

Протокол взаимодействия с провайдером

Версия 0.1

1 ВВЕДЕНИЕ

Данный документ представляет собой описание стандартного протокола взаимодействия Системы приема платежей и провайдеров услуг А2.

2 ОСНОВНЫЕ СВЕДЕНИЯ

В данном разделе приведены требования к интерфейсу провайдера, а также общие сведения о протоколе обмена данными между Системой и провайдером.

2.1 Термины, сокращения и аббревиатуры

Табл. 1. Сокращения, принятые в документе

Термин	Определение
Система	Аппаратно-программный комплекс, выполняющий обработку платежей
Провайдер	Организация, предоставляющая некоторые услуги (организация мобильной связи, доступ в Интернет и др.)
Абонент	Физическое или юридическое лицо, заключившее договор с провайдером на оказание услуг связи
Интерфейс	Совокупность средств взаимодействия с Системой
Операция	Действие, выполняемое над интерфейсом провайдера
HTTP	Протокол передачи данных в сети Интернет
HTTPS	Расширение протокола HTTP, поддерживающее шифрование передаваемых данных на основании протоколов SSL/TLS
XML	XML – формат, предназначенный для хранения структурированных данных и обмена информацией между программами
АСО	Автомат самообслуживания
ПО	Программное обеспечение

2.2 Общие принципы

Взаимодействие Системы и провайдера строится в режиме запрос-ответ, где инициатором запроса всегда является Система, а отвечающей стороной – провайдер.

Каждый платеж в Системе имеет уникальный идентификатор, который передается в каждом запросе. По этому идентификатору производится дальнейшая сверка взаиморасчетов и решение спорных вопросов.

При обработке запросов от Системы провайдер должен выполнить требуемую операцию, а затем передать Системе данные (если это требуется) и результат выполнения операции (см. раздел 4).

2.3 Требования к интерфейсу провайдера

Протокол:	Интерфейс должен принимать запросы по протоколу HTTPS .
IP адреса подсети:	Интерфейс должен принимать запросы с IP адресов: 89.108.104.211; 89.108.104.129; 91.218.95.251; 109.188.77.113;
Метод передачи данных:	Интерфейс должен обрабатывать параметры, передаваемые системой методом POST .
Формат ответа:	Интерфейс должен формировать ответ Системе в формате xml в кодировке utf-8 (если ответ содержит символы национальных алфавитов).
Скорость взаимодействия:	Скорость ответа не должна превышать 60 секунд.
Многопоточность:	Интерфейс провайдера должен поддерживать многопоточную коммуникацию (10-15 одновременных соединений).
Метод проверки подлинности запросов и ответов:	POST запрос и ответ должны содержать в заголовке поле X-Signature с base64 содержимым hmac-sha256 от тела запроса или ответа.
Поле заголовка Content-Type в запросе к провайдеру:	application/x-www-form-urlencoded; charset=utf-8
Поле заголовка Content-Type в ответе провайдера:	text/xml; charset=utf-8

3 ОПИСАНИЕ ОПЕРАЦИЙ

Табл. 2. Описание операций

Операция	Назначение	Параметры	Возвращаемое значение
check	Проверка возможности совершения платежа. Запись информации о платеже у провайдера	Информация о платеже	Информация о возможности принятия платежа
pay	Подтверждение проведения платежа	Информация о платеже	Информация о платеже (в том числе идентификатор транзакции у провайдера)

3.1 Check

При получении запроса `check` провайдер должен проверить наличие в своей базе абонента с указанным идентификатором и выполнить внутренние проверки идентификатора и суммы платежа в соответствии с принятой логикой пополнения лицевых счетов через платежные Системы.

В базе провайдера не должно содержаться двух успешно проведенных платежей с одним и тем же идентификатором платежа (`txn_id` – идентификатор платежа в Системе). При получении запроса со значением `txn_id`, уже существующим в базе провайдера, провайдер должен вернуть результат обработки предыдущего запроса.

3.1.1 Поля запроса

Табл. 3. Обязательные поля запроса «check»

Поле	Описание
<code>command=check</code>	Идентификация типа запроса: осуществить проверку счета абонента
<code>txn_id</code>	Идентификатор платежа в Системе
<code>account</code>	Идентификатор абонента в информационной системе провайдера
<code>sum</code>	Сумма операции

В запросе могут присутствовать дополнительные поля для передачи провайдеру данных, введенных клиентом: адрес, фио и пр.

Пример запроса см. в [разделе 5.1.1](#).

Формат передаваемых в полях данных см. в [Приложении Б](#).

3.1.2 Поля ответа

В ответ провайдер должен вернуть XML, содержащий элемент `response` со следующими полями (см. Табл. 4):

Табл. 4. Поля ответа на запрос «check»

Поле	Описание
Обязательные	
txn_id	Идентификатор транзакции в Системе (txn_id)
result	Код результата операции (result=0 в случае возможности принятия платежа)
Необязательные	
comment	Комментарий к операции

Пример ответа см. в [разделе 5.1.2](#).

3.2 Pay

При получении запроса `pay` провайдер должен произвести пополнение баланса абонента.

3.2.1 Поля запроса

Табл. 5. Обязательные поля запроса «pay»

Поле	Описание
<code>command=pay</code>	Идентификация типа запроса: осуществить пополнение баланса абонента
<code>txn_id</code>	Идентификатор платежа в Системе
<code>txn_date</code>	Дата операции
<code>account</code>	Идентификатор абонента в информационной системе провайдера
<code>sum</code>	Сумма операции

В запросе могут присутствовать дополнительные поля для передачи провайдеру данных, введенных клиентом: адрес, фио и пр.

Пример запроса см. в [разделе 5.2.1](#).

Формат данных см. в [Приложении Б](#).

3.2.2 Поля ответа

В ответ интерфейс провайдера должен вернуть XML, содержащий элемент `response` со следующими полями (Табл. 6):

Табл. 6. Поля ответа на запрос «pay»

Поле	Описание
Обязательные	
txn_id	Идентификатор транзакции в Системе (txn_id)
prv_txn	Уникальный номер операции пополнения баланса абонента (в базе провайдера)
sum	Сумма платежа, переданная провайдеру
result	Код результата операции (result=0 в случае возможности принятия)

	платежа)
Необязательные	
comment	Комментарий к операции

Пример см. в [разделе 5.2.2](#).

3.3 Дополнительные поля ответов

Наряду с обязательными полями ответа существуют дополнительные поля, в которых провайдер может передавать Системе дополнительную информацию:

- [Информация об абоненте и/или операции](#);
- [Идентификатор шлюза](#).

Дополнительные поля могут содержаться в ответе провайдера как на запросы check, так и на запросы pay. В ответе провайдера могут содержаться все дополнительные поля или только часть из них.

Информация об абоненте и/или операции

В необязательных полях `field1`, `field2` ... `fieldN` может содержаться информация, которую необходимо передать Системе.

Эта информация может быть показана пользователю и/или сохранена в Системе при совершении платежа.

Табл. 7. Дополнительные поля ответов.

Элемент	Описание
fields	Блок с информацией об абоненте или об операции
fields.field1	Поле, содержащее информацию
...	
fields.fieldN	
fields.field1.type	Тип поля. Возможно использование следующих типов полей:
...	<ul style="list-style-type: none"> • <code>disp</code> – информация для отображения клиенту при совершении платежа (по умолчанию, если тип не указан); • <code>info</code> – информация для сохранения в Системе; • <code>pvt-data</code> – текст для печати на чеке при совершении платежа (используется только для платежей с АСО).
fields.fieldN.type	
fields.field1.name	Имя поля. Если не указано, присваивается имя <code>disp1,...,dispN</code>
...	
fields.fieldN.name	

Пример см. в [разделе 5.3.1](#).

Информация о шлюзе

При необходимости информацию о шлюзе, в котором будет зафиксирован платеж, можно передать в теге `pay_id`.

Информация о том, какие шлюзы доступны для выбора, уточняется при тестировании Системы.

Пример см. в [разделе 5.3.2](#).

4 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЗАПРОСОВ

При обработке запросов от Системы провайдер должен сопоставить все возникающие в его приложении ошибки с приведенным ниже списком (см. [Приложение А](#)) и возвращать соответствующие коды в поле `result`.

Существуют фатальные и нефатальные ошибки:

- Фатальная ошибка означает, что повторная отправка запроса с теми же параметрами приведет к 100% повторению той же ошибки; следовательно, Система прекращает обработку клиентского запроса и завершает его с ошибкой.

- Нефатальная ошибка означает для Системы, что повторение запроса с теми же параметрами через некоторый промежуток времени, возможно, приведет к успеху.

При получении нефатальной ошибки Система будет повторять запросы, увеличивая интервал, пока операция не завершится успехом или фатальной ошибкой, либо пока не истечет срок жизни платежа в Системе (24 часа). Например, отсутствие связи с сервером провайдера является нефатальной ошибкой. Отсутствие в ответе поля `result` является фатальной ошибкой.

5 ПРИМЕРЫ ЗАПРОСОВ

В примерах предполагается, что платежное приложение провайдера `payment_app.cgi` располагается по адресу `https://service.someprv.ru`, сервер поддерживает протокол HTTPS соединения на порт 8443.

5.1 Check

5.1.1 Запрос

Для проверки состояния абонента Система генерирует запрос следующего вида:
`https://service.someprv.ru:8443/payment_app.cgi`

POST body:

```
command=check&txn_id=1234567&account=4950001111&sum=10.45
```

5.1.2 Ответ

Ответ провайдера должен выглядеть следующим образом:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <txn_id>1234567</txn_id>
  <result>0</result>
  <comment>Some comment</comment>
</response>
```

Код результата `result=0` выполнения запроса `check` свидетельствует о том, что лицевой счет абонента с соответствующим ему номером `txn_id` может быть пополнен на сумму, указанную в запросе.

5.2 Pay

5.2.1 Запрос

Для подтверждения платежа на пополнение лицевого счета Система генерирует запрос следующего вида:
`https://service.someprv.ru:8443/payment_app.cgi`

POST body:

```
command=pay&txn_id=1234567&txn_date=20090815120133&account=4950001111&sum=10.45
```

5.2.2 Ответ

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <txn_id>1234567</txn_id>
  <prv_txn>2016</prv_txn>
  <sum>10.45</sum>
  <result>0</result>
  <comment>OK</comment>
</response>
```

Возвращая `result=0` на запрос `pay`, провайдер сообщает об успешном завершении операции пополнения баланса.

В необязательном поле comment содержится служебный комментарий.

5.3 Дополнительные поля ответов

5.3.1 Информация об абоненте и/или операции

Ответ провайдера тогда должен выглядеть следующим образом:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <txn_id>1234567</txn_id>
  <result>0</result>
  <fields>
    <field1 name="name1" type="disp">value1</field1>
    <field2 name="name2" type="disp">value2</field2>
    ...
    <fieldN name="nameN" type="prt-data">valueN</fieldN>
  </fields>
  <comment></comment>
</response>
```

5.3.2 Информация о шлюзе

Ответ провайдера тогда должен выглядеть следующим образом:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <txn_id>1234567</txn_id>
  <result>0</result>
  <comment>Some comment</comment>
  <pay_id>111</pay_id>
</response>
```

6 ПРОВЕРКА ПОДЛИННОСТИ ЗАПРОСОВ/ОТВЕТОВ

Для обеспечения безопасности обмена данными, администратор системы передает провайдеру секретный ключ, который необходимо использовать при проверке запросов системы и передаче ответа системе.

Провайдер на своей стороне должен проверять подлинность запросов системы с помощью алгоритма hmac-sha256.

Передача контрольной суммы производится в http заголовке, поле X-Signature. Поле должно содержать base64 закодированное значение результата работы функции hmac-sha256.

Проверка контрольной суммы запроса (псевдокод):

```
base64_decode(X_Signature_value) == hash_hmac("sha256", $data, "mysecretkey")
```

Создание контрольной суммы ответа (псевдокод):

```
request.header.add("X-Signature",
  base64_encode(hash_hmac("sha256", $data, "mysecretkey")));
```

7 ЕЖЕДНЕВНАЯ СВЕРКА

До 10:00 по московскому времени Система генерирует и отправляет по указанному адресу электронный реестр принятых платежей за предыдущий день.

Реестр имеет следующую структуру:

txn_id;дата-время;идентификатор абонента;сумма;доп-поле1;доп-поле2

...

txn_id;дата-время;идентификатор абонента;сумма;доп-поле1;доп-поле2

Поля разделены знаком `;`, дробная часть суммы отделена точкой, дата/время Московские, перевод строки может состоять как из символов x0D x0A, так и просто из x0D (см. Пример 1).

Пример 1. Пример реестра

12345678;2018-05-20 12:13:14;0957000059;123.45
12345678;2018-05-20 13:22:34;8002000059;0.01
12345678;2018-05-21 14:55:11;9161234567;123.01
12345689;2018-05-25 14:55:12;0732123456;1000.00

Система включает в реестр только успешно проведенные платежи.

Подтвержденными считаются платежи, которые пришли как при online обмене сообщениями, так и в реестре.

Если в реестре отсутствуют платежи, которые проведены в базе провайдера, или содержатся платежи, которых нет в базе провайдера, либо если не удалось получить реестр, необходимо связаться до 12.00 по контактному номеру Системы, указанному в договоре, для выяснения ситуации и принятия решения.

8 ПРИЛОЖЕНИЕ А: Коды результата

Табл. 8. Список кодов завершения

Код	Комментарий	Фатальность
0	ОК	
1	Временная ошибка. Повторите запрос позже	
4	Неверный формат идентификатора абонента	+
5	Идентификатор абонента не найден	+
7	Прием платежа запрещен провайдером/Счет абонента не активен (При отмене платежа – отказ провайдера в отмене платежа)	+
8	Прием платежа запрещен по техническим причинам	+
79	Счет абонента не активен	+
90	Проведение платежа не окончено (при отмене платежа – отмена еще не подтверждена. Система отправит повторный запрос через некоторое время.)	
241	Сумма слишком мала	+
242	Сумма слишком велика	+
243	Невозможно проверить состояние счета	+
300	Другая ошибка провайдера	+

Знак «+» в столбце **Фатальность** показывает то, как Система будет интерпретировать данную ошибку.

9 ПРИЛОЖЕНИЕ Б: Формат данных

Табл. 9. Форматы полей

Элемент	Формат
account	Строка, содержащая буквы, цифры и спецсимволы, длиной до 200 символов
comment	Текстовое поле

prv_txn	Целое число длиной до 20 знаков
result	Числовой код. Список возможных кодов завершения перечислен в Приложении А
sum	Дробное число с точностью до сотых, в качестве разделителя используется «.» (точка). Если сумма представляет собой целое число, то оно все равно дополняется точкой и нулями, например, «152.00»
txn_date	Дата в формате ГГГММДДЧММСС
txn_id	Целое число длиной до 20 знаков