



НАЧАЛО РАБОТЫ С XML ПРОТОКОЛОМ

вер. 4.76

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

вер. 3.0

МОСКВА
8-495-783-5959

РОССИЯ
8-800-200-0059

ФАКС
8-495-926-4619

WEB
WWW.QIWI.COM

СОДЕРЖАНИЕ

1.	ГЛОССАРИЙ	3
2.	ВВЕДЕНИЕ.....	4
3.	СТРУКТУРА ПРОТОКОЛА.....	5
3.1.	ФОРМАТ XML-ЗАПРОСА	6
3.2.	ФОРМАТ XML-ОТВЕТА	6
3.3.	ФОРМАТ ДАННЫХ	8
4.	КАК ПОДКЛЮЧИТЬСЯ К СИСТЕМЕ	9
4.1.	СОЗДАНИЕ СЕРТИФИКАТА ДЛЯ ВХОДА В ЛИЧНЫЙ КАБИНЕТ	9
4.2.	АВТОРИЗАЦИЯ В ЛИЧНОМ КАБИНЕТЕ	9
4.3.	СОЗДАНИЕ ТЕРМИНАЛА	10
5.	АВТОРИЗАЦИЯ ПО ЦИФРОВОЙ ПОДПИСИ	15
5.1.	РЕГИСТРАЦИЯ КЛЮЧА ДЛЯ ЦИФРОВОЙ ПОДПИСИ	15
5.2.	ФОРМИРОВАНИЕ ЗАПРОСОВ С ЦИФРОВОЙ ПОДПИСЬЮ.....	16
6.	ПРОВЕДЕНИЕ ПЛАТЕЖЕЙ	19
6.1.	СТАТУСЫ ПЛАТЕЖЕЙ	19
6.2.	СЦЕНАРИИ ПРОВЕДЕНИЯ ПЛАТЕЖЕЙ	20
6.2.1.	Одношаговое проведение платежей.....	21
6.2.2.	Двухшаговое проведение платежей.....	24
7.	ЧАСТО ЗАДАВАЕМЫЕ ВОПРОСЫ.....	27
ПРИЛОЖЕНИЕ А.	КОДЫ ОШИБОК	29
ПРИЛОЖЕНИЕ Б.	ТИПЫ ДЕНЕЖНЫХ СРЕДСТВ (MONEYTYPE).....	34
ПРИЛОЖЕНИЕ В.	ТИПЫ ТЕРМИНАЛОВ.....	35
ПРИЛОЖЕНИЕ Г.	РОЛИ ПЕРСОН.....	36
ПРИЛОЖЕНИЕ Д.	ПЛАТЕЖНЫЕ ЗАПРОСЫ	37
	ADDOFFLINEPAYMENT	37
	CHECKPAYMENTREQUISITES	39
	AUTHORIZEPAYMENT	39
	CONFIRMPAYMENT	40
	GETPAYMENTSTATUS	41
ПРИЛОЖЕНИЕ Е.	ОПИСАНИЕ СЦЕНАРИЯ ОПЛАТЫ	42
	КАТАЛОГ ДОСТУПНЫХ ДЛЯ ОПЛАТЫ УСЛУГ.....	42
	Отображение каталога услуг и сценария оплаты.....	43
	Отображение страниц интерфейса оплаты провайдера.....	44
	Дополнительные параметры интерфейса провайдера.....	46
	СПИСОК РИСУНКОВ	48
	СПИСОК ТАБЛИЦ.....	48
	СПИСОК ПРИМЕРОВ.....	48

1. ГЛОССАРИЙ

Табл. 1. Глоссарий

Термин	Определение
<i>Агент</i>	Юридическое лицо или индивидуальный предприниматель, осуществляющие предпринимательскую деятельность, осуществляющие техническое взаимодействие с QIWI в рамках настоящего Протокола, подписавшее Договор о приеме Платежей/договор об осуществлении Переводов/иной вид договора.
<i>Персона</i>	Сотрудник агента: менеджер, кассир, бухгалтер и т.п. – человек, который непосредственно производит разрешенные ему действия (в соответствии с назначенной ему ролью) в <i>Системе</i> от имени агента
<i>Право</i>	Разрешение на выполнение XML-запроса. Права группируются в роли
<i>Роль</i>	Набор прав на выполнение действий с системой. Каждому пользователю системы назначена роль
<i>Поставщик услуг (Провайдер)</i>	Коммерческая организация или индивидуальный предприниматель, предоставляющие потребителям услуги, продающие товары от собственного имени, а также благотворительная организация, созданная для осуществления благотворительной деятельности
<i>Система</i>	Совокупность программно-аппаратных средств QIWI, предназначенная для обеспечения приема и проведения платежей/переводов за услуги Провайдеров.
<i>Терминал</i>	Программно-технические комплексы различных типов, в том числе POS-терминалы (специализированные прикассовые устройства типа Point Of Sale), стационарное оборудование, мобильные устройства карманного типа, переносные терминалы и кассовые аппараты, автоматы самообслуживания, банкоматы, каналы дистанционного банковского обслуживания (интернет-банк, мобильный банк), банковский сервис «Автоплатеж».
<i>Протокол</i>	Набор правил обмена данными между Терминалом и процессингом QIWI для приема платежей, получения сценариев оплаты услуг, формирования отчетности и др.

2. ВВЕДЕНИЕ

Данный документ содержит базовое описание протокола взаимодействия, порядка регистрации в Системе, запросов и ответов, необходимых для проведения платежей, а также ответы на часто задаваемые вопросы.

Чтобы начать принимать платежи через QIWI, необходимо выполнить следующие действия:

1. Ознакомиться с принципами организации протокола взаимодействия и его структурой.
2. Зарегистрировать в Системе персону и терминал для формирования платежей.
3. Создать пару RSA-ключей для подписывания xml-запросов и передать публичный ключ в QIWI.
4. Научиться формировать запросы с использованием ЭЦП.
5. Реализовать логику отправки запросов на проведение платежей.
6. Реализовать логику проверки статуса отправленных платежей.
7. Ознакомиться с ответами на часто задаваемые вопросы.

3. СТРУКТУРА ПРОТОКОЛА

Передача данных между клиентским и серверным ПО построена на основе протокола **HTTP** (RFC 2616: HTTP/1.1). Для передачи данных используется метод **RAW POST** – поток данных отправляется в теле запроса от клиента на сервер. Данные передаются в формате XML (*Extensible Markup Language (XML) 1.0*).

ПРИМЕЧАНИЕ



Клиент имеет возможность передавать запросы в сжатом виде и получать сжатые ответы. Для сжатия используется алгоритм **gzip**.

При передаче сжатых запросов в заголовке HTTP-пакета должна содержаться запись "Content-encoding: gzip".

Для получения ответов в сжатом виде клиент должен указать в заголовке: "Accept-encoding: gzip".

Взаимодействие терминала и процессинга происходит по следующему алгоритму:

1. Терминал формирует [запрос в формате XML](#), включая в него необходимые действия, подписывает запрос, добавляет HTTP-заголовки для авторизации, и передает его по протоколу HTTP на сервер.
2. Сервер осуществляет [авторизацию](#) и разбор XML.
3. В случае успешной авторизации сервер выполняет действия.
4. Сервер возвращает клиенту [ответ в формате XML](#), в том числе результат выполнения запроса (код ошибки).

Отправляемый терминалом запрос состоит из двух частей:

Пример 1. Пример запроса

```
POST /XMLgate/XML.jsp HTTP/1.0
Connection: keep-alive
Content-Type: text/XML
Content-Length: 249
Host: xml1.qiwi.com
Accept: text/html, */*
Accept-Encoding: identity
User-Agent: Dealer v0

<?xml version="1.0" encoding="utf-8"?>
<request>
...
</request>
```

HTTP-заголовки.

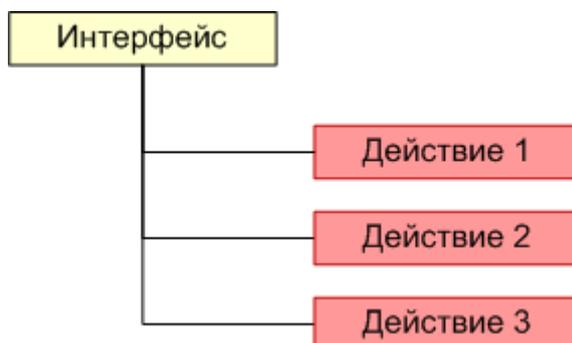
Помимо стандартных HTTP-заголовков в целях авторизации по цифровой подписи используются специальные заголовки, описание которых будет представлено ниже (см. [5.2](#)).

XML-запрос.

Реализация тела XML-протокола основана на интерфейсах и действиях.

- **Интерфейсы (interface)** – группы возможных действий (аналогичны классам объектов в ООП). Каждый интерфейс объединяет несколько действий (см. [Рис. 1](#));
- **Действия (action)** – действия, выполняемые в *Системе* (аналогичны методам объектов в ООП).

Рис. 1. Структура интерфейса



3.1. Формат XML-запроса

В общем виде запрос выглядит следующим образом:

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <client terminal="123" serial="123456"/>
  <interfacel>
    <action1>
      Параметры действия
    </action1>
    <action2>
      Параметры действия
    </action2>
  </interfacel>
  <interface2>
    <action3>
      Параметры действия
    </action3>
  </interface2>
</request>
  
```

где:

- request – корневой тег запроса:
 - client – тег содержит информацию о терминале, с которого отправлен запрос:
 - ⊕ terminal – идентификатор терминала;
 - ⊕ serial – серийный номер терминала (указывается, если он задан в Личном кабинете);
 - interfacel, interface2 – теги интерфейсов. Содержат теги действий:
 - ⊕ action1, action2, action3 – теги действий. Содержат параметры (атрибуты и вложенные теги) действий.

3.2. Формат XML-ответа

В общем виде ответ сервера выглядит следующим образом:

```

<?xml version="1.0" encoding="utf-8"?>
<response result="0">
  <interfacel>
  
```

```

<action1 result="0" result-description="описание ошибки">
  Параметры действия
</action1>
<action2 result="0" result-description="описание ошибки">
  Параметры действия
</action2>
</interface1>
<interface2>
  <action3 result="0" result-description="описание ошибки">
    Параметры действия
  </action3>
</interface2>
</response>

```

где:

- response – корневой тег ответа:
 - result – атрибут отражает успешность/неуспешность выполнения запроса в целом и содержит код ошибки.

ПРИМЕЧАНИЕ



Если ошибок обработки не выявлено, атрибут result возвращает значение "0".

Если запрос не прошел первичную проверку, ответ от сервера содержит ненулевой код ошибки, а в теге response отображается текстовое описание ошибки. При этом будут отсутствовать теги интерфейсов и действий.

Наиболее часто встречающиеся ошибки первичной проверки:

- <response result="202"> – неверно составлен xml запрос.
 - Исправьте причину и повторно отправьте запрос.
 - Возможные причины возникновения ошибки:
 - ❖ отсутствие логина и/или подписи в заголовках запроса;
 - ❖ отсутствие открывающего и/или закрывающего тега;
 - ❖ отсутствуют обязательные параметры;
- <response result="150"> – указанная подпись не соответствует введенному логину.
 - Проверьте корректность логина и/или подписи в заголовках запроса.
- <response result="13"> – процессинг загружен.
 - Отправьте повторно тот же самый запрос через минуту.
- interface1 – тег интерфейса. Содержит теги действий:
 - action1 – тег действия. Содержит параметры действия:
 - ⊕ result – атрибут возвращает код результата выполнения действия (код ошибки). Перечень наиболее распространенных кодов ошибок и рекомендуемые действия по их устранению см. в [Приложении А](#). Полный список кодов ошибок содержится в [Протоколе взаимодействия терминального ПО и процессинга](#).

ПРИМЕЧАНИЕ



Если ошибок при выполнении действия не возникло, атрибут result возвращает значение "0", а в ответе будет отсутствовать атрибут result-description.



Коды ошибок подразделяются на фатальные и не фатальные:

- Фатальная ошибка – отрицательный ответ на клиентский запрос. Все ошибки первичной обработки являются фатальными.
- Не фатальная ошибка – промежуточный статус обработки клиентского запроса, дальнейшие действия зависят от конкретной ошибки.

Наиболее часто встречающиеся ошибки:

- ❖ `<action1 result="133"...>` – нет прав на прием платежей.
[Роль персоны](#), использованная для авторизации, не имеет прав на выполнение данного XML запроса.
- ❖ `<action1 result="202"...>` – ошибка данных запроса.
Проверьте корректность параметров данного действия.
- ❖ `<action1 result="212"...>` – не задана сумма платежа.
Укажите сумму платежа.
- ❖ `<action1 result="295"...>` – ошибка в названии действия или интерфейса.
Проверьте наименование интерфейса / действия.

⊕ `result-description` – атрибут содержит краткое описание произошедшей ошибки.

3.3. Формат данных

Данные в тегах и атрибутах представляются следующим образом:

- Идентификатор платежа – натуральное число длиной до 18 цифр.
- Целые числа (в основном, идентификаторы) представляются в десятичном виде. Например, 1234.
- Суммы платежей представляются в основных единицах валюты.

Например, в рублях: в десятичном виде с двумя знаками после разделителя. Разделителем является точка.

- Кодировка строк должна соответствовать значению атрибута `encoding` тега `xml`.
- Дата и время представляются в соответствии с **ISO 8601:2004**. (Используется следующий формат: "ГГГГ-ММ-ДДТчч:мм:сс", однако может быть использован и другой, соответствующий стандарту).
- При описании логических значений истине соответствует значение `true`, лжи – `false`.
- Бинарные данные должны кодироваться с использованием схемы **Base64**.
- Коды валют представляются как целые числа. Значения кодов соответствуют: **ISO 4217**. Общий справочник кодов валют можно посмотреть [здесь](#).

Российскому рублю соответствует значение 643.

4. КАК ПОДКЛЮЧИТЬСЯ К СИСТЕМЕ

Техническое подключение к Системе выполняется после подписания соответствующего договора между Агентом и QIWI (о приеме платежей/осуществлении переводов/на оказание ИТО-услуг/иных видов договоров).

Для формирования платежей необходимо зарегистрировать хотя бы один терминал в качестве точки оплаты и персону, от имени которой будут проводиться платежи. Эти действия выполняются в Личном кабинете агента.

При активации договора агенту передаются по электронной почте логин и одноразовый пароль персоны для генерации сертификата, необходимого для входа в Личный кабинет агента на сайтах agt.qiwi.com, agent.qiwi.com.

ПРИМЕЧАНИЕ



При регистрации договора создается персона с ролью *Главный менеджер*

4.1. Создание сертификата для входа в Личный кабинет

Генерация сертификата осуществляется при помощи программы QIWI Защита для Windows, которую можно скачать на сайте corp.qiwi.com в разделе **Бизнесу → Действующим агентам → Скачать ПО и документацию → QIWI Защита**.

Для создания сертификата выполните следующие действия:

1. Выберите пункт **Сертификат для доступа на агентский сайт**.
2. Выберите тип агента – агент АО КИВИ.
3. Введите авторизационные данные персоны (**логин и одноразовый пароль для сертификата**).
4. Выберите тип хранилища eToken.
5. Сохраните сертификат в хранилище.

После генерации сертификата необходимо пройти процедуру его активации.

4.2. Авторизация в Личном кабинете

Процедура первичной авторизации в Личном кабинете агента включает несколько этапов:

1. Установите ключ eToken в USB порт компьютера.
2. В браузере введите адрес agt.qiwi.com. Будет открыто окно выбора сертификата.
3. Выберите сертификат *OSMP Agent CA* и нажмите **ОК**.
4. Укажите PIN-код ключа eToken, затем нажмите **ОК**.
5. Подтвердите контактные данные персоны: номер мобильного телефона и электронную почту ([Рис. 2](#))

Рис. 2. Подтверждение контактных данных

Вход на сайт агентов QIWI 

подтвердите свои данные

Владелец сертификата – Фамилия Имя Отчество

1 Мобильный телефон

Отправить код

2 Электронная почта

3 Сертификат доступа

6. Активируйте сертификат, если он не был активирован ранее ([Рис. 3](#)).

Рис. 3. Активация сертификата

Вход на сайт агентов QIWI 

подтвердите свои данные

Владелец сертификата – Фамилия Имя Отчество

✓ Мобильный телефон

+79161234567

✓ Электронная почта

login@mail.ru

3 Сертификат доступа

Отправить SMS с кодом

7. Нажмите кнопку **Отправить SMS с кодом**.
8. Введите код, полученный в SMS-сообщении, в поле **Код подтверждения**.
9. Нажмите кнопку **Подтвердить**. После успешной активации сертификата будет выполнен вход в личный кабинет агента.

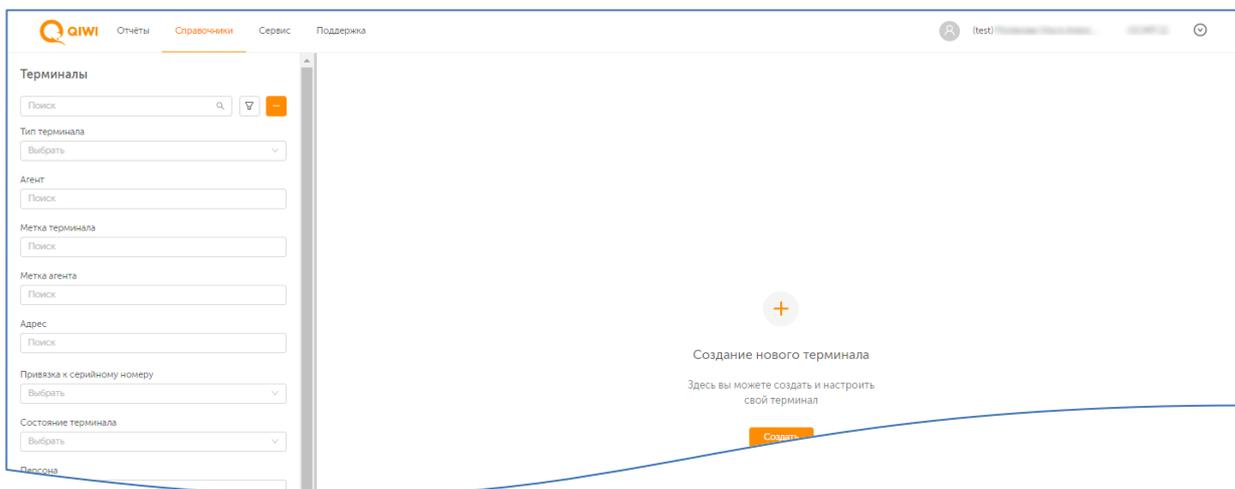
4.3. Создание терминала

Чтобы создать терминал, выполните следующие действия.

1. Авторизуйтесь на сайте agt.qiwi.com.

- Выберите пункт меню **Справочники** → **Терминалы**. Откроется справочник **Терминалы** (Рис. 4).

Рис. 4. Справочник «Терминалы»



- Нажмите кнопку **Создать** в центре экрана или  на панели слева. На экран будет выведена форма создания нового терминала (Рис. 5).

Рис. 5. Создание нового терминала

Форма разделена на вкладки, которые можно заполнять в любом порядке. Не заполнив обязательные поля формы, сохранить новый терминал нельзя. Заполните поля на вкладках данными нового терминала.

В правом нижнем углу формы всегда отображаются кнопки **Заккрыть** и **Сохранить**. Нажмите кнопку **Заккрыть**, чтобы очистить форму и выйти из режима создания терминала, или **Сохранить**, чтобы создать терминал с указанными в форме параметрами.

После успешного создания терминала форма будет переведена в режим редактирования терминала, а ему автоматически будет присвоен идентификатор (номер).

С назначением полей и вкладок, не описанных в текущем документе, можно ознакомиться в [Руководстве пользователя agt.qiwi.com](#).

4.3.1.1. Основная информация

Заполните **обязательные** поля на вкладке:

1. **Тип терминала** – выберите подходящий тип XML-терминала из выпадающего списка. Список существующих в системе типов XML-терминалов и их описание см. в [Приложении В](#).
На экране будут отображены дополнительные вкладки для заполнения: [IP-адреса](#), [Персоны](#).
- **Название терминала** – укажите название терминала, в дальнейшем название можно использовать для поиска этого терминала;
- **Адрес терминала** – в соответствии с требованиями ЦБ РФ необходимо указать полный адрес местонахождения терминала вплоть до дома. В случае, если терминал находится вне какого-либо здания, укажите номер ближайшего здания. Адреса указываются в соответствии с ФИАС. В случае, если адрес существует, но не определяется ФИАС, обратитесь за помощью к своему менеджеру, предоставив подтверждение регистрации по этому адресу;

ПРИМЕЧАНИЕ



Для XML-терминалов после ввода адреса терминала под полем с адресом будет отображен соответствующий этому адресу часовой пояс относительно Москвы в формате MSK+Ч

- **Тип местоположения** – выберите из раскрывающегося списка тип местоположения для терминала.

4.3.1.2. IP-адреса

Настройте диапазоны IP-адресов, с которых будут приниматься запросы на проведение платежей.

Рис. 6. Вкладка «IP-адреса»

Создание нового терминала

Основная информация Метки **IP-адреса**

При необходимости укажите диапазоны IP-адресов, с которых возможен приём платежей

с по IPv4 IPv6 ✕

+ Диапазон

Закрыть
Сохранить

1.24.0

Нажмите кнопку **+ Диапазон**, чтобы добавить новый диапазон IP-адресов. Укажите начало и конец диапазона в полях «с» и «по» соответственно, а также версию IP-протокола для этого диапазона, нажав кнопку **IPv4** или **IPv6**. Чтобы добавить конкретный адрес, а не диапазон, укажите этот адрес и в поле «с», и в поле «по». Терминалу можно добавить несколько диапазонов, указав аналогичные данные для каждого из них.

Чтобы удалить лишний диапазон, нажмите кнопку (корзина) в строке с описанием диапазона.

ПРИМЕЧАНИЕ



После сохранения терминала внести изменения в существующие диапазоны IP-адресов нельзя, можно только заменить их на новые.

4.3.1.3. Персоны

Список существующих персон для терминала представлен в виде таблицы (Рис. 7).

Над таблицей размещена кнопка **+ Добавить персону** – эта кнопка открывает форму создания новой персоны для терминала.

Рис. 7. Вкладка «Персоны»

ФИО	Логин	ID	Роли	Статус	Действия
Центральный Офис доп.	N [redacted]	1 [redacted] 4	Продавец	<input checked="" type="checkbox"/>	Сбросить пароль Редактировать
Д. Любовь Фёдоровна	D [redacted]	1 [redacted] 4	Продавец	<input type="checkbox"/>	Сбросить пароль Редактировать

4.3.1.3.1. Создание и редактирование персон

Чтобы создать новую персону, нажмите кнопку **+ Добавить персону** над списком персон. Откроется форма создания персоны.

Укажите данные новой персоны:

- **Логин** – обязательное поле, введенное значение будет проверено на занятость и на соответствие списку допустимых символов;

ПРИМЕЧАНИЕ



Логин может состоять только из букв латинского алфавита, цифр и символов "-._" (дефис, точка, знак подчеркивания).

- **Фамилия (или псевдоним)** – обязательное поле;
- **Имя, Отчество**;
- **Роли** – проставьте флаг у роли «Продавец».

Рис. 8. Форма создания персоны

Новая персона

Логин

Фамилия (или псевдоним)

Имя

Отчество

Роли

Запрет на изменение настроек печати чека в КИВИ Кассир (ID 153)

Запрет на отображение баланса агента в ПО (ID 154)

Продавец (ID 22)

Метки

Создать

Нажмите кнопку **Создать** в правом нижнем углу формы. Новая персона будет создана и привязана к терминалу, пароль будет отправлен на мобильный телефон персоны, осуществляющей изменения.

Чтобы изменить данные существующей персоны, найдите её в списке и нажмите **Редактировать** в столбце **Действия** (см. [Рис. 7](#)). Откроется форма редактирования персоны.

Рис. 9. Форма редактирования данных персоны

Редактирование персоны

ID 1...4

Логин Н...

Пароль Менялся 27 сентября 2018 г., 14:18
[Сбросить пароль](#)

Статус
Активна

Фамилия (или псевдоним)
Центральный Офис доп.

Имя

Отчество

Роли

Запрет на изменение настроек печати чека в КИВИ Кассир (ID 153)

Запрет на отображение баланса агента в ПО (ID 154)

Продавец (ID 22)

Метки

Сохранить

Вы можете сбросить пароль персоны (пароль будет отправлен на мобильный телефон персоны, осуществляющей изменения), изменить статус, ФИО, набор ролей и метки.

Логин существующей персоны поменять нельзя.

Внесите необходимые изменения и нажмите кнопку **Сохранить** в правом нижнем углу формы.

5. АВТОРИЗАЦИЯ ПО ЦИФРОВОЙ ПОДПИСИ

Авторизация в рамках протокола производится по цифровой подписи пакетов данных. Это безопасный метод авторизации, соответствующий рекомендациям ЦБ РФ.

Для авторизации по цифровой подписи (ЦП) требуется выполнить следующие шаги:

1. Регистрация ключа для цифровой подписи (выполняется один раз).
2. Формирование подписи XML-запроса (выполняется для каждого XML-запроса).

5.1. Регистрация ключа для цифровой подписи

Данные действия выполняются однократно для каждой персоны, которую предполагается использовать при авторизации:

1. Формирование пары RSA ключей с помощью клиентского ПО (например, с помощью утилиты OpenSSL). Длина ключа может быть 1024, 2048 или 4096 бит.

Пример 2. Генерация RSA ключей.

В данном примере для генерации пары RSA ключей используется утилита OpenSSL (подробную информацию по установке и использованию данного приложения можно получить на сайте www.openssl.org).

1. Создание закрытого ключа. Выполните команду:

```
openssl genrsa -out private.key 1024
```

Команда выдает закрытый ключ (1024 бит) в файл private.key.

2. Создание открытого ключа в кодировке Base64. Выполните команду:

```
openssl rsa -in private.key -pubout -out public.key
```

Команда выдает открытый ключ в кодировке Base64 в файл public.key.

2. Закрытый ключ сохраняется в файле в локальном хранилище (например, в файловом хранилище или на eToken).
3. Открытый ключ направляется в QIWI с помощью xml-запроса `setPublicKey` интерфейса `persons`. При этом в параметре запроса `store-type` необходимо указать значение 1.

ВНИМАНИЕ



Для выполнения запроса необходимо использовать одноразовый пароль.

Пример 3. Сохранение открытого ключа.

В данном примере для шифрования ключа используется утилита OpenSSL (подробную информацию по установке и использованию данного приложения можно получить на сайте www.openssl.org).

Например, сформируем ключ от имени персоны с логином "person_login" и одноразовым паролем "person_password":

1. Шифруем одноразовый пароль по алгоритму MD5. Для этого поместите пароль в файл pass.txt и выполните команду:

```
openssl dgst -md5 pass.txt -out passmd5.txt
```

Команда выдает зашифрованный пароль в файл passmd5.txt. Получим строку `e8cfd9bdfc718b22489329048555fb320`.

2. Создаем открытый ключ и помещаем его в файл public.key (см. [Пример 2](#)). Получаем строку вида:

```
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDH7Sxozbq5Gkr5r5qIpQEnOXXw9W9JS+hqaQm4K6douY+IDv49iqqTV96U5D7V4Alg7TACpGCs/2XSE6pyGpc50VEYk3jGDFHQcI3lS6kEQLoZ/gDeHVreCRgRJ2ssoAS1B09rS9vpYWNQKgxFvc0RwIf4aAvW5mKoofJ84o1Z4QIDAQAB
```

3. На основании полученных данных формируем запрос setPublicKey:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth login="person_login" sign="e8cfdbdfc718b22489329048555fb320"
    signAlg="MD5"/>
  <client terminal="номер терминала"/>
  <persons>
    <setPublicKey>
      <store-type>1</store-type>
      <pubkey>MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDH7Sxozbq5Gkr5r5qIp
        QEnOXXw9W9JS+hqaQm4K6douY+IDv49iqqTV96U5D7V4Alg7TACpGCs/2X
        SE6pyGpc50VEYk3jGDFHQcI3lS6kEQLoZ/gDeHVreCRgRJ2ssoAS1B09rS9
        vpYWNQKgxFvc0RwIf4aAvW5mKoofJ84o1Z4QIDAQAB</pubkey>
    </setPublicKey>
  </persons>
</request>
```

Ответ приходит в следующем виде:

```
<?xml version="1.0" encoding="utf-8"?>
<response result="0">
  <persons>
    <setPublicKey result="0"/>
  </persons>
</response>
```

где result=0 говорит о том, что запрос выполнен успешно, и вы можете использовать цифровую подпись для авторизации персоны на сервере.

5.2. Формирование запросов с цифровой подписью

Данные действия выполняются для каждого XML-запроса:

1. Подготовить исходный XML-запрос согласно протоколу.

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <client terminal="1234567" serial="123456"/>
  <interface>
    <action/>
  </interface>
</request>
```

2. Вычислить хэш-функцию от текста XML-запроса с использованием алгоритма SHA1 или SHA256, подписать предварительно созданным (см. раздел 5.1) закрытым ключом и кодировать по схеме Base64 (см. Пример 4).

Пример 4. Формирование цифровой подписи XML-запроса.

В данном примере для шифрования ключа используется утилита OpenSSL (подробную информацию по установке и использованию данного приложения можно получить на сайте www.openssl.org).

1. Формируем исходный XML-запрос:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <client terminal="1234567" serial="123456"/>
  <agents>
```

```
<getBalance/>
</agents>
</request>
```

2. Помещаем текст запроса в файле request.txt и выполняем команду (предполагается, что закрытый ключ хранится в файле private.key):

```
openssl dgst -sha1 -out request.sign -sign private.key request.txt
```

где для шифрования используется алгоритм SHA1. В результате выполнения команды получаем ЦП пакета в файле request.sign.

3. Полученную цифровую подпись шифруем по алгоритму Base64 командой:

```
openssl base64 -in request.sign -out request.enc
```

В результате зашифрованная ЦП пакета будет записана в файл request.enc вида:

```
XTC5UVZ3mjXA+uZq5rb76iNK7x41A2S8expSPsYVsD1N1bSheG01tsbuXZJCcPTUr7RCIFZmDbKv
IGD57f8ipPUrtjObMnd02hSQVpO6J0/+V2UdE/aLBKg7e3pyTIBE9gUOW/W8ySEqV3Mg7gd700Pp
WmHv7Uu7jpKcs5a8Jmc=.
```

3. Добавить в заголовок HTTP запроса следующие параметры, необходимые для выполнения авторизации и проверки целостности данных пакета (см. также [Пример 5](#)):
 - "X-Digital-Sign" – ЦП пакета, закодированная по схеме **Base64**;
 - "X-Digital-Sign-Alg" – алгоритм вычисления ЦП. Если применялся SHA1 - укажите SHA1withRSA, если SHA256 - SHA256withRSA соответственно;
 - "X-Digital-Sign-Login" – логин персоны (может быть закодирован по схеме **Base64**).

ВНИМАНИЕ



Так как открытый ключ привязан к персоне, перед отправкой XML-запроса необходимо проверить, что соответствующая персона имеет разрешение на выполнение данного запроса (см. раздел [Роли персон](#)).

Пример 5. Формирование заголовка запроса с цифровой подписью.

Исходный XML-запрос:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <client terminal="1234567" serial="123456" />
  <agents>
    <getBalance/>
  </agents>
</request>
```

Формируем цифровую подпись запроса (см. [Пример 4](#)):

```
XTC5UVZ3mjXA+uZq5rb76iNK7x41A2S8expSPsYVsD1N1bSheG01tsbuXZJCcPTUr7RCIFZmDbKv
IGD57f8ipPUrtjObMnd02hSQVpO6J0/+V2UdE/aLBKg7e3pyTIBE9gUOW/W8ySEqV3Mg7gd700Pp
WmHv7Uu7jpKcs5a8Jmc=
```

1. Добавляем следующие поля в HTTP заголовок запроса:

```
X-Digital-Sign:
XTC5UVZ3mjXA+uZq5rb76iNK7x41A2S8expSPsYVsD1N1bSheG01tsbuXZJCcPTUr7RC
IFZmDbKvIGD57f8ipPUrtjObMnd02hSQVpO6J0/+V2UdE/aLBKg7e3pyTIBE9gUOW/W8ySEqV3Mg7
```

```
gd700PpWmHv7Uu7jpKcs5a8Jmc=  
X-Digital-Sign-Alg: SHA1withRSA  
X-Digital-Sign-Login: person_login
```

т. к. для шифрования пакета использовался алгоритм SHA1 и закрытый ключ персоны с логином person_login.

2. В результате получим следующий запрос (приведены не все HTTP-заголовки) для отправки в процессинг QIWI:

```
X-Digital-Sign: XTC5UVZ3mjXA+uZq5rb76iNK7x41A2S8expSPsYVsD1NlbSheG01tsbuXZJCc  
PTUr7RCIFZmDbKvIGD57f8ipPUrtjObMnd02hSQVpO6J0/+V2UdE/aLBKg7e3pyTIBE9gUOW/W8yS  
EqV3Mg7gd700PpWmHv7Uu7jpKcs5a8Jmc=  
X-Digital-Sign-Alg: SHA1withRSA  
X-Digital-Sign-Login: person_login
```

```
<?xml version="1.0" encoding="utf-8"?>  
<request>  
  <client terminal="1234567" serial="123456" />  
  <agents>  
    <getBalance/>  
  </agents>  
</request>
```

6. ПРОВЕДЕНИЕ ПЛАТЕЖЕЙ

Для проведения платежей в протоколе используются следующие действия интерфейса **providers**:

Табл. 2. Запросы на проведение платежей

Действие	Описание	Режимы обработки
addOfflinePayment	Запрос на проведение платежа (без дополнительного подтверждения)	Строго синхронно
checkPaymentRequisites	Проверка реквизитов платежа. Авторизация платежа без записи информации в базу	Строго синхронно
authorizePayment	Запрос на авторизацию платежа	Строго синхронно
confirmPayment	Запрос позволяет подтвердить платеж, авторизованный в результате выполнения запроса authorizePayment	Строго синхронно
getPaymentStatus	Запрос на получение статуса ранее отправленного платежа	Строго синхронно

Платежные запросы строго идемпотентны. Контролируются следующие параметры:

- `payment id` – идентификатор платежа;
- `terminal` – идентификатор терминала;
- `service` – идентификатор сервиса, услуги которого оплачиваются;
- `account` – номер счета;
- `amount` – сумма к зачислению;

В случае отправки повторного запроса с другим `payment id` будет создан новый платеж. Важно корректно обрабатывать статусы платежей из ответов платежных запросов (атрибут `status`) и принимать решение о повторной отправке запроса с новым `payment id` строго в случае получения финального статуса.

Краткое описание запросов см. в разделе [Платежные запросы](#), подробное – в [Протоколе взаимодействия терминального ПО и процессинга](#).

6.1. Статусы платежей

При выполнении запросов на авторизацию и проведение платежей в теге `payment` ответа возвращаются следующие данные (см. [Пример 6](#)):

- статус платежа (атрибут `status`). Платеж может находиться в одном из следующих статусов:
 - **3** – «авторизован»;
 - **1** – «проводится»;
 - **2** – «проведен» (финальный статус);
 - **0** – «ошибка проведения» (финальный статус).

- код ошибки проведения платежа (атрибут `result`). Перечень наиболее распространенных кодов ошибок и рекомендуемые действия по их устранению см. в [Приложении А](#). Полный перечень кодов ошибок см. в [Протоколе взаимодействия терминального ПО и процессинга](#).
- логический признак фатальности ошибки (атрибут `fatal`). Наличие признака фатальной ошибки (`fatal="true"`) означает отрицательный ответ на клиентский запрос (статус "0"), тогда как нефатальная ошибка (`fatal="false"`) означает промежуточный статус обработки клиентского запроса (дальнейшие действия зависят от конкретной ошибки).

Пример 6. Ответ на запрос авторизации платежа.

```
<?xml version="1.0" encoding="utf-8"?>
<response result="0">
  <providers>
    <authorizePayment result="0">
      <payment ... fatal="false" result="0" status="3" ...>
        ...
      </payment>
    </authorizePayment>
  </providers>
</response>
```

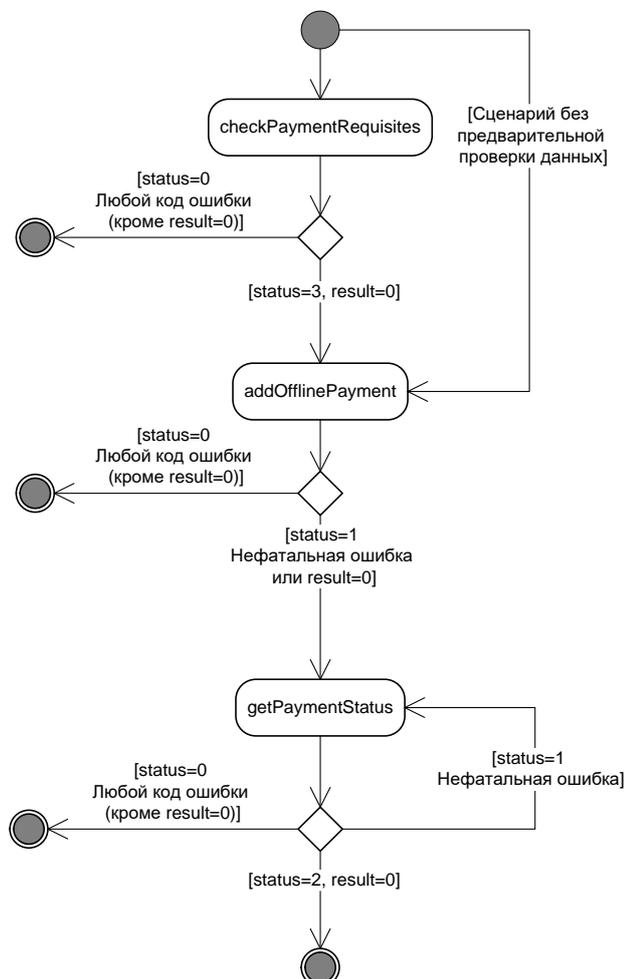
6.2. Сценарии проведения платежей

В *Системе* поддерживаются два метода проведения платежей:

1. [одношаговый \(без дополнительного подтверждения\)](#),
2. [двухшаговый \(авторизация + подтверждение платежа\)](#).

6.2.1. Одношаговое проведение платежей

Рис. 10. Сценарий одношагового проведения платежа



1. Сценарий может выполняться как с предварительной проверкой реквизитов платежа, так и без нее. В первом случае с терминала отправьте запрос **providers.checkPaymentRequisites**, во втором – см. п. 5 ниже.

ВНИМАНИЕ



Предварительная проверка реквизитов требуется, если в [интерфейсе провайдера](#) присутствует страница с атрибутом `useOnline="true"`, например: `<page pageId="36930" orderId="3" nextPage="-1" useOnline="true">`.

В запросе **checkPaymentRequisites** укажите параметры платежа:

- идентификатор платежа на терминале;
- данные о принятых от клиента денежных средствах;
- данные о зачисляемых на счет клиента у провайдера денежных средствах (в том числе идентификатор услуги в QIWI);

- информация о предчеке (при наличии);
 - значения дополнительных (экстра) полей, необходимые для корректной обработки запроса.
2. Сервер проверяет реквизиты платежа, однако не записывает данные в базу.
 3. Сервер отправляет реквизиты платежа поставщику услуг.
 4. **Ответ** от сервера может включать информацию о результатах проверки реквизитов, а также некоторые дополнительные поля, которые необходимо отображать пользователю:
 - данные об успешности выполнения запроса;
 - данные о платеже, реквизиты которого были отправлены на проверку:
 - ⊕ московские дата и время приема платежа в процессинг;
 - ⊕ идентификатор платежа на терминале;
 - ⊕ [код ошибки](#) обработки платежа;
 - ⊕ статус успешной проверки (`status="3"`);

ВНИМАНИЕ

В случае получения финального статуса `status="0"` необходимо рассмотреть код ошибки для выяснения причины ошибки проверки реквизитов.

- ⊕ идентификатор транзакции в процессинге;
 - данные в дополнительных (экстра) полях.
5. В случае успешной авторизации с терминала (возвращен статус `status="3"`) отправьте запрос **providers.addOfflinePayment**. В запросе укажите параметры платежа:
 - идентификатор платежа на терминале;
 - данные о принятых от клиента денежных средствах;
 - данные о зачисляемых на счет клиента у провайдера денежных средствах (в том числе идентификатор услуги в QIWI);
 - информация о чеке (при наличии);
 - значения дополнительных (экстра) полей, необходимые для корректной обработки запроса.
 6. **Ответ** от сервера содержит информацию о результате записи платежа в базу, а также уникальный идентификатор транзакции в *Системе* (в случае успешной записи, т.е. если ответ содержит атрибут `saved="true"`):
 - данные об успешности выполнения запроса;
 - данные о проводимом платеже:
 - ⊕ московские дата и время приема платежа в процессинг;
 - ⊕ идентификатор платежа на терминале;
 - ⊕ [код ошибки](#) обработки платежа;
 - ⊕ статус проведения платежа;

ВНИМАНИЕ

В случае получения финального статуса `status="0"` необходимо рассмотреть код ошибки для выяснения причины ошибки проведения платежа.

- ⊕ идентификатор транзакции в процессинге;
 - данные в дополнительных (экстра) полях.
7. Сервер отправляет реквизиты платежа поставщику услуг.
 8. В случае получения промежуточного статуса `status="1"` необходимо повторно запросить статус платежа для получения информации о результате проведения платежа:
 - 8.1. С терминала отправьте запрос **providers.getPaymentStatus**. В параметрах запроса необходимо указать тот же идентификатор платежа на терминале, что и в запросе **addOfflinePayment** (`<payment id="7473517739148">`).
 - 8.2. **Ответ** от сервера включает информацию о статусе платежа.

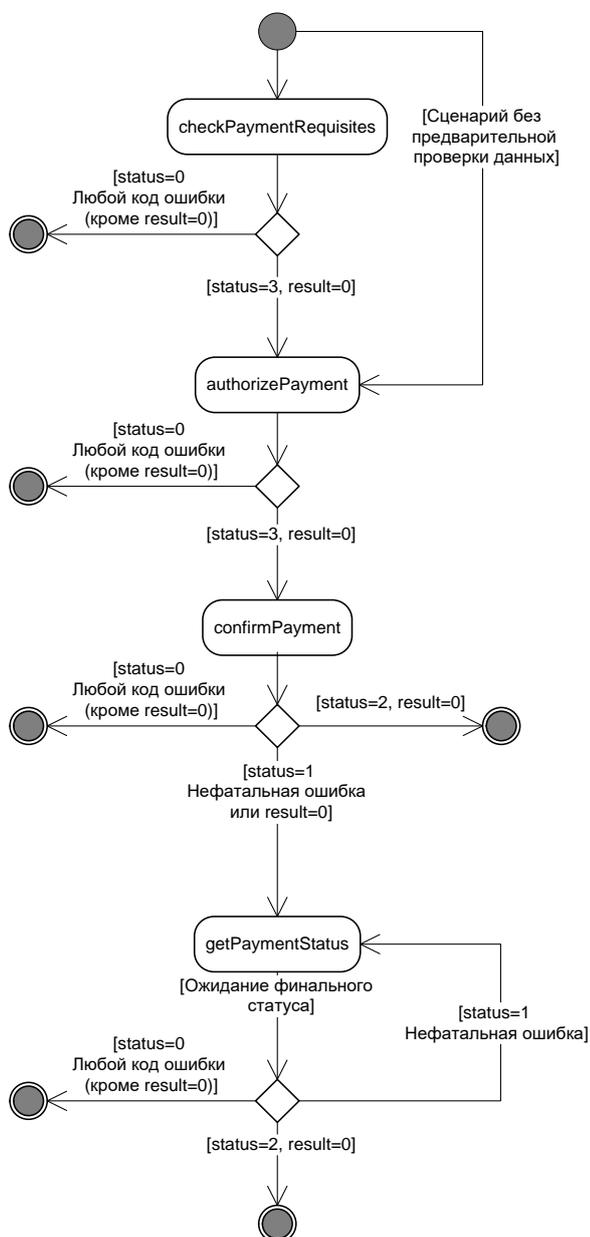
ВНИМАНИЕ

Запрос статуса платежа (`providers.getPaymentStatus`) необходимо отправлять на сервер с интервалом 1 сек. в течение первых 30 сек. с момента получения ответа на платежный запрос, далее необходимо увеличить интервал опроса до 1 минуты и продолжать опрос следующие 10 минут, затем увеличить интервал до 30 минут. При отсутствии финального статуса по истечении 60 минут необходимо обратиться в техническую поддержку.

Запрос необходимо отправлять до тех пор, пока не будет получен финальный статус платежа (обычно в течение 24 часов). До получения финального статуса платежа вывод о том, проведен платеж или нет, сделать нельзя.

6.2.2. Двухшаговое проведение платежей

Рис. 11. Сценарий двухшагового проведения платежа



1. Сценарий может выполняться как с предварительной проверкой реквизитов платежа, так и без нее. В первом случае с терминала отправьте запрос **providers.checkPaymentRequisites**, во втором – см. п. 5 ниже.

ВНИМАНИЕ



Предварительная проверка реквизитов требуется, если в [интерфейсе провайдера](#) присутствует страница с атрибутом `useOnline="true"`, например: `<page pageId="36930" orderId="3" nextPage="-1" useOnline="true">`.

В запросе **checkPaymentRequisites** укажите следующие параметры:

- идентификатор платежа на терминале;
 - данные о принятых от клиента денежных средствах;
 - данные о зачисляемых на счет клиента у провайдера денежных средствах (в том числе идентификатор услуги в QIWI);
 - информация о предчеке (при наличии);
 - значения дополнительных (экстра) полей, необходимые для корректной обработки запроса.
2. Сервер проверяет реквизиты платежа, однако не записывает данные в базу.
 3. Сервер отправляет реквизиты платежа поставщику услуг.
 4. **Ответ** от сервера содержит информацию о результатах проверки реквизитов, а также некоторые дополнительные поля, которые необходимо отображать пользователю:
 - данные об успешности выполнения запроса;
 - данные о платеже, реквизиты которого были отправлены на проверку:
 - ⊕ московские дата и время приема платежа в процессинг;
 - ⊕ идентификатор платежа на терминале;
 - ⊕ [код ошибки](#) обработки платежа;
 - ⊕ статус успешной проверки (`status="3"`);

ВНИМАНИЕ

В случае получения финального статуса `status="0"` необходимо рассмотреть код ошибки для выяснения причины ошибки проверки реквизитов.

- ⊕ идентификатор транзакции в процессинге;
 - данные в дополнительных (экстра) полях.
5. С терминала отправьте запрос авторизации платежа **providers.authorizePayment**.
В запросе **authorizePayment** укажите следующие параметры:
 - идентификатор платежа на терминале;
 - данные о принятых от клиента денежных средствах;
 - данные о зачисляемых на счет клиента у провайдера денежных средствах (в том числе идентификатор услуги в QIWI);
 - информация о чеке (при наличии);
 - значения дополнительных (экстра) полей, необходимые для корректной обработки запроса.
 6. Выполняется авторизация платежа у поставщика услуг.
 7. **Ответ** от сервера содержит информацию о результате авторизации платежа у поставщика услуг, уникальный идентификатор транзакции в *Системе*, а также некоторые дополнительные поля, которые необходимо отображать пользователю:
 - данные об успешности выполнения запроса;
 - данные о платеже, реквизиты которого были отправлены на проверку:
 - ⊕ московские дата и время приема платежа в процессинг;
 - ⊕ идентификатор платежа на терминале;

- ⊕ [код ошибки](#) обработки платежа;
- ⊕ статус успешной проверки (`status="3"`);

ВНИМАНИЕ

В случае получения финального статуса `status="0"` необходимо рассмотреть код ошибки для выяснения причины ошибки авторизации.

- ⊕ идентификатор транзакции в процессинге;
 - данные в дополнительных (экстра) полях.
8. В случае успешной авторизации с терминала (возвращен статус `status="3"`) отправьте запрос на проведение платежа **providers.confirmPayment**.
- В запросе указывается идентификатор платежа на терминале из предыдущего запроса **authorizePayment**, см. выше п. 5.
9. Сервер отправляет платеж поставщику услуг и получает от него ответ.
10. **Ответ** от сервера включает информацию о результате отправки платежа поставщику услуг, а также уникальный идентификатор транзакции в *Системе*:
- данные об успешности выполнения запроса;
 - данные о платеже, реквизиты которого были отправлены на проверку:
 - ⊕ московские дата и время приема платежа в процессинг;
 - ⊕ идентификатор платежа на терминале;
 - ⊕ [код ошибки](#) обработки платежа;
 - ⊕ статус платежа (`status="1"`);

ВНИМАНИЕ

В случае получения финального статуса `status="0"` необходимо рассмотреть код ошибки для выяснения причины ошибки проведения платежа.

- ⊕ идентификатор транзакции в процессинге.
11. В случае получения нефинального статуса `status="1"` необходимо повторно запрашивать статус платежа для получения информации о том, проведен платеж успешно или нет:
- 11.1. С терминала отправьте запрос **providers.getPaymentStatus**. В параметрах запроса необходимо указать идентификатор платежа на терминале из запроса **confirmPayment** (`<payment id="7473517739148">`).
- 11.2. **Ответ** от сервера включает информацию о текущем статусе платежа.

ВНИМАНИЕ

Запрос статуса платежа (`providers.getPaymentStatus`) необходимо отправлять на сервер с интервалом 1 сек. в течение первых 30 сек. с момента получения ответа на платежный запрос, далее необходимо увеличить интервал опроса до 1 минуты и продолжать опрос следующие 10 минут, затем увеличить интервал до 30 минут. При отсутствии финального статуса по истечении 60 минут необходимо обратиться в техническую поддержку.

Запрос необходимо отправлять до тех пор, пока не будет получен финальный статус платежа (обычно в течение 24 часов). До получения финального статуса платежа вывод о том, проведен платеж или нет, сделать нельзя.

7. ЧАСТО ЗАДАВАЕМЫЕ ВОПРОСЫ

Как посмотреть список провайдеров, их маски ввода и т.д.?

Для получения списка доступных провайдеров и их сценариев оплаты (см. раздел [0](#)) необходимо использовать запрос `getUIProviders`.

Формат ответа на запрос и примеры обработки ответов для разных провайдеров можно посмотреть в документе [Протокол взаимодействия терминального ПО и процессинга](#).

На какие адреса необходимо отправлять запросы по протоколу?

Запросы по протоколу можно отправлять на любой из следующих адресов:

- <https://xml1.qiwi.com/xmlgate/xml.jsp>
- <https://xml2.qiwi.com/xmlgate/xml.jsp>

Существуют ли тестовые сервера или тестовые провайдеры?

Для получения доступа к тестовой среде обратитесь к своему менеджеру сопровождения.

Какие способы авторизации существуют? Для чего необходим одноразовый пароль персоны?

Авторизация происходит [по цифровой подписи пакетов данных](#) — это безопасный метод, соответствует рекомендациям ЦБ РФ. Одноразовый пароль используется для передачи публичного ключа в QIWI.

Как можно узнать минимальную и максимальную суммы платежа по провайдеру?

Минимальную/максимальную сумму платежа по провайдеру можно получить с помощью запроса `getUIProviders` (см. атрибуты `minSum`, `maxSum` корневого тега `providers`).

ВНИМАНИЕ



Узнать минимальную/максимальную сумму платежа можно только по провайдерам, подключенным у агента.

Если минимальная/максимальная сумма в настройках провайдера не указана, то по умолчанию она должна попадать в диапазон 0,01-15000,00 рублей.

В ответе на запрос возвращается ошибка 300

Рекомендуемые действия:

- Проверить правильность составления XML-запроса.

- Узнать, нет ли проблем с проведением платежей по данному провайдеру. Оповещения о задержке платежей публикуются на главной странице сайта agt.qiwi.com в разделе **Сообщения службы технической поддержки**.
- Проверить наличие необходимых экстра-полей в запросе платежа (подробнее см. раздел [Платежные запросы](#)).
- Повторить запрос с тем же набором данных (при проведении платежа с тем же идентификатором платежа) до получения ответа со статусом/[кодом ошибки](#), на основе которого можно сделать вывод о прохождении платежа/выполнении действия.
- Если ничего не помогло, обратиться за помощью в техническую поддержку

Сколько платежей можно отправлять в одном пакете?

В одном пакете можно отправить от 1 до 50 платежей.

ВНИМАНИЕ



Данное правило действует только для запроса `addOfflinePayment`.

Какова размерность уникального ID платежа на терминале?

Максимальная разрядность идентификатора платежа – 18 цифр. Максимальное значение идентификатора 9 223 372 036 854 775 807.

Как получить реестр совершенных платежей для проведения сверки?

Для этого существует несколько способов:

- выгрузить список транзакций по xml-протоколу запросом `getPayments`;
- получить реестр проведенных платежей на электронную почту;
- выполнить автоматическую сверку через xml-протокол или по электронной почте;
- воспользоваться Личным кабинетом.

Для получения подробной информации обратитесь к своему менеджеру сопровождения.

ПРИЛОЖЕНИЕ А. КОДЫ ОШИБОК

Приложение содержит список наиболее распространенных кодов ошибок обработки запросов, а также их фатальность и рекомендуемые действия, которые необходимо предпринять для исправления возникшей ошибки.

Действиям, направленным на исправление ошибки, присвоены следующие типы:

- 1 – нет ошибки;
- 2 – рекомендуется дождаться окончания проведения (запросить статус платежа или отправить платеж повторно с тем же ID). Если ошибка повторяется в финальном статусе, необходимо заново сформировать платеж;
- 3 – платеж необходимо перепровести вручную;
- 4 – требуется исправить возникшую ошибку и заново сформировать платеж;
- 5 – требуется исправить ошибку данных, введенных пользователем, и заново сформировать платеж.

Табл. 3. Список кодов ошибок обработки запросов

ID	Текст	Описание ошибки	Фатальность	Тип действия
0	ОК	Платеж успешно проведен	0	1
1	Провайдер временно недоступен	Провайдер не успевает обработать ваш запрос в указанное время.	0	2
3	Техническая ошибка, нельзя отправить запрос провайдеру	Техническая ошибка сервера. Приостановите прием платежей по данному провайдеру. Сообщите о возникновении ошибки в службу поддержки. В течение 5-10 минут проблема будет устранена	0	2
4	Неверный формат счета/телефона	Вы неверно ввели номер счета или номер телефона. Попробуйте снова внимательно ввести номер счета или телефона. Если ошибка повторяется – обратитесь в службу поддержки	1	5
5	Номер не принадлежит оператору	Номер счета или телефона, который клиент Вам сообщил, не существует в базе данных провайдера, или Вы неверно его ввели. Проверьте, правильно ли Вы выбрали провайдера, правильно ли ввели номер телефона. Попробуйте ввести данные еще раз. Возможно, клиент сообщил Вам ошибочный номер	1	5
7	Прием платежа запрещен, обратитесь к оператору	Проблема во взаимоотношениях клиента с провайдером. Провайдер запретил прием платежа для этого клиента. Сообщите клиенту об этой проблеме, попросите его самостоятельно связаться с провайдером и решить этот вопрос	1	4

ID	Текст	Описание ошибки	Фатальность	Тип действия
8	Прием платежа запрещен по техническим причинам	Технические проблемы при взаимодействии с провайдером. Возможно, провайдер производит технические работы и временно прекратил прием платежей. Приостановите прием платежей по данному провайдеру. Сообщите о возникновении ошибки в службу поддержки.	1	4
13	Сервер занят, повторите запрос через минуту	Временная проблема с обработкой запроса на стороне QIWI, повторите отправку запроса с теми же реквизитами через минуту	0	2
15	Истекло время ожидания платежа в очереди	Истекло время ожидания платежа в очереди на отправку провайдеру.	1	4
16	Превышен суточный лимит на сумму операций	Превышен суточный лимит на сумму операций	1	4
19	Транзакция не подтверждена в течение 24 часов	Истекло время ожидания запроса на подтверждение платежа	1	4
23	Не удалось выставить шлюз	Не удалось назначить платежу шлюз провайдера. Приостановите прием платежей по данному провайдеру. Сообщите о возникновении ошибки в службу поддержки. В течение 5-10 минут проблема будет устранена	0	2
42	Провайдер не определен	Невозможно определить провайдера сотовой связи автоматически. Укажите провайдера вручную	1	4
77	Платеж запрещен провайдером	Прием данного платежа запрещен провайдером. Сообщите клиенту об этой проблеме, попросите его самостоятельно связаться с провайдером и решить этот вопрос	1	4
85	Отмена платежа невозможна	Запрет на отмену платежа	1	3
90	Проведение платежа не окончено	Проведение платежа не окончено. Результат проведения будет известен позже.	0	2
100	Ошибка сервера	Техническая ошибка сервера. Приостановите прием платежей по данному провайдеру. Сообщите о возникновении ошибки в службу поддержки. В течение 5-10 минут проблема будет устранена	0	2
130	Работа с данным провайдером не реализована	В настоящий момент прием платежей для выбранного провайдера невозможен. Сообщите клиенту о невозможности оплаты	1	5

ID	Текст	Описание ошибки	Фатальность	Тип действия
133	Нет прав на прием платежей	Персона, от имени которой посылается запрос, не имеет права на выполнение данного действия	1	4
150	Неверный пароль или нет прав на этот терминал	Указан неверный номер терминала, логин или неверно сформирована подпись запроса. Проверьте авторизационные данные	1	4
152	Невозможно выполнить операцию. Неодноразовый пароль.	Невозможно выполнить операцию. Требуется одноразовый пароль.	1	4
153	Персона временно заблокирована. 10 неудачных попыток пароля.	Персона временно заблокирована (на час, после 10 неудачных авторизаций в течение часа)	1	4
154	Проведение платежа запрещено СБ	Проведение платежа запрещено службой безопасности	1	4
155	Прием платежа для данного провайдера запрещен	Прием платежа в пользу данного провайдера запрещен для данного терминала или агента	1	4
158	Невозможно выполнить операцию. Запрещено использование постоянного пароля.	В запросе используется запрещенный метод авторизации. Используйте метод с цифровой подписью.	1	4
172	Данные плательщика не прошли проверку	Данные плательщика не прошли проверку, платеж отклонен	1	5
173	Превышен лимит суммы платежа без идентификации	Превышен лимит суммы платежа без идентификации, для совершения таких платежей обратитесь к своему курирующему менеджеру	1	4
201	Слишком много запросов с терминала	Превышено число одновременных запросов с терминала	0	2
202	Ошибка данных запроса	<p>Ошибка параметров запроса – возможно, не соблюден формат протокола. Исправьте причину и повторно отправьте запрос.</p> <p>Возможные причины возникновения ошибки:</p> <ul style="list-style-type: none"> отсутствие логина и/или подписи запроса; отсутствие открывающего и/или закрывающего тега; отсутствуют обязательные параметры 	1	4

ID	Текст	Описание ошибки	Фатальность	Тип действия
203	Транзакция не найдена в базе данных	Транзакция не найдена в базе данных	1	4
204	Переданы не все обязательные поля	В платеже отсутствуют обязательные поля, заданные в конфигурации провайдера	1	5
205	Отсутствуют обязательные данные плательщика	В платеже отсутствуют данные плательщика, требуемые законодательством	1	5
212	Не задана сумма платежа (или лимит терминала)	В платеже не указана сумма к зачислению на счет клиента. Попробуйте указать данные и повторить операцию. Если ошибка повторяется, обратитесь в службу поддержки.	1	4
213	Не задана сумма списания	Сумма, полученная от клиента, не указана в данных платежа.	1	4
214	Запросы с данного IP-адреса запрещены	IP-адрес, с которого выполняется запрос, не находится в списке разрешенных для терминала или агента	1	4
215	Транзакция с таким номером уже есть в базе	Нарушение логики работы. Ошибка возникает, когда в запросе с одним и тем же идентификатором платежа отличаются реквизиты. Платеж, на котором возникла эта ошибка, нужно провести снова.	1	4
216	Ошибка при сохранении транзакции на сервере	Ошибка при сохранении данных о транзакции в БД	0	2
217	Одинаковые номера терминальной транзакции в пакете	Одинаковые номера терминальной транзакции в одном пакете платежей	1	4
220	Недостаток средств у агента	Проблема в финансовых взаимоотношениях между Вами и платежной системы. На Вашем ЛС недостаточно средств для приема этого платежа. Произведите пополнение ЛС	0	2
241	Сумма слишком мала	Сумма слишком мала	1	4
242	Сумма слишком велика	Сумма слишком велика	1	4
244	Терминал не зарегистрирован у оператора	Терминал не зарегистрирован у оператора. Сообщите о возникновении ошибки в службу поддержки	0	2
246	Терминал привязан к другому компьютеру	Серийный номер терминала указан неверно	1	4

ID	Текст	Описание ошибки	Фатальность	Тип действия
252	Превышена максимальная сумма оплаты для терминала	Сумма превышает максимально установленную для терминала	1	4
271	Ошибка в ответе провайдера	Технические проблемы при взаимодействии с провайдером. Возможно, провайдер производит технические работы. Приостановите прием платежей по данному провайдеру. Сообщите о возникновении ошибки в службу поддержки.	0	2
272	Временно нет связи с провайдером	Временно нет связи с провайдером. Приостановите прием платежей по данному провайдеру. Сообщите о возникновении ошибки в службу поддержки.	0	2
290	Ошибка блокировки баланса для списания средств	При прохождении финансового контроля не может быть заблокирован баланс агента	0	2
291	Ошибка при блокировке баланса	Ошибка при блокировке баланса агента	0	2
295	Ошибка в названии интерфейса или действия	Проверьте корректность наименования указанного в запросе интерфейса или действия	1	4
300	Другая (неизвестная) ошибка провайдера	Повторите запрос с тем же набором данных (при проведении платежа – с тем же идентификатором платежа) до получения ответа со статусом/кодом ошибки, на основе которого можно сделать вывод о прохождении платежа/выполнении действия	0	2
507	Платёж отменён	Платёж отменён в системе	0	4
515	Прием платежей вне режима работы терминала запрещен	Прием платежей вне установленного времени работы терминала запрещен	1	4

ПРИЛОЖЕНИЕ Б. ТИПЫ ДЕНЕЖНЫХ СРЕДСТВ (MONEYTYPE)

ПРИМЕЧАНИЕ

При отсутствии атрибута источником платежа будут считаться наличные

moneyType	Описание	Категория
0	Наличный платеж без комиссии	Иное
10	Наличный платеж с комиссией	Иное
1	Карта своего банка без комиссии	Карты
11	Карта своего банка с комиссией	Карты
2	Карта другого банка без комиссии	Карты
21	Карта другого банка с комиссией	Карты
3	Интернет-банк без комиссии	Иное
31	Интернет-банк с комиссией	Иное
4	Мобильный банк без комиссии	Иное
41	Мобильный банк с комиссией	Иное
5	Электронные деньги без комиссии	Иное
51	Электронные деньги с комиссией	Иное
7	Платеж со счета без комиссии	Счет
71	Платеж со счета с комиссией	Счет

ПРИЛОЖЕНИЕ В. ТИПЫ ТЕРМИНАЛОВ

В системе существуют следующие типы XML-терминалов:

- XML Касса - операционная касса банка, касса в салоне связи или в магазине с приемом оплаты наличными, обслуживается кассиром;
- XML Банкомат - банкомат с приемом оплаты только банковской картой и с возможностью выдачи наличных, с самостоятельным доступом клиента;
- XML Банкомат Cash-In - банкомат с приемом оплаты наличными и банковской картой, с возможностью выдачи наличных, с самостоятельным доступом клиента;
- XML Терминал самообслуживания - терминал с приемом оплаты наличными и банковской картой, с самостоятельным доступом клиента;
- XML POS-терминал (оплата картой) - отдельное устройство для приема оплаты банковской картой, обслуживается кассиром;
- XML POS-терминал (оплата наличными) - отдельное устройство для приема оплаты наличными, обслуживается кассиром;
- XML Мобильный банкинг - (только для кредитных организаций) прием оплаты со счета клиента в мобильном приложении, с самостоятельным доступом клиента;
- XML Интернет банкинг - (только для кредитных организаций) прием оплаты со счета клиента на интернет-сайте, с самостоятельным доступом клиента;
- XML Интернет-портал - (только для кредитных организаций) прием оплаты с электронного кошелька клиента;
- XML Автоплатежи - прием оплаты путем автоматического списания с банковского счета клиента;
- XML протокол - тип XML-терминала без детализации;
- Спецпроекты QIWI - для использования данного типа терминала свяжитесь с вашим менеджером.

Набор типов XML-терминалов, доступный конкретному агенту, зависит от роли агента в системе.

ПРИЛОЖЕНИЕ Г. РОЛИ ПЕРСОН

Право – разрешение на выполнение XML-запроса. Права группируются в роли. **Роль** – набор разрешенных на выполнение действий – прав. Каждому пользователю системы назначается, как минимум, одна роль:

1. Если персоне назначена некоторая роль, то все XML-запросы, разрешенные этой роли, могут быть выполнены данной персоной.
2. Некоторые XML-запросы могут выполнять персоны с любыми ролями (например, запрос `getPayments`).

Пример 7. Права и роли персоны

Роль **Мониторинг** имеет права на просмотр баланса субагентов, мониторинг терминалов, просмотр статистики и некоторые другие операции.

Роль **Продавец** имеет права на просмотр баланса субагентов и прием платежей.

В [Протоколе взаимодействия терминального ПО и процессинга](#) для каждого XML-запроса указан набор ролей, имеющих права на его выполнение. Кроме того, запрос может быть разрешен для выполнения любым зарегистрированным пользователям системы.

Просмотреть роли, назначенные вашей персоне, вы можете с помощью следующих запросов:

1. Запросом `getPersonInfo` получить список идентификаторов ролей, назначенных персоне;
2. Запросом `getRoles` узнать названия ролей, соответствующих полученным идентификаторам.

ПРИЛОЖЕНИЕ Д. ПЛАТЕЖНЫЕ ЗАПРОСЫ

Для проведения платежей в протоколе используются действия интерфейса **providers**, описанные ниже.

addOfflinePayment

Запрос

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <client terminal="111" serial=""/>
  <providers>
    <addOfflinePayment>
      <payment id="301">
        <extras ev_paytype="5" PAY_TYPE="5".../>
        <from currency="643" amount="0.01"/>
        <to currency="643" service="3" amount="0.01" account="1111111111"
moneyType="1"/>
        <receipt id="1" date="2023-08-16T10:38:19"/>
      </payment>
    </addOfflinePayment>
  </providers>
</request>
```

Для выполнения запроса необходимо указать дополнительные параметры (формат параметров см. в п. 3.3, символом * отмечены обязательные поля):

- **payment*** – тег, содержащий информацию о платеже. Включает атрибуты:
 - **id** – идентификатор платежа;
 - **comment** – комментарий (необязательный атрибут);

ПРИМЕЧАНИЕ



Можно использовать несколько тегов **payment** внутри **addOfflinePayment**.

- **extras** – тег, в атрибутах которого указываются экстр-поля платежа. В одном теге может быть указано несколько экстр-полей в соответствии со спецификацией [интерфейса провайдера](#);
- **from*** – тег содержит информацию о сумме от клиента. Включает атрибуты:
 - **amount** – сумма;
 - **currency** – код валюты денежных средств (значения кодов соответствуют **ISO 4217**);
- **to*** – тег содержит информацию о назначении платежа и сумме, которая должна быть перечислена на счет. Включает атрибуты:
 - **service** – идентификатор сервиса, услуги которого оплачиваются;
 - **account** – номер счета;
 - **amount** – сумма к зачислению;
 - **currency** – код валюты суммы к зачислению (значения кодов соответствуют **ISO 4217**);
 - **moneyType** – тип денежных средств, с помощью которых клиент совершил оплату. Возможные значения см. в [Приложении Б](#);

- receipt – тег, содержит информацию о чеке. Включает атрибуты:
- id – идентификатор чека;
- date – дата и время печати чека на терминале.

Ответ

```
<?xml version="1.0" encoding="utf-8"?>
<response result="0">
  <providers>
    <addOfflinePayment result="0">
      <payment date="2023-08-16T10:38:21+04:00" fatal="false" id="301"
        result="0" status="1" uid="7777777777777777"/>
    </addOfflinePayment>
  </providers>
</response>
```

Ответ содержит следующие данные:

- payment – тег, содержащий информацию о платеже. Включает атрибуты:
 - date – дата и время (московские) приема платежа в процессинг;
 - id – идентификатор платежа;
 - status – текущий статус платежа (подробнее см. в разделе [6.1](#));
 - result – код ошибки проведения платежа (подробнее см. в [Приложении А](#));
 - fatal – признак фатальной ошибки:
 - ⊕ true - отрицательный ответ на платежный запрос;
 - ⊕ false - промежуточный статус обработки платежа (дальнейшие действия зависят от конкретной ошибки);
 - uid – идентификатор транзакции в QIWI.

ПРИМЕЧАНИЕ



Внутри тега <payment> может быть тег <extras>, содержащий в качестве атрибутов экстра-поля ответа.

В одном теге <extras> может быть указано несколько экстра-полей. Экстра-поля провайдера передаются агенту в соответствии со спецификацией интерфейса провайдера.

checkPaymentRequisites

Запрос

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <client terminal="111" serial=""/>
  <providers>
    <checkPaymentRequisites>
      <payment id="501">
        <from currency="643" amount="11.00"/>
        <to currency="643" service="2" amount="11.00" account="111" moneyType="1" />
        <receipt id="132" date="2023-08-16T12:43:01"/>
      </payment>
    </checkPaymentRequisites>
  </providers>
</request>
```

Параметры запроса соответствуют параметрам `addOfflinePayment`.

Ответ

```
<?xml version="1.0" encoding="utf-8"?>
<response result="0">
  <providers>
    <checkPaymentRequisites result="0">
      <payment date="2023-08-16T12:43:02+04:00" id="501" fatal="true"
        result="5" status="0" uid="7777777777777777">
        <extras AUTH_HOST="app9.osmp.ru" disp1="[param1]"
          disp2="[param2]"/>
      </payment>
    </checkPaymentRequisites>
  </providers>
</response>
```

Ответ содержит данные, аналогичные ответу на запрос `addOfflinePayment`.

authorizePayment

Запрос

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <client terminal="111" serial=""/>
  <providers>
    <authorizePayment>
      <payment id="301">
        <extras ev_paytype="5" PAY_TYPE="5" .../>
        <from amount="0.01" currency="643"/>
        <to service="2" account="1919" amount="0.01" currency="643" moneyType="1"/>
        <receipt id="132" date="2023-09-01T15:54:16"/>
      </payment>
    </authorizePayment>
  </providers>
</request>
```

Параметры запроса соответствуют параметрам `addOfflinePayment`.

Ответ

```
<?xml version="1.0" encoding="utf-8"?>
<response result="0">
  <providers>
    <authorizePayment result="0">
      <payment date="2023-09-01T15:54:38+04:00" id="301" result="0"
        fatal="false" status="3" uid="7777777777777777">
        <extras AUTH_HOST="app8.osmp.ru" PARSER_HOST="app8.osmp.ru"/>
      </payment>
    </authorizePayment>
  </providers>
</response>
```

Ответ содержит данные, аналогичные ответу на запрос `addOfflinePayment`.

confirmPayment

Запрос

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <client terminal="111" serial=""/>
  <providers>
    <confirmPayment>
      <payment id="301"/>
    </confirmPayment>
  </providers>
</request>
```

Для выполнения запроса необходимо указать дополнительные параметры (формат параметров см. в п. 3.3, символом * отмечены обязательные поля):

- `payment*` – тег описывает один платеж. Включает атрибут:
 - `id` – идентификатор платежа (должен совпадать с `id` в предшествующем запросе [authorizePayment](#)).

ПРИМЕЧАНИЕ



Можно использовать несколько тегов `payment` внутри `confirmPayment`.

Ответ

```
<?xml version="1.0" encoding="utf-8"?>
<response result="0">
  <providers>
    <confirmPayment result="0">
      <payment date="2023-09-01T15:57:34+04:00" id="301"
        result="0" fatal="false" status="1" uid="7777777777777777">
        <extras AUTH_HOST="app8.osmp.ru"/>
      </payment>
    </confirmPayment>
  </providers>
</response>
```

Ответ содержит данные, аналогичные ответу на запрос `addOfflinePayment`.

getPaymentStatus

Запрос

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <client terminal="111" serial=""/>
  <providers>
    <getPaymentStatus>
      <payment id="301"/>
    </getPaymentStatus>
  </providers>
</request>
```

Для выполнения запроса необходимо указать дополнительные параметры (формат параметров см. в п. [3.3](#), символом * отмечены обязательные поля):

- `payment*` – тег, содержащий информацию о платеже. Включает атрибут:
 - `id` – идентификатор платежа.

ПРИМЕЧАНИЕ



Для получения информации о нескольких платежах можно использовать несколько тегов `payment` внутри `getPaymentStatus`.

Ответ

```
<?xml version="1.0" encoding="utf-8"?>
<response result="0">
  <providers>
    <getPaymentStatus result="0">
      <payment date="2023-12-04T19:23:10+04:00" fatal="false" id="301"
        result="0" status="2" uid="10625592616002">
        <extras AUTH_HOST="app2.osmp.ru" PARSER_HOST="app5.osmp.ru"/>
      </payment>
    </getPaymentStatus>
  </providers>
</response>
```

Ответ содержит следующие данные:

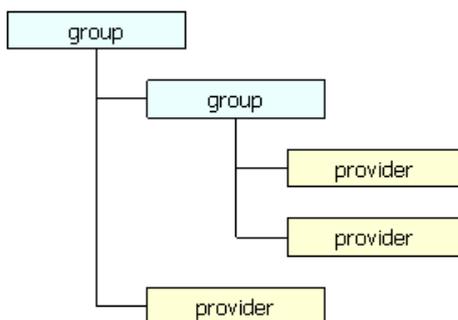
- `payment` – тег, содержащий информацию о платеже. Включает атрибуты:
 - `date` – дата и время (московские) приема платежа в процессинг;
 - `id` – идентификатор платежа;
 - `status` – текущий статус платежа (подробнее см. в разделе [6.1](#));
 - `result` – код ошибки проведения платежа (подробнее см. в [Приложении А](#));
 - `fatal` – признак фатальной ошибки:
 - ⊕ `true` - отрицательный ответ на платежный запрос;
 - ⊕ `false` - промежуточный статус обработки платежа;
 - `uid` – идентификатор транзакции в QIWI;
 - `value-date` - дата и время (московские) списания суммы платежа с баланса агента.

ПРИЛОЖЕНИЕ Е. ОПИСАНИЕ СЦЕНАРИЯ ОПЛАТЫ

Каталог доступных для оплаты услуг

Каталог доступных для оплаты услуг (провайдеров) представляет собой иерархическую структуру групп (категорий услуг) и содержащихся в них провайдеров (Рис. 12), а также параметры отображения страниц сценария оплаты провайдера (порядок и элементы управления для каждой страницы см. на Рис. 13).

Рис. 12. Структура XML-тегов каталога услуг



Получение каталога услуг производится с помощью следующих действий интерфейса `providers`:

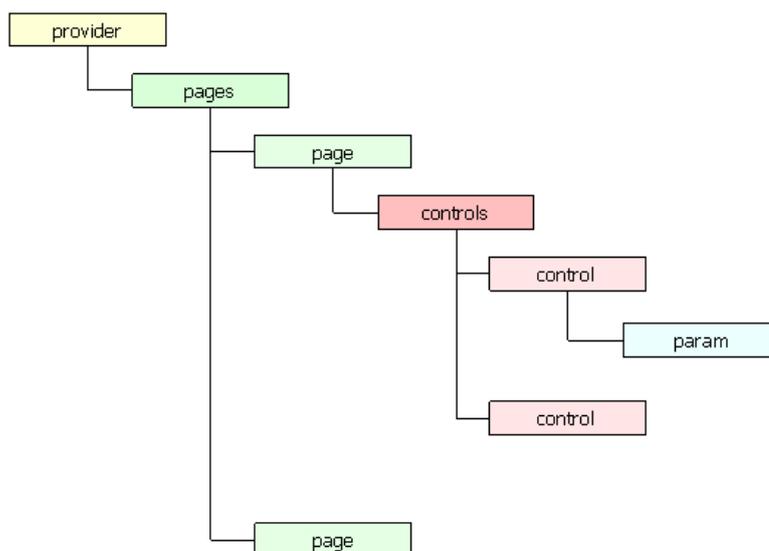
- `getUIGroups` – получение структуры групп и провайдеров для формирования списка услуг в интерфейсе терминала (описание структуры см. в [Протоколе взаимодействия терминального ПО и процессинга](#)).

ВНИМАНИЕ



Запрос возвращает только разрешенных для оплаты на текущем терминале провайдеров.

Рис. 13. Структура XML-тегов страниц сценария оплаты провайдера



- `getUIProviders` – получение подробной информации об отображении сценария оплаты каждого провайдера, включая порядковый номер провайдера, а также элементы управления: маски полей ввода и др.

Для каждого провайдера сформирована последовательность страниц сценария оплаты, определен порядок их отображения, а также элементы управления страницами (описание формата элементов страниц см. в [Протоколе взаимодействия терминального ПО и процессинга](#)).

Отображение каталога услуг и сценария оплаты

Рассмотрим пример получения каталога услуг и построения сценария оплаты:

Пример 8. Интерфейс терминала

```
<?xml version="1.0" encoding="utf-8"?>
<response result="0">
  <providers>
    <getUIGroups result="0">
      <group id="20" logo="logos/2975749923198187786" logo_crc="2035F071"
logo_size="2921" name="Сотовая связь" orderId="1" tag="visible,ranges">
        <logos>
          <logo crc="2035F071" path="logos/2975749923198187786" size="2921"
type="standard"/>
        </logos>
        <provider id="2" orderId="2" showInTop="2" tag="visible,ranges"/>
        <provider id="1" orderId="3" showInTop="1" tag="visible,ranges"/>
      </group>
      <group id="112" logo="logos/6174676531169676367" logo_crc="143CCC70"
logo_size="4541" name="Интернет, ТВ и Телефония" orderId="41" tag="visible">
        <logos>
          <logo crc="143CCC70" path="logos/6174676531169676367" size="4541"
type="standard"/>
        </logos>
        <group id="103" logo="logos/1564789233217259209" logo_crc="143CCC70"
logo_size="4541" name="Интернет" orderId="42" tag="visible">
          </group>
          <group id="40" logo="logos/4084770143860797557" logo_crc="9E1D3902"
logo_size="3675" name="ТВ" orderId="624" tag="visible">
            </group>
            <group id="107" logo="logos/7930600709753489404" logo_crc="46EB2E16"
logo_size="5848" name="Телефония" orderId="705" tag="visible">
              </group>
              <group id="114" logo="logos/3388469817160941103" logo_crc="10D5E119"
logo_size="4194" name="Хостинг и домены" orderId="776" tag="visible">
                </group>
                <provider id="71635" orderId="803"
tag="visible,new_visibility_root_rus,161_Ф3"/>
                <provider id="2935" orderId="804"
tag="visible,new_visibility_root_rus,161_Ф3"/>
              </group>
            ...
          </getUIGroups>
        </providers>
      </response>
```

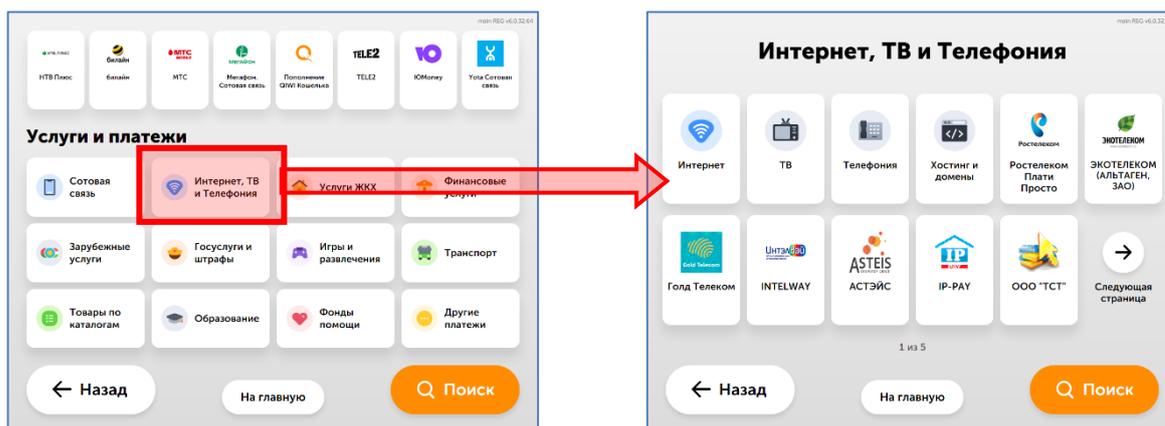
В соответствии со списком групп, полученным в запросе `getUIGroups`, страница категорий услуг (упорядоченных по параметру `orderId`) выглядит следующим образом ([Рис. 14](#)):

3. Группа **Сотовая связь** (ID=20).
4. Группа **Интернет, ТВ и Телефония** (ID=112) и т.д.

Внутри группы **Интернет, ТВ и Телефония** подгруппы и провайдеры также упорядочены по параметру `orderId`:

5. Интернет (ID=103)
6. ТВ (ID=40)
7. Телефония (ID=107) и т.д.

Рис. 14. Внешний вид каталога услуг в интерфейсе терминала (пример)



Отображение страниц интерфейса оплаты провайдера

Для каждого провайдера (см. [Пример 9](#)) определяются страницы сценария оплаты услуги, порядок их отображения, а также элементы управления для страниц интерфейса:

- `page` – содержит идентификатор страницы и порядковый номер ее отображения в интерфейсе провайдера,
 - `control` – элементы управления, например:
 - ⊕ цифровая клавиатура (`type="keyboard"`),
 - ⊕ поле ввода номера телефона, лицевого счета или иного реквизита платежа (`type="text_input"`);
 - ⊕ атрибуты элемента, например, для поля ввода номера:
 - ❖ маска ввода номера;
 - ❖ регулярное выражение для проверки номера;
 - ❖ заголовок поля.

Например, для провайдера Yota Интернет реализованы следующие страницы ([Пример 9](#), [Рис. 15](#)):

Пример 9. Описание интерфейса оплаты провайдера Yota Интернет

```
<?xml version="1.0" encoding="utf-8"?>
<response result="0">
  <providers>
    <getUIProviders result="0">
      <provider curId="643" fiscalName="Интернет" grpId="103" id="1516"
inn="7701725181" jDocDate="01.04.2014" jDocNum="13-0132" jDocType="qiwi" jName="ООО
&quot;Скартел&quot;" keywords="ета|йота|skartel|iota|yotta" lName="ООО
&quot;Скартел&quot;" logo="logos/!anonymous/yota_bill.gif" logo_crc="EE3B781F"
```

```

logo_size="3580" maxSum="15000" minSum="1" ratingId="0" receiptName="Интернет"
sName="Yota Интернет" supportPhone="8-800-550-00-07"
tag="visible,161_Ф3,Аффилированные провайдеры Мегафон">
  <logos>
    <logo crc="EE3B781F" path="logos!/anonymous/yota_bill.gif" size="3580"
type="standard"/>
  </logos>
  <constParams/>
  <pages>
    <page orderId="1" pageId="1405" useOnline="true">
      <controls>
        <control footer="Уважаемые абоненты! Обращаем ваше внимание на то, что
с 25 сентября произвести оплату можно исключительно по номеру Лицевого Счета
(формат 10 цифр)" header="Введите номер счета" mask="&lt;!^\\d+${10}&gt;"
name="account" nobr="false" orderId="1" regexp="^[0-8]{1}\\d{9}$" strip="false"
type="text_input"/>
        <control layout="DG" orderId="2" type="keyboard"/>
      </controls>
    </page>
  </pages>
</provider>
</getUIProviders>
</providers>
</response>

```

В данном случае интерфейс содержит только одну страницу с единственным полем ввода `<control type="text_input" .../>`, в котором указаны:

- заголовок поля `header="Введите номер счета"`,
- маска ввода `mask="<!^\\d+${10}>"`
- регулярное выражение для проверки номера `regexp="^[0-8]{1}\\d{9}$"`
- комментарий `footer="Уважаемые абоненты! Обращаем ваше внимание на то, что с 25 сентября произвести оплату можно исключительно по номеру Лицевого Счета (формат 10 цифр)"`,

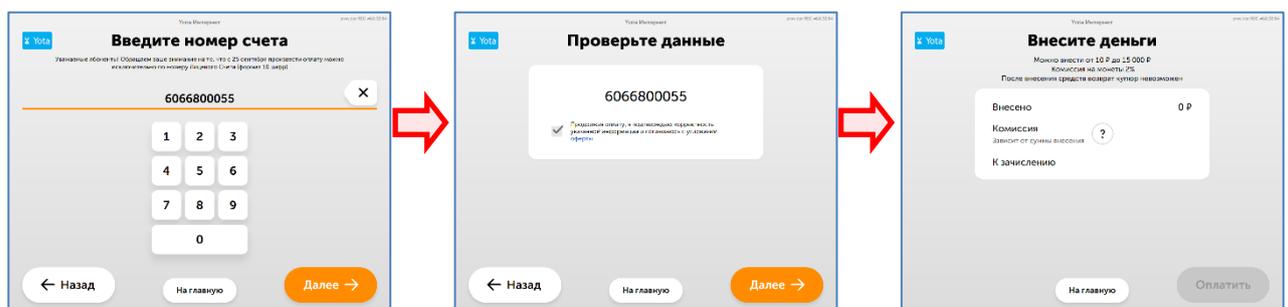
Также на странице отображается цифровая клавиатура `<control type="keyboard" orderId="2" layout="DG"/>`.

ПРИМЕЧАНИЕ



Страницы подтверждения реквизитов платежа, выбора способа оплаты, внесения денег/ввода суммы платежа являются служебными и одинаковы для всех сценариев оплаты, поэтому не содержатся в описании.

Рис. 15. Внешний вид страницы провайдера Yota Интернет в терминале (пример)



Дополнительные параметры интерфейса провайдера

В описании страниц интерфейса оплаты провайдера могут присутствовать дополнительные реквизиты (экстра-поля) для корректной обработки платежа провайдером. Эти параметры необходимо передавать в [платежных запросах](#) в специальном теге `<extras ...>`.

Список необходимых экстра-полей содержится в ответе на запрос `getUIProviders`. Из ответа на запрос необходимо извлечь следующие атрибуты (параметры экстра-полей) конфигурации провайдера внутри тега `<provider id="...">` (см. также [Пример 10](#)):

- данные из вложенного тега `constParams`, например:
 - ⊕ `ev_bank="osmp" (<param name="ev_bank" value="osmp"/>);`
 - ⊕ `PT="1" (<param name="PT" value="1"/>);`
- данные во вложенном теге `control` с атрибутом `disp_type="extra"` или поля ввода, кроме стандартного поля номера телефона/счета `account`. Например:
 - ⊕ `ev_fname="имя плательщика, введенное в сценарии оплаты" из элемента интерфейса в теге <control type="text_input" orderId="2" name="ev_fname" mask="<!^\.$\{1,20}>" strip="false" nobr="false" regexp="^\.{1,20}$" disp_type="extra" header="Введите имя"/>.`

ПРИМЕЧАНИЕ



Если минимальное количество цифр в регулярном выражении для проверки дополнительного поля ввода (атрибут `regexp`) – 0 (например, `regexp="^\d{0,10}$"`), то данное поле ввода передавать в запросах не обязательно.

- данные из вложенного тега `param` внутри тега `control`. Например:
 - ⊕ `account_type="2" (<param name="_extra_account_type" value="2"/>);`
 - ⊕ `oper_type="3" (<param name="_extra_oper_type" value="3"/>);`

ВНИМАНИЕ



Если имя дополнительного параметра начинается с префикса `_extra_`, необходимо удалить префикс перед помещением параметра в список экстра-полей платежного запроса.

Пример 10. Интерфейс провайдера с дополнительными полями

```
<?xml version="1.0" encoding="utf-8"?>
<response result="0">
  <providers>
    <getUIProviders result="0">
      ...
    <provider buttonName="ОТП Банк" curId="643" fiscalName="Др.услуги" grpId="362"
      id="95389" inn="3123011520" jDocDate="01.12.2012" jDocNum="БАМ-БА-38/2012"
      jDocType="bank" jName="КИВИ Банк (АО)" keywords="jng|банк|ОТП|БАНК
      ОТП|отпбанк|отпОтпбанк|Отпбанк" lName="ОТП БАНК АО"
      logo="logos/!anonymous/prv95389_bc2762a0.gif" logo_crc="BC2762A0" logo_size="654"
      maxSum="15000" minSum="1" ratingId="1" receiptName="Другие услуги" sName="ОТП Банк
      для КО XML" supportPhone="74957835400"
      tag="visible,Открыть_XML_KO,Disable_retzn_prv_group">
      <logos>
        <logo crc="BC2762A0" path="logos/!anonymous/prv95389_bc2762a0.gif"
          size="654" type="standard"/>
      </logos>
    </provider>
  </getUIProviders>
</providers>
</response>
```

```

</logos>
<constParams>
  <param name="account_type" value="2"/>
</constParams>
<pages>
  <page orderId="1" pageId="259706">
    <controls>
      <control layout="DG" orderId="2" type="keyboard"/>
      <control footer="Комиссия КИВИ Банк (АО): 1,8% от суммы
платежа, но не менее 120 руб.\nМаксимальная сумма данного платежа составляет 15 000
руб.\nПлатежи проводятся в течение одного-двух рабочих дней.\nСумма платежей на
один номер счета/карты/договора в течение 7 дней не должна превышать 600 000
рублей.\nВозможно погашать кредиты только в рублях и пополнять только рублевые
счета резидентов." header="Введите номер счёта" mask="&lt;!^\d{20}&gt;"
name="account" nobr="false" orderId="1" regexp="^(40817810|42309810)\d{12}$"
strip="false" type="text_input"/>
    </controls>
  </page>
  <page orderId="2" pageId="259708">
    <controls>
      <control disp_type="extra" header="Введите фамилию"
mask="&lt;!^\.{1,20}&gt;" name="lname" nobr="false" orderId="1" regexp="^\.{1,20}$"
strip="false" type="text_input"/>
      <control disp_type="extra" header="Введите имя"
mask="&lt;!^\.{1,20}&gt;" name="fname" nobr="false" orderId="2" regexp="^\.{1,20}$"
strip="false" type="text_input"/>
      <control disp_type="extra" header="Введите отчество"
mask="&lt;!^\.{1,20}&gt;" name="mname" nobr="false" orderId="3" regexp="^\.{1,20}$"
strip="false" type="text_input"/>
      <control layout="ALCR" orderId="4" type="keyboard"/>
    </controls>
  </page>
</pages>
</provider>
...
</getUIProviders>
</providers>
</response>

```

Таким образом, при оплате услуги из примера 10 плательщик в качестве реквизитов платежа укажет номер счета, фамилию, имя и отчество, а в [платеже](#) тег `<extras ...>` будет заполнен следующим образом:

```
<extras account_type="2" lname="Иванов" fname="Иван" mname="Иванович" />.
```

СПИСОК РИСУНКОВ

Рис. 1. Структура интерфейса	6
Рис. 2. Подтверждение контактных данных	10
Рис. 3. Активация сертификата	10
Рис. 4. Справочник «Терминалы».....	11
Рис. 5. Создание нового терминала.....	11
Рис. 6. Вкладка «IP-адреса».....	12
Рис. 7. Вкладка «Персоны»	13
Рис. 8. Форма создания персоны	14
Рис. 9. Форма редактирования данных персоны.....	14
Рис. 10. Сценарий одношагового проведения платежа	21
Рис. 11. Сценарий двухшагового проведения платежа.....	24
Рис. 12. Структура XML-тегов каталога услуг	42
Рис. 13. Структура XML-тегов страниц сценария оплаты провайдера	42
Рис. 14. Внешний вид каталога услуг в интерфейсе терминала (пример).....	44
Рис. 15. Внешний вид страницы провайдера Yota Интернет в терминале (пример).....	45

СПИСОК ТАБЛИЦ

Табл. 1. Глоссарий	3
Табл. 2. Запросы на проведение платежей	19
Табл. 3. Список кодов ошибок обработки запросов.....	29

СПИСОК ПРИМЕРОВ

Пример 1. Пример запроса.....	5
Пример 2. Генерация RSA ключей.....	15
Пример 3. Сохранение открытого ключа.	15
Пример 4. Формирование цифровой подписи XML-запроса.	16
Пример 5. Формирование заголовка запроса с цифровой подписью.	17
Пример 6. Ответ на запрос авторизации платежа.	20
Пример 7. Права и роли персоны	36
Пример 8. Интерфейс терминала	43
Пример 9. Описание интерфейса оплаты провайдера Yota Интернет.....	44
Пример 10. Интерфейс провайдера с дополнительными полями.....	46