Протокол информационно-технологического взаимодействия
между Оператором и Получателем по http протоколу.
Версия 3 Редакция 2.2 от 01.05.2018г.

Для осуществления качественного обслуживания Плательщика сервис должен обеспечить выполнение следующих требований:

1. проверить правильности реквизитов платежа, указанных Плательщиом, перед началом процедуры оплаты;
2. передать информацию Получателю о принятом платеже;
3. передать Плательщику результат пополнения Получателем абонентского лицевого счета или результат оплаты покупки в интернет-магазине или сервисе.

Описание интерфейса, в полной мере реализующего эти требования:

Сервис передает Получателю введенные клиентом данные, Получатель проверяет их по базе своих абонентов или номеров заказов интернет-магазина/сервиса и отвечает сервису есть такой абонент/заказ или нет. Реализация: http post запрос сервиса на страницу (скрипт) сервера Получателя (п.1).

После подтверждения сервисом правильности указанных реквизитов Плательщик приступает к оплате. После получения оплаты сервис извещает Получателя о приеме платежа отправкой http извещения на страницу/скрипт Получателя. Реализация: http post запрос сервиса на страницу сервера Получателя (п.2). Защита информации от несанкционированного изменения – хеширование данных методом MD5.

Получив http извещение сервиса, Получатель должен пополнить абонентский лицевой счет или отметить оплату заказа и вернуть сервису ответ, содержащий результат операции пополнения или отметки об оплате.
Реализация: ответ на запрос сервиса на страницу сервера Получателя (п.2).

п.1. Проверка реквизитов абонента **сервисом**.

Метод передачи данных – POST
Передаваемые поля:

details - строка состоящая из реквизитов платежа, разделенных символом «;» (количество реквизитов, значения и формат реквизитов зависят от количества запрашиваемых Сервисом у Плательщика данных и определяется дополнительно между Сервисом и Получателем);
amount - строка состоящая из суммовых реквизитов платежа, разделенных символом «;»(количество реквизитов, значения и формат реквизитов реквизитов зависят от количества запрашиваемых Сервисом у Плательщика данных и определяется дополнительно между Сервисом и Получателем);
requesttype – дополнительное поле, указывающее тип запроса. В хеш не включается. Значение поля в запросе: "accpres" - проверка реквизитов абонента.

hash - строка из всех полей и секретного слова в конце без пробелов между полями. Метод хеширования - MD5 32 бита.

Принципалом проверяется наличие лицевого счета абонента в учетной системе и возможность его пополнения. Сервер Принципала возвращает ответ на запрос проверки реквизитов.

Варианты ответа сервера Принципала:
accpres1 - все реквизиты указаны правильно;
accpres2 - несоответствие указанных реквизитов;
accpres3 - нет такого лицевого счета абонента;
accpres4 - ошибка при проверке реквизитов, повторить запрос позже (при получении такого ответа, так же как при отсутствии ответа от сервера, пользователю будет предложено произвести попытку оплаты через некоторое время);
accpres5 - несовпадение хеша (нарушение целостности данных).

п.2. Отправка сервисом извещения о приеме платежа и получение ответа с кодом завершения операции.

Метод передачи данных - POST
Передаваемые поля:

details - строка состоящая из реквизитов платежа, разделенных символом «;» (количество реквизитов, значения и формат реквизитов зависят от количества запрашиваемых Сервисом у Плательщика данных и определяется дополнительно между Сервисом и Получателем);
amount - строка состоящая из суммовых реквизитов платежа (десятичная точка, 2 знака после точки), разделенных символом «;»(количество реквизитов, значения и формат реквизитов зависят от количества запрашиваемых Сервисом у Плательщика данных и определяется дополнительно между Сервисом и Получателем);
date - дата и время формирования заказа на сайте сервиса (YYYY-MM-DD HH:MM:SS);
order - номер заказа в Сервисе.
requesttype – дополнительное поле, указывающее тип запроса. В хеш не включается. Значение поля в запросе: "accpay" – извещение о приеме платежа.

hash - строка из всех полей и секретного слова в конце без пробелов между полями. Метод хеширования - MD5 32 бита.

Варианты ответа сервера Принципала:
accpay1 - средства зачислены;
accpay2 - средства будут зачислены вручную;
accpay3 - средства не зачислены, ошибка в реквизитах платежа;
accpay4 - ошибка при зачислении средств, повторить запрос позже (не регламентированная ситуация);
accpay5 - несовпадение хеша (нарушение целостности данных).

Пример ответа сервера Принципала на php:

```
<?php
…
Тело скрипта, пополняющего лицевой счет абонента или делающего отметку об оплате заказа.
Пополнение успешно или заказ отмечен как оплаченный.
…
echo ' accpay1';
?>
```

Проверка целостности и достоверности данных с использованием хеширования.

В данном примере используется вычисление хеша по алгоритму MD5.
Секретное слово известно только отправителю и получателю (Указывается в настройках на сайте Сервиса).
Хеш отправляется получателю вместе с данными как значение поля hash.

Порядок отправки данных:

1. Формирование строки, содержащей данные + секретное слово;
2. Вычисление хеша этой строки;
3. Отправка данных и хеша.

Порядок приема и проверки данных:

1. Чтение принятых данных и хеша;
2. Формирование строки, содержащий полученные данные + секретное слово;
3. Вычисление хеша данных;
4. Сравнение вычисленного хеша с полученным. Если хеши совпадают – то принятые данные соответствуют отправленным, если не совпадают – полученные данные отличаются от отправленных (т.е. изменены).

Примеры реализации на php:

Пример 1. Вычисление хеша при отправке данных:

Отправляемые данные: details, amount, date, order.
Секретное слово: “SecretWord“

```
<?php
      $str="$details$amount$date$orderSecretWord";
      $hash=md5($str) ;
?>
```

hash отправляется получателю вместе с данными.

Пример 2. Проверка целостности данных при приеме:

Полученные данные: details, amount, date, order, hash.

```
<?php
     $str="$details$amount$date$orderSecretWord ";
     $myhash=md5($str) ;
    if ($myhash!=$ hash) {
       exit;  // неправильные данные обрабатывать не надо или скрипт ответа серверу или инфы админа.
       }
    else {
       // скрипт обработки данных, коррекции счета абонента, ответа сервису
       }
?>
```