика и постобработчика. Может быть настроено несколько провайдеров, по которым будет отправляться callback, поэтому секций **modules** может быть несколько.
3. Измените параметры в соответствии с вашими настройками.
4. Для запуска шлюза выполните скрипты, которые создают в БД таблички для очереди. Скрипты передаются вместе со шлюзом.
5. Перезапустите шлюзы.

**Пример настройки обработчика:**

```
<startup-run  
class="ru.softlogic.processing.gates.callback.UniversalCallbackProcessor">  
  <param name="provider-id" value="1"/>  
  <param name="protocol" value="http"/>  
  <param name="host" value="host"/>  
  <param name="port" value="80"/>  
  <param name="url" value="/url"/>  
</startup-run>
```

**Пример настройки постобработчика:**

```
<module  
class="ru.softlogic.processing.gates.callback.UniversalCallbackPostprocessor">  
  <param name="provider-id" value="1"/>  
</module>
```

Параметры постобработчика:

1. **provider-id** — провайдер, по которому будут собирать платежи и отправляться в очередь.

Параметры обработчика приведены в таблице 8.16.2.1.

Таблица 8.16.2.1 — Параметры обработчика

Параметр	Обяз.	Описание
<b>provider-id</b>	Да	Провайдер по которому будет происходить отправка
<b>protocol</b>	Да	Протокол (http/https)
<b>host</b>	Да	Хост
<b>port</b>	Да	Порт
<b>url</b>	Да	Адрес доставки

### 8.16.3 ФОРМАТ CALLBACK

Callback предоставляется в теге `<payment>`:

```
<payment
  id="383942"
  process_time="2021-11-19T17:32:11+0700"
  server_time="2021-11-19T17:32:11+0700"
  state="60"
  substate="0"
  trans="14918251"
  sum_outcome="10000">
  <attribute name="processing-provider-full-name"
    value="Эмулятор платёжного провайдера"/>
</payment>
```

Структура элемента `<payment>` приведена в таблице 8.16.3.1.

Таблица 8.16.3.1 — Структура элемента `<payment>`

Параметр	Обяз.	Описание
<code>id</code>	Да	Идентификатор операции агента
<code>process_time</code>	Да	Время обработки операции в системе
<code>server_time</code>	Да	Время обработки операции на сервере
<code>state</code>	Да	Статус платежа в системе
<code>substate</code>	Да	Субстатус платежа в системе
<code>trans</code>	Да	Номер транзакции ПЦ
<code>sum_outcome</code>	Да	Сумма платежа
<code>processing-provider-full-name</code>	Да	Полное название провайдера в системе

## 9 МОДУЛЬ УПРАВЛЕНИЯ ПОДПИСКАМИ

### 9.1 НАЗНАЧЕНИЕ МОДУЛЯ УПРАВЛЕНИЯ ПОДПИСКАМИ

Модуль по управлению подписками предназначен для автоматического получения по клиентам агентов (партнеров процессинга) данных по задолженностям, штрафам, другим начислениям, а также управлением услугами автоплатежей по порогу баланса.

Данный модуль не активирован в ПЦ по умолчанию, и для возможности его использования необходимо уточнить информацию у администратора процессинга.

Подключение, изменение параметров, отключение услуги подписки выполняется клиентом в точках обслуживания Партнеров (Банкоматы, Терминалы самообслуживания, Каналы ДБО). Услуга автоплатежа доступна для абонентов сотовых операторов, список которых предоставляется Партнеру отдельно. Остальные услуги для получения начислений, например штрафы ГИБДД, предоставляются партнеру дополнительно. Изменение параметров, отключение услуги подписки выполняются клиентом у того же Партнера, у которого он подключил услугу.

## 9.2 СОВЕРШЕНИЕ АВТОПЛАТЕЖЕЙ И ПОЛУЧЕНИЕ НАЧИСЛЕНИЙ

ПЦ контролирует баланс лицевого счета абонента и при его снижении до уровня порога пополнения уведомляет Партнера о необходимости автоплатежа. Партнер проверяет достаточность средств на счете клиента, а также выполнение иных условий, предусмотренных установленным Партнером правилам по предоставлению услуги (например, контроль лимитов операций по счету). Если предварительные проверки не позволяют Партнеру совершить автоплатеж, он уведомляет об этом ПЦ, а ПЦ, в свою очередь, информирует абонента о невозможности совершения автоплатежа.

Остальные виды подписок обрабатываются подобным образом, за тем лишь исключением, что ПЦ периодически опрашивает поставщиков услуг на предмет наличия штрафов/задолженностей/платежей.

### 9.3 СУЩНОСТИ ПОДПИСКИ

Объекты сущности подписки хранятся в БД ПЦ.

**Клиент** — объект, который подписан на сбор начислений по одной или нескольким услугам. Подписчик характеризуется:

1. Идентификатор клиента во внешней системе (например, номер телефона).

**Подписка** — объект, характеризующий подписку клиента на получение счетов по какой-либо услуге. Подписка характеризуется:

1. Уникальный идентификатор подписки.
2. Наименование подписки. Понятное наименование, задается клиентом или внешней системой.
3. Идентификатор клиента.
4. Идентификатор сервиса.
5. Дата оформления подписки.
6. Дата последнего получения начисления по подписке.
7. Статус подписки. Подписки, которые в данный момент неактивны, но ранее по ним были получены начисления — имеют статус архивных, не удаляются.
8. Атрибуты подписки. Набор атрибутов, каждый из которых определяется в зависимости от конкретной услуги, на которую оформлена подписка. Например, номер телефона, порог баланса и сумма автоплатежа.

**Атрибут подписки** — элемент, характеризующийся кодом, значением и опционально — титулом (понятным названием). Вводом атрибутов занимается система Партнера.

---

Один клиент может иметь более одной подписки на один и тот же сервис — с разными атрибутами и наименованием.

**Начисление** — полученный от поставщика услуги выставленный счет по услуге либо счет по срабатыванию триггера по автоплатежу. Каждое начисление — это элемент, описывающий счет (при наличии периода или идентификатора начисления) или задолженность. Начисление характеризуется:

1. Уникальный идентификатор начисления.
2. Идентификатор сервиса.
3. Идентификатор подписки.
4. Идентификатор клиента (соответствует ИД подписки).
5. Идентификатор начисления (период, номер счета в системе провайдера).
6. Дополнительные атрибуты. **Атрибут счета** — элемент, характеризующийся кодом и значением. Обработкой атрибутов занимается приложение внешней системы.
7. Дата выставления начисления.
8. Статус начисления.

## 9.4 ТИПЫ ЗАПРОСОВ

Корневым элементом запроса к серверу процессинга является элемент `request`. Корневым элементом ответа является элемент `response`. Запрос может включать один из следующих элементов:

1. Проверка параметров подписки на допустимость — `<verify_subscriber>`.
2. Подписка клиента на получение начислений — `<subscription>`.
3. Отключение клиента от получения начислений — `<unsubscription>`.
4. Изменение параметров подписки — `<change_subscription>`.
5. Получение текущих подписок клиента — `<get_subscriptions>`.
6. Получение списка начислений по абонентам — `<get_notifications>`.
7. Результат списания со счета абонента — `<process_notification>`.
8. Получение результата проведения платежа по начислению — `<get_result>`.

Ответ процессинга может включать следующие элементы:

1. Результат операции (`<result>`), который в свою очередь, в зависимости типа запроса и результата может содержать один или несколько вложенных элементов:
  - 1) Начисления (`<notification>`).
  - 2) Элемент подписки (`<subscription_item>`).

## 9.5 ПОСЛЕДОВАТЕЛЬНОСТЬ ВЗАИМОДЕЙСТВИЙ

После обращения клиента к партнеру и регистрации в системе партнера, клиент однозначно идентифицирован партнером. Партнер использует уникальный идентификатор клиента в своей системе (например, номер телефона, ID клиента) для идентификации подписчика в системе подписки.

Для оформления подписки партнер запрашивает согласно имеющейся на его стороне Форме, атрибуты (реквизиты) для осуществления подписки и выполняет запрос проверки параметров будущей подписки на допустимость — `<verify_subscriber>`. Результатом выполнения запроса является код ответа (либо успех, либо неудача), в случае успеха — переход к оформлению подписки.

**Для оформления подписки** Партнер отправляет в ПЦ запрос на подключение клиента на услугу получения начислений, используя запрос `<subscription>` и атрибуты согласно полученным либо введенным ранее. Атрибутный набор передается без каких-либо изменений.

**Для получения списка начислений требующих оплаты**, Партнер с заданной периодичностью, опрашивает сервер ПЦ, отправляя запрос `<get_notifications>`. Либо при создании подписки партнер указывает, что он хочет получать оповещения о начислениях на свой Web-сервер.

При получении не пустого списка Партнер пытается списать со счета абонента средства для оплаты начисления, предварительно, при необходимости запросив подтверждение у клиента. По результатам списания Партнер отправляет на сервер запрос `<process_notification>` с результатом списания.

Во всех случаях, Партнер для каждого начисления располагает о нем всей необходимой информацией, включая уникальный идентификатор начисления в системе ПЦ.

В случае успешного списания, Партнер по номеру начисления может отследить его проведение с помощью запроса `<get_result>`.

**Для прекращения подписки** Партнер вызывает метод `<unsubscribe>`, передавая в качестве параметра идентификатор подписчика, идентификатор сервиса и идентификатор подписки.

## 9.6 ТИП РЕЗУЛЬТАТА

Ответы сервера ПЦ генерируются в xml-формате, и содержат обязательные атрибуты **code** и **message**, в которых передаются код ответа и комментарий соответственно. Успешным ответом считается ответ с кодом 0. Коды ошибок приведены в приложении [«А. Результат данных, возвращаемых ПЦ»](#) (Таблица А.5.1).

## 9.7 СПЕЦИФИКАЦИЯ ЗАПРОСОВ

### 9.7.1 ПРОВЕРКА ДАННЫХ АБОНЕНТА ПЕРЕД ОФОРМЛЕНИЕМ ПОДПИСКИ

Запрос `<verify_subscriber>`. Обработчик запроса инициирует обращение к API методу ядра шлюзов, что в свою очередь вызывает обращение к реализации этого метода в шлюзе текущего поставщика услуги, согласно направлений проведения. Запрос проходит в онлайн.

Таблица 9.7.1.1 — Передаваемые параметры

Название	Формат	Обяз.	Описание
<code>service</code>	int2	Да	Номер услуги, справочник предоставляется ПЦ
<code>&lt;attribute&gt;</code>	элемент	Нет	Список атрибутов, по значениям которым выполняется поиск и фильтрация кандидатов. Коды атрибутов, и возможные значения предоставляются ПЦ для каждой услуги. Структура аналогична элементу из запроса <code>&lt;payment&gt;</code> — таблица 8.3.5.2

#### Возвращаемые параметры:

Код ответа. В случае успеха — один элемент, на который допустимо оформить подписку. Элемент в свою очередь, содержит набор элементов (атрибутов), по которым необходимо выполнять запрос на проверку и на оформление подписки (секция

<subscription\_info>), и набор элементов (атрибутов), несущих информационный характер (секция <addinfo>) — опционально.

Такое поведение позволяет осуществлять подтверждение пользователем корректности введенных данных и дальнейшее намерение оформить подписку. Атрибуты из <addinfo> не нужно передавать при запросе оформления подписки, т. к. шлюзом используются только атрибуты, которые система вернула в секции <subscription>, любые другие атрибуты — игнорируются.

#### ПРИМЕР ЗАПРОСА:

```
<verify_subscriber service="1">  
  <attribute name="account" value="554002382735"/>  
</verify_subscriber>
```

#### УСПЕШНЫЙ РЕЗУЛЬТАТ:

```
<result code="0" message="Успех">  
  <candidate Name="Ivanov I I">  
    <subscription_info>  
      <attribute name="account" value="220803347713"/>  
    </subscription_info>  
    <addinfo>  
      <attribute name="fio" value="Ivanov I I"/>  
      <attribute name="details" value="Lenina 55-134"/>  
    </addinfo>  
  </candidate>  
</result>
```

#### ОШИБКА:

```
<result code="60"  
  message="По указанным реквизитам подписка не может быть оформлена"/>
```

## 9.7.2 ПОДКЛЮЧЕНИЕ АБОНЕНТА НА УСЛУГУ ПОЛУЧЕНИЯ НАЧИСЛЕНИЙ

Запрос <subscription>. Обработчик запроса предварительно выполняет поиск в БД подписок на предмет наличия подписки на указанные реквизиты у этого подписчика. ПЦ создает сущность подписки в БД ПЦ и возвращает объект с идентификатором созданной сущности.

Таблица 9.7.2.1 — Передаваемые параметры

Название	Тип	Формат	Обяз.	Описание
<b>identifier</b>	Атрибут	varchar(50)	Да	Уникальный идентификатор клиента на стороне Партнера (например, номер телефона)
<b>service</b>	Атрибут	int2	Да	Номер услуги, справочник предоставляется ПЦ
<b>name</b>	Атрибут	varchar (100)	Да	Наименование подписки для идентификации клиентом (вводится клиентом или генерируется Партнером на основании введенных данных/полученных при проверке атрибутов)
<b>notification-method</b>	Атрибут	int2	Нет	Метод получения начислений по подписке партнером. 0 — Партнер сам забирает начисления, используя метод <code>get_notifications</code> , 1 — Партнер настраивает на своей стороне Web-обработчик, на который ПЦ будет отправлять начисления. В случае отсутствия

Название	Тип	Формат	Обяз.	Описание
				считается как 0
<attribute>	Элемент	-	Нет	Список атрибутов абонента, необходимых для осуществления подписки. Коды атрибутов и возможные значения предоставляются ПЦ для каждой услуги. Структура аналогична элементу из запроса <payment> — таблица 8.3.5.2

Ответ на запрос содержит код результата и в случае успеха — идентификатор оформленной подписки для возможности выполнения в дальнейшем с ней манипуляций (отказа от подписки, изменения условий).

#### ПРИМЕР ЗАПРОСА (ПОДКЛЮЧЕНИЕ ПОДПИСКИ НА ВЫСТАВЛЕНИЕ СЧЕТА):

```
<subscription identificator="9132331122" service="72"  
    name="Квартира сестры - Ленина 55">  
    <attribute name="account" value="220803347713"/>  
</subscription>
```

#### Успех:

```
<result code="0" message="Успех">  
    <subscription id="100500565">  
</result>
```

#### Ошибка:

```
<result code="60"  
    message="По указанным реквизитам подписка не может быть  
оформлена"/>
```

#### ПРИМЕР ЗАПРОСА (ПОДКЛЮЧЕНИЕ УСЛУГИ АВТОПОПОЛНЕНИЯ):

```
<subscription identificator="1122" service="1">  
    <attribute name="phone" value="9132134455"/>  
    <attribute name="sum" value="10000"/>
```

```
<attribute name="threshold" value="5000"/>
</subscription>
```

**Успех:**

```
<result code="0" message="Успех">
  <subscription id="100500568">
</result>
```

**Ошибка:**

```
<result code="12" message="Ошибка регистрации"/>
```

### 9.7.3 ОТКЛЮЧЕНИЕ АБОНЕНТА ОТ УСЛУГИ ПОЛУЧЕНИЯ НАЧИСЛЕНИЙ

Запрос `<unsubscribe>` — отключение подписки.

Таблица 9.7.3.1 — Передаваемые параметры

Название	Формат	Обяз.	Описание
<b>identifier</b>	varchar(50)	Да	Уникальный идентификатор абонента (подписчика) в разрезе услуги (например, номер телефона)
<b>service</b>	int2	Да	Номер услуги, справочник предоставляется ПЦ
<b>subscription</b>	int4	Нет	Идентификатор подписки, которую следует прекратить

Запрос затрагивает либо все подписки абонента, либо (если передан уникальный идентификатор конкретной подписки в системе ПЦ) только указанную подписку по запрошенному сервису и подписчику. Идентификаторы подписок Партнер может сохранять либо в своей информационной системе при оформлении подписки, либо запрашивать актуальный при помощи метода `<get_subscriptions>`.

В БД ПЦ выполнение метода либо удаляет соответствующие подписки (если по ним не было ни одного счета), либо меняет статус подписки на «Архивный», после чего подписка перестает быть доступна при помощи метода `<get_subscriptions>` и перестает обрабатываться системными шлюзами.

Ответ на запрос содержит код результата, и в случае успеха — элементы `<subscription>` с идентификаторами подписок, которые были прекращены. В случае ошибки, если ошибка вызвана несуществующим уникальным идентификатором подписки либо несоответствием идентификатора подписки идентификатору абонента, ответ содержит элемент `<subscription>` с уникальным идентификатором подписки, который вызвал ошибку.

#### ПРИМЕР ЗАПРОСА — ОТКАЗ ОТ ВСЕХ ПОДПИСОК НА СЕРВИС:

```
<unsubscribe identificator="9132331122" service="1"/>
```

##### Успех:

```
<result code="0" message="Успех">  
  <subscription id="100400568">  
</result>
```

##### Ошибка:

```
<result code="12" message="Абонент не найден"/>
```

#### ПРИМЕР ЗАПРОСА — ОТКАЗ ОТ ОДНОЙ ИЗ ПОДПИСОК НА СЕРВИС:

```
<unsubscribe identificator="9132331122" service="1"  
  subscription="100400568"/>
```

##### Успех:

```
<result code="0" message="Успех">  
  <subscription id="100400568">  
</result>
```

##### Ошибка:

```
<result code="40" message="Идентификатор подписки не соответствует номеру  
абонента"/>
```

#### 9.7.4 ИЗМЕНЕНИЕ ПАРАМЕТРОВ ПОДПИСКИ

Запрос `<change_subscription>`. Обработчик запроса предварительно выполняет поиск в БД подписок на предмет наличия подписки на вновь запрашиваемые реквизиты у этого подписчика. В случае успеха — обработчик изменяет атрибуты подписки в БД ПЦ.

Таблица 9.7.4.1 — Передаваемые параметры

Название	Тип	Формат	Обяз.	Описание
<code>identifier</code>	Атрибут	<code>varchar(50)</code>	Да	Уникальный идентификатор абонента (подписчика) в разрезе услуги (например, номер телефона)
<code>service</code>	Атрибут	<code>int2</code>	Да	Номер услуги, справочник предоставляется ПЦ
<code>subscription</code>	Атрибут	<code>int4</code>	Нет	Идентификатор подписки, которую следует изменить. Не передается, если Партнер оперирует с единственной подпиской по сервису
<code>name</code>	Атрибут	<code>varchar(100)</code>	Нет	Новое наименование подписки, если не задано — не изменяется
<code>&lt;attribute&gt;</code>	Элемент	-	Нет	Список атрибутов абонента, необходимых для осуществления подписки. Коды атрибутов и значения предоставляются ПЦ для каждой услуги. Структура аналогична элементу из

Название	Тип	Формат	Обяз.	Описание
				запроса <payment> — таблица 8.3.5.2

Запрос позволяет либо оперировать с конкретной подпиской, передавая опционально ее уникальный идентификатор, либо — не передавать уникальный идентификатор, и в этом случае предполагается, что у абонента по сервису подписка единственная. Если это не так — система вернет соответствующую ошибку, которая означает, что у абонента несколько подписок на сервис и следует передать дополнительно уникальный идентификатор подписки. Идентификаторы подписок Партнер может сохранять либо в своей информационной системе при оформлении подписки, либо запрашивать актуальный перечень по абоненту при помощи метода <get\_subscriptions>.

В БД ПЦ выполнение метода удаляет все существующие атрибуты подписки и добавляет новые взамен старых. Атрибуты следует передавать полностью все, т. к. метод не аддитивный.

Ответ на запрос содержит код результата.

#### ПРИМЕР ЗАПРОСА — ИЗМЕНЕНИЕ ЕДИНСТВЕННОЙ ПОДПИСКИ:

```
<change_subscription identifier="9132331122" service="1">  
  <attribute name="sum" value="20000"/>  
  <attribute name="threshold" value="10000"/>  
</change_subscription>
```

#### Успех:

```
<result code="0" message="Успех"/>
```

#### Ошибка:

```
<result code="12" message="Абонент не найден"/>
```

**ПРИМЕР ЗАПРОСА — ИЗМЕНЕНИЕ ОДНОЙ ИЗ ПОДПИСОК НА СЕРВИС:**

```
<change_subscription identificator="9132331122" service="72"  
                        name="Бабушкина избушка" subscription="100400568">  
  <attribute name="acount" value="2203346722"/>  
</change_subscription>
```

**Успех:**

```
<result code="0" message="Успех"/>
```

**Ошибка:**

```
<result code="40"  
        message="Идентификатор подписки не соответствует номеру  
                абонента"/>
```

### 9.7.5 ПОЛУЧЕНИЕ СПИСКА ДЕЙСТВУЮЩИХ ПОДПИСОК

Запрос: `<get_subscriptions>`. Обработчик запроса возвращает активные (не архивные) подписки по идентификатору подписчика.

Таблица 9.7.5.1 — Передаваемые параметры `<get_notifications>`

Название	Формат	Обяз.	Описание
<code>identificator</code>	<code>varchar(50)</code>	Да	Уникальный идентификатор абонента (подписчика)
<code>service</code>	<code>int2</code>	Нет	Номер услуги, справочник предоставляется ПЦ

Запрос возвращает результат и объекты `<subscription>`, позволяет получить все выставленные счета по услуге, опционально указав номер абонента. Возвращает список объектов `<subscription>`, найденных согласно переданным параметрам для поиска — подписчик и сервис (опционально). Каждый элемент списка описывает один элемент, описывающий текущую подписку.

Таблица 9.7.5.2 — Объект `<subscription>`

Название	Тип	Формат	Обяз.	Описание
<code>id</code>	Атрибут	<code>int4</code>	Да	Уникальный идентификатор подписки
<code>service</code>	Атрибут	<code>int2</code>	Да	Номер услуги, справочник предоставляется ПЦ
<code>name</code>	Атрибут	<code>varchar(100)</code>	Да	Имя подписки
<code>dset</code>	Атрибут	<code>time</code>	Да	Дата подписки
<code>dlastbill</code>	Атрибут	<code>time</code>	Нет	Время последнего выставленного

Название	Тип	Формат	Обяз.	Описание
				начисления
<attribute>	Элемент	-	Да	Атрибуты, на которые оформлена подписка, минимум 1 элемент. Структура аналогична элементу из запроса <payment> — таблица 8.3.5.2

#### ПРИМЕР ЗАПРОСА:

```
<get_subscriptions identificator="9132735185"/>
```

#### Успех:

```
<result code="0" message="Успех">
  <subscriptions>
    <subscription id="10050032"
      Name="Автоплатеж МТС" service = "1"
      dset ="2007-10-12T12:00:00+0300"
      dlastbill ="2007-10-15T12:00:00+0300">
      <attribute name="account" value="9132735185"/>
      <attribute name="sum" value="10000"/>
      <attribute name="threshold" value="20000"/>
    </subscription>
    ...
    <subscription id="10050056"
      Name="квартира сестры" service = "27"
      dset ="2007-10-12T12:00:00+0300">
      <attribute name="account" value="554002382735"/>
      <attribute name="fio" value="Ivanova E S"/>
      <attribute name="details" value="Lenina 55-134"/>
    </subscription>
  </subscriptions>
</result>
```

#### Ошибка:

```
<result code="50" message="Данные не найдены"/>
```

## 9.7.6 ПОЛУЧЕНИЕ СПИСКА НАЧИСЛЕНИЙ

В зависимости от переданного метода получения начисления при создании подписки (<notification-method>) Партнер может самостоятельно забирать новые начисления, используя метод <get\_notifications>, либо настроить на своей стороне Web-сервер для получения начислений от ПЦ.

Передаваемые параметры для метода <get\_notifications> приведены в таблице 9.7.6.1.

Таблица 9.7.6.1 — Передаваемые параметры

Название	Формат	Обяз.	Описание
<b>service</b>	int2	Да	Номер услуги, справочник предоставляется ПЦ
<b>identifier</b>	varchar(50)	Нет	Уникальный идентификатор абонента. Опциональное поле, предназначено для фильтрации начислений
<b>subscription</b>	int4	Нет	Идентификатор подписки

ПРИМЕР ЗАПРОСА:

```
<get_notifications service="1"/>
```

Запрос возвращает объекты выставленного счета (<notifications>) в виде списка из объектов Notification. Каждый объект <notification> представляет собой одно начисление по услуге и включает в себя атрибуты, приведенные в таблице 9.7.6.2.

Таблица 9.7.6.2 — Атрибуты выставленного начисления

Название	Тип	Формат	Обяз.	Описание
<b>id</b>	Атрибут	int4	Да	Уникальный идентификатор начисления
<b>identifier</b>	Атрибут	varchar(50)	Да	Уникальный идентификатор абонента
<b>subscription</b>	Атрибут	int4	Да	Идентификатор подписки
<b>service</b>	Атрибут	int2	Да	Номер услуги, по которой выставлен счет
<b>period</b>	Атрибут	varchar2 (50)	Да	Идентификатор начисления (период начисления) — период, за который выставлен счет/задолженность. Может быть пустым
<b>description</b>	Атрибут	varchar(100)	Да	Наименование счета. Если не переопределено на стороне поставщика — наименование подписки
<b>sum</b>	Атрибут	int8	Да	Сумма выставленного счета (в копейках)
<attribute>	Элемент	-	Нет	Дополнительные атрибуты выставленного счета. Структура аналогична элементу из запроса <payment> — таблица 8.3.5.2

Запрос позволяет получить все выставленные счета по абоненту и услуге либо все выставленные счета по конкретной подписке.

---

**ПРИМЕР ОТВЕТА:**

```
<result code="0" message="Успех">
<notifications>
  <notification id="77043" identificator="9132131122" service="1"
    subscription="12333333" sum="10000" period=""
    description="Автоплатеж по услуге МТС">
  </notification>
  <notification id="77056" identificator="9132131122" service="72"
    sum="354017" subscription="133333333" period="jun 2015"
    description="Квартира сестры">
    <attribute name="fio" title="Фамилия" value="Ivanov I I"/>
    <attribute name="counter1" title="Показание счетчика ГВС"
      value="105"/>
    <attribute name="counter2" title="Показание счетчика ХВС"
      value="223"/>
  </notification>
  ...
</notifications>
</result>
```

**ОШИБКА:**

```
<result code="0" message="Нет абонентов требующих пополнения"/>
```

Список пополнений включает в себя уникальный идентификатор начисления, который нужно использовать в запросах `<get_result>`, `<process_notification>`; уникальный идентификатор абонента; сумму пополнения в копейках; а также текстовое сообщение, которое можно показать абоненту.

Счет будет присутствовать в списке до тех пор, пока:

1. Начисление не будет погашено в другом месте, и статус начисления не примет значение «Архивный» (устанавливается системным шлюзом опроса поставщиков).
2. Начисление не будет погашено партнером, о чем Партнер уведомит ПЦ запросом `<process_notification>`, установив счету статус «Оплачен».

3. Начисление не будет отклонено Партнером (плательщиком), о чем Партнер уведомит ПЦ запросом `<process_notification>`, установив счету статус «Аннулирован».

В случае с методом получения начислений на стороне Партнера необходимо настроить Web-сервер для приема запросов по протоколу HTTP/HTTPS, предоставить полный URL для отправки запросов, а также, при необходимости, авторизационные данные (Сертификат, Логин/Пароль для Basic авторизации, и т. п.).

Запрос на сторону Партнера будет отправляться в формате XML в теле POST-запроса и иметь ту же структуру, что и ответ на `<get_notifications>`. Пример:

```
<notifications>
  <notification id="77043"
    identifier="9132131122"
    service="1"
    subscription="12333333"
    sum="10000"
    period=""
    description="Автоплатеж по услуге МТС">
  </notification>
  ...
</notifications>
```

В случае успешной регистрации начислений на стороне Партнера ответ должен вернуться в теле HTTP-ответа и содержать элемент `<result>` с кодом ответа в атрибуте **code** (0 — успех, 1 — ошибка).

```
<result code="0" message="Успех"/>
```

Или:

```
<result code="1" message="Ошибка сохранения начислений в БД"/>
```

### 9.7.7 РЕЗУЛЬТАТ СПИСАНИЯ

Запрос `<process_notification>`. Позволяет изменить статус начисления в системе ПЦ.

Варианты использования:

1. Начисление погашено партнером (через ПЦ или иную систему), о чем Партнер уведомит ПЦ запросом `<process_notification>`, установив счету статус «Оплачен». В данном случае, при проведении платежа через ПЦ, необходимо передавать атрибут **bill-id**, в котором указан идентификатор начисления (см. раздел [8.3.5](#)).
2. С клиента списаны денежные средства по начислению, и необходимо провести платеж в ПЦ, в данном случае платеж будет создан ПЦ самостоятельно.
3. Начисление отклонено Партнером (плательщиком), о чем Партнер уведомит ПЦ запросом `<process_notification>`, установив счету статус «Аннулирован».

Таблица 9.7.7.1 — Передаваемые параметры `<process_notification>`

Название	Формат	Обяз.	Описание
<b>identifier</b>	varchar(50)	Да	Уникальный идентификатор абонента
<b>service</b>	int2	Да	Номер услуги
<b>id</b>	int4	Да	Идентификатор начисления
<b>action</b>	varchar(10)	Да	Статус счета, который следует присвоить. Допустимые варианты: <ul style="list-style-type: none"><li>• PAID — оплачен партнером;</li><li>• CHARGED — списано с клиента;</li><li>• DECLINED — отклонен без оплаты.</li></ul>

ПРИМЕР ЗАПРОСА:

```
<process_notification identificator="9132735185" service="1"  
id="10050032" action="PAID"/>
```

ПРИМЕР ОТВЕТА:

```
<result code="0" message="Успех"/>
```

## 9.7.8 ПОЛУЧЕНИЕ СТАТУСА ПЛАТЕЖА

Передаваемые параметры приведены в таблице 9.7.8.1.

Таблица 9.7.8.1 — Передаваемые параметры

Название	Формат	Обяз.	Описание
id	int4	Да	Уникальный идентификатор начисления

ПРИМЕР ЗАПРОСА:

```
<get_result id="12345"/>
```

УСПЕШНЫЙ РЕЗУЛЬТАТ:

```
<result code="0" message="Успех" state="60" error="0"/>
```

ОБРАБОТКА:

```
<result code="0" message="Проведение" state="40" error="0"/>
```

ОШИБКА:

```
<result code="0" message="Ошибка" state="80" error="1"/>
```

В ответе дополнительно присутствует состояние платежа по начислению и номер ошибки (Приложение [«А. Результат данных, возвращаемых ПЦ»](#), Таблица А.2.1) в случае, когда платеж не проводится. Возможные состояния платежа:

1. 40 — «В обработке».
2. 60 — «Успешно проведен».
3. 80 — «Ошибка».

**ПРИЛОЖЕНИЕ А. РЕЗУЛЬТАТ ДАННЫХ, ВОЗВРАЩАЕМЫХ ПЦ****А.1 КОДЫ СТАТУСОВ И СУБСТАТУСОВ ПЛАТЕЖА**

Коды статусов и субстатусов платежа приведены в таблице А.1.1.

Таблица А.1.1— Коды статусов и субстатусов платежа

Статус	Субстатус	Финальный	Описание
0	0	Нет	Новый. Статус, в котором платежи вставляются в БД
0	1	Нет	Готов к обработке. Подготовка к обработке и перенос в очередь
0	2	Нет	Определение провайдера. Определение провайдера согласно направлениям проведения платежа
0	3,4	Нет	Fraud-control. Статус Fraud-контроля, если система контроля включена, платежи добавляются в это состояние
0	5	Нет	Подтверждение. Статус, в котором платежи загружаются в кабинет через раздел «Диспетчерская — Загрузка платежей». Платежи будут находиться в нем, пока пользователь не подтвердит проведение (кнопка <b>«Провести все отложенные»</b> )
0	6	Нет	Провайдер не задан. В данный статус попадают платежи по сервисам, для которых не определено направление проведения

Статус	Субстатус	Финальный	Описание
0	7	Нет	Таймаут. Статус, в который будут попадать платежи на время, указанное в настройках сервиса как «отсрочка Проведения платежей»
0	8	Нет	Отложен. Статус платежа в случае включенной у провайдера настройки «Использовать отложенное проведение». Будет находиться в нем до ручного проведения
0	9	Нет	Ожидает подтверждения. Данный статус получает платеж, полученный с внешнего шлюза (external) от внешней системы с флагом dealed. Находится в данном статусе до тех пор, пока внешняя система не пришлет подтверждение
0	11	Нет	Вознаграждение не задано. В данный статус попадают платежи по сервисам, которые не добавлены в профиль вознаграждения
10	любой	Нет	Платеж заблокирован (по результатам применения правил Fraud-мониторинга)
20	1	Нет	Готов к списанию
20	2,3	Нет	Списание средств со счета
20	4	Нет	Недостаточно средств на счете
20	5	Нет	Ожидание списания банковской операции
30		Нет	Предварительная верификация платежа
30	1	Нет	Готов к предварительной верификации (модулями предварительной проверки)
30	2	Нет	Предварительная верификация, в обработке

Статус	Субстатус	Финальный	Описание
30	3	Нет	Предварительная верификация, неизвестный. Верификация закончилась неоднозначной ошибкой. После истечения глобального таймаута (1 день) операция переводится в статус «Проведение» (40) субстатус «Ошибка» (4) для возврата резерва, а затем в финальный статус «Ошибка» субстатус «Отклонен у провайдера» (5)
30	4	Нет	В статус попадают операции, которые не прошли проверку модулем предварительной проверки при условии, что для модуля установлена опция <b>«Ставить в ручной контроль»</b> . Операция находится в статусе либо до истечения глобального таймаута (5 дней), либо до ручного перевода в другой статус. После истечения таймаута операция сначала переводится шлюзами в статус «Проведение» (40) субстатус «Ошибка» (4) для возврата резерва агенту, а затем — в финальный статус «Ошибка» субстатус «Отклонен у провайдера» (5). Вручную операцию возможно перевести в статус «Проведение» (40) субстатус «Готов к проведению» (0) — в этом случае операция будет переведена в проведение или в статус «Верификация» (30) субстатус «Заблокирован. Отменен» (5)
30	5	Нет	В этот статус платежи могут быть переведены только вручную из статуса

Статус	Субстатус	Финальный	Описание
			«Верификация» (30) субстатуса «Заблокирован. Отменен» (4). Платежи в этом статусе сначала переводятся шлюзами в статус «Проведение» (40) субстатус «Ошибка» (4) для возврата резерва агенту, а затем в финальный статус «Ошибка» субстатус «Отклонен у провайдера» (5)
40	1	Нет	Готов к проведению
40	2,3	Нет	Проведение
40	4,5,6,7	Нет	Ошибка проведения
40	8	Нет	Ожидается ответ от внешнего поставщика
40	9	Нет	Ожидание подтверждения от внешней системы
40	10	Нет	Ожидание недостающих атрибутов, необходимых для завершения проведения (для операций в разрывном статусе)
60		Да	Статус успешного проведения (финальный)
80		Да	Статус ошибки (финальный)
80	1,2,3	Да	Платеж отменен вручную
80	4	Да	Недостаточно средств на счете
80	5	Да	Ошибка проведения
80	6,7,8	Да	Другая ошибка
80	9	Да	Возврат средств
80	10	Да	Корректируемая ошибка
80	11	Да	Средства недовнесены
80	12	Да	Заблокирован пользователем

---

Статус	Субстатус	Финальный	Описание
-1	0	Нет	Ошибка вставки операции в БД (нефинальный)
-2	0	Да	Платеж не найден при проверке статуса (финальный)

Финальными считаются статусы 60,80,-2. В случае статуса 80 о причине ошибки мы можем узнать из элемента code, возможные значения приведены в таблице А.2.1.

В том случае, если в ответе возвращается статус или субстатус, отличный от представленных в таблице, необходимо считать статус платежа нефинальным и продолжать запрашивать статус.

## А.2 НОМЕРА ОШИБОК

Номера ошибок приведены в таблице А.2.1.

Таблица А.2.1 — Номера ошибок

Номер ошибки	Описание
-100	Дублирующиеся атрибуты переданы
-2	Ошибка платежного инструмента. Описание ошибок приведено в таблице А.7.1
1,2	Неверно указан номер
3	Сумма вне допустимого диапазона
4	Сервер провайдера недоступен
5	Ошибка авторизации
6	Сервер оператора недоступен
7	Общая ошибка провайдера
8	Нехватка средств
9	Неверные параметры проведения
10	Фатальная ошибка провайдера
30	Недостаточно средств для добавления платежа
33	Сервис недоступен для данного агента
50	Не прошел верификацию
<b>Для онлайн-проверки:</b>	
-100	Дублирующиеся атрибуты переданы
1000	Неверный номер
1001	Сервер поставщика недоступен
1002	Прием платежей запрещен провайдером

---

Номер ошибки	Описание
1003	Проверка через сервер поставщика недоступна. Вы можете совершить платеж, если уверены в правильности параметров
1004	Задолженности не найдены
1005	Платеж перенаправлен на другой сервис
1006	Поставщик не может выполнить проверку, но вы можете совершить платеж, если уверены в правильности реквизитов
1007	Оплата невозможна по причине отсутствия персональных данных у оператора
1008	Отклонено системой Fraud-мониторинга. Проверка сервисом Fraud-мониторинга при онлайн-проверке выполняется, если в разделе «Справочники — Системные параметры» на вкладке «Система Fraud-мониторинга» установлен флажок «Использовать ЧБ при онлайн-проверке»
1100	Недостаточно денежных средств на балансе агента для выполнения запроса

### А.3 НОМЕРА ОШИБОК АТТРИБУТА SERVICE ПРИ ADVANCED-ЗАПРОСЕ

Номера ошибок атрибута **service** при advanced-запросе приведены в таблице А.3.1.

Таблица А.3.1 — Номера ошибок атрибута service при advanced-запросе

Номер ошибки	Описание
0	Успешно
1	Абонент по указанным реквизитам не найден
2	Запрет оплаты указанного номера со стороны поставщика услуги
3	Невозможно разобрать ответ поставщика. Попробуйте выполнить попытку позднее
4	Онлайн-запросы для данного сервиса не реализованы
5	Задолженности отсутствуют
6	Отсутствует или неверен один из аргументов
7	Под указанные условия поиска попадает слишком большое число элементов. Уточните параметры запроса
8	По указанным условиям поиска не найдено ни одной записи. Измените условия поиска
9	Некая ошибка с текстовым описанием. Текстовое описание возвращается в элементе message: <code>&lt;input key="message" value="Необходимо выполнить восстановление карты"/&gt;</code>
10	Не удалось выполнить проверку. Самостоятельно принять решение о продолжении или прекращении обработки платежа
11	Оплата невозможна по причине отсутствия персональных данных у оператора
100	Недостаточно денежных средств на балансе агента для

Номер ошибки	Описание
	выполнения запроса

**А.4 КОДЫ СТАТУСОВ ОБРАБОТКИ ЗАЯВОК НА ОТМЕНУ**

Коды статусов обработки заявок на отмену приведены в таблице А.4.1.

Таблица А.4.1 — Коды статусов обработки заявок на отмену

Статус	Финальный	Описание
-1	Да	Не найден пользователь
-2	Да	У пользователя другой агент (не такой, как у точки)
-3	Да	Пользователь заблокирован
-4	Да	Платеж не найден
-5	Да	Платеж не в успехе (в неверном статусе)
-6	Да	Неверная сумма отмены (возможно больше суммы платежа)
-10	Нет	Неизвестная ошибка (возвращается в случае, если произошло исключение при обработке)
0	Нет	Новый статус заявки
1	Нет	В обработке
2	Да	Выполнена (Успех)
3	Да	Отклонена
10	Нет	Отмена по шлюзу (обработку отмены выполняет шлюз поставщика, а не менеджер ПЦ)
11	Нет	Отмена по шлюзу, обработка
12	Нет	Отмена по шлюзу, неизвестный
13	Да	Отмена по шлюзу, успех
14	Да	Отмена по шлюзу, ошибка

**A.5 КОДЫ ОТВЕТОВ МОДУЛЯ УПРАВЛЕНИЯ ПОДПИСКАМИ**

Коды ответов модуля управления подписками приведены в таблице A.5.1.

Таблица A.5.1 — Коды ответов модуля управления подписками

<b>Статус</b>	<b>Описание</b>
0	Успех
10	Неверный тип запроса
11	Такой абонент уже зарегистрирован
12	Абонент не найден
13	Платеж не найден
14	Платеж еще не вставлен или не произошло списание
100	Ошибка разбора параметров запроса
101	Ошибка работы с БД

## А.6 ТИПЫ СОБЫТИЙ

Типы событий в системе мониторинга приведены в таблице А.6.1.

Таблица А.6.1 — Типы событий в системе мониторинга.

ID	Тип события	Причина возникновения
1	Старт автомата	Запуск терминального ПО
2	Ошибка автомата	Общая ошибка терминального ПО. Подробности в комментарии к событию
3	Л/с — Заканчиваются средства	Заканчиваются средства на лицевом счету Агента. Порог срабатывания задается в настройках Агента
4	Л/с — Закончились средства	Закончились средства на лицевом счету Агента
5	Л/с — Пополнение	Совершено пополнение лицевого счета Агента. В комментарии к событию сумма пополнения и подробности
6	Купюроприемник — Наполнение	Купюроприемник близок к переполнению. Максимальная вместимость купюроприемника задается в настройках точки в бэк-офисе системы. Коэффициент срабатывания 0.75 от максимальной вместимости
7	Купюроприемник — Переполнение	Купюроприемник переполнен. Событие генерируется самим купюроприемником на уровне устройства
8	Купюроприемник — Вынута кассета	Вынута кассета купюроприемника
9	Купюроприемник — Зажевывание в голове	Зажевана купюра в голове купюроприемника
10	Купюроприемник — Зажевывание в кассете	Зажевана купюра в кассете купюроприемника
11	Купюроприемник —	Ошибки, о которых сообщает само устройство

	Ошибка мотора кассеты	
12	Купюроприемник — Ошибка транспортного мотора	
13	Купюроприемник — Ошибка выравнивающего мотора	
14	Купюроприемник — Ошибка инициализации стекера	
15	Купюроприемник — Ошибка оптического канала	
16	Купюроприемник — Ошибка магнитного канала	
17	Купюроприемник — Ошибка емкостного канала	
18	Купюроприемник — Ошибка связи с устройством	Ошибка взаимодействия с устройством через порт (Ошибка ввода/вывода)
19	Купюроприемник — Другая Ошибка	Прочая ошибка купюроприемника
20	Купюроприемник — Работоспособность восстановлена	Работоспособность купюроприемника восстановлена после любого сбоя
21	Принтер — Неисправность	Неисправность принтера, подробности в комментарии к событию
22	Принтер — Работоспособность восстановлена	Работоспособность принтера восстановлена
23	Связь — Длительное	Терминал в течение длительного времени не

	отсутствие	выходит на связь. Периодичность выхода на связь задается в параметрах точек в бэк-офисе системы
24	Связь — Работоспособность восстановлена	Связь с терминалом восстановлена
25	Купюроприемник — Возможное зажевывание	Данное событие не связано с сообщением от устройства о зажевывании купюры. А возникает в том случае, когда несколько человек подряд не могут совершить платеж, по причине того, что купюроприемник не принимает купюры. Количество попыток оплаты – 3, количество попыток внесения купюры – 10
26	Вход в сервисное меню	Выполнен вход в сервисное меню терминала. В комментариях указывается логин техника
27	Провайдеры низкий баланс	Низкий баланс на счету Провайдера/Агрегатора. Порог срабатывания задается в настройках провайдера. Уведомление отправляется в случае, если через данного провайдера есть хотя бы одно активное направление проведения
28	Неизвестный номер	Номер отсутствует в номерных емкостях, но платеж в итоге прошел. Используется при отсутствии онлайн-проверки номера, а при проверке его по базе емкостей номеров сотовых операторов (включая БДПН)
29	Ошибка справочников	Ошибка обработки справочной информации, полученной с сервера процессинга
30	Инкассация с гашением	Инкассация со сбросом счетчиков на терминале и отправкой пакета инкассации на сервер с покупюрной раскладкой
31	Инкассация без гашения	Инкассация без сброса счетчиков на терминале. Не инкассация по сути, а лишь вытаскивание кассеты, например, для устранения проблем (зажевана купюра в кассете). Данное событие можно не создавать
32	Ошибка обновления	Ошибка обновления терминального ПО. Подробности в комментарии

33	Мало места на диске	Мало места на диске терминала. Свободного места осталось менее 300 Мб
34	Отсутствие платежей	Отсутствие платежей в течение длительного времени. Период максимального простоя задается в параметрах точки
35	Заканчивается срок действия сертификата провайдера	Если работа с провайдером ведется с использованием сертификата, отслеживается его срок действия. Событие генерируется при сроке окончания, наступающем менее чем через две недели
38	Зажевано на номер	Если удается определить номер, который вводился перед зажевыванием купюры, в комментариях данного события он будет отображен
40	Стекло откалибровано некорректно	В параметрах точек возможно задать реакцию на снятие кассеты при необходимости проверки калибровки экрана. В данном случае, при снятии кассеты на экране, открывается специальное окно с мишенями по углам, на которые нужно попасть техникой. Если тест не пройдет, на сервер улетает данное событие
41	Платеж сверх суточного лимита	Платеж сверх суточного лимита, заданного в настройках точки, заблокирован
42	Обновление ПО	Успешно выполнено обновление терминального ПО
43	Пополнение сим карты точки	Выполнено автоматическое пополнение сим-карты терминала путем создания платежа на номер сим-карты. Создается в случае настройки отслеживания баланса сим-карты, а также задания порога баланса для пополнения. Настраивается в параметрах точки
44	Предупреждение на автомате	Общее предупреждение терминального ПО. Подробности в комментарии к событию. Например, ошибка разбора какого-либо сценария услуги либо ошибка разбора шаблона чека
45	Заканчиваются карты	При продаже ПИН-кодов с хранением их в базе процессинга, возникает при остатке ниже заданного порога срабатывания
47	Большое количество	Порог задается в настройках сервиса и срабатывает

	ошибок проведения	при большом количестве отказов проведения платежа по сервису
48	Большое количество платежей в проведении	Порог задается в настройках провайдера и срабатывает, когда в очереди проведения платежей по Провайдеру/Агрегатору обрабатывается много платежей (свыше порога)
49	Долго висит платеж в проведении	Порог задается в настройках провайдера в часах и срабатывает, если платеж обрабатывается дольше заданного периода
51	Нештатная ситуация в шлюзе	Нештатная ситуация в каком-либо шлюзе (с Провайдером или системном шлюзе). Подробности в комментарии к событию
52	Сработал датчик	Сработал какой-либо датчик на терминале. Например датчик удара и открытия двери. Подробности в комментарии к событию
53	Купюроприемник требует ремонта	Генерируется при высоком проценте отказа приема купюр купюроприемником два дня подряд. Генерируется на второй день. Процент отказа за два дня должен быть более 60%, причем за второй день больше, чем за первый
54	Низкое качество связи	На терминале ведется статистика ошибок сетевого взаимодействия. В случае 50%-го превышения два дня подряд, на второй день происходит генерация данного события, причем за второй день процент ошибок должен быть больше, чем за первый
55	Экран выключен	Экран терминального ПО выключен по расписанию, которое настраивается в настройках точки
56	Экран включен	Экран терминального ПО включен по расписанию, которое настраивается в настройках точки
57	Ошибка в системе шлюзов	Неперехваченное исключение в системе шлюзов. Какой-то шлюз завершил работу в нештатном режиме. Приоритет события высокий
58	Почтомат — Открытие дверцы	Устройство сгенерировало событие «Открытие дверцы»
59	Почтомат — Закрытие дверцы	Устройство сгенерировало событие «Закрытие дверцы»

60	Почтомат — Неисправность	Возникла неисправность устройства
61	Неподдерживаемая или недопустимая версия Java	Версия Java проверяется раз в сутки, и при необходимости генерируется соответствующее событие. Информацию о поддерживаемых и неподдерживаемых версиях определяют разработчики
62	Версия Java требующая обновления	Версия Java проверяется раз в сутки, и при необходимости генерируется соответствующее событие. Информацию о версиях, требующих обновления, определяют разработчики
63	Блокированные платежи	Доступно в системах с установленным модулем Fraud-контроля. События генерируются в разрезе агентов, в качестве комментария используется количество заблокированных платежей. События генерируются раз в 4 часа для конкретного агента
64	Диспенсер карт — Ошибка	Событие возникает в случае возникновения ошибок, приводящих к невозможности работы диспенсера карт. В комментарии к событию отображаются коды ошибок
65	Диспенсер карт — Работоспособность восстановлена	Событие возникает в случае восстановления работоспособности диспенсера карт
66	Диспенсер карт — Предупреждение	Событие возникает в случае возникновения некритичных ошибок в работе диспенсера карт. В комментарии к событию отображаются коды предупреждений
67	Диспенсер купюр — Ошибка	Событие возникает в случае возникновения ошибок, приводящих к невозможности работы диспенсера купюр. В комментарии к событию отображаются коды ошибок
68	Диспенсер купюр — Работоспособность восстановлена	Событие возникает в случае восстановления работоспособности диспенсера купюр
69	Диспенсер купюр — Предупреждение	Событие возникает в случае возникновения некритичных ошибок в работе диспенсера купюр. В

		комментарии к событию отображаются коды предупреждений
70	Диспенсер монет — Ошибка	Событие возникает в случае возникновения ошибок, приводящих к невозможности работы диспенсера монет. В комментарии к событию отображаются коды ошибок
71	Диспенсер монет — Работоспособность восстановлена	Событие возникает в случае восстановления работоспособности диспенсера монет
72	Диспенсер монет — Предупреждение	Событие возникает в случае возникновения некритичных ошибок в работе диспенсера монет. В комментарии к событию отображаются коды предупреждений
73	Карт-ридер — Ошибка	Событие возникает в случае возникновения ошибок, приводящих к невозможности работы карт-ридера. В комментарии к событию отображаются коды ошибок
74	Карт-ридер — Работоспособность восстановлена	Событие возникает в случае восстановления работоспособности карт-ридера
75	Карт-ридер — Предупреждение	Событие возникает в случае возникновения некритичных ошибок в работе карт-ридера. В комментарии к событию отображаются коды предупреждений
76	Пинпад — Ошибка	Событие возникает в случае возникновения ошибок, приводящих к невозможности работы пинпада. В комментарии к событию отображаются коды ошибок
77	Пинпад — Работоспособность восстановлена	Событие возникает в случае восстановления работоспособности пинпада
78	Пинпад — Предупреждение	Событие возникает в случае возникновения некритичных ошибок в работе пинпада. В комментарии к событию отображаются коды предупреждений

79	Платеж заблокирован	Доступно в системах с установленным модулем Fraud-контроля. Для платежей в заблокированном состоянии, по которым еще не было отправлено уведомление, генерируется событие с типом «Платеж заблокирован». Для одной операции событие формируется один раз. Анализ наличия платежей в заблокированном состоянии осуществляется 1 раз в 30 секунд
80	Оператор сотовой связи не сопоставлен	При формировании обновленной базы соответствия наименования сотового оператора символьному коду ( <i>operators-matching.csv</i> ) в случае, если найдены не сопоставленные операторы, генерируется данное событие
81	Провайдер отключен динамической системой маршрутизации	Событие генерируется, когда провайдер отключается согласно настройкам модуля динамической маршрутизации, например, при снижении баланса провайдера или получении от провайдера определенной ошибки
82	Провайдер включен динамической системой маршрутизации	Событие генерируется, когда провайдер включается после отключения согласно настройкам модуля динамической маршрутизации
83	Создана заявка на отмену операции	При создании заявки на отмену автоматически создается событие данного типа. Атрибуты события: идентификатор отменяемой операции, id1, комментарий к заявке на отмену (при наличии)
84	Большое количество сетевых ошибок по провайдеру	Событие информирует о некорректном поведении шлюза. Событие генерируется раз в 3 часа в ситуации, когда процент сетевых ошибок за текущий день превышает 70%
85	Устройство находится на сервисном обслуживании	Возникает, когда устройство переходит в статус "SERVICING", например, при обновлении купюроприемника
86	Пополнение баланса	Событие генерируется для тех провайдеров, чей

	провайдера	баланс контролируется системой. Событие генерируется, если баланс провайдера с момента последнего запроса баланса изменился в большую сторону. Комментарий к событию: «Создано перечисление (ID = <ID>) провайдеру <ID провайдера> на сумму <сумма>»
87	Проблемы с безопасностью	Событие генерируется, если определены некоторые проблемы безопасности на терминале
88	Лицо, причастное к экстремистской/террористической деятельности	Событие генерируется, если на сервисе настроена проверка через модуль предварительной проверки по базе террористов, или шлюз упрощенной идентификации, или шлюз, выполняющий проверку по advanced-запросам
89	Некорректная инкассация	Поддерживается только ТПО 5. Событие генерируется, когда при синхронизации удаленной инкассации невозможно обновить счетчики купюр для терминала на сервере
90	Новая удаленная инкассация	Поддерживается только ТПО 5. Событие генерируется, когда ТПО получена инкассация с сервера
91	Удаленная инкассация обработана	Поддерживается только ТПО 5. Событие генерируется, когда ТПО обработана инкассация, полученная с сервера
92	Новое удаленное внесение	Поддерживается только ТПО 7. Событие отправляется терминалом после того, как будет принято техником. В комментариях может быть указано как success (количество купюр в боксах в кабинете увеличится), так и по различным причинам failed, тогда количество купюр в боксах в кабинете не изменится
96	Удаленное внесение обработано	Поддерживается только ТПО 7. Событие генерируется, когда ТПО приняло серверное внесение
97	Неизвестное событие	Если ТПО отправлено событие с типом, неизвестным коннектору, то создается событие с типом «Неизвестное событие», комментарий сохраняется

---

98	СКНО — потеря связи	Событие генерируется, если на терминале используется СКНО, а связь с СКНО не удалось установить и результат проверки связи отличается от предыдущего
99	СКНО — связь восстановлена	Событие генерируется, если на терминале используется СКНО, а связь с СКНО удалось установить, и результат проверки связи отличается от предыдущего
100	СКНО — терминал заблокирован	Событие генерируется, если на терминале используется СКНО, а связь с СКНО не удалось установить в течение 30 минут и терминал в результате заблокировался
101	Заявка на инкассацию создана	Событие генерируется, если в системе еще нет заявки на инкассацию некоторого же бокса некоторого устройства некоторой точки, а наполненность этого бокса превышает заданный порог
102	Невозможно создать заявку на инкассацию - отсутствуют или не настроены инкассаторы	Событие генерируется, если не найдена персона или email для отправки заявки на инкассацию

**A.7 КОДЫ ОШИБОК ПРОВАЙДЕРА ПЛАТЕЖНОГО ИНСТРУМЕНТА**

Коды ошибок провайдера платежного инструмента приведены в таблице A.7.1.

Таблица A.7.1 — Коды ошибок провайдера платежного инструмента

<b>Статус</b>	<b>Описание</b>
1	Внутренняя ошибка процессинга (исключение и т.п.)
2	Не задан код платёжного инструмента
3	Не найден платёжный инструмент по коду
4	Не найдено направление проведения платёжного инструмента
5	Таймаут обработки операции (операция долго находилась в нефинальном статусе)
6	Недостаточно средств (на карте, на счёте плательщика)
7	Операция запрещена провайдером
8	Внутренняя ошибка на стороне провайдера (результат операции неизвестен)
9	Авторизация не выполнена
10	Списание не выполнено
11	Списание или авторизация в процессе отмены
12	Списание или авторизация отменена
13	Реверс не выполнен
14	Авторизованная транзакция не найдена (при списании, при реверсе)

## ПРИЛОЖЕНИЕ В. ГЕНЕРАЦИЯ И ПРОВЕРКА ЭП

### В.1 ПРИМЕРЫ ГЕНЕРАЦИИ И ПРОВЕРКИ ЭП НА JAVA

Пример генерации и проверки подписи в формате RSA:

```
/**
 * Подписывает строку
 * @param message - Строка для подписи
 * @return подпись в Base64
 */
public String sign(String message) throws SignatureException{
    try {
        Signature sign = Signature.getInstance("SHA1withRSA");
        sign.initSign(privateKey);
        sign.update(message.getBytes("UTF-8"));
        return new String(Base64.encodeBase64(sign.sign()), "UTF-8");
    } catch (Exception ex) {
        throw new SignatureException(ex);
    }
}

/**
 * Проверяет подпись
 * @param message строка для проверки
 * @param signature подпись в Base64
 * @return true если подпись верна
 * @throws java.security.SignatureException
 */
public boolean verify(String message, String signature) throws
SignatureException{
    try {
        Signature sign = Signature.getInstance("SHA1withRSA");
        sign.initVerify(publicKey);
        sign.update(message.getBytes("UTF-8"));
        return sign.verify(Base64.decodeBase64(signature.getBytes("UTF-8")));
    } catch (Exception ex) {
        throw new SignatureException(ex);
    }
}
```

---

**Пример генерации и проверки подписи в формате ГОСТ Р 34.10-2012:**

```
// Используемые классы
import ru.CryptoPro.JCP.JCP;
import ru.CryptoPro.JCP.Key.GostPublicKey;
import ru.CryptoPro.JCP.tools.Control.KeyUsageControlUtility;

import java.nio.charset.Charset;
import java.security.*;

/**
 * Подписывает строку
 * @param message - Строка для подписи
 * @return подпись в Base64
 */
public String sign(String message) throws AuthException {
    try {
        String signatureAlgorithm =
getSignatureAlgorithm(privateKey.getAlgorithm());
        Signature signature = Signature.getInstance(signatureAlgorithm,
JCP.PROVIDER_NAME);
        signature.initSign(privateKey);
        signature.update(message.getBytes(charset));
        return Base64.encodeBase64String(signature.sign());
    } catch (Exception ex) {
        throw new AuthException("Ошибка формирования подписи: " + ex,
ex);
    }
}

/**
 * Проверяет подпись
 * @param message строка для проверки
 * @param authParam подпись в Base64
 * @return true, если подпись верна
 * @throws java.security.SignatureException
 */
public boolean verify(String message, String authParam) throws
SignatureException {
    try {
        String signatureAlgorithm =
getSignatureAlgorithm(publicKey.getAlgorithm());
```

```
        Signature signature = Signature.getInstance(signatureAlgorithm,
JCP.PROVIDER_NAME);
        signature.initVerify(publicKey);
        signature.update(message.getBytes(charset));
        return signature.verify(Base64.decodeBase64(authParam));
    } catch (Exception ex) {
        throw new SignatureException("Signature verification error", ex);
    }
}

/**
 * Определяет алгоритм подписи на основе алгоритма ключа.
 * @param keyAlgorithm - название алгоритма ключа
 * @return название алгоритма подписи
 */
private String getSignatureAlgorithm(String keyAlgorithm) {
    switch (keyAlgorithm) {
        case JCP.GOST_EL_2012_512_NAME:
        case JCP.GOST_DH_2012_512_NAME:
            return JCP.CRYPTOPRO_SIGN_2012_512_NAME;
        case JCP.GOST_EL_2012_256_NAME:
        case JCP.GOST_DH_2012_256_NAME:
            return JCP.CRYPTOPRO_SIGN_2012_256_NAME;
        case JCP.GOST_EL_DEGREE_NAME:
        case JCP.GOST_EL_DH_NAME:
            return JCP.CRYPTOPRO_SIGN_NAME;
        default:
            throw new IllegalArgumentException("Key algorithm not
supported: " + keyAlgorithm);
    }
}
}
```

## В.2 ПРИМЕР ГЕНЕРАЦИИ И ПРОВЕРКИ ЭП НА PHP

```
/**
 * @return string
 * @param string $msg
 * @param string $Message
 * @desc подписывает $Message и возвращает электронную подпись. В случае
 * ошибки генерирует исключение
 */
public function Sign($Message) {
    $Signature='';
    $KeyId = openssl_get_privatekey($this->PrivateKey,$this-
    >PrivateKeyPass);
    $Res = openssl_sign($Message, $Signature, $KeyId);
    openssl_free_key($KeyId);
    if (!$Res)
        throw new Exception("Sign error!");
    return base64_encode($Signature);
}
/**
 * @return none
 * @param string $Message
 * @param string $Digest
 * @desc проверяет соответствие сообщения $Message и подписи $Digest,
 * генерирует исключение в случае неудачной проверки
 */
public function Verify($Message,$Signature)
{
    $KeyHash = openssl_get_publickey($this->PublicKey);
    $Res=openssl_verify($Message, base64_decode($Signature), $KeyHash);
    openssl_free_key($KeyHash);
    if ($Res!=1)
        throw new Exception("Sign verify error");
}
```

**ПРИЛОЖЕНИЕ С. ПАРАМЕТРЫ КОННЕКТОРА**

Список параметров коннектора приведен в таблице С.1. Данные параметры задаются в конфигурационном файле коннектора — `/srv/glassfish3/config.xml`.

Таблица С.1 — Параметры коннектора

Параметр	Описание
<code>&lt;app_dir&gt;...&lt;/app_dir&gt;</code>	Путь к директории приложения. Относительно данной директории задаются пути директорий сценариев и форм
<code>&lt;forms&gt;...&lt;/forms&gt;</code>	Путь к директории с формами (относительно директории приложения)
<code>&lt;scenarios&gt;...&lt;/scenarios&gt;</code>	Путь к директории со сценариями (относительно директории приложения)
<code>&lt;private-key&gt;...&lt;/private-key&gt;</code>	Закрытый ключ сервера, который используется для подписи запросов
<code>&lt;public-key&gt;...&lt;/public-key&gt;</code>	Открытый ключ сервера, который используется для подписи запросов
<code>&lt;cyber-private-key&gt;...&lt;/cyber-private-key&gt;</code>	Закрытый ключ для веб-сервиса, работающего по протоколу Киберплат
<code>&lt;cyber-private-password&gt;...&lt;/cyber-private-password&gt;</code>	Пароль закрытого ключа для веб-сервиса, работающего по протоколу Киберплат
<code>&lt;input-model-scheduler-enable&gt;...&lt;/input-model-scheduler-enable&gt;</code>	Позволяет включить загрузку модели форм
<code>&lt;use-file-form-loader&gt;...&lt;/use-file-form-loader&gt;</code>	Позволяет использовать файловый загрузчик для загрузки модели форм
<code>&lt;paylogic-encode-cyrillic&gt;...&lt;/paylogic-encode-cyrillic&gt;</code>	Возможные значения:

Параметр	Описание
	<ol style="list-style-type: none"><li><b>true</b> — позволяет экранировать кириллицу в xml-файле.</li><li><b>false</b> — экранируются только символы, которые не являются алфавитными.</li></ol>
<code>&lt;paylogic-message-in-response&gt;...&lt;/paylogic-message-in-response&gt;</code>	При установлении значения <b>true</b> в ответы на запросы проведения и статуса платежа будет добавлено поле <b>message</b> с описанием кода ошибки. Для этого необходимо указать: <pre>&lt;paylogic-message-in response&gt; true &lt;/paylogic-message-in-response&gt;</pre>
<code>&lt;paylogic-verify-separate-key-title&gt;...&lt;/paylogic-verify-separate-key-title&gt;</code>	Позволяет отключить замену <code>key</code> на <code>keyTitle</code> и <code>value</code> на <code>valueTitle</code> в ответе на запрос онлайн-проверки по протоколу <code>paylogic2</code> . <pre>&lt;paylogic-verify-separate-key- title&gt; true &lt;/paylogic-verify-separate-key- title&gt;</pre>
<b>dynamicRequestUrl</b>	URL-адрес движка динамических форм. Пример задания параметра: <pre>dynamicRequestUrl = getStrParam("/config/dynamic- request-url", doc)</pre>
<code>&lt;dynamic-request-protocols&gt;...&lt;/dynamic-request-protocols&gt;</code>	Протоколы, поддерживаемые обработчиком динамических форм. По умолчанию задано значение <b>all</b>
<code>&lt;dynamic-request-timeout&gt;...&lt;/dynamic-request-timeout&gt;</code>	Таймаут запроса к движку динамических

Параметр	Описание
	форм. По умолчанию значение равно 60 с
<code>&lt;gates-host&gt;...&lt;/gates-host&gt;</code>	Хост шлюзов. По умолчанию значение равно 127.0.0.1
<code>&lt;duplication-check&gt;...&lt;/duplication-check&gt;</code>	Параметр, позволяющий выполнить контроль дублей запросов. Для включения проверки установите следующее значение параметра: <pre>&lt;duplication-check&gt;   PL_STATUS, PL_VERIFY, PL_ADVANCED,   PL_FORMS, PL_BALANCE &lt;/duplication-check&gt;</pre>
<code>&lt;bank-security-config&gt;...&lt;/bank-security-config&gt;</code>	Путь к конфигурационному файлу <code>log4j.properties</code> с настройками маскирования