API сервиса http.

Метод GET.

1)Для отправки sms через партнерский http интерфейс методом GET требуется сформировать следующий запрос http://api.super-gate.ru:5762/http_receiver?log=login&pwd=pass&time=ГГГГ-ММ-ДД ЧЧ:ММ:СС&count=X&dn1=msisdn1&dn2=msisdn2…&dnX=msisdnX&sn=signature&text=message где

login – логин партнера, введенный при регистрации на сайте Super-gate. pass – пароль партнера, введенный при регистрации на сайте Super-gate.

time – дата и время отправки сообщения (по Москве). Возможны 2 варианта указания времени. В виде строки (как указано в примере) и в формате unixtime. Оба варианта являются равноправными и автоматически распознаются при обработке запроса.

X – количество номеров абонентов в сформированном запросе. msisdn1 – Номер первого абонента. msisdn2 –Номер второго абонента.

…

msisdnX –Номер последнего абонента. signature – подпись сообщения (от кого пришло).

message – текст сообщения.

!!! обратите внимание на кодировку, текст должен быть в windows-1251

После обработки полученного запроса будет выдан ответ

В случае отсутствия ошибок с кодом 200 и текстом msg_id_1=id_1;msg_id_2=id_2;… msg_id_X= id_X

где

id_1 – уникальный идентификатор, присвоенный платформой агрегатора сообщению msisdn1. id_2 – уникальный идентификатор, присвоенный платформой агрегатора сообщению msisdn2. id_X – уникальный идентификатор, присвоенный платформой агрегатора сообщению msisdnX.

Примечание: возвращаемые идентификаторы служат для получения результатов отправки sms.

В случае обнаружения ошибки будет возвращен ее код:

400 - неизвестная ошибка

401 – партнер не авторизован

402 – недостаточно средств на счете

405 – запрошенный метод не поддерживается

413 – размер пакета больше максимального

512 – некорректная подпись отправителя

511– неразрешенная подпись отправителя

3)Для получения результатов отправки sms методом GET требуется сформировать следующий запрос http://api.super-gate.ru:5761/check_delivery? log=login&pwd=pass&count=X&msg_id1=id_1;&msg_id2=id_2;…msg_idX=id_X где

login – логин партнера, введенный при регистрации на сайте Super-gate.

pass – пароль партнера, введенный при регистрации на сайте Super-gate.

X – количество уникальных идентификаторов по которым требуется получить отчет о результатах.

id_1 – уникальный идентификатор №1.

id_2 – уникальный идентификатор №2.

id_X – уникальный идентификатор №X.

После обработки полученного запроса будет выдан ответ id_1=Res;id_2=Res;…id_X=Res

где id_1, id_2…id_X НЕ ТЕКСТОВЫЕ ОБОЗНАЧЕНИЯ, а реальные уникальные идентификаторы, переданные платформе агрегатора в предыдущем запросе.

Res – одно из следующих значений

-1 — неизвестный уникальный идентификатор

1 – запрос находится в обработке

2 – сообщение успешно доставлено абоненту

4 - – сообщение не удалось доставить абоненту

Примеры:

простейшая функция отправки смс на PHP:

```
/**
* Функция отправки СМС через сервис api.super-gate.ru
* @param type $phone //10-значный номер телефона
* @param type $text // текст СМС
 */
function sendSmsUserSGate($phone, $text){
 $phone = substr(trim($phone), -10);
 //обратите внимание на кодировку, текст должен быть в windows-1251
 //в php можете использовать iconv();
 //$text = iconv("Кодировка текста", "windows-1251", $text);
 $text = urlencode($text);// кодируем строку для корректной передачи

 $sphone = 'my_signature';// доступное имя отправителя
$time = time();// время отправки сообщения.
 // Возможны 2 варианта указания времени.
 // В виде строки (ГГГГ-ММ-ДД ЧЧ:ММ:СС) и в формате unixtime.
 // Оба варианта являются равноправными и автоматически распознаются при обработке
запроса.
 //
 // формирование запроса
 $url = 'http://api.super-gate.ru:5762/http_receiver?log=my_login&pwd=my_pass&time='.
$time.'&count=1&dn1='.$phone.'&sn='.$sphone.'&text='.$text;
 // отправка запроса
 $res = file_get_contents($url);
 // запись результата, для просмотра статуса.
writeLogSms($res,$phone,$text);// можете использовать БД
 // в $res записывается строка вида: msg_id_1..x = msg_id. необходимо регулярным выражением
вытащить значение
 // Пример:  msg_id_1=247188065 , для проверки статуса необходимо извлечь 247188065
 // preg_match("|msg_id_(\d)=(\d{0,})|u", $res, $out);
 // в результирующем массиве $out = Array ( [0] => msg_id_1=157435033 [1] => 1 [2] => 157435033
) нам нужен $out[2]

}
```

Простейшая функция проверки статуса СМС-сообщения отправленного через API на PHP

```
/**
* Функция просмотра статуса СМС через сервис api.super-gate.ru
* @param type sms_id уникальный идентификатор СМС, полученный в результате выполнения
   предыдущей функции.
 */ function checkStatusSmsSGate($sms_id =
'') {

        if ($sms_id != '') {
                //Формирование url для запроса статуса
                $url = 'http://api.super-gate.ru:5761/check_delivery?
```

```
log=my_login&pwd=my_pass&count=1&msg_id1=' . $sms_id;
                //запрос статуса возвращает строку id_1=Res, где Res - статус
                $status = file_get_contents($url);
                //обработка результата, Пишем в $out значение
                статуса preg_match("|".$sms_id."=(\d{1})|u", $status,
                $out); return $out[1];//статус
                // в $out[1] содержится одно из следующих значений:
                // -1 — неизвестный  уникальный идентификатор
                // 1 – запрос находится в обработке
                // 2 – сообщение успешно доставлено абоненту
                // 4 - – сообщение не удалось доставить абоненту
        }
        return false;
}
```

Простейший вариант функции проверки баланса на PHP

```
 /**
* Функция проверки баланса
* @return type
 */
function chekBalansSGate() {
        $url = 'http://super-gate.ru/gateway/index.php?
login=my_login&password=my_pass&method=check_balance&type=int';
        $result = file_get_contents($url);
        return $result;
}
```

Построение дерева (массива) деф-кодов. В итоге вернет большой ассоциативный массив.

```
/**
 * @return array
 */
function setTreeDef()
{
   // Глобальное подключение к SQL
   global $link;
   $phone_codes_array = array();

   // Запрос деф-кодов из MYSQL (code, MCC, MNC)
   $query = 'SELECT * FROM my_table';
   $result = mysqli_query($link, $query);
   while ($res = mysqli_fetch_assoc($result)) {
      // Длина строки кода
      $len_send = strlen($res['code']);
      $tmp = &$phone_codes_array;
```

```
        for ($j = 0; $j < $len_send; $j++) {
            $tmp = &$tmp[$res['code'][$j]];
        }
        $tmp['v'] = array('MCC' => $res['MCC'], 'MNC' => $res['MNC']);
    }
    return $phone_codes_array;
}
```

Определение оператора по номеру телефона используя дерево деф-кодов.

```
/**
 * @param $phone
 * @param $phone_codes_array
 * @return array
 */
function getPhoneMccMnc($phone, $phone_codes_array)
{
    $phone = (string)$phone;
    $len_phone = strlen($phone);
    $tmp = &$phone_codes_array;
    $MCC = 0;
    $MNC = 0;
    for ($j = 0; $j < $len_phone; $j++) {
        if (is_array($tmp)) {
            $tmp = &$tmp[$phone[$j]];
            if (isset($tmp['v'])) {
                $MCC = $tmp['v']['MCC'];
                if ($tmp['v']['MNC'])
                    $MNC = $tmp['v']['MNC'];
            }
        } else
            break;
    }
    return array('MCC' => $MCC, 'MNC' => $MNC);
}
```

Получение идентификатора параметра из глобальной переменной по полному имени, с привязкой к пользователю.

```
/**
 * @param $name
 * @param $ID_USER
 * @return int|mixed|string
 */
function getIdFromName($name, $ID_USER)
{
    if ($name) {
        global $link, $GLOBAL_CACHE_DATA;
```

```
    // Проверяет на наличие данного идентификатора.
    if (isset($GLOBAL_CACHE_DATA['getIdName'][$ID_USER][$name]))
      return $GLOBAL_CACHE_DATA['getIdName'][$ID_USER][$name];
    else {
      // Ищем запись в БД.
      $query = 'select id from my_table where id_user="' . $ID_USER . '" and name="' . $name . '"';
      $result = mysqli_query($link, $query);
      if (mysqli_num_rows($result)) {
        $row = mysqli_fetch_assoc($result);
        $id = $row['id'];
      } else {
        // Вставляем запись в БД в случае отсутствия.
        $query = 'insert into my_table set name="' . $name . '", id_user="' . $ID_USER . '"';
        mysqli_query($link, $query);
        $id = mysqli_insert_id($link);
      }
      $GLOBAL_CACHE_DATA['getIdName'][$ID_USER][$name] = $id;
      return $id;
    }
  } else
    return 0;
}
```

Отправка XML

```
/**
 * @param $xml
 * @param $href
 * @return bool|string
 */
function sendXML($xml, $href)
{
  $xml = str_replace("
", '', $xml);

  $ch = curl_init();
  curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-type: text/xml; charset=utf-8'));
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
  curl_setopt($ch, CURLOPT_CRLF, true);
  curl_setopt($ch, CURLOPT_POST, true);
  curl_setopt($ch, CURLOPT_POSTFIELDS, $xml);
  curl_setopt($ch, CURLOPT_URL, $href);
  $result = curl_exec($ch);
  curl_close($ch);

  return $result;
```

```
}
```

Отправка GET

```
/**
 * @param $url
 * @return bool|string
 */
function sendGet($url)
{
   $ch = curl_init();
   curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
   curl_setopt($ch, CURLOPT_CRLF, true);
   curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
   curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
   curl_setopt($ch, CURLOPT_URL, $url);
   $result = curl_exec($ch);
   curl_close($ch);

   return $result;
}
```

Перевод объекта в массив

```
/**
 * @param $XML
 * @return array
 */
function ObjInArray($XML)
{
   $array = array();
   $array_obj = (array)$XML;
   $count_k = array();
   foreach ($XML as $k => $v) {
      if (!isset($count_k[$k]))
         $count_k[$k] = 0;
      $tmp =& $array[$k][$count_k[$k]];
      if (is_string($array_obj[$k]))
         $tmp['v'] = $array_obj[$k];
      elseif (is_string($array_obj[$k][$count_k[$k]]))
         $tmp['v'] = $array_obj[$k][$count_k[$k]];
      else {
         $res = ObjInArray($v);
         if (count($res) > 0)
            $tmp['v'] = $res;
      }
      $atr = (array)$v;
```

```
        if (isset($atr['@attributes']))
            $tmp['a'] = $atr['@attributes'];
        $count_k[$k]++;
    }
    return $array;
}
```

Построение ссылки для WAP PUSH

```
/**
 * @param $url
 * @return string
 */
function setUrlWAP($url)
{
    $num_1 = strpos($url, '//');
    if ($num_1 === FALSE)
        $return_url = chr(0x0B);
    else {
        $href = substr($url, 0, $num_1);
        if ($href == 'http:' or $href == 'https:') {
            $www = substr($url, $num_1 + 2, 4);
            if ($href == 'http:' and $www == 'www.') {
                $return_url = chr(0x0D);
                $url = substr($url, 11);
            } elseif ($href == 'http:' and $www != 'www.') {
                $return_url = chr(0x0C);
                $url = substr($url, 7);
            } elseif ($www == 'www.') {
                $return_url = chr(0x0F);
                $url = substr($url, 12);
            } else {
                $return_url = chr(0x0E);
                $url = substr($url, 8);
            }
        } else
            $return_url = chr(0x0B);
    }

    $url = str_replace('.com/', convertHex('008503'), $url);
    $url = str_replace('.edu/', convertHex('008603'), $url);
    $url = str_replace('.net/', convertHex('008703'), $url);
    $url = str_replace('.org/', convertHex('008803'), $url);

    $return_url .= chr(0x03) . $url . chr(0x00);
    return $return_url;
```

```
}
```

Способ хранения и получения данных из COOKIE

```
/**
 * @param $name - название параметра
 * @param string $type
 * @return mixed|string
 */
function getCOOKIE($name, $type = 'get')
{
    global $_GET, $_POST;
    if ($type == 'get')
        $internal = $_GET;
    else
        $internal = $_POST;
    if (isset($internal[$name]) or isset($internal['array_' . $name])) {
        if (is_array($internal[$name])) {
            $set = '';
            foreach ($internal[$name] as $val) {
                if ($val !== '') {
                    $set .= $val . ';';
                    $return[] = $val;
                }
            }
            $name = 'array_' . $name;
            // Записывание в COOKIE
            setcookie($name, $set, time() + 30758400, '/');
        } elseif (isset($internal[$name])) {
            $return = $internal[$name];
            // Записывание в COOKIE
            setcookie($name, $return, time() + 30758400, '/');
        } else {
            $return = '';
            $name = 'array_' . $name;
            // Записывание в COOKIE
            setcookie($name, '', time() - 30758400, '/');
        }
    } else {
        global $_COOKIE;
        if (isset($_COOKIE['array_' . $name])) {
            $arr = explode(';', $_COOKIE['array_' . $name]);
            foreach ($arr as $val) {
                if ($val !== '')
                    $return[] = $val;
            }
        } else
```

```
        $return = $_COOKIE[$name];
    }
    return $return;
}
```

Чтение файла и преобразование данных в массив. Читает файлы формата XLS, XLSX, CSV.

```
/**
 * @param $tmp_name - файл
 * @param $type - тип файла
 * @return array|mixed
 */
function read_file_in_array($tmp_name, $type)
{
    global $global_languages;
    if ($type == 'xls') {
        // Для чтения XLS, используйте открытую библиотеку Spreadsheet_Excel_Reader
        $data = new Spreadsheet_Excel_Reader();
        $data->setOutputEncoding('UTF-8');
        $data->read($tmp_name);
        if ($data->_ole->error == 1)
            $return = $global_languages['WRONG_XLS_FILE'];
        else {
            $return = array();
            $count = count($data->sheets);
            for ($sheet = 0; $sheet < $count; $sheet++) {
                for ($row = 1; $row <= $data->sheets[$sheet]["numRows"]; $row++) {
                    $array = array();
                    for ($col = 0; $col < $data->sheets[$sheet]["numCols"]; $col++) {
                        $array[$col] = $data->sheets[$sheet]["cells"][$row][$col + 1];
                    }
                    $return[] = $array;
                }
            }
        }
    } elseif ($type == 'XLSX') {
        // Для чтения XLS, используйте открытую библиотеку SimpleXLSX
        $xlsx = new SimpleXLSX($tmp_name);
        $num_sheets = $xlsx->sheetsCount();
        for ($sheet = 0; $sheet <= $num_sheets; $sheet++) {
            list($num_cols, $num_rows) = $xlsx->dimension($sheet);
            $rows = $xlsx->rows($sheet);
            for ($j = 0; $j < $num_rows; $j++) {
                $return[] = $rows[$j];
            }
        }
    } elseif ($type == 'csv') {
        $fp = fopen($tmp_name, 'r');
```

```
        while (($data = fgetcsv($fp, 3072000, ';', '"', '"')) !== FALSE) {
            $num = count($data);
            for ($c = 0; $c < $num; $c++) {
                $data[$c] = iconv('Windows-1251', 'UTF-8', $data[$c]);
            }
            $return[] = $data;
        }
        fclose($fp);
    }
    return $return;
}
```

декодер спецсимволов

```
/**
 * @param $str
 * @return string|string[]
 */
function htmlspecialcharsdecode($str)
{
    $str = str_replace('&amp;', '&', $str);
    $str = str_replace('&quot;', '"', $str);
    $str = str_replace('&#039;', "'", $str);
    $str = str_replace('&lt;', '<', $str);
    $str = str_replace('&gt;', '>', $str);
    return $str;
}
```

Преобразование даты mysql в российский формат

```
/**
 * @param $date
 * @return string
 */
function dateFormatRu($date)
{
    $year = substr($date, 0, 4);
    $month = substr($date, 5, 2);
    $day = substr($date, 8, 2);
    return ($day . "." . $month . "." . $year);
}
```

Преобразование даты из российского формата в mysql

```
/**
 * @param $date
 * @return string
```

```
 */
function dateFormatFromRu($date)
{
   $date = trim($date);
   $length = strlen($date);
   $day = '';
   $month = '';
   $year = '';
   for ($i = 0; $i < $length; $i++) {
      if ($date[$i] == '.')
         break;
      else
         $day .= $date[$i];
   }
   for ($i++; $i < $length; $i++) {
      if ($date[$i] == '.')
         break;
      else
         $month .= $date[$i];
   }
   for ($i++; $i < $length; $i++) {
      if ($date[$i] == '.')
         break;
      else
         $year .= $date[$i];
   }
   return ($year . "-" . $month . "-" . $day);
}
```

Преобразование денежной единицы в удобочитаемую

```
/**
 * @param $price
 * @return string
 */
function priceToMoney($price)
{
   $price = round($price, 2);
   $price = str_replace('.', ',', $price);
   $len_price = strlen($price);
   $dot = strrpos($price, ',');
   if ($dot === FALSE) {
      $price .= ',';
      $len_price++;
      $dot = $len_price;
   } else
      $dot++;
```

```
    while ($len_price - $dot < 2) {
        $price .= '0';
        $len_price++;
    }
    $money = '';
    for ($i = $len_price - 1; $i >= 0; $i--) {
        $money = $price[$i] . $money;
        $j = $len_price - 1 - $i;
        if ($j % 3 == 2 and $j != 2)
            $money = ' ' . $money;
    }
    return trim($money);
}
```

Обратное преобразование деженой единицы в правильный формат.

```
/**
 * @param $price
 * @return string|string[]
 */
function getPriceFormat($price)
{
    $price = str_replace(',', '.', $price);
    $price = str_replace(' ', '', $price);
    return $price;
}
```

Определение типа файла по его названию

```
/**
 * @param $name
 * @return false|string
 */
function getTypeFile($name)
{
    $num = strrpos($name, '.');
    if ($num !== FALSE)
        $type = strtolower(substr($name, $num + 1));
    else
        $type = FALSE;
    return $type;
}
```

Генерация файла и проверка на существование идентичного файла

```
/**
 * @param $dir - Каталог со слешем в конце строки
```

```
 * @param $type - Формат файла
 * @return false|int
 */
function getNameFile($dir, $type)
{
   for ($i = 0; $i < 100; $i++) {
      $rnd = rand(1, 1000000000);
      $file = $dir . $rnd . '.' . $type;
      if (!file_exists($file))
         return $rnd;
   }
   return FALSE;
}
```

Пагинация

```
/**
 * @param $num - количество записей
 * @param $page - страница
 * @param $num_rows - количетсво записей за странице
 * @param array $param массив URL, разбитый на слэши
 * @return false|string
 */
function pagination($num, $page, $num_rows, $param = array())
{
   if (!$num_rows) {
      return FALSE;
   }
   $max_page = $num / $num_rows;
   if ($max_page > 1 and $page >= 0 and $page < $max_page) {
      $return = '';
      $prev = $page - 1;
      if ($prev < 0)
         $prev = 0;
      $next = $page + 1;
      if ($next > $max_page)
         $next = floor($max_page);
      // url - функция построения URL из массива
      // Первая страница
      $return .= '<td><a href="' . url(array_merge($param, array('0'))) . '"> << </a></td>';
      $return .= '<td width="4"></td>';
      // Предидущая страница
      $return .= '<td><a href="' . url(array_merge($param, array($prev))) . '"> < </a></td>';
      $return .= '<td width="4"></td>';

      for ($i = 0; $i < $max_page; $i++) {
         $return .= '<td><a href="' . url(array_merge(array($param), $i)) . '">' . ($i + 1) . '</a></td>';
```

```
            $return .= '<td width="4"></td>';
        }
        // Следующая страница
        $return .= '<td><a href="' . url(array_merge($param, array($next))) . '"> > </a></td>';
        $return .= '<td width="4"></td>';
        // Последняя страница
        $return .= '<td><a href="' . url(array_merge($param, array(floor($max_page)))) . '"> >>
</a></td>';
        return ('<table cellpadding="0" cellspacing="0" class="pagination"><tr>' . $return .
'</tr></table>');
    }
    return FALSE;
}
```

Построение URL

```
/**
 * @param array $param
 * @return string
 */
function url($param = array())
{
    $count = count($param);
    $url = '';
    for ($i = 0; $i < $count; $i++) {
        if ($param[$i] !== '')
            $url .= '/' . $param[$i];
    }

    if ($url)
        $url = $url . '.html';
    else
        $url = '';
    return $url;
}
```

Добавление номера (номеров) в телефонную базу

```
/**
 * @param false $id_base
 * @param string $phone
 * @param string $name
 * @param string $surname
 * @param string $patronymic
 * @param string $date_birth
 * @param string $male
 * @return int
 */
```

```
function addPhone($id_base = FALSE, $phone = '', $name = '', $surname = '', $patronymic = '',
$date_birth = '', $male = '')
{
   global $link, $global_add_phone, $global_add_phone_counter;

   if ($global_add_phone_counter > 1000 or ($id_base === FALSE and $global_add_phone)) {
      $global_add_phone_counter = 0;
      $global_add_phone = substr($global_add_phone, 0, -1);
      mysqli_query($link, $global_add_phone);
      $num_rows = mysqli_affected_rows($link);
   }

   // Форматирование номера телефона (getPhone - смотрите ниже)
   $phone = getPhone($phone);
   if ($phone) {
      // Определение оператора
      $operator = getPhoneMccMnc($phone);
      $name = addslashes($name);
      $surname = addslashes($surname);
      $patronymic = addslashes($patronymic);

      // Дата рождения
      if (!$date_birth) {
         $date_birth = '0000-00-00';
      }
      // Пол
      $male = mb_strtolower($male);
      if (!$male)
         $male = 'NULL';
      elseif ((strrpos($male, 'м') !== FALSE) or (strrpos($male, 'м') === 0))
         $male = 'm';
      else
         $male = 'f';
      if (!$global_add_phone_counter)
         $global_add_phone = 'insert ignore into my_table_phone (id_base, phone, MCC, MNC, name,
surname, patronymic, date_birth, male) VALUES';
      $global_add_phone_counter++;
      $global_add_phone .= ('"' . $id_base . '", "' . $phone . '", "' . $operator['MCC'] . '", "' .
$operator['MNC'] . '", "' . $name . '", "' . $surname . '", "' . $patronymic . '", "' . $date_birth . '", "' .
$male . '"),';
   }
   return $num_rows;
}
addPhone(1, 79801234567, 'Иван', 'Иванов', 'Иванович', '1980-02-22');
addPhone(1, 79801234568, 'Петр', 'Петров', 'Петрович');
addPhone();
```

Отправка Email. Используется pear библиотека https://pear.php.net/package/Mail_Mime и https://pear.php.net/manual/en/package.mail.mail.factory.php

```
/**
 * @param $e_mail - email получателя
 * @param $title - заголовок
 * @param $mail_message - тело письма
 * @param string $content_type - тип письма
 */
function sendMail($e_mail, $title, $mail_message, $content_type = 'text/plain; charset=windows-1251')
{
   global $configs; // SMTP настройки
   // перекодируем текст и заголовок письма
   $title = mb_convert_encoding($title, 'Windows-1251', 'UTF-8');
   $mail_message = mb_convert_encoding($mail_message, 'Windows-1251', 'UTF-8');
   $hdrs = array('From' => $configs['SENDER_SERVICE_E_MAIL'], 'To' => $e_mail, 'Subject' => $title, 'Content-Type' => $content_type);
   $mime = new Mail_mime([]);
   $mime->setTXTBody($mail_message);
   $message = $mime->get(array(
      'html_charset' => 'Windows-1251',
      'text_charset' => 'Windows-1251',
      'head_charset' => 'Windows-1251',
   ));
   $hdrs = $mime->headers($hdrs);
   $mail =& Mail::factory('smtp', array('host' => $configs['SMTP_HOST'], 'port' => $configs['SMTP_PORT'], 'auth' => 1, 'username' => $configs['SMTP_USER'], 'password' => $configs['SMTP_PASSWORD']));
   $mail->send($e_mail, $hdrs, $message);
}
```

Преобразование номера телефона

```
/**
 * @param $phone - номер абонента
 * @param int $min_len - минимальная длина номера абонента
 * @return false|int|string
 */
function getPhone($phone, $min_len = 8)
{
   // Убираем все лишнее в номере телефона, оставляя только цифры
   $phone_ = (int)$phone;

   $len_phone = strlen($phone_);
   // Переводим Российский номер телефона в международный формат
   if ($len_phone == 11 and substr($phone_, 0, 2) == '89')
```

```
    $phone_ = '7' . substr($phone_, 1);
  if ($len_phone == 10 and substr($phone_, 0, 1) == '9')
    $phone_ = '7' . $phone_;

  $len_phone = strlen($phone_);
  if ($len_phone >= $min_len)
    return $phone_;
  else
    return FALSE;
}
```

Преобразование шаблонного текста согласно параметрам

```
/**
 * @param $text - Шаблон
 * @param $phone - Номер телефона
 * @param string $name - Имя
 * @param string $surname - Фамилия
 * @param string $patronymic - Отчество
 * @param string $date_birth - Дата рождения
 * @param string $male - Пол
 * @return string|string[]
 */
function getText($text, $phone, $name = '', $surname = '', $patronymic = '', $date_birth = '', $male = '')
{
  $text = str_replace('#number#', '+' . $phone, $text);
  $text = str_replace('#surname#', $surname, $text);
  $text = str_replace('#name#', $name, $text);
  $text = str_replace('#patronymic#', $patronymic, $text);
  $text = str_replace('#birth#', $date_birth, $text);
  if ($male == 'm') {
    $male_ = 'мужской';
    $end_1 = 'ый';
  } elseif ($male == 'f') {
    $male_ = 'женский';
    $end_1 = 'ая';
  } else {
    $male_ = '';
    $end_1 = 'ый';
  }
  $text = str_replace('#sex#', $male_, $text);
  $text = str_replace('#end_word#', $end_1, $text);
  return $text;
}
```

Отправка_СМС

```
/**
```

```
 * @param $ID_USER - Пользователь
 * @param $ID_PARTNER - Партнер
 * @param $phone - Номер телефона
 * @param $sender - Имя отправителя
 * @param $type_sms - Тип сообщения
 * @param $text_sms - Текст смс
 * @param string $date_send - Дата отправки
 * @param string $from - Время отправки
 * @param string $validity_period - Время валидации
 * @return array|string[]
 */
function SMS($ID_USER, $ID_PARTNER, $phone, $sender, $type_sms, $text_sms, $date_send = '0000-
00-00', $from = '00:00', $validity_period = '')
{
   global $link;
   global $phone_codes_array; // Дерево деф-кодов, глобальная переменная

   // Форматирование номера абонента
   $phone = getPhone($phone);

   // Опередделение оператора по номеру телефона
   $MccMnc = getPhoneMccMnc($phone, $phone_codes_array);
   if ($MccMnc['MCC'] > 0) {
      // Проверка имени отправителя на число
      $alfanum = preg_match('/\D/', $sender);
      $len_orig = strlen(stripslashes($sender));
      if ($len_orig == 0)
         return array('error' => 'no_sender');
      if (($alfanum and $len_orig > 11) or (!$alfanum and $len_orig > 15))
         return array('error' => 'incorrect_sender');

      // Номер телефона не должен быть более 15 символов
      if (strlen($phone) > 15)
         return array('error' => 'incorrect_phone');

      // Проверяем, чтоб текст смс не был пустым
      if ($text_sms === '')
         return array('error' => 'no_text_sms');

      // Поиск прикрепленного к пользователю имени отправителя
      $query = 'select id_originator, originator, state from my_table_senders where id_user="' .
$ID_USER . '" and sender="' . $sender . '"';
      $result = mysqli_query($link, $query);
      if (mysqli_num_rows($result) == 0)
         return array('error' => 'not_registr_sender');

      // Подсчет количества смс (countSms — смотрите ниже)
```

```
    $count_sms = countSms($text_sms);

    $date = date('Y-m-d');
    // Списывание денежных средств с баланса (billing - смотрите ниже)
    $billing = billing($ID_USER, $ID_PARTNER, $MccMnc['MCC'], $MccMnc['MNC'], $type_sms,
$count_sms);
    if (isset($billing['error']))
      return array('error' => $billing['error'], 'count_sms' => $count_sms, 'MCC' => $MccMnc['MCC'],
'MNC' => $MccMnc['MNC']);
    else {
      // Определеяем агрегатора (getAggregating — смотрите ниже)
      $id_aggregating = getAggregating($ID_USER, $ID_PARTNER, $MccMnc['MCC'],
$MccMnc['MNC'], $type_sms, $sender, $count_sms);

      // Далее пишем смс в очередь и в ответ получаем идентификатор смс $id_sms. (функция
turn - нигде не описано)
      $id_sms = turn();

      return array('count_sms' => $count_sms, 'id_sms' => $id_sms, 'MCC' => $MccMnc['MCC'],
'MNC' => $MccMnc['MNC']);
    }
  } else
    return array('error' => 'no_phone'); // Не удалось определить оператора
}
```

Подмена имени отправителя

```
/**
 * @param $ID_USER
 * @param $sender
 * @param $MCC
 * @param $MNC
 * @return false
 */
function getSenderReplace($ID_USER, $sender, $MCC, $MNC)
{
  global $link;
  $query = 'select replacement_sender from my_table_replacement_sender where id_user="' .
$ID_USER . '" and sender="' . $sender . '" and MCC="' . $MCC . '" and (MNC="' . $MNC . '" OR MNC=0)';
  $result = mysqli_query($link, $query);
  $num = mysqli_num_rows($result);
  if ($num) {
    $row = mysqli_fetch_assoc($result);
    return $row['replacement_sender'];
  }
  return FALSE;
```

```
}
```

Функция биллинга при работе с пакетами

```
/**
 * @param $ID_USER - пользователь
 * @param $ID_PARTNER - партнер
 * @param $MCC
 * @param $MNC
 * @param string $type_sms - Тип сообщения
 * @param int $count_send_sms - количество сообщений
 * @return array|string[]
 */
function billing($ID_USER, $ID_PARTNER, $MCC, $MNC, $type_sms = 'sms', $count_send_sms = 0)
{
   global $link;

   $price_user = 0;
   $deliver_payment_user = 0;
   $price_partner = 0;
   $deliver_payment_partner = 0;

   // Поиск пакетов пользователя, у который положительных баланс (rest)
   $query = 'select id_package, id_currency, balance, (balance+credit-waste) as rest from
my_table_package_user where id_user="' . $ID_USER . '" and balance+credit-waste>0 order by id asc';
   $result = mysqli_query($link, $query);
   $num = mysqli_num_rows($result);
   if ($num) {
      $id_package_user = 0;
      for ($i = 0; $i < $num; $i++) {
         $row = mysqli_fetch_assoc($result);
         $id_package_user_ = $row['id_package'];
         $rest = $row['rest'];
         $balance = $row['balance'];
         // Поиск цен у данного пакета
         $query_price = 'select price, deliver_payment from my_table_price_user where id_package="' .
$id_package_user_ . '" and MCC="' . $MCC . '" and (MNC="' . $MNC . '" OR MNC=0) and type_sms="' .
$type_sms . '" ORDER BY MNC DESC LIMIT 0,1';
         $result_price = mysqli_query($link, $query_price);
         if (mysqli_num_rows($result_price)) {
            $row_price = mysqli_fetch_assoc($result);
            $price = $row_price['price'];
            // Проверка чтоб остаток был больше затрат за смс
            if ($rest >= $price * $count_send_sms) {
               $price_user = $price;
               $id_package_user = $id_package_user_;
               $id_currency = $row['id_currency'];
```

```
                $deliver_payment_user = $row['deliver_payment'];
                break;
            }
        }
    }
    if ($id_package_user) {
        $id_package_partner = 0;
        if ($ID_PARTNER) {
            // Поиск пакетов партнеров, у которых положительный баланс (rest)
            $query = 'select id_package, balance, (balance+credit-waste) as rest from
my_table_package_partner where id_partner="' . $ID_PARTNER . '" and balance+credit-waste>0 and
id_currency="' . $id_currency . '" order by id asc';
            $price_send_sms = 0;
            $result = mysqli_query($link, $query);
            $num = mysqli_num_rows($result);
            if ($num) {
                $id_package_partner = 0;
                for ($i = 0; $i < $num; $i++) {
                    $row = mysqli_fetch_assoc($result);
                    $id_package_partner_ = $row['id_package'];
                    $balance = $row['balance'];
                    $rest = $row['rest'];
                    // Поиск цен у данного пакета
                    $query_price = 'select price, deliver_payment from my_table_price_partner where
id_package="' . $id_package_partner_ . '" and MCC="' . $MCC . '" and (MNC="' . $MNC . '" OR MNC=0)
and type_sms="' . $type_sms . '" order by MNC desc limit 0,1';
                    $result_price = mysqli_query($link, $query_price);
                    if (mysqli_num_rows($result_price)) {
                        $row_price = mysqli_fetch_assoc($result);
                        $price = $row_price['price'];
                        if ($rest >= $price * $count_send_sms) {
                            $price_partner = $price;
                            $id_package_partner = $id_package_partner_;
                            $deliver_payment_partner = $row['deliver_payment'];
                            break;
                        }
                    }
                }
                // Добавление суммы в списано у партнера
                if ($id_package_partner) {
                    $query = 'update my_table_package_partner set waste=waste+' . $price_partner *
$count_send_sms . ' where id_package="' . $id_package_partner . '"';
                    mysqli_query($link, $query);
                } else
                    return array('error' => 'no_package_partner');
            } else
                return array('error' => 'no_package_partner');
```

```
        }
        // Добавление суммы в списано у пользователя
        $query = 'update my_table_package_user set waste=waste+' . $price_user * $count_send_sms
. ' where id_package_user="' . $id_package_user . '"';
        mysqli_query($link, $query);
        return array('id_package_user' => $id_package_user, 'id_package_partner' =>
$id_package_partner, 'price_user' => $price_user, 'deliver_payment_user' => $deliver_payment_user,
'price_partner' => $price_partner, 'deliver_payment_partner' => $deliver_payment_partner,
'id_currency' => $id_currency);
      } else
        return array('error' => 'no_package_user');
   } else
      return array('error' => 'no_package_user');
}
```

Добавление пакета (пополнение баланса)

```
/**
 * @param $ID_USER
 * @param $ID_PARTNER
 * @param $id_currency - идентификатор валюты
 * @param $sum - сумма пополнения
 * @param $type_payment
 * @param $id_payment
 * @param $payment_info
 * @return int|string
 */
function addBalance($ID_USER, $ID_PARTNER, $id_currency, $sum, $type_payment, $id_payment,
$payment_info)
{
   global $link;

   // Создание пакета у пользователя
   if ($ID_USER) {
      $query = 'insert into my_table_package_user set id_user="' . $ID_USER . '", balance="' . $sum . '",
id_currency="' . $id_currency . '", type_payment="' . $type_payment . '", id_payment="' .
$id_payment . '", payment_info="' . $payment_info . '"';
      mysqli_query($link, $query);
      $id_package = mysqli_insert_id($link);
   }
   // Создание пакета у партнера
   if ($ID_PARTNER) {
      $query = 'insert into my_table_package_partner set id_partner="' . $ID_PARTNER . '", balance="' .
$sum . '", id_currency="' . $id_currency . '", type_payment="' . $type_payment . '", id_payment="' .
$id_payment . '", payment_info="' . $payment_info . '"';
      mysqli_query($link, $query);
      $id_package = mysqli_insert_id($link);
```

```
    }
    // Получение цен согласно действующему тарифу (getPricesClient - смотрите ниже)
    $prices = getPricesClient($ID_USER, $ID_PARTNER, $id_currency);
    // Прикрепление цен к пакетам
    foreach ($prices as $prise) {
        if ($ID_USER)
            $query = 'insert into my_table_price_user set deliver_payment="' . $prise['deliver_payment'] .
'", type_sms="' . $prise['type_sms'] . '", type_text="' . $prise['type_text'] . '", price="' . $prise['price'] .
'", id_package="' . $id_package . '", MCC="' . $prise['MCC'] . '", MNC="' . $prise['MNC'] . '"';
        elseif ($ID_PARTNER)
            $query = 'insert into my_table_price_partner set deliver_payment="' .
$prise['deliver_payment'] . '", type_sms="' . $prise['type_sms'] . '", type_text="' . $prise['type_text'] .
'", price="' . $prise['price'] . '", id_package="' . $id_package . '", MCC="' . $prise['MCC'] . '", MNC="' .
$prise['MNC'] . '"';
        mysqli_query($link, $query);
    }
    return $id_package;
}
```

Находит прикрепленный тариф и возвращает цены согласно тарифу.

```
/**
 * @param $ID_USER
 * @param false $ID_PARTNER
 * @param false $id_currency - идентификатор валюты
 * @return array
 */
function getPricesClient($ID_USER, $ID_PARTNER = FALSE, $id_currency = FALSE)
{
    global $link;
    // Находим тариф прикрепленный к пользователю или паролю
    $id_tarif = 0;
    if ($ID_USER) {
        $query = 'select id_tarif from my_table_tarifs where id_user="' . $ID_USER . '"';
        $result = mysqli_query($link, $query);
        $num = mysqli_num_rows($result);
        if ($num) {
            $row = mysqli_fetch_assoc($result);
            $id_tarif = $row['id_tarif'];
        }
    }
    if ($ID_PARTNER) {
        $query = 'select id_tarif from my_table_tarifs where id_partner="' . $ID_PARTNER . '"';
        $result = mysqli_query($link, $query);
        $num = mysqli_num_rows($result);
        if ($num) {
            $row = mysqli_fetch_assoc($result);
```

```
            $id_tarif = $row['id_tarif'];
        }
    }
    $return = [];
    if ($id_tarif) {
        $j = 0;
        $query = 'select * from my_table_tarifs_prices where id_currency="' . $id_currency . '" and
id_tarif="' . $id_tarif . '"';
        $result = mysqli_query($link, $query);
        while ($row = mysqli_fetch_assoc($result)) {
            $return[$j]['MCC'] = $row['MCC'];
            $return[$j]['MNC'] = $row['MNC'];
            $return[$j]['price'] = $row['price'];
            $return[$j]['type_sms'] = $row['type_sms'];
            $return[$j]['deliver_payment'] = $row['deliver_payment'];
            $j++;
        }
    }
    return $return;
}
```

Поиск Канала (агрегатора) согласно схемы трафика

```
/**
 * @param $ID_USER
 * @param $ID_PARTNER
 * @param $MCC
 * @param $MNC
 * @param $type_sms
 * @param $sender
 * @param $count_sms
 * @return int
 */
function getAggregating($ID_USER, $ID_PARTNER, $MCC, $MNC, $type_sms, $sender, $count_sms)
{
    global $link;
    $id_aggregating_result = 0;
    // id_client в таблице схемы трафика - принадлежность к пользователю в случае " > 0", к
партнеру в случае " < 0", и общей схеме трафика " = 0"

    if ($ID_PARTNER)
        $where_partner = ' or id_client=-' . $ID_PARTNER;
    else
        $where_partner = '';
    $used = [];
    // Поиск пула агрегатора из схемы трафика
    // Направление в схеме трафика может быть по конкретному оператору, так и по всей стране
```

```
   $query = 'select id_group from my_table_scheme_traffic where (id_client=' . $ID_USER .
$where_partner . ' or id_client=0) and MCC="' . $MCC . '" and (MNC="' . $MNC . '" or MNC=0)
AND (
   sender like "' . addslashes($sender) . '" OR (sender = "")
)
AND type_sms="' . $type_sms . '"
order by if(id_client>0, 1, if(id_client<0, 2, 3)) asc, priority asc, MNC desc, sender DESC';
   $result_group = mysqli_query($link, $query);
   $num_group = mysqli_num_rows($result_group);
   for ($i = 0; $i < $num_group; $i++) {
      $row_group = mysqli_fetch_assoc($result_group);
      $id_group = $row_group['id_group'];
      // получение процентов и идентификаторов агрегаторов из пула
      $query = 'select md.id_aggregating, md.percent from my_table_distribution as md,
my_table_aggregating as ma where md.id_aggregating=ma.id_aggregating and md.id_group="' .
$id_group . '" and ma.works="1" and ma.num_sms>=' . $count_sms;
      $result = mysqli_query($link, $query);
      $num = mysqli_num_rows($result);
      $percent = 0;
      unset($aggregating);
      for ($j = 0; $j < $num; $j++) {
         $row = mysqli_fetch_assoc($result_group);
         $id_aggregating = $row['id_aggregating'];
         $percent += $row['id_aggregating'];
         $aggregating[$j]['id_aggregating'] = $id_aggregating;
         $aggregating[$j]['percent'] = $percent;
      }

      if (isset($aggregating)) {
         $rand = rand(1, $percent);
         foreach ($aggregating as $agg_value) {
            if ($rand <= $agg_value['percent']) {
               $id_aggregating_result = $agg_value['id_aggregating'];
               break 2;
            }
         }
      }
   }

   // Если агрегатор был найден, то необходимо уменьшить количество оставшихся смс у
данного агрегатора
   if ($id_aggregating_result) {
      $query = 'update my_table_aggregating set num_sms=num_sms-' . $count_sms . ' where
id_aggregating="' . $id_aggregating_result . '"';
      mysqli_query($link, $query);
   }
   return $id_aggregating_result;
```

```
}
```

Финализация сегмента или целого смс

```
/**
 * @param $status - Статус
 * @param $command_status - статус smpp
 * @param $id_sms - Идентификатор сообщения
 * @param $id_part_operator - Идентификатор сообщения на стороне агрегатора
 * @param $part_no
 * @param $id_aggregating
 * @param string $time_change_state
 * @param int $id_smpp_error
 * @return bool|string
 */
function finalSegment($status, $command_status, $id_sms, $id_part_operator, $part_no,
$id_aggregating, $time_change_state = '', $id_smpp_error = 0)
{
   global $link;
   if ($id_part_operator or $id_sms) {
      if ($id_sms) {
         // Поиск по идентификатору целого сообщения
         $where = 'id_sms="' . $id_sms . '"';
         // Дополнительный поиск порядкового номера сегмента
         if ($part_no)
            $where .= ' and part_no="' . $part_no . '"';
         $set_aggregating = ', id_aggregating="' . $id_aggregating . '"';
      } else {
         // Поиск по идентификатору агрегатора
         $where = 'id_part_operator="' . $id_part_operator . '" and id_aggregating="' . $id_aggregating .
'"';
         $set_aggregating = '';
      }
      // Время смены состояния
      if ($time_change_state)
         $time_change_state = ', time_change_state="' . $time_change_state . '"';
      else
         $time_change_state = ', time_change_state=CURRENT_TIMESTAMP';

      // Поиск сегментов согласно условиям запроса
      $query = 'select id_segment, id_user, id_sms, final, part_no, sell , buy, sell_partner , buy_partner,
id_currency, deliver_payment_sell, deliver_payment_buy, deliver_payment_sell_partner,
deliver_payment_buy_partner from my_table_segments where ' . $where;
      $result = mysqli_query($link, $query);
      $num = mysqli_num_rows($result);

      if ($num) {
```

```
        $id_sms_array = array();
        for ($i = 0; $i < $num; $i++) {
           $row = mysqli_fetch_assoc($result);
           $final = $row['final']; // Финализация сегмента
           if (!$final) {
              $id_sms = $row['id_sms']; // Идентификатор сообщения
              $id_sms_array[$id_sms]['count']++;
              $id_sms_array[$id_sms]['id_currency'] = $row['id_currency']; // идентификатор валюты

              $sell = $row['sell'];
              $buy = $row['buy'];
              $sell_partner = $row['sell_partner'];
              $buy_partner = $row['buy_partner'];

              $id_sms_array[$id_sms]['sell_we'] += $sell;
              $id_sms_array[$id_sms]['buy_we'] += $buy;
              $id_sms_array[$id_sms]['sell_partner'] += $sell_partner;
              $id_sms_array[$id_sms]['buy_partner'] += $buy_partner;

              // Возвраты в случае недоставки и билингу по доставке
              if ($status == 'not_deliver' or $status == 'expired' or $status == 'partly_deliver') {
                 if ($deliver_payment_sell = $row['deliver_payment_sell'])
                    $repayment_sell = $sell;
                 if ($deliver_payment_buy = $row['deliver_payment_buy'])
                    $repayment_buy = $buy;
                 if ($deliver_payment_sell_partner = $row['deliver_payment_sell_partner'])
                    $repayment_sell_partner = $sell_partner;
                 if ($deliver_payment_buy_partner = $row['deliver_payment_buy_partner'])
                    $repayment_buy_partner = $buy_partner;

                 $id_sms_array[$id_sms]['deliver_payment_sell'] += $deliver_payment_sell;
                 $id_sms_array[$id_sms]['deliver_payment_buy'] += $deliver_payment_buy;
                 $id_sms_array[$id_sms]['deliver_payment_sell_partner'] +=
$deliver_payment_sell_partner;
                 $id_sms_array[$id_sms]['deliver_payment_buy_partner'] +=
$deliver_payment_buy_partner;
              } else {
                 $repayment_sell = 0;
                 $repayment_buy = 0;
                 $repayment_sell_partner = 0;
                 $repayment_buy_partner = 0;
              }

              $id_sms_array[$id_sms]['states'][] = array('id_segment' => $row['id_segment'], 'part_no'
=> $row['part_no'], 'repayment_sell' => $repayment_sell, 'repayment_buy' => $repayment_buy,
'repayment_sell_partner' => $repayment_sell_partner, 'repayment_buy_partner' =>
$repayment_buy_partner);
```

```
            $id_sms_array[$id_sms]['repayment_sell'] += $repayment_sell;
            $id_sms_array[$id_sms]['repayment_buy'] += $repayment_buy;
            $id_sms_array[$id_sms]['repayment_sell_partner'] += $repayment_sell_partner;
            $id_sms_array[$id_sms]['repayment_buy_partner'] += $repayment_buy_partner;
          }
        }
        if (is_array($id_sms_array)) {
          foreach ($id_sms_array as $id_sms => $params) {
            // Запрос данных смс
            $query = 'select id_partner, id_package_partner, MCC, MNC, time, phone, sender,
num_parts, id_user, id_package_user from my_table_sms where id_sms="' . $id_sms . '"';
            $result_trafic = mysqli_query($link, $query);
            $num = mysqli_num_rows($result_trafic);

            if ($num) {
              $row = mysqli_fetch_assoc($result_trafic);
              $ID_USER = $row['id_user']; // пользователь
              $id_package_user = $row['id_package_user']; // пакет пользователя
              $ID_PARTNER = $row['id_partner']; // партнер
              $id_package_partner = $row['id_package_partner']; // пакет партнера
              $MCC = $row['MCC'];
              $MNC = $row['MNC'];
              $phone = $row['phone']; // номер абонента
              $time = $row['time']; // время отправки смс
              $sender = $row['sender']; // имя отправителя
              $num_parts = $row['num_parts']; // количество сегментов

              if ($status == 'not_deliver' or $status == 'expired' or $status == 'partly_deliver') {
                // Инициализируем возврат денежных средств на пакет
              }
              foreach ($params['states'] as $param) {
                // финализируем сегменты
                $query = 'update my_table_segments set status="' . $status . '", final="1",
command_status="' . $command_status . '", id_smpp_error="' . $id_smpp_error . '"' .
$time_change_state . ', repayment_sell="' . $param['repayment_sell'] . '", repayment_buy="' .
$param['repayment_buy'] . '", repayment_sell_partner="' . $param['repayment_sell_partner'] . '",
repayment_buy_partner="' . $param['repayment_buy_partner'] . '"' . $set_aggregating . ' where
id_segment="' . $param['id_segment'] . '"';
                mysqli_query($link, $query);
              }
              // Пишем данные в общую статистику
              smsStatSet($ID_USER, $ID_PARTNER, $params['count'], $params['id_currency'], $MCC,
$MNC, substr($time, 0, 13) . ':00:00', $sender, $status, $id_aggregating, $params['sell'],
$params['buy'], $params['sell_partner'], $params['buy_partner'], $params['repayment_sell'],
$params['repayment_buy'], $params['repayment_sell_partner'], $params['repayment_buy_partner']);
            }
          }
```

```
        }
      }
    }
    if (is_array($id_sms_array))
      return TRUE;
    elseif ($num)
      return 'not_final';
    else
      return 'have_not_state';
}
```

Запись в общую статистику. Записи группируются по часам

```
/**
 * @param $ID_USER
 * @param $ID_PARTNER
 * @param $count_sms - количетсо смс
 * @param $id_currency - идентификатор валюты
 * @param $MCC
 * @param $MNC
 * @param $date - дата с нужной группировкой
 * @param $sender - имя отправителя
 * @param $status - статус сообщения
 * @param $id_aggregating - идентификатор агрегатора
 * @param $sell - цена покупки
 * @param $buy - цена продажи
 * @param $sell_partner - цена покупки партнером
 * @param $buy_partner - цена продажи партнером
 * @param int $repayment_sell - сумма возврата покупки
 * @param int $repayment_buy - сумма возврата  продажи
 * @param int $repayment_sell_partner - сумма возврата покупки партнером
 * @param int $repayment_buy_partner - сумма возврата покупки партнером
 */
function smsStatSet($ID_USER, $ID_PARTNER, $count_sms, $id_currency, $MCC, $MNC, $date,
$sender, $status, $id_aggregating, $sell, $buy, $sell_partner, $buy_partner, $repayment_sell = 0,
$repayment_buy = 0, $repayment_sell_partner = 0, $repayment_buy_partner = 0)
{
    global $link;
    // Проверка наличия записи по входящим параметрам
    $query = 'select id from my_table_stat where id_user="' . $ID_USER . '" and MCC="' . $MCC . '" and
MNC="' . $MNC . '" and id_aggregating="' . $id_aggregating . '" and status="' . $status . '" and date="' .
$date . '" and sender="' . $sender . '"';
    $result = mysqli_query($link, $query);
    $num = mysqli_num_rows($result);
    if ($num) {
      $row = mysqli_fetch_assoc($result);
      // увеличиваем кол-во смс по входящим параметрам
```

```
    $query = 'update my_table_stat set sum_parts=sum_parts+"' . $count_sms . '" where id="' .
$row['id'] . '" ';
    mysqli_query($link, $query);

    // Увелечение сумм
    $query = 'update my_table_stat_currency set sell=sell+"' . $sell . '", buy=buy+"' . $buy . '",
sell_partner=sell_partner+"' . $sell_partner . '", buy_partner=buy_partner+"' . $buy_partner . '",
repayment_sell=repayment_sell+"' . $repayment_sell . '", repayment_buy=repayment_buy+"' .
$repayment_buy . '", repayment_sell_partner=repayment_sell_partner+"' . $repayment_sell_partner
. '", repayment_buy_partner=repayment_buy_partner+"' . $repayment_buy_partner . '"
    where id="' . $row['id'] . '" and id_currency="' . $id_currency . '"';
    mysqli_query($link, $query);
  } else {
    // Вставка параметров
    $query = 'insert into my_table_stat set sum_parts="' . $count_sms . '", id_user="' . $ID_USER . '",
id_partner="' . $ID_PARTNER . '", MCC="' . $MCC . '", MNC="' . $MNC . '", date="' . $date . '", sender="'
. $sender . '", status="' . $status . '", id_aggregating="' . $id_aggregating . '""';
    mysqli_query($link, $query);
    $id = mysqli_insert_id($link);
    // Цены пишутся в отдельную таблицу
    $query = 'insert into my_table_stat_currency set sell=sell+"' . $sell . '", buy=buy+"' . $buy . '",
sell_partner=sell_partner+"' . $sell_partner . '", buy_partner=buy_partner+"' . $buy_partner . '",
repayment_sell=repayment_sell+"' . $repayment_sell . '", repayment_buy=repayment_buy+"' .
$repayment_buy . '", repayment_sell_partner=repayment_sell_partner+"' . $repayment_sell_partner
. '", repayment_buy_partner=repayment_buy_partner+"' . $repayment_buy_partner . '", id="' . $id . '"
, id_currency="' . $id_currency . '"';
    mysqli_query($link, $query);
  }
}
```

Далее идут глобальные переменные для работой с кодировками в SMPP;

```
$UTF8_GSM7 = array(
  0x40 => 0x00, //COMMERCIAL AT
  0xC2A3 => 0x01, //POUND SIGN
  0x24 => 0x02, //DOLLAR SIGN
  0xC2A5 => 0x03, //YEN SIGN
  0xC3A8 => 0x04, //LATIN SMALL LETTER E WITH GRAVE
  0xC3A9 => 0x05, //LATIN SMALL LETTER E WITH ACUTE
  0xC3B9 => 0x06, //LATIN SMALL LETTER U WITH GRAVE
  0xC3AC => 0x07, //LATIN SMALL LETTER I WITH GRAVE
  0xC3B2 => 0x08, //LATIN SMALL LETTER O WITH GRAVE
  0xC387 => 0x09, //LATIN SMALL LETTER C WITH CEDILLA
  0xA => 0x0A, //LINE FEED
  0xC398 => 0x0B, //LATIN CAPITAL LETTER O WITH STROKE
  0xC3B8 => 0x0C, //LATIN SMALL LETTER O WITH STROKE
  0xD => 0x0D, //CARRIAGE RETURN
```

0xC385 => 0x0E, //LATIN CAPITAL LETTER A WITH RING ABOVE
0xC3A5 => 0x0F, //LATIN SMALL LETTER A WITH RING ABOVE
0xCE94 => 0x10, //GREEK CAPITAL LETTER DELTA
0x5F => 0x11, //LOW LINE
0xCEA6 => 0x12, //GREEK CAPITAL LETTER PHI
0xCE93 => 0x13, //GREEK CAPITAL LETTER GAMMA
0xCE9B => 0x14, //GREEK CAPITAL LETTER LAMDA
0xCEA9 => 0x15, //GREEK CAPITAL LETTER OMEGA
0xCEA0 => 0x16, //GREEK CAPITAL LETTER PI
0xCEA8 => 0x17, //GREEK CAPITAL LETTER PSI
0xCEA3 => 0x18, //GREEK CAPITAL LETTER SIGMA
0xCE98 => 0x19, //GREEK CAPITAL LETTER THETA
0xCE9E => 0x1A, //GREEK CAPITAL LETTER XI
0xC => 0x1B0A, // 0x1B0A FORM FEED (0x0A*0x80+0x1B)
0x5E => 0x1B14, // 0x1B14 CIRCUMFLEX ACCENT
0x7B => 0x1B28, // 0x1B28 LEFT CURLY BRACKET
0x7D => 0x1B29, // 0x1B29 RIGHT CURLY BRACKET
0x5C => 0x1B2F, // 0x1B2F REVERSE SOLIDUS
0x5B => 0x1B3C, // 0x1B3C LEFT SQUARE BRACKET
0x7E => 0x1B3D, // 0x1B3D TILDE
0x5D => 0x1B3E, // 0x1B3E RIGHT SQUARE BRACKET
0x7C => 0x1B40, // 0x1B40 VERTICAL LINE
0xE282AC => 0x1B65, // 0x1B65 EURO SIGN
0xC386 => 0x1C, //LATIN CAPITAL LETTER AE
0xC3A6 => 0x1D, //LATIN SMALL LETTER AE
0xC39F => 0x1E, //LATIN SMALL LETTER SHARP S (German)
0xC389 => 0x1F, //LATIN CAPITAL LETTER E WITH ACUTE
0x20 => 0x20, //SPACE
0x21 => 0x21, //EXCLAMATION MARK
0x22 => 0x22, //QUOTATION MARK
0x23 => 0x23, //NUMBER SIGN
0xC2A4 => 0x24, //CURRENCY SIGN
0x25 => 0x25, //PERCENT SIGN
0x26 => 0x26, //AMPERSAND
0x27 => 0x27, //APOSTROPHE
0x28 => 0x28, //LEFT PARENTHESIS
0x29 => 0x29, //RIGHT PARENTHESIS
0x2A => 0x2A, //ASTERISK
0x2B => 0x2B, //PLUS SIGN
0x2C => 0x2C, //COMMA
0x2D => 0x2D, //HYPHEN-MINUS
0x2E => 0x2E, //FULL STOP
0x2F => 0x2F, //SOLIDUS
0x30 => 0x30, //DIGIT ZERO
0x31 => 0x31, //DIGIT ONE
0x32 => 0x32, //DIGIT TWO
0x33 => 0x33, //DIGIT THREE

0x34 => 0x34, //DIGIT FOUR
0x35 => 0x35, //DIGIT FIVE
0x36 => 0x36, //DIGIT SIX
0x37 => 0x37, //DIGIT SEVEN
0x38 => 0x38, //DIGIT EIGHT
0x39 => 0x39, //DIGIT NINE
0x3A => 0x3A, //COLON
0x3B => 0x3B, //SEMICOLON
0x3C => 0x3C, //LESS-THAN SIGN
0x3D => 0x3D, //EQUALS SIGN
0x3E => 0x3E, //GREATER-THAN SIGN
0x3F => 0x3F, //QUESTION MARK
0xC2A1 => 0x40, //INVERTED EXCLAMATION MARK
0x41 => 0x41, //LATIN CAPITAL LETTER A
0x42 => 0x42, //LATIN CAPITAL LETTER B
0x43 => 0x43, //LATIN CAPITAL LETTER C
0x44 => 0x44, //LATIN CAPITAL LETTER D
0x45 => 0x45, //LATIN CAPITAL LETTER E
0x46 => 0x46, //LATIN CAPITAL LETTER F
0x47 => 0x47, //LATIN CAPITAL LETTER G
0x48 => 0x48, //LATIN CAPITAL LETTER H
0x49 => 0x49, //LATIN CAPITAL LETTER I
0x4A => 0x4A, //LATIN CAPITAL LETTER J
0x4B => 0x4B, //LATIN CAPITAL LETTER K
0x4C => 0x4C, //LATIN CAPITAL LETTER L
0x4D => 0x4D, //LATIN CAPITAL LETTER M
0x4E => 0x4E, //LATIN CAPITAL LETTER N
0x4F => 0x4F, //LATIN CAPITAL LETTER O
0x50 => 0x50, //LATIN CAPITAL LETTER P
0x51 => 0x51, //LATIN CAPITAL LETTER Q
0x52 => 0x52, //LATIN CAPITAL LETTER R
0x53 => 0x53, //LATIN CAPITAL LETTER S
0x54 => 0x54, //LATIN CAPITAL LETTER T
0x55 => 0x55, //LATIN CAPITAL LETTER U
0x56 => 0x56, //LATIN CAPITAL LETTER V
0x57 => 0x57, //LATIN CAPITAL LETTER W
0x58 => 0x58, //LATIN CAPITAL LETTER X
0x59 => 0x59, //LATIN CAPITAL LETTER Y
0x5A => 0x5A, //LATIN CAPITAL LETTER Z
0xC384 => 0x5B, //LATIN CAPITAL LETTER A WITH DIAERESIS
0xC396 => 0x5C, //LATIN CAPITAL LETTER O WITH DIAERESIS
0xC391 => 0x5D, //LATIN CAPITAL LETTER N WITH TILDE
0xC39C => 0x5E, //LATIN CAPITAL LETTER U WITH DIAERESIS
0xC2A7 => 0x5F, //SECTION SIGN
0xC2BF => 0x60, //INVERTED QUESTION MARK
0x61 => 0x61, //LATIN SMALL LETTER A
0x62 => 0x62, //LATIN SMALL LETTER B

```
    0x63 => 0x63, //LATIN SMALL LETTER C
    0x64 => 0x64, //LATIN SMALL LETTER D
    0x65 => 0x65, //LATIN SMALL LETTER E
    0x66 => 0x66, //LATIN SMALL LETTER F
    0x67 => 0x67, //LATIN SMALL LETTER G
    0x68 => 0x68, //LATIN SMALL LETTER H
    0x69 => 0x69, //LATIN SMALL LETTER I
    0x6A => 0x6A, //LATIN SMALL LETTER J
    0x6B => 0x6B, //LATIN SMALL LETTER K
    0x6C => 0x6C, //LATIN SMALL LETTER L
    0x6D => 0x6D, //LATIN SMALL LETTER M
    0x6E => 0x6E, //LATIN SMALL LETTER N
    0x6F => 0x6F, //LATIN SMALL LETTER O
    0x70 => 0x70, //LATIN SMALL LETTER P
    0x71 => 0x71, //LATIN SMALL LETTER Q
    0x72 => 0x72, //LATIN SMALL LETTER R
    0x73 => 0x73, //LATIN SMALL LETTER S
    0x74 => 0x74, //LATIN SMALL LETTER T
    0x75 => 0x75, //LATIN SMALL LETTER U
    0x76 => 0x76, //LATIN SMALL LETTER V
    0x77 => 0x77, //LATIN SMALL LETTER W
    0x78 => 0x78, //LATIN SMALL LETTER X
    0x79 => 0x79, //LATIN SMALL LETTER Y
    0x7A => 0x7A, //LATIN SMALL LETTER Z
    0xC3A4 => 0x7B, //LATIN SMALL LETTER A WITH DIAERESIS
    0xC3B6 => 0x7C, //LATIN SMALL LETTER O WITH DIAERESIS
    0xC3B1 => 0x7D, //LATIN SMALL LETTER N WITH TILDE
    0xC3BC => 0x7E, //LATIN SMALL LETTER U WITH DIAERESIS
    0xC3A0 => 0x7F, //LATIN SMALL LETTER A WITH GRAVE
);

$UTF8_ISO_8859_1 = array(
    0x20 => 0x20,
    0x21 => 0x21,
    0x22 => 0x22,
    0x23 => 0x23,
    0x24 => 0x24,
    0x25 => 0x25,
    0x26 => 0x26,
    0x27 => 0x27,
    0x28 => 0x28,
    0x29 => 0x29,
    0x2A => 0x2A,
    0x2B => 0x2B,
    0x2C => 0x2C,
    0x2D => 0x2D,
    0x2E => 0x2E,
```

0x2F => 0x2F,
0x30 => 0x30,
0x31 => 0x31,
0x32 => 0x32,
0x33 => 0x33,
0x34 => 0x34,
0x35 => 0x35,
0x36 => 0x36,
0x37 => 0x37,
0x38 => 0x38,
0x39 => 0x39,
0x3A => 0x3A,
0x3B => 0x3B,
0x3C => 0x3C,
0x3D => 0x3D,
0x3E => 0x3E,
0x3F => 0x3F,
0x40 => 0x40,
0x41 => 0x41,
0x42 => 0x42,
0x43 => 0x43,
0x44 => 0x44,
0x45 => 0x45,
0x46 => 0x46,
0x47 => 0x47,
0x48 => 0x48,
0x49 => 0x49,
0x4A => 0x4A,
0x4B => 0x4B,
0x4C => 0x4C,
0x4D => 0x4D,
0x4E => 0x4E,
0x4F => 0x4F,
0x50 => 0x50,
0x51 => 0x51,
0x52 => 0x52,
0x53 => 0x53,
0x54 => 0x54,
0x55 => 0x55,
0x56 => 0x56,
0x57 => 0x57,
0x58 => 0x58,
0x59 => 0x59,
0x5A => 0x5A,
0x5B => 0x5B,
0x5C => 0x5C,
0x5D => 0x5D,

0x5E => 0x5E,
0x5F => 0x5F,
0x60 => 0x60,
0x61 => 0x61,
0x62 => 0x62,
0x63 => 0x63,
0x64 => 0x64,
0x65 => 0x65,
0x66 => 0x66,
0x67 => 0x67,
0x68 => 0x68,
0x69 => 0x69,
0x6A => 0x6A,
0x6B => 0x6B,
0x6C => 0x6C,
0x6D => 0x6D,
0x6E => 0x6E,
0x6F => 0x6F,
0x70 => 0x70,
0x71 => 0x71,
0x72 => 0x72,
0x73 => 0x73,
0x74 => 0x74,
0x75 => 0x75,
0x76 => 0x76,
0x77 => 0x77,
0x78 => 0x78,
0x79 => 0x79,
0x7A => 0x7A,
0x7B => 0x7B,
0x7C => 0x7C,
0x7D => 0x7D,
0x7E => 0x7E,
0xC2A0 => 0xA0,
0xC2A1 => 0xA1,
0xC2A2 => 0xA2,
0xC2A3 => 0xA3,
0xC2A4 => 0xA4,
0xC2A5 => 0xA5,
0xC2A6 => 0xA6,
0xC2A7 => 0xA7,
0xC2A8 => 0xA8,
0xC2A9 => 0xA9,
0xC2AA => 0xAA,
0xC2AB => 0xAB,
0xC2AC => 0xAC,
0xC2AD => 0xAD,

0xC2AE => 0xAE,
0xC2AF => 0xAF,
0xC2B0 => 0xB0,
0xC2B1 => 0xB1,
0xC2B2 => 0xB2,
0xC2B3 => 0xB3,
0xC2B4 => 0xB4,
0xC2B5 => 0xB5,
0xC2B6 => 0xB6,
0xC2B7 => 0xB7,
0xC2B8 => 0xB8,
0xC2B9 => 0xB9,
0xC2BA => 0xBA,
0xC2BB => 0xBB,
0xC2BC => 0xBC,
0xC2BD => 0xBD,
0xC2BE => 0xBE,
0xC2BF => 0xBF,
0xC380 => 0xC0,
0xC381 => 0xC1,
0xC382 => 0xC2,
0xC383 => 0xC3,
0xC384 => 0xC4,
0xC385 => 0xC5,
0xC386 => 0xC6,
0xC387 => 0xC7,
0xC388 => 0xC8,
0xC389 => 0xC9,
0xC38A => 0xCA,
0xC38B => 0xCB,
0xC38C => 0xCC,
0xC38D => 0xCD,
0xC38E => 0xCE,
0xC38F => 0xCF,
0xC390 => 0xD0,
0xC391 => 0xD1,
0xC392 => 0xD2,
0xC393 => 0xD3,
0xC394 => 0xD4,
0xC395 => 0xD5,
0xC396 => 0xD6,
0xC397 => 0xD7,
0xC398 => 0xD8,
0xC399 => 0xD9,
0xC39A => 0xDA,
0xC39B => 0xDB,
0xC39C => 0xDC,

```
    0xC39D => 0xDD,
    0xC39E => 0xDE,
    0xC39F => 0xDF,
    0xC3A0 => 0xE0,
    0xC3A1 => 0xE1,
    0xC3A2 => 0xE2,
    0xC3A3 => 0xE3,
    0xC3A4 => 0xE4,
    0xC3A5 => 0xE5,
    0xC3A6 => 0xE6,
    0xC3A7 => 0xE7,
    0xC3A8 => 0xE8,
    0xC3A9 => 0xE9,
    0xC3AA => 0xEA,
    0xC3AB => 0xEB,
    0xC3AC => 0xEC,
    0xC3AD => 0xED,
    0xC3AE => 0xEE,
    0xC3AF => 0xEF,
    0xC3B0 => 0xF0,
    0xC3B1 => 0xF1,
    0xC3B2 => 0xF2,
    0xC3B3 => 0xF3,
    0xC3B4 => 0xF4,
    0xC3B5 => 0xF5,
    0xC3B6 => 0xF6,
    0xC3B7 => 0xF7,
    0xC3B8 => 0xF8,
    0xC3B9 => 0xF9,
    0xC3BA => 0xFA,
    0xC3BB => 0xFB,
    0xC3BC => 0xFC,
    0xC3BD => 0xFD,
    0xC3BE => 0xFE,
    0xC3BF => 0xFF
);
```

Подсчет количества смс

```
/**
 * @param $text_sms
 * @return false|int
 */
function countSms($text_sms)
{
    // Удаляем экранирование символов
    $text_sms = stripslashes($text_sms);
```

```
    // Определяем кодировку по тексту смс (getCHRmessage - смотрите ниже)
    $encoding = getCHRmessage($text_sms);
    if ($encoding == 0x8) {
        $len_text_sms = mb_strlen($text_sms);
        if ($len_text_sms > 70)
            $count_sms = ceil(($len_text_sms / 67));
        else
            $count_sms = 1;
    } else {
        // Подсчитывает количествол символов в формате GSM (convertEncoding - смотрите ниже)
        $len_text_sms1 = strlen(convertEncoding($text_sms, 0x0, TRUE)); // GSM
        // Подсчитывает количествол символов в формате Latin (convertEncoding - смотрите ниже)
        $len_text_sms2 = strlen(convertEncoding($text_sms, 0x3, TRUE)); // Латиница
        if ($len_text_sms1 > $len_text_sms2)
            $len_text_sms = $len_text_sms1;
        else
            $len_text_sms = $len_text_sms2;

        if ($len_text_sms > 160)
            $count_sms = ceil(($len_text_sms / 153));
        else
            $count_sms = 1;
    }
    return $count_sms;
}
```

Определение кодировки по тексту смс

```
/**
 * @param $text_sms
 * @return int
 */
function getCHRmessage($text_sms)
{
    // Массивы с кодом символов, согласно их кодировке
    global $UTF8_GSM7, $UTF8_ISO_8859_1;
    $len = mb_strlen($text_sms);
    $not_GSM7 = 0;
    $not_ISO = 0;
    for ($i = 0; $i < $len; $i++) {
        // Определяем код символа
        $k = mbGetORD(mb_substr($text_sms, $i, 1));
        if (!isset($UTF8_ISO_8859_1[$k]) and !isset($UTF8_GSM7[$k]))
            return 0x8;
        elseif (!isset($UTF8_GSM7[$k]))
            $not_GSM7++;
        elseif (!isset($UTF8_ISO_8859_1[$k]))
```

```
            $not_ISO++;
    }
    if ($not_GSM7 > $not_ISO)
        return 0x3;
    else
        return 0x0;
}
```

Определяет код символа

```
/**
 * @param $char
 * @return float|int
 */
function mbGetORD($char)
{
    $len = strlen($char);
    $code = 0;
    for ($i = 0; $i < $len; $i++) {
        $code += pow(0x100, $len - 1 - $i) * ord($char[$i]);
    }
    return $code;
}
```

Определяет символ по коду

```
/**
 * @param $ord
 * @return string
 */
function mbGetCHR($ord)
{
    if ($ord == 0)
        return chr(0);
    $char = '';
    while ($ord) {
        $rest = $ord % 0x100;
        $char = chr($rest) . $char;
        $ord = ($ord - $rest) / 0x100;
    }
    return $char;
}
```

Перекодировка текста смс

```
/**
 * @param $string
 * @param int $charset
 * @param bool $to
```

```
 * @return string
 */
function convertEncoding($string, $charset = 0x0, $to = TRUE)
{
   if ($charset === 0x0) {
      global $UTF8_GSM7;
      $conv = $UTF8_GSM7;
   } else {
      global $UTF8_ISO_8859_1;
      $conv = $UTF8_ISO_8859_1;
   }
   if (!$to)
      $conv = array_flip($conv);
   $len = strlen($string);
   $return = '';
   for ($i = 0; $i < $len; $i++) {
      $char1 = ord($string[$i]);
      if ($i < $len - 1)
         $char2 = $char1 * 0x100 + ord($string[$i + 1]);
      if ($i < $len - 2)
         $char3 = $char2 * 0x100 + ord($string[$i + 2]);
      if ($i < $len - 2 and isset($conv[$char3]) and $char1) {
         $tmp = $conv[$char3];
         $i += 2;
      } elseif ($i < $len - 1 and isset($conv[$char2]) and $char1) {
         $tmp = $conv[$char2];
         $i += 1;
      } elseif (isset($conv[$char1]))
         $tmp = $conv[$char1];
      else
         $tmp = -0x01;
      if ($tmp >= 0)
         $return .= mbGetCHR($tmp); // Определяем символ по коду
   }
   return $return;
}
```

Получение текста конкретного сегмента

```
/**
 * @param $text_sms
 * @param $part
 * @return string
 */
function getTextSegment($text_sms, $part)
{
   $text_sms = str_replace("\r\n", "\n", $text_sms);
```

```
    $text_sms = stripslashes($text_sms);
    $len_text_sms = mb_strlen($text_sms);
    // Определяем кодировку сообщения
    $encoding = getCHRmessage($text_sms);
    if ($encoding == 0x8) {
        if ($len_text_sms > 70)
            $len_sms = 67;
        else
            $len_sms = 70;
    } else {
        // Подсчитывает количествол символов в формате GSM
        $len_text_sms1 = strlen(convertEncoding($text_sms, 0x0, TRUE)); // GSM
        // Подсчитывает количествол символов в формате Latin
        $len_text_sms2 = strlen(convertEncoding($text_sms, 0x3, TRUE)); // Latin
        if ($len_text_sms1 > $len_text_sms2)
            $len_text_sms = $len_text_sms1;
        else
            $len_text_sms = $len_text_sms2;

        if ($len_text_sms > 160)
            $len_sms = 153;
        else
            $len_sms = 160;
    }
    $part = mb_substr($text_sms, ($part - 1) * $len_sms, $len_sms);
    return $part;
}
```

Преобразование HEX

```
/**
 * @param $_message
 * @return string
 */
function convertHex($_message)
{
    $len = strlen($_message);
    $return = '';
    for ($i = 0; $i < $len; $i += 2) {
        $tmp = hexdec(substr($_message, $i, 2));
        $return .= chr($tmp);
    }
    return $return;
}
```

Получение числового статуса smpp из текстового статуса

```
/**
 * @param $status
 * @return false|int
 */
function getStatusFromShortName($status)
{
   switch ($status) {
      case 'ENROUTE':
         $status_text = 0x1;
         break;
      case 'DELIVRD':
         $status_text = 0x2;
         break;
      case 'DELIVER':
         $status_text = 0x2;
         break;
      case 'EXPIRED':
         $status_text = 0x3;
         break;
      case 'DELETED':
         $status_text = 0x4;
         break;
      case 'UNDELIV':
         $status_text = 0x5;
         break;
      case 'ACCEPTD':
         $status_text = 0x6;
         break;
      case 'UNKNOWN':
         $status_text = 0x7;
         break;
      case 'REJECTD':
         $status_text = 0x8;
         break;
      default:
         $status_text = FALSE;
   }
   return $status_text;
}
```

Преобразование времени в smpp формат

```
/**
 * @param $time - Дата и время в формате mysql
 * @return string - возвращает время в SMPP формате для отчетов о доставке
 */
function get_time_smpp($time)
```

```
{
   $time = substr($time, 2, 2) . substr($time, 5, 2) . substr($time, 8, 2) . substr($time, 11, 2) .
substr($time, 14, 2);
   return $time;
}
```

Преобразование SMPP даты+времени в SQL формат

```
/**
 * @param $time - Дата и время в формате smpp
 * @return false|string
 */
function get_time_from_smpp($time)
{
   $zone = substr($time, 15);
   if ($zone == 'R') {
      $time_ = mktime(date('H') + substr($time, 6, 2), date('i') + substr($time, 8, 2), date('s') +
substr($time, 10, 2), date('m') + substr($time, 2, 2), date('d') + substr($time, 4, 2), date('Y') +
substr($time, 0, 2));
   } else {
      $time_ = mktime(substr($time, 6, 2), substr($time, 8, 2), substr($time, 10, 2), substr($time, 2, 2),
substr($time, 4, 2), '20' . substr($time, 0, 2));
      $time_zote = substr($time, 13, 2);
      if ($zone == '-')
         $time_ += ($time_zote * 900);
      else
         $time_ -= ($time_zote * 900);
   }
   $time = date('Y-m-d H:i:s', $time_);
   return $time;
}
```

Преобразование даты+время в число секунд

```
/**
 * @param $date
 * @return false|int
 */
function get_time($date)
{
   return mktime(substr($date, 11, 2), substr($date, 14, 2), substr($date, 17, 2), substr($date, 5, 2),
substr($date, 8, 2), substr($date, 0, 4));
}
```

Преобразование времени валидации mysql формата в smpp формат

```
/**
 * @param $time
 * @return string
```

```
 */
function get_validity_period($time)
{
   if ($time && $time != '0000-00-00 00:00' && $time != '0000-00-00 00:00:00')
      // Преобразование даты и врменеи формата mysql в число секунд (get_time - смотрите
выше)
      $time = get_time($time);
   else
      $time = time() + 259200;
   $time_now = time();
   $validity_period = $time - $time_now;
   if ($validity_period < 600)
      $validity_period = 600;
   $day = floor($validity_period / 86400);
   if ($day > 31)
      $day = 31;
   $validity_period = $validity_period % 86400;
   $hour = floor($validity_period / 3600);
   $validity_period = $validity_period % 3600;
   $minute = floor($validity_period / 60);
   $second = $validity_period % 60;
   // функция add_zero - добавляет нули вначале
   $validity_period = '0000' . add_zero($day, 2) . add_zero($hour, 2) . add_zero($minute, 2) .
add_zero($second, 2) . '000R';
   return $validity_period;
}
```

Заполнение строки нулями. нули добавляются в начало строки

```
/**
 * @param $number - изначальное число
 * @param $num - количество цифр
 * @return false|mixed|string
 */
function add_zero($number, $num)
{
   $len = strlen($number);
   if ($len > $num) {
      $number = substr($number, -$num);
      $len = $num;
   }
   for ($i = 0; $i < $num - $len; $i++) {
      $number = '0' . $number;
   }
   return $number;
}
```

Разделение длинных SMS на части (7-ми битовые)

```
/**
 * @param $text
 * @param int $message_sequence
 * @return array
 */
function split_message($text, $message_sequence = 0)
{
   $max_len = 153;
   $res = array();
   $len_text = strlen($text);
   if ($len_text <= 160)
      $result[] = $text;
   else {
      if ($message_sequence)
         $msg_sequence = $message_sequence;
      else
         $msg_sequence = ++$this->_message_sequence;

      $num_messages = ceil($len_text / $max_len);
      for ($i = 0; $i < $num_messages; $i++) {
         $ttext = substr($text, $i * $max_len, $max_len);
         $udh = pack("cccccc", 5, 0, 3, $msg_sequence, $num_messages, $i + 1);
         $result[] = $udh . $ttext;
      }
   }
   return $result;
}
```

Разделение длинных SMS на части (16-ти битовые)

```
/**
 * @param $text
 * @param int $message_sequence
 * @return array
 */
function split_message_unicode($text, $message_sequence = 0)
{
   $max_len = 67;
   $res = array();
   $len_text = mb_strlen($text, 'UCS-2BE');
   if ($len_text <= 70)
      $result[] = $text;
   else {
      if ($message_sequence)
         $msg_sequence = $message_sequence;
```

```
      else
         $msg_sequence = ++$this->_message_sequence;

      $num_messages = ceil($len_text / $max_len);
      for ($i = 0; $i < $num_messages; $i++) {
         $ttext = mb_substr($text, $i * $max_len, $max_len, 'UCS-2BE');
         $udh = pack("cccccc", 5, 0, 3, $msg_sequence, $num_messages, $i + 1);
         $result[] = $udh . $ttext;
      }
   }
   return $result;
}
```

Распаковка пакета

```
/**
 * @param $spec - формат пакета (Ncommand_length/Ncommand_id/Ncommand_status/Nsequence_number более подробно смитрите документацию по php функции unpack())
 * @param $data - пакет
 * @return array возвращает TRUE - в случае удачни, если не задан формат тела пакета, массив прочитанных параметров, если задан формат и FALSE - в случае неудачи
 */
function unpack_smpp($spec, $data)
{
   $res = array();
   $specs = explode("/", $spec);
   $pos = 0;
   reset($specs);
   while (list(, $sp) = each($specs)) {
      $subject = substr($data, $pos);
      $type = substr($sp, 0, 1);
      $var = substr($sp, 1);
      switch ($type) {
         case "N":
            $temp = unpack("Ntemp2", $subject);
            $res[$var] = $temp["temp2"];
            $pos += 4;
            break;
         case "c":
            $temp = unpack("ctemp2", $subject);
            $res[$var] = $temp["temp2"];
            $pos += 1;
            break;
         case "C":
            $temp = unpack("Ctemp2", $subject);
            $res[$var] = $temp["temp2"];
```

```
            $pos += 1;
            break;
        case "a":
            $pos2 = strpos($subject, chr(0)) + 1;
            $temp = unpack('a' . ($pos2 - 1) . 'temp2', $subject);
            $res[$var] = $temp["temp2"];
            $pos += $pos2;
            break;
        case "o":
            $pos2 = end($res);//[count($res)-1];
            $temp = unpack("a{$pos2}temp2", $subject);
            $res[$var] = $temp["temp2"];
            $pos += $pos2;
            break;
      }
   }

   $len_data = strlen($data);
   for ($i = 0; $pos < $len_data; $i++) {
      $subject = substr($data, $pos);
      $tag = substr($subject, 0, 2);
      $tag = unpack("ntag", $tag);
      $size = substr($subject, 2, 2);
      $size = unpack("nsize", $size);
      $value = substr($subject, 4, $size['size']);
      $pos = $pos + $size['size'] + 4;
      $res['size'][$tag['tag']] = $size['size'];
      $res['value'][$tag['tag']] = $value;
   }
   return $res;
}
```