Протокол взаимодействия (онлайн)

Требования к интерфейсу провайдера

1. Интерфейс должен принимать запросы по протоколу HTTPS с IP-адресов подсети:
   194.187.247.152
   194.187.245.108
   197.187.244.108
2. Интерфейс должен обрабатывать параметры, передаваемые системой методом GET
3. Интерфейс должен формировать ответ системе в формате в JSON или 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 **15 секунд**, в противном случае система разрывает соединение по таймауту.
5. Если предполагаемое количество платежей за услуги подключаемого провайдера, ожидается интенсивным (до 10 платежей в минуту и более), необходимо, чтобы интерфейс поддерживал многопоточную коммуникацию до 10-15 одновременных соединений.

Основные принципы работы интерфейса

В информационном обмене участвуют:

- со стороны Банка – Kaspi.kz
- со стороны провайдера или же Компании поставщика услуг (далее КПУ) - уполномоченные подразделения компаний

1. Каждый платеж в системе Kaspi.kz имеет уникальный идентификатор, который передается провайдеру в переменной **txn_id** – ID запроса, целое число длиной до 18 знаков. По этому идентификатору производится дальнейшая сверка взаиморасчетов и решение спорных вопросов.
2. 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 бухгалтерских взаиморасчетов. Так как в системе Kaspi.kz 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 31.12.2018 в 23:59:59, учитывая задержку на обработку данных и пересылку информации по каналам связи, система смогла отправить запрос провайдеру 1.1.2019 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.
3. Провайдер идентифицирует своего абонента или же заказ по уникальному идентификатору (номер лицевого счета, телефона, номер накладной, 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. Возможна передача следующих дополнительных полей в строке запроса:
   data1,data2,…,dataN – дополнительные поля, передаваемые провайдером - строки, содержащие буквы, цифры и спецсимволы.
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 выполнить внутренние проверки идентификатора, суммы платежа в соответствии с принятой логикой пополнения лицевых счетов через платежные системы. При проведении платежа (запрос «pay»), провайдер должен произвести пополнение баланса абонента. **txn_id** у «check» и «pay» запросов **будут различными**.

5. Сумма платежа принимается от клиента плательщика и передается провайдеру в тенге 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 – «200.00», в запросе **check** значение в переменной **sum** является фиктивным, передается с терминала по умолчанию и **его обрабатывать не нужно**.
6. В случае если любой из запросов провайдеру завершается ошибкой, то провайдер возвращает код ошибки (см. пункт 9).
7. В базе провайдера не должно содержаться двух успешно проведенных платежей с одним и тем же номером **txn_id**. Если система повторно присылает запрос с уже существующим в базе провайдера **txn_id**, то провайдер должен вернуть результат обработки предыдущего запроса.
8. 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>
 <sum></sum>
 <result></result>
 <comment></comment>
</response>
```

Либо в **JSON** формате:

```
{
      "txn_id": "",
   "prv_txn_id": "",
      "result": "",
      "sum": "",
      "comment": ""
}
```

- **<response>** – тело ответа
- **<txn_id>** – номер транзакции в системе, который передается провайдеру в переменной txn_id.
- **<prv_txn>** – уникальный номер оплаты (в базе провайдера), целое число длиной до 20 знаков. Этот элемент должен возвращаться провайдером после запроса на оплату (запроса «pay»). При ответе на запрос на проверку состояния абонента (запрос «check») его возвращать не нужно – не обрабатывается.
- **<sum>** – необходимая сумма платежа по запрашиваемому лицевому счету, номеру заказа и т.д., 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 – «200.00».
- **<result>** – код результата завершения запроса.
- **<comment>** – необязательный элемент – комментарий завершения операции.
- При необходимости, дополнительную информацию о платеже можно передать в теге **fields**.

Ответ провайдера в формате **XML** должен выглядеть так:

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

Либо в **JSON** формате:

```
{
      "txn_id": "1234567",
      "result": 0,
      "comment": "",
"fields":   {
         "field1": {
            "@name": "name1",
            "#text": "value1"
         },
         "field2": {
            "@name": "name2",
            "#text": "value2"
         },
         "fieldN": {
            "@name": "nameN",
            "#text": "valueN"
         }
      }
}
```

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

9. Возможные коды состояния/ошибки запросов:

0	абонент/счёт/заказ найден и доступен для пополнения/оплаты
1	"абонент/счёт не найден" или "заказ не найден", если запрос check был на проверку состояния
2	заказ отменен
3	заказ уже оплачен
4	платеж в обработке
5	Другая ошибка провайдера

Пример запроса на проверку состояния счета абонента

В качестве примера платежное приложение провайдера payment_app.cgi, располагается по адресу example.com, сервер поддерживает HTTPS соединения на порт 443. Для проверки состояния абонента, система kaspi.kz генерирует запрос следующего вида:

https://example.com/payment_app.cgi?command=check&txn_id=1234567&account=4957835959&sum=200.00

Запрос содержит переменные:

- **command=check** – запрос на проверку состояния абонента/номера заказа
- **txn_id=1234567** – внутренний номер запроса в системе Kaspi.kz
- **account=4957835959** – идентификатор абонента/номера заказа в информационной системе провайдера
- **sum=200.00** – сумма к зачислению на лицевой счет абонента (в запросе check значение в переменной sum является фиктивным, передается с терминала по умолчанию и **его обрабатывать не нужно**).

Возможна передача следующих дополнительных полей в строке запроса:

data1,data2,…,dataN – дополнительные поля, передаваемые провайдером - строки, содержащие буквы, цифры и спецсимволы.

В строке запроса могут быть переданы одно или несколько дополнительных полей:

https://example.com/payment_app.cgi?command=check&txn_id=1234567&account=4957835959&sum=200.00&data1=123456

Ответ провайдера в формате **XML** должен выглядеть так:

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

Либо в **JSON** формате:

```
{
    "txn_id": "123467",
    "result": 0,
    "comment": ""
}
```

Возвращение **result=0** на запрос **«check»** свидетельствует о том, что лицевой счет абонента с соответствующим ему номером **txn_id** может быть пополнен. После успешной проверки состояния счета абонента система переходит к формированию и отправке запроса на пополнение баланса (запрос «pay»).

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

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

Ответ провайдера в формате **XML** должен выглядеть так:

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

```
   <field1 name="name1"> value1</field1>
   <field2 name="name2"> value2</field2>
   <fieldN name="nameN"> valueN</fieldN>
  </fields>
  <comment></comment>
</response>
```

Либо в **JSON** формате:

```
{
      "txn_id": "1234567",
      "result": 0,
      "comment": "",
"fields":   {
         "field1": {
            "@name": "name1",
            "#text": "value1"
         },
         "field2": {
            "@name": "name2",
            "#text": "value2"
         },
         "fieldN": {
            "@name": "nameN",
            "#text": "valueN"
         }
      }
}
```

В необязательных полях **field1, field2… fieldN** содержится информация, которую необходимо передать системе. Эта информация может быть показана пользователю при совершении платежа. В ответе провайдера могут содержаться все дополнительные поля или только часть из них.

Пример запроса на пополнение лицевого счета

Для подтверждения платежа, система генерирует запрос следующего вида:

https://example.com/payment_app.cgi?command=pay&txn_id=1234567&txn_date=20110101120005&account=4957835959&sum=500.00

Запрос содержит переменные:

- **command=pay** – запрос на пополнение баланса абонента/номера заказа
- **txn_id=1234567** – внутренний номер платежа в системе kaspi.kz
- **txn_date=20110101120005** – дата учета платежа в системе kaspi.kz
- **account=4957835959** – идентификатор абонента/номера заказа в информационной системе провайдера
- **sum=500.00** – сумма к зачислению на лицевой счет абонента (сумма, внесенная абонентом в терминал)

Пример ответа в формате **XML**:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<response>
  <txn_id>1234567</txn_id>
  <prv_txn>2016</prv_txn>
  <sum>500.00</sum>
  <result>0</result>
  <comment>OK</comment>
</response>
```

Пример ответа в **JSON** формате:

```
{
      "txn_id": "1234567",
      "prv_txn_id": "2016",
      "result": 0,
      "sum": 500.00,
      "comment": "OK"
}
```

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.

Общая схема взаимодействия