



# WebPay™ Developer Guide

Developer Guide  
v. 2.1.2 – October 2017

## Contents

Contents .....	2
Introduction .....	3
About This Guide.....	3
Target Audience .....	3
Development Procedure .....	3
Introduction to Sandbox Development Environment.....	4
Test Environment Parameters.....	4
Web Application Interface.....	4
Web Application Main Screen .....	5
Company Profile.....	6
Setting the Secret Key .....	7
View of Financial Transactions .....	8
Payment Module Development .....	9
The Principle of Order Placement and Payment.....	9
Forming the Order for Payment .....	10
Payment Form Fields.....	11
Example of payment form.....	17
Order Electronic Signature .....	18
Order Payment.....	19
Upon Payment Notification.....	20
Payment Verification.....	22
Upon Buyer's Return to the Internet Resource .....	22
Transaction Types .....	24
Changing of payment page address.....	26
Changing of WebPay™ store identifier .....	26
Changing the value of wsb_test .....	26
Support .....	27

## Introduction

### About This Guide

This guide describes the development process of the payment module for payment acceptance on the Internet using WebPay™ system and the use of the test environment (Sandbox) to test the module under development.

### Target Audience

This document is prepared for Web developers and specialists who want to test the developed payment module before connecting to WebPay™ system.

### Development Procedure

1. Read this guide carefully;
2. Sign in the Web application Sandbox (<https://sandbox.webpay.by/>) using parameters from the letter received from WebPay Support Service;
3. Get acquainted with Sandbox interface and fill in the required fields (see [Company Profile](#));
4. Start the development of the payment module according to this documentation;
5. Perform module testing;
6. Receive the letter with new parameters to move the payment module from the test system to the real payment system;
7. Make modifications in the payment module.

## Introduction to Sandbox Development Environment

WebPay™ Sandbox is an independent Web application which is the prototype of the real system designed to test and get acquainted with functionality of the real WebPay™ system.

In terms of functionality, it does not differ from the production copy except that it does not perform real processing of bank cards i. e. actions do not lead to actual transfer of funds on the cards.

The development and testing of the payment modules are always to be performed in the test environment.

### Test Environment Parameters

<b>Test environment URL</b>	https://sandbox.webpay.by
<b>User Name</b>	is sent in the letter
<b>Password</b>	is sent in the letter
<b>Store unique identifier</b>	is sent in the letter

### Web Application Interface

The interface of the Web application of Sandbox test system and of the real system differs only in the presence of the special banner noting that the environment is a test environment. Before starting the development, it is recommended to sign in the Web application, get acquainted with its contents.

## Web Application Main Screen

After entering the Web application you will see the following page:

The screenshot displays the WebPay application interface. At the top, a yellow banner reads "Sandbox Environment © WEBPAY". Below this, the WebPay logo and name are visible on the left, and the server time "13 19" and user "test" are shown on the right. A navigation menu on the left lists various sections: Company Home, Accounts, Invoices, Products, Transactions, Log, Reports, and Log out. The main content area is titled "Statistics" and features a form with the following elements:

- Time interval:** Two date pickers with a minus sign between them.
- Currency:** A dropdown menu showing "\*\*\* All currencies \*\*\*".
- Consider the dependence (?):** A checkbox that is currently unchecked.
- Submit:** A button to execute the query.

This page displays some basic statistics on your billing account; also here you can display statistics for the selected time interval. Navigation provides the ability to control your account is displayed on the left.

## Company Profile

Before starting, you need to verify the basic billing parameters; to do this go to the Profile page and pay attention to the required fields.

**Sandbox Environment** © WEBPAY

**WEBPAY** національний оператор

Server time: 13:20 Logged in as: test

### Company Profile

Billing name: **Test Billing**  
Billing ID: **581682424**

Company name:  \*

Legal person:  \*

Company site:  \*

Company country:  \*

Company street address 1:  \*

Company street address 2:

Company city:  \*

Company state/province:

Company postal/ZIP code:  \*

Company phone:  \*

Company fax:

Company contact email:  \*

This is the 'from' address for all your outgoing emails and ingoing service emails.

\* - required fields

On this page you need to change all the fields marked with \* (asterisk). The Company Contact Email field is required to send you notices of the transactions completed on your billing.

The next page is Company Settings. On this page you can see mail server settings and other account settings. You can use your own mail server instead of using WebPay™ mail server. To change mail server, please, contact WEBPAY manager and send data for connection using e-mail.

 **Sandbox Environment** © **WEBPAY**


Server time: 13:21    Logged in as: test

- Company Home
  - Profile
  - Preferences
  - Statistics
- Accounts
- Invoices
- Products
- Transactions
  - List Pending
  - List
  - Search
- Log
  - List
  - Search
- Reports
- Log
  - Logout

### Company Preferences

Billing name: **Test Billing**

Billing ID: **581682424**

SMTP host:  \*

Secret key:

Notification URL:

Default language:  \*

Default time format ①:  \*

Default date format ①:  \*

Decimal placement:  \*

\* - required fields

The Secret Key field is described in details below.

### Setting the Secret Key

The secret key is necessary to form the electronic signature for each of your payments (see [Order Electronic Signature](#)). This parameter is mandatory and without it will not be possible to pay for your goods or services. To set the secret key, fill in the Secret Key field. This field can contain a random sequence of characters.

SMTP host:  \*

Secret key:

Notification URL:

## View of Financial Transactions

To view all financial transactions, use the Payments menu item.

Sandbox Environment © WEBPAY									
WEBPAY									
Server time: 13 22 Logged in as: test									
Transactions									
ID	Account	Payment method	Payment amount	Status	Means of payment	Batch date	Invoice	Notes	
226374439	* Guest *	test	BYR 100,000.00	Authorized	434179xxxxxx0051	02/19/2015 3:54 PM	#24742		
947515273	* Guest *	test	BYR 10,000.00	Authorized	434179xxxxxx0051	10/08/2014 11:14 AM	#20785		
245733031	* Guest *	test	BYR 10,000.00	Authorized	434179xxxxxx0051	10/08/2014 11:10 AM	#20783		
239949399	* Guest *	test	BYR 10,000.00	Authorized	434179xxxxxx0051	10/08/2014 11:06 AM	#20781		
707149788	* Guest *	test	BYR 10,000.00	Authorized	434179xxxxxx0051	10/07/2014 12:53 PM	#20679		
588458400	* Guest *	test	BYR 10,000.00	Authorized	434179xxxxxx0051	10/07/2014 12:53 PM	#20677		
565504725	* Guest *	test	BYR 10,000.00	Authorized	434179xxxxxx0051	10/07/2014 12:24 PM	#20669		
871346157	* Guest *	test	BYR 10,000.00	Authorized	434179xxxxxx0051	05/21/2013 4:37 PM	#11277		
690599142	* Guest *	test	BYR 10,000.00	Authorized	434179xxxxxx0051	05/20/2013 9:25 AM	#11134		
577558019	* Guest *	system	BYR 204,250.00	Completed		04/16/2013 2:31 PM	#8		
329282911	* Guest *	system	\$ 10,000.00	Completed		04/11/2013 12:35 PM	#1		
643234510	* Guest *	test	BYR 1,000.00	Authorized	547087xxxxxx8950	03/25/2013 8:58 PM	#10325		
666705718	* Guest *	test	BYR 10,000.00	Authorized	411111xxxxxx1111	07/20/2011 1:35 PM	#6516		
249215002	* Guest *	system_fee	BYR -500.00	System		07/15/2011 3:49 PM			System Fee, Transaction # 772555824
819525659	* Guest *	test	BYR 10,000.00	Completed		07/15/2011 3:49 PM	#4122		Parent # 772555824
300126518	* Guest *	test	BYR 10,000.00	Completed		07/15/2011 3:49 PM	#4227		Parent # 763793234
609082969	* Guest *	system_fee	BYR -500.00	System		07/15/2011 3:49 PM			System Fee, Transaction # 763793234

This page displays all financial payments (transactions). Each transaction is linked to the Invoice; the invoice can include several transactions, such as:

- **Authorized** - payment authorization state (the funds are blocked on the buyer's bank card). In this state the buyer does not see the charged off amount, but this amount has not been transferred to your bank account yet;
- **Completed** - the funds were debited from the bank card;
- **System** — system transaction.

The Invoice field contains the number are linked to the transactions. To view all the transactions associated with this number and the values with which the payment was made, click on the link. The page containing information about this invoice will open.

**Invoice # 20785**

Creation date: 10/08/2014 11:14 AM  
 Invoice name: 949154496  
 Invoice PO: test\_order\_1412671824572  
 Account: \* Guest \*  
 Billing status: Pending

Invoice total	Billed amount	Auth amount	Amount due
BYR 10,000.00	BYR 0.00	BYR 10,000.00	BYR 0.00

Invoice URL: <https://sandbox.webpay.by/?bqdOp5NdMHcR3B4705MTK8HOecnBqIvphkEeKlouyB89PvsN6iZRpt3KL3%2BEFpgrO9Iv3vfVvG1>

Due date: 10/08/2014 11:14 AM  
 Grace period (days): 3  
 Language: russian

card: 434179xxxxxx0051  
 approval: 000000  
 3D secure: no

Product code	Product Name	Base Price	Quantity	Total
custom	test	BYR 10,000.00	1	BYR 10,000.00

ID	Batch date	Payment amount	Status
947515273	10/08/2014 11:14 AM	BYR 10,000.00	Authorized

1 of 1 record(s) displayed

## Payment Module Development

Payment module development is performed on the server side of the connected client and involves creation or modification of the script source code of the finalized Web application.

### The Principle of Order Placement and Payment

1. Order placement. Generation of the html page with the ordinary html form. The fields needed to make the payment should be specified and filled in as the fields of the form (see [Payment Form Fields](#));
2. Moving to WebPay™ payment form (<https://securesandbox.webpay.by>), for entering the plastic card details (the move is done via POST request);
3. Plastic card data entry;
4. Order payment;
5. Return to the payment module html page (if the buyer does not click the Back button on the payment form, the special notice will be sent (see [Payment Notification](#));

6. Verification of payment parameters;
7. Provision of services (delivery of goods).

### Forming the Order for Payment

To make payment for the order it is necessary to generate the form with special fields (values and description of fields are provided in Table [Payment Form Fields](#)), and redirect the buyer to the payment page using POST method. For testing the address is <https://securesandbox.webpay.by>, for real payments the address is <https://payment.webpay.by>

## Payment Form Fields

Field Name	Mandatory name	Field description	Comment
<b>Основные поля формы оплаты</b>			
<b>*scart</b>	Yes	This field does not contain the value and denotes the request type. <b>Example:</b> <input type='hidden' name='*scart'>	
<b>wsb_storeid</b>	Yes	WebPay™ store identifier. This identifier is created during registration in WebPay™ and is sent in the letter. <b>Example:</b> <input type='hidden' name='wsb_storeid' value='123456789'>	
<b>wsb_store</b>	No	The store name which will be displayed on the payment form. <b>Example:</b> <input type='hidden' name='wsb_store' value='Store name'>	Maximum field length is 64 characters.
<b>wsb_order_num</b>	Yes	The unique order identifier assigned by the store. <b>Example:</b> <input type='hidden' name='wsb_order_num' value='ORDER_1234'>	Maximum field length is 64 characters.
<b>wsb_currency_id</b>	Yes	Currency identifier. Alphabetic three-digit currency code according to ISO4271. <b>Example:</b> <input type='hidden' name='wsb_currency_id' value='BYN'>	Valid values are <b>BYN, USD, EUR, RUB</b>
<b>wsb_version</b>	Yes	Payment form version.	The current version number: <b>2</b>

Field Name	Mandatory name	Field description	Comment
		<b>Example:</b> <code>&lt;input type='hidden' name='wsb_version' value='2'&gt;</code>	
<b>wsb_language_id</b>	No	Payment form language identifier.  <b>Example:</b> <code>&lt;input type='hidden' name='wsb_language_id' value='russian'&gt;</code>	Valid values are <b>russian, english</b> . If this field is not specified, the payment form will determine the language on the base of buyer's browser settings.
<b>wsb_seed</b>	Yes	Random sequence of characters involved in the order electronic signature formation.  <b>Example:</b> <code>&lt;input type='hidden' name='wsb_seed' value='11123232001'&gt;</code>	see Order Electronic Signature
<b>wsb_signature</b>	Yes	Order control value (electronic signature) - the result of function <b>sha1</b> (for version 2, see <b>wsb_version</b> field) or <b>md5</b> , if the protocol version is not specified. This value is a hex-sequence.  <b>Example:</b> <code>&lt;input type='hidden' name='wsb_signature' value='8d4395dab7598c5f4b94d5bc4780f4af'&gt;</code>	see Order Electronic Signature
<b>wsb_return_url</b>	No	URL to which the buyer returns in case of successful payment  <b>Example:</b> <code>&lt;input type='hidden' name='wsb_return_url' value='http://yoursite.com/compete'&gt;</code>	<i>The Order Identifier values (<b>wsb_order_num</b>) and the Transaction Number special value in the WebPay™ are added to this URL.</i>
<b>wsb_cancel_return_url</b>	No	URL to which the buyer returns in case of unsuccessful payment.	<i>The Order Identifier values (<b>wsb_order_num</b>) are added to</i>

Field Name	Mandatory name	Field description	Comment
		<p><b>Example:</b>  <code>&lt;input type='hidden' name='wsb_cancel_return_url' value='http://yoursite.com/cancel'&gt;</code></p>	this URL.
<b>wsb_notify_url</b>	No	<p>This URL is activated regardless of whether the URL in the <b>wsb_return_url</b> field was followed or not. The main purpose of this URL is to inform about the successful payment if the user did not press the Finish button on the payment form.</p> <p><b>Attention!</b> Sending notifier is only possible on standard ports 80 (http) and 443 (https)</p> <p><b>Example:</b>  <code>&lt;input type='hidden' name='wsb_notify_url' value='http://yoursite.com/notify'&gt;</code></p>	see Payment Notification
<b>wsb_test</b>	Yes	<p>This field indicates test payment performance.  1 — to make test payment;  0 - to make real payment.</p> <p><b>Example:</b>  <code>&lt;input type='hidden' name='wsb_test' value='1'&gt;</code></p>	In Sandbox test environment the value of this field is always assumed to be <b>1</b> .
<b>wsb_customer_name</b>	No	<p>Customer's name field.</p> <p><b>Example:</b>  <code>&lt;input type='hidden' name='wsb_customer_name' value='John Smith'&gt;</code></p>	Maximum field length is 255 characters.
<b>wsb_customer_address</b>	No	<p>Customer's address field.</p> <p><b>Example:</b>  <code>&lt;input type='hidden' name='wsb_customer_address' value=' Minsk, Nezavisimosty av., 10-25'&gt;</code></p>	Maximum field length is 255 characters.

Field Name	Mandatory name	Field description	Comment
<b>wsb_service_date</b>	No	Service date field.  <b>Example:</b> <input type='hidden' name='wsb_service_date' value='Delivery to the July 1, 2016'>	Maximum field length is 255 characters.
<b>List of products for payment</b>			
<b>wsb_invoice_item_name[{n}]</b>	Yes	Product unit name.  <b>Example:</b> <input type='hidden' name='wsb_invoice_item_name[0]' value='Product 1'> <input type='hidden' name='wsb_invoice_item_name[1]' value='Product 2'> ... <input type='hidden' name='wsb_invoice_item_name[n-1]' value='Product n'>	Index {n} is to start at 0 and increase by 1 for each subsequent position.
<b>wsb_invoice_item_quantity[{n}]</b>	Yes	The number of product units, the integer that represents the number of product units for each item.  <b>Example:</b> <input type='hidden' name='wsb_invoice_item_quantity[0]' value='2'> <input type='hidden' name='wsb_invoice_item_quantity[1]' value='1'> ... <input type='hidden' name='wsb_invoice_item_quantity[n-1]' value='1'>	Index {n} is to start at 0 and increase by 1 for each subsequent position.
<b>wsb_invoice_item_price[{n}]</b>	Yes	The price of the product unit, the integer that represents the price of each product (BYN, USD, EUR, RUB with 2 digits separated by point or comma from the integer part).  <b>Example:</b> <input type='hidden' name='wsb_invoice_item_price[0]' value='10'> <input type='hidden' name='wsb_invoice_item_price[1]' value='2'>	Index {n} is to start at 0 and increase by 1 for each subsequent position.

Field Name	Mandatory name	Field description	Comment
		... <input type='hidden' name='wsb_invoice_item_price[n-1]' value='4'>	
<b>wsb_tax</b>	No	The value of this field is the tax amount in Belarusian rubles added to the total order amount. <b>Example:</b> <input type='hidden' name='wsb_tax' value='1'>	
<b>wsb_shipping_name</b>	No	The field which determines the name (method) of delivery. <b>Example:</b> <input type='hidden' name='wsb_shipping_name' value='Courier delivery'>	
<b>wsb_shipping_price</b>	No	The value of this field is the amount of delivery added to the total order amount. <b>Example:</b> <input type='hidden' name='wsb_shipping_price' value='2'>	
<b>wsb_discount_name</b>	No	Discount name field. <b>Example:</b> <input type='hidden' name='wsb_discount_name' value='Discount card'>	
<b>wsb_discount_price</b>	No	The value of this field is the discount amount deducted from the total order amount. <b>Example:</b> <input type='hidden' name='wsb_discount_price' value='0.58'>	

Field Name	Mandatory name	Field description	Comment
<b>wsb_discount_promo_code</b>	No	This field contains the value of the promotional code discount for the order. <b>Example:</b> <input type='hidden' name='wsb_discount_promo_code' value='ABC001'>	
<b>wsb_total</b>	Yes	This field is calculated. The value of this field is the total order payment amount. Rules for total amount calculation:  $\text{wsb\_total} = \text{wsb\_invoice\_item\_quantity}[0] * \text{wsb\_invoice\_item\_price}[0] + \text{wsb\_invoice\_item\_quantity}[1] * \text{wsb\_invoice\_item\_price}[1] + \dots$ $\text{wsb\_invoice\_item\_quantity}[N] * \text{wsb\_invoice\_item\_price}[n] + \text{wsb\_tax} + \text{wsb\_shipping\_price} - \text{wsb\_discount\_price}$ <b>Example:</b> <input type='hidden' name='wsb_total' value='28.42'>	The payment will not be made if <b>wsb_total</b> and the calculated value of products do not match. Buyer will be shown the error message.
<b>Additional fields</b>			
<b>wsb_order_tag</b>	No	Order information tag. <b>Example:</b> <input type='hidden' name='wsb_order_tag' value='main'>	Maximum field length is 64 characters.
<b>wsb_email</b>	No	Customer's email address. <b>Example:</b> <input type='hidden' name='wsb_email' value='yourname@yoursite.com'>	This field value will be automatically filled in the relevant field of the payment form.

**NOTE!**

All text fields are to be encoded in UTF-8 or windows-125.

## Example of payment form

```

<form action="https://securesandbox.webpay.by/" method="post">
<input type="hidden" name="*scart">
<input type="hidden" name="wsb_version" value="2">
<input type="hidden" name="wsb_language_id" value="russian">
<input type="hidden" name="wsb_storeid" value="11111111" >
<input type="hidden" name="wsb_store" value="Store name" >
<input type="hidden" name="wsb_order_num" value="ORDER-12345678" >
<input type="hidden" name="wsb_test" value="1" >
<input type="hidden" name="wsb_currency_id" value="BYN" >
<input type="hidden" name="wsb_seed" value="1242649174">
<input type="hidden" name="wsb_customer_name" value="John Smith">
<input type="hidden" name="wsb_customer_address" value="Minsk, Nezavisimosty av., 10-25">
<input type="hidden" name="wsb_service_date" value="Delivery to the July 1, 2016">
<input type="hidden" name="wsb_return_url" value="http://your site url.com/success.php">
<input type="hidden" name="wsb_cancel_return_url" value="http://your site url.com/cancel.php">
<input type="hidden" name="wsb_notify_url" value="http://your site url.com/notify.php">
<input type="hidden" name="wsb_email" value="no@body.tld" >
<input type="hidden" name="wsb_invoice_item_name[]" value="Product 1">
<input type="hidden" name="wsb_invoice_item_quantity[]" value="2">
<input type="hidden" name="wsb_invoice_item_price[]" value="10">
<input type="hidden" name="wsb_invoice_item_name[]" value="Product 2">
<input type="hidden" name="wsb_invoice_item_quantity[]" value="1">
<input type="hidden" name="wsb_invoice_item_price[]" value="0.5">
<input type="hidden" name="wsb_total" value="21.95" >
<input type="hidden" name="wsb_signature"
value="912702512e447846add6fa4985c7a2f271de52e6" >
<input type="hidden" name="wsb_tax" value="1050">
<input type="hidden" name="wsb_shipping_name" value="Courier delivery" >
<input type="hidden" name="wsb_shipping_price" value="0.98" >
<input type="hidden" name="wsb_discount_name" value="Discount card" >
<input type="hidden" name="wsb_discount_price" value="0.58" >
<input type="submit" value="Buy">
</form>

```

## Order Electronic Signature

The electronic signature is formed to prevent changes in the payment form and is required to be present in each order form. Orders without electronic signatures will not be considered by WebPay™.

To form the electronic signature, it is required to set the value of the Secret Key field in your billing account settings (see [Setting the Secret Key](#)).

It is required to specify the following fields in each order form:

**wsb\_seed** - a random character sequence (you can use the current time, for example, unixtime);

**wsb\_signature** - the electronic signature itself. It is to be formed according to the following rule using the values of the following fields:

<b>wsb_seed</b>
<b>wsb_storeid</b>
<b>wsb_order_num</b>
<b>wsb_test</b>
<b>wsb_currency_id</b>
<b>wsb_total</b>
<b>Secret Key</b>

The fields are to be joined into a single line without inverting the joint order. Further, depending on the specified protocol version (`wsb_version`), MD5 (if the version is not specified) or SHA1 (for version 2) of the joint line is calculated. Example of electronic signature formation (in PHP programming language):

```
$wsb_seed = 1242649174;
$wsb_storeid = 11111111;
$wsb_order_num = "ORDER-12345678";
$wsb_test = 1;
$wsb_currency_id = "BYN";
$wsb_total = 21.95;
$SecretKey = "12345678901234567890";
// Joint line value: 124264917411111111ORDER-123456781BYN21.9512345678901234567890

// For protocol version 2 (wsb_version = 2)
$wsb_signature = sha1($wsb_seed.$wsb_storeid.$wsb_order_num.$wsb_test.$wsb_currency_id.$wsb_total.
$SecretKey); // 912702512e447846add6fa4985c7a2f271de52e6

// If the version is not specified
$wsb_signature = md5($wsb_seed.$wsb_storeid.$wsb_order_num.$wsb_test.$wsb_currency_id.$wsb_total.
$SecretKey); // 94993a8063f8ee3c205fe555f8f46319
```

## Order Payment

After formation of the order fields and clicking the Pay button, the buyer should be redirected to the payment page of WebPay™ (test or real) where he will get the opportunity to enter the details of his bank card.

English ▾

**Test payment.** In the "Test payment" write-off money from card processing center in the treatment is not performed.

CARD NUMBER  
434179xxxxxx0051

CARDHOLDER NAME  
Enter your name

CARD EXPIRATION DATE  
MM YY

CVV CODE ⓘ  
---

EMAIL ADDRESS  
Enter email

ORDER NUMBER  
164710775

FOR PAYMENT  
30,00 BYN

Submit

Further information transfer occurs via closed banking networks with the highest level of reliability.  
The service does not transmit Your card data to third parties.

расчет belqi Verified by VISA MasterCard SecureCode PCI DSS COMPLIANT

Transactions are protected by the international safety standard for credit cards PCI DSS

VERIFIED BY GeoTrust Web Pay OOO CLICK 08.11.16 07:01 UTC

After the successful order payment the buyer should click the Finish Payment and Return button, in this case the payment form will forward him to the URL specified in the **wsb\_return\_url** field.

In case of payment error and clicking the Finish button the buyer will be redirected to the URL specified in the **wsb\_cancel\_return\_url** field. In each of these cases, the URL will be appended with the **wsb\_order\_num** values, and in the case of successful payment the URL will also be appended with WebPay™ transaction number value.

### NOTE!

There may be situations when the buyer does not press the Back button on the payment form. To notify of the successful operation, after a while WebPay™ will send special notice.

## Payment Notification

Upon return to the Internet resource page in case of successful payment (the value specified in the **wsb\_return\_url** field is used), the billing sends the order number (**wsb\_order\_num** field value) and the transaction number (**wsb\_tid** field) relevant to the made payment in the GET request parameters.

If the buyer did not click the Back button on the payment form, the billing notifies the Internet resource of executing the operation at the URL specified in the **wsb\_notify\_url** field (if it is specified in the form field or in the settings of your billing account).

In case of notification, the Internet resource should answer with code 200 ("HTTP/1.0 200 OK"); if the Internet resource server does not respond positively and 1 hour has passed since the start of sending notifying requests, the alert email is sent to the store address. 30 days later, if the Internet resource could not receive the notification, sending of requests is terminated, and the Internet resource is notified of it by email.

In the process of development, it is necessary to take into account that the payment notification can come (**wsb\_notify\_url**) and the customer can return to the Internet resource page specified in the field (**wsb\_return\_url**).

### Upon Payment Notification

Two methods are available to receive payment notification:

1. Standard POST request sent from WEBPAY to **wsb\_notify\_url** (default);
2. SOAP request (can be activated upon request to support).

After the successful payment, WebPay™ sends a specially crafted POST request to the address specified in the **wsb\_notify\_url** field of the Internet resource. This request contains payment details. The Internet resource is required to verify the received information in accordance with the requirements of the order execution and answer to the query with the code: "HTTP/1.0 200 OK".

Fields contained in the request:

<b>batch_timestamp</b>	Time of transaction execution
<b>currency_id</b>	Transaction currency
<b>amount</b>	Transaction amount
<b>payment_method</b>	Method of transaction execution (possible values: cc - bank card, test - executed without real card processing)
<b>order_id</b>	Order number in WebPay™
<b>site_order_id</b>	Order number (name) assigned by the store
<b>transaction_id</b>	Transaction number
<b>payment_type</b>	Transaction type <sup>1</sup>

<sup>1</sup> payment\_method 1 and 4 equals to success payment

<b>rrn</b>	Transaction number in Visa/MasterCard
<b>wsb_signature</b>	Electronic signature <sup>2</sup>

**wsb\_signature** is a hex-sequence and is the result of MD5 function. MD5 function argument is the text sequence obtained by simple concatenation of the following fields:

<b>batch_timestamp</b>
<b>currency_id</b>
<b>amount</b>
<b>payment_method</b>
<b>order_id</b>
<b>site_order_id</b>
<b>transaction_id</b>
<b>payment_type</b>
<b>rrn</b>
<b>«Secret key»</b>

The fields are to be concatenated in the specified order.

For SOAP mechanism (can be activated upon request to support), query structure is described in the scheme xsd:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://ws.webpay.by/notifier">
  <xs:element name="NotifierRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="BatchTimestamp" type="xs:int"/>
        <xs:element minOccurs="1" maxOccurs="1" name="CurrencyId" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="Amount" type="xs:decimal"/>
        <xs:element minOccurs="1" maxOccurs="1" name="PaymentMethod" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="OrderId" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="SiteOrderId" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="TransactionId" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="PaymentType" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="RRN" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="WsbSignature" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="Action" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="RC" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="Approval" type="xs:string"/>
        <xs:element minOccurs="0" name="ServiceNumber" type="xs:string"/>
        <xs:element minOccurs="0" name="ServiceAccount" type="xs:string"/>
        <xs:element minOccurs="0" name="ServiceTransactionId" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

<sup>2</sup> The electronic signature is calculated if the Secret Key is specified in billing settings

```

<xs:element minOccurs="0" name="ServiceReceipt" type="xs:string"/>
<xs:element minOccurs="0" name="ServiceResultCode" type="xs:string"/>
<xs:element minOccurs="0" name="CustomerId" type="xs:string"/>
<xs:element minOccurs="0" name="Card" type="xs:string"/>
<xs:element minOccurs="0" name="CardToken" type="xs:string"/>
<xs:element minOccurs="0" name="RecurringToken" type="xs:string"/>
<xs:element minOccurs="0" name="OfferExpDate" type="xs:string"/>
<xs:element minOccurs="0" name="RCText" type="xs:string"/>
<xs:element minOccurs="0" name="OperationType" type="xs:string"/>
<xs:element minOccurs="0" name="Cardholder" type="xs:string"/>
<xs:element minOccurs="0" name="TerminalId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="NotifierResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="code" type="xs:string"/>
      <xs:element name="codeDescription" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

## Payment Verification

Before product (service) delivery, the Internet resource is required to verify the payment made by the buyer.

### Upon Buyer's Return to the Internet Resource

To verify the payment upon return to the Internet resource page specified in the **wsb\_return\_url** field, it is required to run the billing API command «**get\_transaction**».

The following example demonstrates the code for API command execution in WebPay™ system. Pay attention that the request to the test environment should be sent to <https://sandbox.webpay.by>, and the request to the real environment should be sent to <https://billing.webpay.by>.

Example of the request for payment verification (PHP):

```
// API Request
$postdata = '*API=&API_XML_REQUEST='.urlencode('
    <?xml version="1.0" encoding="ISO-8859-1" ?>
    <wsb_api_request>
        <command>get_transaction</command>
        <authorization>
            <username>your_username</username>
            <password>your_md5_password</password>
        </authorization>
        <fields>
            <transaction_id>123456789
            </transaction_id>
        </fields>
    </wsb_api_request>
');
$curl = curl_init ("https://sandbox.webpay.by");
curl_setopt ($curl, CURLOPT_HEADER, 0);
curl_setopt ($curl, CURLOPT_POST, 1);
curl_setopt ($curl, CURLOPT_POSTFIELDS, $postdata);
curl_setopt ($curl, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt ($curl, CURLOPT_RETURNTRANSFER, 1);
curl_setopt ($curl, CURLOPT_SSL_VERIFYHOST, 0);
$response = curl_exec ($curl);
curl_close ($curl);
echo $response;
```

In the result of query execution, XML document containing the following fields comes:

<b>transaction_id</b>	Transaction number
<b>batch_timestamp</b>	Time of transaction execution
<b>currency_id</b>	Transaction currency
<b>amount</b>	Transaction amount
<b>payment_method</b>	Method of transaction execution. (possible values: <b>cc</b> - bank card, <b>test</b> - executed without real card processing)
<b>payment_type</b>	Transaction type
<b>order_id</b>	Order number in WebPay™
<b>order_num</b>	Order number (name) assigned by the store
<b>rrn</b>	Transaction number in Visa/MasterCard
<b>wsb_signature</b>	Electronic signature <sup>3</sup>

**wsb\_signature** is a hex-sequence and is the result of MD5 function. MD5 function argument is the text sequence obtained by simple concatenation of the following fields:

<sup>3</sup> The electronic signature is calculated if the Secret Key is specified in billing settings

<b>transaction_id</b>
<b>batch_timestamp</b>
<b>currency_id</b>
<b>amount</b>
<b>payment_method</b>
<b>payment_type</b>
<b>order_id</b>
<b>rrn</b>
<b>«Secret key»</b>

The fields are to be concatenated in the specified order.

### Transaction Types

Transactions are divided according to the type (**payment\_type** field) into the following groups:

Type	Value
<b>1</b>	Completed
<b>2</b>	Declined
<b>3</b>	Pending
<b>4</b>	Authorized
<b>5</b>	Refunded
<b>6</b>	System
<b>7</b>	Voided
<b>8</b>	Failed
<b>9</b>	Partial Voided
<b>10</b>	Recurrent

#### **NOTE!**

When writing the code to handle additional notices it is required to consider the transaction type. This notification type is used to notify the Internet resource of the successfully completed transactions and the transactions that were subsequently canceled by the Internet resource or the bank in accordance with the contract (refunding).

Thus, the successful payment corresponds to the following types:

<b>1</b>	<b>Completed</b>
<b>4</b>	Authorized
<b>10</b>	Recurrent

While the canceled payments (refunding) correspond to the following types:

<b>5</b>	<b>Refunded</b>
<b>7</b>	Voided
<b>9</b>	Partial Voided

Please also take into account the fact that for the test and real environments maximum and minimum payment amounts are different. For the test environment the minimum payment amount is 0.1 BYN and the maximum payment amount is 10 000 BYN. For the real environment, the transaction limit is set according to the terms of the contract.

## Migrating from test environment to real

Please note when migrating from the test environment to real environment a number of conditions should be satisfied.

### Changing of payment page address

To make payment for the order it is necessary to generate the form with special fields and redirect the buyer to the payment page using POST method and payment page address <https://payment.webpay.by>.

Addresses of payment pages:

Type of payment environment	Address
Test	<a href="https://secur sandbox.webpay.by">https://secur sandbox.webpay.by</a>
Real	<a href="https://payment.webpay.by">https://payment.webpay.by</a>

### Changing of WebPay™ store identifier

WebPay™ store identifier value is transferred using field **wsb\_storeid**. This identifier is created during registration in WebPay™ and is sent in the letter. It is necessary to replace this parameter with the value using in the real environment.

### Changing the value of wsb\_test

When you made test payments in the test environment, field **wsb\_test** were used with the value **1** to indicate using test environment. To make payment in the real environment it is necessary to replace the value of this field with **0**.

## Support

If you have any questions about or during the development of payment module using this Guide, please feel free to contact us using e-mail address: [support@webpay.by](mailto:support@webpay.by). We recommend to specify in e-mail your company name, store identifier and screenshots of mistakes or problems.