

ИНТЕРФЕЙС ПОДКЛЮЧЕНИЯ ПРОВАЙДЕРОВ



**1. Общие сведения**

**2. Требования к интерфейсу провайдера**

**3. Основные принципы работы интерфейса**

**4. Пример запроса на проверку счета абонента.**

**Приложение А. Список кодов завершения**

**Приложение В. Описание полей и их размеров**

## **1. Общие сведения**

Данный документ содержит описание протокола взаимодействия сервера обработки платежей с серверами провайдеров. Документ предназначен для IT служб провайдеров, перед которыми стоит цель подключиться к процессинговой системе ООО «Свифт Гарант» (платежная система Банк24).

В документе будет приведено описание и примеры запросов отправляемых на серверы провайдеров при обработке платежей, а также описание и примеры ожидаемых ответов.

## 2. Требования к интерфейсу провайдера

1. Интерфейс должен принимать запросы по протоколу HTTPS с IP адреса 62.149.15.210.
2. Интерфейс должен обрабатывать XML в кодировке UTF-8, передаваемый системой методом POST.
3. Интерфейс должен формировать ответ системе в формате XML в кодировке UTF-8.
4. Обмен информацией ведется в режиме запрос-ответ, при этом скорость ответа не должна превышать 60 секунд, в противном случае система разрывает соединение по таймауту.
5. Если предполагаемое количество платежей за услуги подключаемого провайдера, ожидается достаточно интенсивным (до 10 платежей в минуту и более), желательно, чтобы интерфейс спокойно переносил многопоточковую коммуникацию до 10-15 одновременных соединений.
6. Если реализация протокола на PHP, то XML запрос с нашего сервера нужно забирать методом `file_get_contents("php://input")`. Для того чтобы работал этот метод нужно в настройках PHP установить флаг `always_populate_raw_post_data = On`, который находится в файле `php.ini`

### 3. Основные принципы работы интерфейса

1. Каждый запрос в системе имеет уникальный идентификатор, который передается провайдеру в переменной **transactionID** – целое 18-ти разрядное число (идентификатор никак не участвует во взаиморасчетах и используется только при решении спорных вопросов, для уникальной идентификации каждого XML запроса посланного с сервера).

2. Каждый запрос на проведение платежа в системе имеет уникальный идентификатор платежа, который передается провайдеру в переменной **payID** – строка длиной до 64 знаков. По этому идентификатору производится дальнейшая сверка взаиморасчетов и решение спорных вопросов.

1. Сумма платежа принимается от абонента и передается провайдеру в переменной **amount** – целое число, которое отображает сумму в **копейках**. Например – сумма платежа «152.25» будет передаваться как «15225».

2. В запросе на проведение платежа, система передает дату платежа (под датой платежа в системе подразумевается дата начала операции на сервере) в переменной **payTimestamp** – дата в формате **YYYYMMDDHHMISS**. Эту дату необходимо использовать для проведения бухгалтерских взаиморасчетов. Например, ситуация: система начала проведение платежа 31.12.2009 в 23:59:59, учитывая задержку на обработку данных и пересылку информации по каналам связи, запрос был получен провайдером 01.01.2010 00:00:02, соответственно платеж будет учтен в системе провайдера в другом отчетном периоде, что вызовет проблемы при проведении сверок. Чтобы избежать такой ситуации система передает провайдеру дату, в которой нужно учитывать платеж.

3. 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.

4. Передача информации о платеже провайдеру 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 (запрос «check»), провайдер должен проверить наличие в своей базе абонента с указанным идентификатором и возможность проведения платежа по данному абоненту. При проведении платежа (запрос «pay»), провайдер должен произвести пополнение баланса абонента.

5. В случае если любой из запросов провайдеру завершается ошибкой, то провайдер возвращает код ошибки в соответствии с таблицей приведенной, ниже (Приложение А). Все ошибки имеют признак фатальности. Для системы 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. Система будет повторять запросы, завершающийся нефатальной ошибкой, пока операция не завершится успехом или фатальной ошибкой. Отсутствие связи с сервером провайдера является нефатальной ошибкой. Отсутствие в ответе элемента <result> (некорректный XML, страница Service temporarily unavailable и т.д.) – является фатальной ошибкой.

6. **В базе провайдера не должно содержаться двух успешно проведенных платежей с одним и тем же номером payID.** Если наш сервер присылает повторный запрос на пополнение с **payID** который уже существует, провайдер должен вернуть положительный ответ (**result=0**).

## 4. Пример запроса на проверку счета абонента.

Для проверки состояния абонента, система генерирует XML запрос следующего вида:

```
<?xml version="1.0" encoding="UTF-8"?>
<commandCall>
  <login>platezhka</login>
  <password>1234567</password>
  <command>check</command>
  <transactionID>1234567890123</transactionID>
  <payID>55830367279006</ payID >
  <payElementID>0</payElementID>
  <account>1234567890</account>
</commandCall >
```

Запрос содержит переменные:

**command**=check – запрос на проверку состояния абонента

**login** = platezhka – логин используемый для идентификации системы

**password** =1234567 – пароль используемый для идентификации системы

**account**=1234567890 – идентификатор абонента в информационной системе провайдера

**transactionID**=1234567890123 – номер транзакции на стороне системы

**payID** = 55830367279006 – внутренний номер платежа в системе

**payElementID**=0 - номер сервиса в системе провайдера, используется для случаев если провайдер предоставляет возможность пополнения нескольких слуг. Например: интернет и телефон (если сервис один - поле не анализируется)

Ответ провайдера должен выглядеть так:

```
<?xml version="1.0" encoding="UTF-8"?>
<commandResponse>
  <extTransactionID>1234567</extTransactionID>
  <account>1234567890</account>
  <result>0</result>
  <comment></comment>
</commandResponse>
```

Возвращение **result**=0 на запрос «check» свидетельствует о том, что лицевой счет абонента может быть пополнен. После успешной проверки состояния счета абонента система переходит к формированию и отправке запроса на пополнение баланса (запрос «pay»). Другие коды тега **result** смотрите в "Приложение А. Список кодов завершения".

В необязательном поле **comment** содержится служебный комментарий.

Поле **extTransactionID** ответа содержит номер запроса на стороне провайдера (если у провайдера запросы не логируются, то номер запроса не передается).

При необходимости, дополнительную информацию о платеже можно передать в теге **fields**.

Ответ провайдера тогда должен выглядеть так:

```
<?xml version="1.0" encoding="UTF-8"?>
<commandResponse>
  <account>1234567890</account>
  <result>0</result>
  <fields>
    <field1 name="FIO">Иванов Иван Петрович</field1>
    <field2 name="balance">152.17</field2>
    ...
    <fieldN name="nameN">valueN</fieldN>
  </fields>
  <comment></comment>
</commandResponse>
```

В необязательных полях **field1**, **field2**... **fieldN** содержится информация, которую необходимо передать системе. Эта информация может быть показана пользователю при совершении платежа или использоваться при взаиморасчетах.

На текущий момент поддерживаются следующие наименования служебных полей:

**Message** - произвольный текст который отобразится клиенту на экране.

**FIO** - фио клиента.

**Balance** - остаток на счету

**FixedPrice** - заполняется, если клиенту необходимо заплатить фиксированную сумму платежа.

**SubProviderId** - если в компании есть несколько получателей платежей, то в этом параметре передается идентификатор получателя на который нужно перечислять денежные средства.

**pay\_min\_override** - минимальная сумма платежа (параметр не обязательный, передается только если нужно ограничивать минимальную сумму платежа)

**pay\_max\_override** - максимальная сумма платежа (параметр не обязательный, передается только если нужно ограничивать максимальную сумму платежа)

5. **Пример запроса на пополнение лицевого счета**  
Для подтверждения платежа, система генерирует XML запрос следующего вида:

```
<?xml version="1.0" encoding="UTF-8"?>
<commandCall>
  <login>platezhka</login>
  <password>1234567</password>
  <command>pay</command>
  <transactionID>1234567890123</transactionID>
  <payTimestamp>20101008162022</ payTimestamp >
  <payID>55830367279006</ payID >
  <payElementID>0</payElementID>
  <account>1234567890</account>
  <amount>9800</amount>
  <terminalId>11352</terminalId>
</commandCall >
```

Запрос содержит переменные:

- command**= pay – запрос на пополнение баланса абонента
- login** = platezhka – логин используемый для идентификации системы
- password** = 1234567 – пароль используемый для идентификации системы
- account**= 1234567890 – идентификатор абонента в информационной системе провайдера
- amount** = 9800 – сумма к зачислению на лицевой счет абонента
- payID** = 55830367279006 – внутренний номер платежа в системе
- payTimestamp** = 20101008162022 – дата учета платежа в системе
- transactionID**=1234567890123 – номер транзакции на стороне системы
- terminalId**=11352 – номер терминала на котором было произведено пополнение
- payElementID**=0 - номер сервиса в системе провайдера (если сервис один - поле не анализируется)

Ответ провайдера должен выглядеть так:

```
<?xml version="1.0" encoding="UTF-8"?>
< commandResponse >
  <extTransactionID>1234567</extTransactionID>
  <account>1234567890</account>
  <result>0</result>
  <comment>OK</comment>
```



```
</ commandResponse >
```

Возвращая result=0 на запрос «pay», провайдер сообщает об успешном завершении операции пополнения баланса. Система полностью завершает обработку данной транзакции.

В необязательном поле comment содержится служебный комментарий.

Поле extTransactionID ответа содержит номер транзакции на стороне провайдера.

**Внимание!** Если наш сервер присылает повторный запрос на пополнение с payID который уже существует, провайдер должен вернуть положительный ответ (result=0).

## Приложение А. 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	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	+
90	Проведение платежа не окончено	
300	Другая ошибка провайдера	+

## Приложение В. Описание полей и их размеров

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.

Поле	Тип данных	Описание
login	varchar(50)	Логин
password	varchar(128)	Пароль
command	varchar(10)	Тип команды
transactionID	numeric(18,0)	Уникальный идентификатор запроса
payTimestamp	varchar(14)	дата начала платежа в формате <b>YYYYMMDDHHMISS</b>
payID	varchar(64)	Уникальный номер платежа.
terminalId	int	Номер терминала на котором было проведено пополнение
payElementID	int	Номер сервиса в системе провайдера
account	varchar(50)	Идентификатор абонента в информационной системе провайдера
amount	int	целое число, которое отображает сумму в копейках. Например – сумма платежа «152.25» будет передаваться как «15225».
result	int	Результат выполнения операции.
extTransactionID	int	номер транзакции на стороне провайдера
comment	varchar(128)	служебный комментарий
field1, field2... fieldN	varchar(128)	Необязательные поля, в которых провайдер передает дополнительную информацию о введенном идентификаторе абонента