

API обмена CRM с партнеркой

Изменения:

- 2014-05-12 Добавлен вывод журнала звонков в методы getOrderStatus и getOrderStatusR
- 2014-05-29 Добавлено поле ip адреса в метод addOrder
- 2014-06-10 В ответе метода addOrder возвращает ID заказа
- 2014-07-16 Добавлено описание поля kolvo метода addOrder
- 2014-11-17 Добавлено описание метода getOrders

Основные положения:

1. Для обмена между CRM и партнеркой используется протокол [http](#) метод POST.
2. Инициатором связи является партнерка. Т.е. она является клиентом.
3. Клиент осуществляет запрос на точку входа, находящуюся в зоне CRM системы. URL запроса будет выглядеть так:
`http://<путь до корня CRM>/exapi`
4. Сообщения передаются в модифицированном [JSON-RCP](#) 1.0 .
5. На каждое сообщение сервер обязан дать ответ, если сообщение прошло проверку подписи.
6. Если подпись сообщения не прошла проверку, то сервер не дает никакого ответа, соединение разрывается.
7. При превышении определенного количества неудачных запросов клиент банится.

Описание протокола и отличия от стандартного JSON-RCP 1.0

Модификация протокола заключается в добавлении идентификатора отправителя и md5 подписи к каждому запросу. Данная подпись позволяет однозначно идентифицировать отправителя и удостовериться что он имеет права выполнить то или иное действие. Алгоритм формирования сообщения будет следующим:

1. Сформировать объект запроса в формате стандартного JSON-RPC 1.0 .
2. Преобразовать полученный объект в json-строку.
3. Осуществить конкатенацию полученной json-строки с идентификатором отправителя и секретным ключом обмена.
4. Вычислить хэш md5 строки, полученной на предыдущем шаге.
5. Сформировать объект со следующими полями:

sender	string(64)	Идентификатор отправителя
sign	string(32)	Подпись, хэш md5, полученная на шаге 4.
request	string	Json строка, полученная на шаге 2

6. Из данного объекта сформировать json строку

Сформированное сообщение должно быть установлено в теле запроса POST.

Структура объекта запроса в формате стандартного JSON-RPC 1.0 :

method	string	Наименование метода (т. е. функции), который должен быть выполнен сервером
params	array	Массив (!) параметров
id	string	Идентификатор запроса, в идеале должен быть уникальным. Сервер в ответе на запрос добавляет этот идентификатор, чтобы клиент знал на какой запрос это ответ. Может быть пустой строкой.

Сервер (CRM система) должна ответить на запрос клиента сообщением, сформированному по следующему алгоритму:

1. Сформировать объект ответа в формате стандартного JSON-RPC 1.0
2. Преобразовать полученный объект в json-строку
3. Осуществить конкатенацию полученной json-строки с идентификатором клиента (отправителя) и секретным ключом обмена.
4. Вычислить хэш md5 строки, полученной на предыдущем шаге.
5. Сформировать объект со следующими полями:

sign	string(32)	Подпись, хэш md5, полученная на шаге 4.
answer	string	Json строка, полученная на шаге 2

6. Из данного объекта сформировать json строку.

Структура объекта ответа в формате JSON-RCP 1.0 :

result	mixed	Результат выполнения метода. Может быть любого типа (объект, массив, строка, число и т.п.)
error	null или string	Описание ошибки или NULL, если ошибки нет.
id	string	Идентификатор запроса.

Данная json строка должна быть отправлена серверу в теле ответа. Сервер устанавливает content-type для ответа application/json .

Примеры реализации (для того, чтобы было все понятно)

Клиент:

```
<?php
$now = time(); //текущее время в unix timestamp

$url = "http://path_to_crm/excapi"; //url запроса, т.е. точка входа
$clientID = "partner_1"; //идентификатор клиента
$secret = "This is my secret phrase"; //секретный ключ

//Шаг 1. Создаем объект запроса
$req = array(
    'method' => 'addOrder',
    'params' => array(
        array(
            'order_id' => 123, // параметры запроса
            'good_id' => 123,
            //... прочие параметры
        )
    ),
    'id' => ($now * 1000 + rand(0, 999)) // псевдо-случайное id запроса
);

//Шаг 2. Получаем json строку из объекта запроса
$req_str = json_encode($req);

//Шаг 3 и 4. Получаем подпись запроса.
$chk_str = $req_str . $clientID . $secret;
$sign = md5($chk_str);

//Шаг 5 и 6. Сформируем тело запроса
$body_obj = array(
    'sender' => $clientID,
    'sign' => $sign,
    'request' => $req_str
);
```

```
$body_str = json_encode($body_obj);

//Сформируем и выполним сам запрос
$httpreq = new HttpRequest($url, HTTP_METH_POST);
$httpreq->setContentType('application/json');
$httpreq->setBody($body_str);
//получим ответ от сервера
$response = $httpreq->send();
```

Сервер:

```
<?php
//получим тело запроса, от клиента
$post_data = file_get_contents("php://input");
//преобразуем строку запроса в php Объект
$body = json_decode($post_data);
//проверим подпись сообщения
$clientID = $body->sender;
$secret = getClientSecret($clientID);
$chk_str = $body->request . $clientID . $secret;
$validSign = md5($chk_str);
if(!$chk_str || $validSign != $body->sign) {
    /*клиент не прошел аутентификацию, шлем его в прекрасное далеко,
    если наглеет не первый раз, то вообще баним его */
}
//получаем объект запроса
$req = json_decode($body->request);
//выполним запрос
list($res, $err) = execJsonRpc($req);

//Шаг 1 и 2. Сформируем объект ответа и преобразуем его в json строку
$ans = array(
    'result' => $res,      // результат
    'error' => $err,      // ошибка
    'id' => $body->id      // идентификатор запроса
);
$ans_str = json_encode($ans);
//Шаг 3 и 4. Сформируем подпись сообщения.
$chk_str = $ans_str . $clientID . $secret;
$sign = md5($chk_str);
//Шаг 5 и 6. Сформируем тело ответа.
$ans_body = array(
    'sign' => $sign,
    'answer' => $ans_str
);
$ans_body_str = json_encode($ans_body);
```

```
//Отправить ответ клиенту
```

```
header('Content-Type: application/json');
```

```
echo $ans_body_str;
```

Методы и параметры:

метод	addOrder	добавить заказ в CRM																														
параметры	объект	<p>Объект со следующей структурой:</p> <table> <tr> <td>ip</td><td>string</td><td>IP адрес</td></tr> <tr> <td>order_id</td><td>string</td><td>Номер заказа</td></tr> <tr> <td>good_id</td><td>string</td><td>Наименование товара</td></tr> <tr> <td>kolvo</td><td>unsigned integer</td><td>Количество продукта</td></tr> <tr> <td>affiliate_id</td><td>string</td><td>Ид аффилиата</td></tr> <tr> <td>country_kod</td><td>string</td><td>код страны</td></tr> <tr> <td>fio</td><td>string</td><td>ФИО</td></tr> <tr> <td>address</td><td>string</td><td>Адрес</td></tr> <tr> <td>phone</td><td>string</td><td>Номер телефона</td></tr> <tr> <td>comment</td><td>string</td><td>Примечание к заказу</td></tr> </table> <p>Пример: {'order_id': 'order 1', 'good_id': 'good 1', ..., 'comment': 'bla bla bla'}</p>	ip	string	IP адрес	order_id	string	Номер заказа	good_id	string	Наименование товара	kolvo	unsigned integer	Количество продукта	affiliate_id	string	Ид аффилиата	country_kod	string	код страны	fio	string	ФИО	address	string	Адрес	phone	string	Номер телефона	comment	string	Примечание к заказу
ip	string	IP адрес																														
order_id	string	Номер заказа																														
good_id	string	Наименование товара																														
kolvo	unsigned integer	Количество продукта																														
affiliate_id	string	Ид аффилиата																														
country_kod	string	код страны																														
fio	string	ФИО																														
address	string	Адрес																														
phone	string	Номер телефона																														
comment	string	Примечание к заказу																														
возвращает	integer	ID заказа в нашей CRM, если он успешно добавлен																														
	false	заказ не добавлен																														

метод	getOrderStatus	получить статус заказа(ов) из CRM
параметры	массив	Массив содержащий список идентификаторов заказов. Пример: ['order 1', 'order 2', 'order 3', ..., 'order N']
	integer, optional	0 или 1, 0 по умолчанию, если 1, то ответ будет в том же формате что и у метода getOrderStatusR
возвращает	массив	<p>Массив, содержащий статусы заказов (массив со следующей структурой:</p> <pre>[<номер заказа>, <статус заказа>, <кол-во звонков по заказу>, <коммент к заказу>, <коммент к звонку>, <ревизия входа>, <ревизия обновления>, [//журнал звонков { 'date' : <дата звонка ISO 8601>, 'state': <статус 0 — не дозвон, 1 — дозвон>, }]]</pre>

		<pre> 'recall': <перезвонить дата ISO 8601>, 'comment': <комментарий к звонку> }, ...]]) по индексам соответствующим тем что в параметре запроса. Пример: [["ml1030002849","pending","0","","", ...], ..., ["ml1030002825","rejected","0","","", ...]]; </pre>
--	--	---

Расшифровка статусов заказов:

Значение статуса	Расшифровка статуса
pending	В ожидании
confirmed	Подтвержден
rejected	Отказ
paid	Оплачен
delivered	Доставлен
return	Возврат
duplicated	Дубль

Данные значения статусов передаются клиенту в ответ на запрос getOrderStatus

метод	getOrderStatusR	получить статус заказа(ов) из CRM
параметры	integer	Номер ревизии от которого будут возвращены изменения
	integer, optional	0 или 1, 1 по умолчанию, если 0, то ответ будет дан как у getOrderStatus
возвращает	объект	Объект имеющий следующую структуру: <pre> { 'rev': <номер актуальной ревизии>, 'orders': [{ 'nmb': <номер заказа>, 'status': <статус заказа>, 'call_cnt': <колво звонков по заказу>, 'comment': <комментарий>, 'call_comment': <комментарий к звонку>, 'add_rev': <ревизия входа>, 'upd_rev': <ревизия обнов>, 'call_log': [{ 'date' : <дата звонка ISO 8601>, 'state': <статус 0 — не дозвон, 1 — дозвон>, 'recall': <перезвонить дата ISO 8601>, 'comment': <комментарий к звонку> }, </pre>

		<pre> ...] }, ...] }</pre>
--	--	--

метод	getOrders	возвращает заказы партнера
параметры	datetime	Дата время в формате понимаемом функцией PHP date_create
возвращает	объект	<p>Массив имеющий следующую структуру:</p> <pre> Array ([0] => Array ([number] => km-1000389297 [date] => 2014-09-11 15:30:39 [fio] => иванов валерий Владимирович [phone1] => +79167802802 [phone2] => [zipCode] => 143054 [city] => Московская обл. Одинцовский р-н. дач. поселок Рампа [street] => нет улицы [house] => --- [flat] => [deliveryCost] => 390 [deliveryDate] => 2014-09-13 [deliveryStime] => 10:00:00 [deliveryEtime] => 18:00:00 [approveDate] => 2014-09-11 15:42:03 [goodItems] => Array ([0] => Array ([goodName] => Police 1101 [goodArticle] => Police 1101 [price] => 1590 [quantity] => 1)))) [1] => Array ([number] => km-1000389311 [date] => 2014-09-11 15:36:07</pre>

		<div>[fio] => Безбородов Сергей Кузьмич [phone1] => +79039101840 [phone2] => +79130937023 [zipCode] => 656006 [city] => г. Барнаул, Алтайский край. [street] => ул. Балтийская [house] => д. 4 [flat] => 43 [deliveryCost] => 390 [deliveryDate] => [deliveryStime] => [deliveryEtime] => [approveDate] => 2014-09-11 15:41:03 [goodItems] => Array ([0] => Array ([goodName] => Летающая Фея [goodArticle] => Летающая [price] => 1990 [quantity] => 1))))</div>
--	--	---