

Приложение №2

к Договору № ____

Редакция 1.9

1 Требования к интерфейсу провайдера

1. Интерфейс должен принимать запросы по протоколу HTTP или HTTPS
2. Интерфейс должен обрабатывать параметры, передаваемые системой методом GET
3. Интерфейс должен формировать ответ системе в формате XML в кодировке UTF-8 (если ответ содержит символы национальных алфавитов)
4. Обмен информацией ведется в режиме запрос-ответ, при этом скорость ответа не должна превышать 60 секунд, в противном случае система разрывает соединение по тайм ауту.
5. Если предполагаемое количество платежей за услуги подключаемого провайдера, ожидается достаточно интенсивным (до 10 платежей в минуту и более), желательно, чтобы интерфейс спокойно переносил много потоковую коммуникацию до 10-30 одновременных соединений.

2 Основные принципы работы интерфейса

1. Каждый платеж в системе Pegas имеет уникальный идентификатор, который передается провайдеру в переменной **txn_id** – целое число длиной до 20 знаков. По этому идентификатору производится дальнейшая сверка взаиморасчетов и решение спорных вопросов.
2. Сумма платежа принимается от абонента и передается провайдеру в гривнах в переменной **sum** – дробное число с точностью до сотых, в качестве разделителя используется «.» (точка). Если сумма представляет целое число, то оно все равно дополняется точкой и нулями, например – «152.00»
3. В запросе на добавление платежа, система передает дату платежа (под датой платежа в системе подразумевается дата получения запроса от клиента) в переменной **txn_date** – дата в формате ГГГГММДДЧЧММСС. Эту дату необходимо использовать для проведения сверок и бухгалтерских взаиморасчетов. Так как в системе Pegas учет платежей ведется по дате получения запроса от клиента, то и расчеты с провайдером необходимо вести по этой дате. Например ситуация: клиент прислал в систему запрос 31.12.2005 в 23:59:59, учитывая задержку на обработку данных и пересылку информации по каналам связи, система смогла отправить запрос провайдеру 1.1.2006 00:00:05, соответственно платеж будет учтен в системе провайдера в другом отчетном периоде, что вызовет некоторые проблемы при проведении сверок. Чтобы избежать такой ситуации система передает провайдеру дату, в которой нужно учитывать платеж.
4. Провайдер идентифицирует своего абонента по уникальному идентификатору (номер лицевого счета, телефона, логин и т.д.). Перед отправкой провайдеру, идентификатор проходит проверку корректности в соответствии с регулярным выражением, которое должен предоставить провайдер. Идентификатор абонента передается в переменной **account** – строка, содержащая буквы, цифры и спецсимволы, длиной до 200 символов.
5. В случае идентификации абонента по нескольким реквизитам платежа поле **account** будет содержать набор свойств. (Пример: account=param1^value1;param2^value2;param3^value3)
6. Оплата услуг провайдера производится системой в 2 этапа – проверка состояния абонента и непосредственно проведение платежа. Тип запроса передается системой в переменной **command** – строка, принимающая значения «check» и «pay». При проверке статуса, 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. При проведении платежа, провайдер должен произвести пополнение баланса абонента.

7. В случае, если любой из запросов провайдеру завершается ошибкой, то провайдер возвращает код ошибки в соответствии с таблицей приведенной ниже. Все ошибки имеют признак фатальности. Фатальная ошибка означает для системы, 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. Не фатальная ошибка – означает для системы, что повторение запроса с теми же параметрами через некоторый промежуток времени, возможно приведет к успеху. Система будет повторять запросы, завершающиеся не фатальной ошибкой, постоянно увеличивая интервал, пока операция не завершится успехом или фатальной ошибкой или пока не истечет срок жизни клиентского запроса – 24 часа. Отсутствие связи с сервером провайдера является не фатальной ошибкой. Отсутствие в ответе элемента `<result>` (некорректный XML, страница Service temporarily unavailable и т.д.) – является фатальной ошибкой. Клиентские запросы получают отказ с ошибкой 300 – Другая ошибка провайдера.
8. В базе провайдера не должно содержаться двух успешно проведенных платежей с одним и тем же номером **txn_id**. Если система присылает повторно запрос с уже существующим в базе провайдера **txn_id**, то провайдер должен вернуть результат обработки предыдущего запроса.
9. Провайдер возвращает ответ на запросы системе в формате XML со следующей структурой:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <txn_id></txn_id>
  <prv_txn></prv_txn>
  <result></result>
  <comment></comment>
</response>
```

`<response>` - тело ответа

`<txn_id>` - номер транзакции в системе Pegas, который передается провайдеру в переменной `txn_id`.

`<prv_txn>` - Уникальный номер операции пополнения баланса абонента. Этот элемент должен возвращаться после запроса на пополнение баланса. При запроса на проверку состояния абонента, его возвращать не обязательно – он все равно не обрабатывается.

`<result>` - код результата завершения запроса.

`<comment>` - необязательный элемент – комментарий завершения операции.

3 Допустимые запросы

Платежное приложение провайдера «app.cgi», располагается по адресу «provider.com», сервер поддерживает HTTPS соединения на порт 8443.

3.1 Запрос на проверку состояния абонента

Для проверки состояния абонента, система генерирует запрос следующего вида:

```
https://provider.com:8443/app.cgi?command=check&account=1234567
```

Запрос содержит переменные:

command =check	– запрос на проверку состояния абонента
prv_id =1	– идентификатор провайдера (опционально)
account =1234567	– идентификатор абонента в информационной системе провайдера

Использование **prv_id** опционально. Если у Вас несколько услуг с одинаковыми идентификаторами абонента, то идентификатор услуги будет определен этой переменной.

ООО «Пегас Крым»

Адрес: Украина, г. Симферополь, ул. Маяковского 12

Телефон: +38 (067) 650-6688, +38 (0652) 600-894, Факс: +38 (0652) 600-894

E-mail: info@pegaspay.com.ua, В Интернет: <http://www.pegaspay.com.ua>

Ответ провайдера должен выглядеть так:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <result>0</result>
  <name>Абонент И.О</name>
  <balance>10.55</balance>
</response>
```

Или так:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <result>7</result>
  <comment>DB error</comment>
</response>
```

result=0 означает, что указанный в запросе лицевой счет абонента, может быть пополнен. После успешной проверки состояния, система переходит к отправке запроса на пополнение баланса.

result=7 означает, что прием платежа для данного абонента запрещен по техническим причинам. Данная ошибка является фатальной. Клиентский запрос получает отказ с соответствующим кодом ошибки.

3.2 Запрос на пополнение лицевого счета

```
https://provider.com:8443/app.cgi?
command=pay&txn_id=1234567&txn_date=20050815120133&account=1234567&sum=10.45
```

Запрос содержит переменные:

command=pay	– запрос на пополнение баланса абонента
txn_id=1234567	– внутренний номер платежа в системе Pegas
txn_date=20050815120133	– дата учета платежа в системе Pegas
prv_id=1	– идентификатор провайдера (опционально)
account=1234567	– идентификатор абонента в информационной системе провайдера
sum=10.45	– сумма к зачислению на лицевой счет абонента

Пример ответа:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <txn_id>1234567</txn_id>
  <prv_txn>2016</prv_txn>
  <result>0</result>
</response>
```

Или так:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <txn_id>1234567</txn_id>
  <prv_txn>2016</prv_txn>
  <result>1</result>
  <comment>Temporarily database error</comment>
</response>
```

result=0, провайдер сообщает об успешном завершении операции пополнения. Система полностью завершает обработку транзакции, сообщая абоненту об успехе.

ООО «Пегас Крым»

Адрес: Украина, г. Симферополь, ул. Маяковского 12

Телефон: +38 (067) 650-6688, +38 (0652) 600-894, Факс: +38 (0652) 600-894

E-mail: info@pegaspay.com.ua, В Интернет: <http://www.pegaspay.com.ua>

result=1 – временная ошибка обработки запроса с дополнительным комментарием в поле <comment>. Система будет повторять посылку запроса, пока он не завершится успехом, отказом или пока не истечет время жизни платежа.

3.3 Запрос на отмену

```
https://provider.com:8443/app.cgi?command=cancel&prv_txn=2016
```

Запрос содержит переменные:

command=cancel – запрос на отмену транзакции
prv_txn=2016 – внутренний номер платежа в системе провайдера

Пример ответа:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <prv_txn>2016</prv_txn>
  <result>0</result>
</response>
```

Или так:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <prv_txn>2016</prv_txn>
  <result>1</result>
  <comment>Temporarily database error</comment>
</response>
```

result=0, провайдер сообщает об успешном завершении операции отмены. Система полностью завершает обработку транзакции, сообщая абоненту об успехе.

result=1 – временная ошибка обработки запроса с дополнительным комментарием в поле <comment>. Система будет повторять посылку запроса, пока он не завершится успехом либо отказом.

3.4 Запрос на получение сверки платежей

```
https://provider.com:8443/app.cgi?command=verify&date=20050815
```

Запрос содержит переменные:

command=verify – запрос на получение сверки
date=20050815 – дата в формате YYYYMMDD

Пример ответа:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <verify>
    <payment txn_id="12345" prv_txn="2016" account="1234567" amount="10.45"
date="15.08.2005 12:01:33"/>
    <payment txn_id="12346" prv_txn="2017" account="1234568" amount="5.10"
date="15.08.2005 12:01:34"/>
  </verify>
</response>
```

```
<result>0</result>
</response>
```

Или так:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <result>1</result>
  <comment>Temporarily database error</comment>
</response>
```

result=0, провайдер сообщает об успешном завершении операции. Система полностью завершает обработку транзакции, сообщая абоненту об успехе.

result=1 – временная ошибка обработки запроса с дополнительным комментарием в поле <comment>. Система будет повторять посылку запроса, пока он не завершится успехом либо отказом.

3.5 Запрос состояния счета (квота платежей)

```
https://provider.com:8443/app.cgi?command=balance
```

Метод реализуется в случае работы с провайдером по предоплате.

Пример ответа:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <result>0</result>
  <balance>100.12</balance>
</response>
```

Или так:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <result>1</result>
  <comment>Temporarily database error</comment>
</response>
```

4 Ежедневная сверка

В 3:00 по местному времени система начинает процедуру генерации и рассылки ежедневных реестров провайдером. Система генерирует и отправляет по указанному адресу электронный реестр принятых платежей за предыдущий день в приведенном ниже формате:

```
<email адрес, на который будет послана сверка>
<txn_id>;<account>;<сумма>;<терминал>;<дата время активации>;<дата время создания>
.....
<txn_id>;<account>;<сумма>;<терминал>;<дата время активации>;<дата время создания>
Total payments: <кол-во платежей>
Total amount: <общая сумма>
```

Поля разделены знаком «;», дробная часть суммы отделена точкой, дата/время местное, перевод строки может состоять как из символов 0x0D 0x0A (\r\n), так и просто из 0x0D (\n).

Например:

test@pegaspay.com.ua

```
Payments report:

12345;1234567;10.45;1;15.08.2005 12:01:33;15.08.2005 12:01:33
12346;1234568;5.10;1;15.08.2005 12:01:34;15.08.2005 12:01:34

Total payments: 2
Total amount: 15.55
```

Система включает в реестр только успешно проведенные платежи.

При реализации пункта 3.4 происходит автоматическая сверка платежей и в случае нахождения расхождений, к отчету прикрепляется список с найденными разногласиями.

Например:

```
Diff errors at: 15.08.2005

Remote service contains unknown payments:

2015;1234567;10.00

Amount: 10.00

Total diff amount: 10.00
```

Если в реестре отсутствуют платежи, которые проведены в базе провайдера, то провайдер должен отменить эти платежи у себя в базе. В случае, если в реестре содержатся платежи, которых нет в базе провайдера, то необходимо связаться с контактным лицом в компании Pegas, указанным в договоре для выяснения ситуации и принятия решения.

5 Список кодов завершения

При обработке запросов от системы, провайдер должен сопоставить все возникающие в его приложении ошибки с приведенным ниже списком и возвращать соответствующие коды в элементе <result>. Знак «●» в столбце фатальность – показывает то, как система будет интерпретировать данную ошибку.

Код ошибки	Комментарий	Фатальность
0	ОК	
1	Временная ошибка. Повторите запрос позже	
4	Неверный формат идентификатора абонента	●
5	Идентификатор абонента не найден (Ошиблись номером)	●
7	Прием платежа запрещен провайдером	●
8	Прием платежа запрещен по техническим причинам	●
79	Счет абонента не активен	●
241	Сумма слишком мала	●
242	Сумма слишком велика	●
243	Невозможно проверить состояние счета	●
251	Отмена платежа невозможна	●
300	Другая ошибка провайдера	●