

merchantReceipt

PHP.

merchantReceipt. - , merhantReceipt MerchantHelper.

```
class Item{
    public $price;
    public $quantity;
    public $title;
    public $tax;
    public function __construct($base_price, $quantity, $title, $tax,
$discount) {
        $this->price = $base_price - round($base_price * $discount, 2);
        $this->quantity = $quantity;
        $this->tax = $tax;
        $this->title = $title;
    }
}

class MerchantReceiptHelper{
    private $common_price=0;
    private $items = array();
    private $recipient_amount;
    private $discount;
    private $group;
    private $inn;
    private $customer_contact;

    public function __construct($recipient_amount, $inn,
$customer_contact, $group=null, $discount_value=0, $order_price=0) {
        $this->recipient_amount = $recipient_amount;
        $this->group = $group;
        $this->inn = $inn;
        $this->customer_contact = $customer_contact;
        $this->setDiscount($discount_value, $order_price);
    }

    public function addItem($base_price, $quantity, $title, $tax) {
        $item = new Item($base_price, $quantity, $title, $tax, $this-
>discount);
        $this->common_price += $item->price*$item->quantity;
        array_push($this->items, $item);
    }

    private function setDiscount($discount_value, $order_price){
        $discount = $discount_value;
        if($this->recipient_amount != $order_price){
            $discount += $order_price - $this->recipient_amount;
        }
        if ($discount) {
            $discount = $discount / ($order_price+$discount_value);
        }
        $this->discount = $discount;
        if(empty($discount)){
            $this->discount = 0;
        }
    }

    public function generateMerchantReceipt(){
        $merchantReceipt = [
            "inn" => $this->inn,
            "group" => $this->group,
            "content" => [
                "type" => "1",
```

-
-
-
-
- 1. ■ :
 ■
 ■
- 2. ' :
 ■
 ■
- 3. ' :
 ■
 ■
- MerchantReceipt
 ■
 MerchantHel
 per
- merchantRe
 ceipt
-

•

```

        "positions" => $this->getItems(),
        "customerContact" => $this->customer_contact
    ]
];
return htmlentities(json_encode($merchantReceipt), ENT_QUOTES);
}

private function getItems(){
    $positions = array();
    $sum_difference = round($this->recipient_amount - $this->common_price, 2);

    foreach ($this->items as $item){
        if (abs($sum_difference) != 0) {
            echo $sum_difference;
            if ($item->quantity == 1){
                $item->price += $sum_difference;
            } elseif ($item->quantity > 1){
                $item->quantity--;
                $positions[] = self::createElementForPositions(1,
$item->price+$sum_difference, $item->tax, $item->title);
            } else {
                $item->price = $item->price + round($sum_difference
/$item->quantity,2);
            }
            $sum_difference = 0;
        }
        $positions[] = $this->createElementForPositions($item->quantity, $item->price, $item->tax, $item->title);
    }

    return $positions;
}

private function createElementForPositions($quantity, $price, $tax, $title){
    return array(
        'quantity' => $quantity,
        'price' => $price,
        'tax' => $tax,
        'text' => $title
    );
}
}

```

\$inn	
\$customer_contact	: email
\$group	, null, - Main
\$tax	1 6
\$products	,
\$recipient_amount	, IntellectMoney

\$discount_amount	.
	' - '
	.
	- 0.
\$order_price	
\$merchant_helper	MerchantHelper

, (, ,)

```
$inn = '58465584558245';//
$customer_contact = "mail@mail.ru";//email
$group = "Main";//
$tax = 1; //
```

1.

:

- 1. 200.
- 2. 3 (1 100 50)

```
$products = array();//
$products[0] = array(
    "PRICE" => 100.00,
    "QUANTITY" => 1,
    "NAME" => " 1"
);
$products[1] = array(
    "PRICE" => 50.00,
    "QUANTITY" => 1,
    "NAME" => " 2"
);
$products[2] = array(
    "PRICE" => 50.00,
    "QUANTITY" => 1,
    "NAME" => " 3"
);
```

, ,

```
$recipient_amount = 200.00; // , IntellectMoney
$discount_amount = 0;// , -
$order_price = 200.00; //
```

2. ,

:

- 1. 200.
- 2. 3 (1 100 50).
- 3. 180 .

```

$products = array();//
$products[0] = array(
    "PRICE" => 100.00,
    "QUANTITY" => 1,
    "NAME" => " 1"
);
$products[1] = array(
    "PRICE" => 50.00,
    "QUANTITY" => 1,
    "NAME" => " 2"
);
$products[2] = array(
    "PRICE" => 50.00,
    "QUANTITY" => 1,
    "NAME" => " 3"
);

```

’ ’

```

$recipient_amount = 20.00; // 200 - 180 = 20
$discount_amount = 0; // , ,
$order_price = 200.00; //

```

3. .

CMS .

:

1. 222 30 .
2. 3 (1 122 30 50) .
3. 22 23 (10%)
4. 120 7, 50 50 ()
5. 180 .
6. 20 7 IntellectMoney.

```

$products = array();//
$products[0] = array(
    "PRICE" => 122.3,
    "QUANTITY" => 1,
    "NAME" => " 1"
);
$products[1] = array(
    "PRICE" => 50.00,
    "QUANTITY" => 1,
    "NAME" => " 2"
);
$products[2] = array(
    "PRICE" => 50.00,
    "QUANTITY" => 1,
    "NAME" => " 3"
);

```

’ ’

```
$recipient_amount = 20.07; //  
$discount_amount = 22.23; // , ,  
$order_price = 222.3; //
```

MerchantReceipt

MerchantHelper

```
$merchant_helper = new MerchantReceiptHelper($recipient_amount, $inn,  
$customer_contact, $group, $discount_amount, $order_price);
```

```
foreach ($products as $product){  
    $merchant_helper->addItem($product["PRICE"], $product['QUANTITY'],  
$product['NAME'], $tax);  
}
```

merchantReceipt

```
echo $merchant_helper->generateMerchantReceipt();
```

	positions	merchantReceipt

1	<pre> Array ([0] => Array ([quantity] => 1 [price] => 100 [tax] => 1 [text] => 1) [1] => Array ([quantity] => 1 [price] => 50 [tax] => 1 [text] => 2) [2] => Array ([quantity] => 1 [price] => 50 [tax] => 1 [text] => 3)) </pre>	<pre> {"inn":"58465584558245","group":"Main","content": {"type":"1","positions": [{"quantity":1,"price":100,"tax":1,"text": "\u0422\u043e\u0432\u0430\u0440 1"}, {"quantity":1,"price":50,"tax":1,"text": "\u0422\u043e\u0432\u0430\u0440 2"}, {"quantity":1,"price":50,"tax":1,"text": "\u0422\u043e\u0432\u0430\u0440 3"}], "customerContact":"mail@mail.ru"}} </pre>
---	---	---

2

```
Array (  
  [0] => Array  
(  
  [quantity] => 1  
  
  [price] => 10  
  
  [tax] => 1  
  
  [text] => 1  
  )  
  [1] => Array  
(  
  
  [quantity] => 1  
  
  [price] => 5  
  
  [tax] => 1  
  
  [text] => 2  
  )  
  [2] => Array  
(  
  
  [quantity] => 1  
  
  [price] => 5  
  
  [tax] => 1  
  
  [text] => 3  
  )  
)
```

```
{"inn":"58465584558245","group":"Main","  
content":  
{"type":"1","positions":  
[{"quantity":1,"price":10,"tax":1,"text":  
"\u0422\u043e\u0432\u0430\u0440 1"},  
{"quantity":1,"price":5,"tax":1,"text":  
"\u0422\u043e\u0432\u0430\u0440 2"},  
{"quantity":1,"price":5,"tax":1,"text":  
"\u0422\u043e\u0432\u0430\u0440 3"}],  
"customerContact":"mail@mail.ru"}}
```

3	<pre> Array ([0] => Array ([quantity] => 1 [price] => 11.05 [tax] => 1 [text] => 1) [1] => Array ([quantity] => 1 [price] => 4.51 [tax] => 1 [text] => 2) [2] => Array ([quantity] => 1 [price] => 4.51 [tax] => 1 [text] => 3) </pre>	<pre> {"inn":"58465584558245","group":"Main","content": {"type":"1","positions": [{"quantity":1,"price":11.05,"tax":1,"text":" \u0422\u043e\u0432\u0430\u0440 1"}, {"quantity":1,"price":4.51,"tax":1,"text":" \u0422\u043e\u0432\u0430\u0440 2"}, {"quantity":1,"price":4.51,"tax":1,"text":" \u0422\u043e\u0432\u0430\u0440 3"}], "customerContact":"mail@mail.ru"}} </pre>
---	--	---

```

<?php
class Item{
    public $price;
    public $quantity;
    public $title;
    public $tax;
    public function __construct($base_price, $quantity, $title, $tax,
$discount) {
        $this->price = $base_price - round($base_price * $discount, 2);
        $this->quantity = $quantity;
        $this->tax = $tax;
        $this->title = $title;
    }
}

class MerchantReceiptHelper{
    private $common_price=0;
    private $items = array();
    private $recipient_amount;
    private $discount;
    private $group;
    private $inn;
    private $customer_contact;

    public function __construct($recipient_amount, $inn,
$customer_contact, $group=null, $discount_value=0, $order_price=0) {
        $this->recipient_amount = $recipient_amount;

```

```

        $this->group = $group;
        $this->inn = $inn;
        $this->customer_contact = $customer_contact;
        $this->setDiscount($discount_value, $order_price);
    }

    public function addItem($base_price, $quantity, $title, $tax) {
        $item = new Item($base_price, $quantity, $title, $tax, $this->discount);
        $this->common_price += $item->price*$item->quantity;
        array_push($this->items, $item);
    }

    private function setDiscount($discount_value, $order_price){
        $discount = $discount_value;
        if($this->recipient_amount != $order_price){
            $discount += $order_price - $this->recipient_amount;
        }
        if ($discount) {
            $discount = $discount / ($order_price+$discount_value);
        }
        $this->discount = $discount;
        if(empty($discount)){
            $this->discount = 0;
        }
    }

    public function generateMerchantReceipt(){
        $merchantReceipt = [
            "inn" => $this->inn,
            "group" => $this->group,
            "content" => [
                "type" => "1",
                "positions" => $this->getItems(),
                "customerContact" => $this->customer_contact
            ]
        ];
        return htmlentities(json_encode($merchantReceipt), ENT_QUOTES);
    }

    private function getItems(){
        $positions = array();
        $sum_difference = round($this->recipient_amount - $this->common_price, 2);

        foreach ($this->items as $item){
            if (abs($sum_difference) != 0) {
                echo $sum_difference;
                if ($item->quantity == 1){
                    $item->price += $sum_difference;
                } elseif ($item->quantity > 1){
                    $item->quantity--;
                    $positions[] = self::createElementForPositions(1,
                    $item->price+$sum_difference, $item->tax, $item->title);
                } else {
                    $item->price = $item->price + round($sum_difference
                    /$item->quantity,2);
                }
                $sum_difference = 0;
            }
            $positions[] = $this->createElementForPositions($item->quantity,
            $item->price, $item->tax, $item->title);
        }

        return $positions;
    }

    private function createElementForPositions($quantity, $price, $tax,
    $title){
        return array(
            'quantity' => $quantity,

```

```

        'price' => $price,
        'tax' => $tax,
        'text' => $title
    );
}
}

$inn = '58465584558245';//
$customer_contact = "mail@mail.ru";//email
$group = "Main";//
$tax = 1; //

$products = array();//
$products[0] = array(
    "PRICE" => 100.00,
    "QUANTITY" => 1,
    "NAME" => " 1"
);
$products[1] = array(
    "PRICE" => 50.00,
    "QUANTITY" => 1,
    "NAME" => " 2"
);
$products[2] = array(
    "PRICE" => 50.00,
    "QUANTITY" => 1,
    "NAME" => " 3"
);
$recipient_amount = 200.00; //
$discount_amount = 0;// , ,
$order_price = 200.00; //
$merchant_helper = new MerchantReceiptHelper($recipient_amount, $inn,
$customer_contact, $group, $discount_amount, $order_price);
foreach ($products as $product){
    $merchant_helper->addItem($product["PRICE"], $product['QUANTITY'],
$product['NAME'], $tax);
}
echo $merchant_helper->generateMerchantReceipt();

$products2 = array();//
$products2[0] = array(
    "PRICE" => 100.00,
    "QUANTITY" => 1,
    "NAME" => " 1"
);
$products2[1] = array(
    "PRICE" => 50.00,
    "QUANTITY" => 1,
    "NAME" => " 2"
);
$products2[2] = array(
    "PRICE" => 50.00,
    "QUANTITY" => 1,
    "NAME" => " 3"
);
$recipient_amount = 20.00; //
$discount_value = 0;// , ,
$order_price = 200.00; //
$merchant_helper2 = new MerchantReceiptHelper($recipient_amount, $inn,
$customer_contact, $group, $discount_value, $order_price);
foreach ($products2 as $product){
    $merchant_helper2->addItem($product["PRICE"], $product['QUANTITY'],
$product['NAME'], $tax);
}
echo $merchant_helper2->generateMerchantReceipt();

$products3 = array();//
$products3[0] = array(

```

```
        "PRICE" => 122.3,  
        "QUANTITY" => 1,  
        "NAME" => " 1"  
    );  
$products3[1] = array(  
    "PRICE" => 50.00,  
    "QUANTITY" => 1,  
    "NAME" => " 2"  
);  
$products3[2] = array(  
    "PRICE" => 50.00,  
    "QUANTITY" => 1,  
    "NAME" => " 3"  
);  
$recipient_amount = 20.07; //  
$discount_value = 22.23; // , ,  
$order_price = 222.3; //  
$merchant_helper3 = new MerchantReceiptHelper($recipient_amount, $inn,  
$customer_contact, $group, $discount_amount, $order_price);  
foreach ($products3 as $product){  
    $merchant_helper3->addItem($product["PRICE"], $product['QUANTITY'],  
$product['NAME'], $tax);  
}  
echo $merchant_helper3->generateMerchantReceipt();
```