



SMS-payments and services

sms:bank

information leaflet



<http://smscoin.net>



sms:bank

Table of contents

- 1. Service description**
- 2. Functionality**
- 3. Installation and configuration**
- 4. Technical description**
- 5. Implementation details**
 1. Form that initiates the transaction
 2. Interface language setup
 3. Payment status pages
 4. Payment result handler
- 6. Samples and source code**
- 7. FAQ**
- 8. Contacts**



sms:bank

Service description

sms:bank is a mechanism based on payment gateway operation principle. This service enables you to create a totally automatized service combining interaction with visitors according to well-known payment systems. One-time effort on your side to install the service makes you free from constant watch over each and every user updating their virtual accounts on your website. As for users, they will use a much more convenient balance replenishment mechanism when you can write any amount off the account for whatever service there is you offer on your website. The same service can be used for one-time payment as well, and in this case you enable user to choose the amount of the transfer.

How it works

sms:bank mechanism is based on a concept of any payment gateway and composed of the same parts. A form is stored on a website and following its activation your user launches the transaction and is directed to payment page stored on SmsCoin server. This page displays payment details and instructions for its implementation. In case user admits all the data, he sends an SMS to a specific short code and confirms the payment on the same page. Besides, user is able to cancel the payment. The moment the process is finished, user is redirected back to your website where transaction results page is generated.

Installation and set up process

In case your service was already integrated with one of the payment gateways, it won't be too hard to add **sms:bank** to general system. Any way, **sms:bank** installation does require good knowledge of PHP or any other script language which performs tasks on server side. If you have the necessary experience, the implementation process will take about half an hour to two hours, depending on the scope of work that has to be performed. Detailed instructions for installation and set up process of **sms:bank** service can be found on [technical description](#) page.



SMS-payments and services

Technical info

sms:bank service is implemented like a transaction gateway; the end user initiates transaction by submitting the form found on your site, and afterwards makes the payment itself by sending an SMS with certain text to the given number. In this kind of implementation there are at least three key-points:

- Form that initiates the transaction;
- Payment status pages (Success URL and Fail URL, to which messages informing of successful operation or error accordingly; must be provided during the setup);
- Payment result processor (Result URL, which is used for informing of fulfilled operations; must be given during the setup).

Please note that for sms:bank installation process you will need to use one of the programming languages, for ex. PHP.



Implementation details

Form that initiates the transaction

An example of HTML-markup form that gives the user an opportunity to initiate the transaction is given below.

```
<form action="http://bank.smscoin.com/bank/" method="POST">
<input name="s_purse" type="hidden" value="1" />
<input name="s_order_id" type="hidden" value="1234" />
<input name="s_amount" type="hidden" value="0.1" />
<input name="s_clear_amount" type="hidden" value="0" />
<input name="s_description" type="hidden" value="Demo
payment" />
<input name="s_sign" type="hidden"
value="0123456789abcdef0123456789abcdef" />
<input type="submit" value="Pay" />
</form>
```



SMS-payments and services

Your implementation might be quite different from the one shown below. You can add your own fields as well. In any case the server must accept the following parameters:

Parameter	Type	Description
<code>s_purse</code>	int	Your sms:bank ID in the system
<code>s_order_id</code>	int	Operation id (any number; used primarily in order to compare initiated transaction with its' result)
<code>s_amount</code>	float	The payment amount, USD
<code>s_clear_amount</code>	int(1)	End user fee calculation algorithm. If 0, then <code>s_amount</code> is the end user price, otherwise the <code>s_amount</code> is your profit from the operation; the price user pays is varied due to rate issues in different countries.
<code>s_description</code>	char(127)	Operation description (random line, not longer then 127 characters, serves as a convenient method for distinguishing target payments). Processed as UTF-8.
<code>s_sign</code>	char(32)	MD5 hash lines conjoined through double (":") <code>s_purse</code> , <code>s_order_id</code> , <code>s_amount</code> , <code>s_clear_amount</code> , <code>s_description</code> and <code>secret_code</code> parameters (in that order), where <code>secret_code</code> is your sms:bank secret code.

If the value of `s_amount` amount does exceed the rate available in one country or another, such country is automatically excluded from the list of the countries served by your sms:bank.



SMS-payments and services

Interface language setup

Russian is set as default language in all our services. In case part of your users is non-Russian speakers, or for some other reasons you have decided to change the default language, simply replace the following address

```
http://bank.smscoin.com/bank/
```

with

```
http://bank.smscoin.com/language/english/bank/
```

Instead of `english` you can specify any language supported by the system.



SMS-payments and services

Payment status pages

Payment status pages should accept the following parameters from the server:

Parameter	Type	Description
<code>s_purse</code>	int	Your sms:bank id in the system.
<code>s_order_id</code>	int	Operation id (any number; used primarily in order to compare initiated transaction with its' result).
<code>s_amount</code>	float	The exact payment amount, with account of processing algorithm (see below), USD.
<code>s_clear_amount</code>	int(1)	End user fee calculation algorithm. If 0, then <code>s_amount</code> is the end user price, otherwise the <code>s_amount</code> is your profit from the operation; the price user pays is varied due to tarification issues in different countries.
<code>s_status</code>	int(1)	The payment status: 1 - success, 0 - fail.
<code>s_sign</code>	char(32)	MD5 hash lines conjoined through double (":") <code>secret_code</code> , <code>s_purse</code> , <code>s_order_id</code> , <code>s_amount</code> , <code>s_clear_amount</code> and <code>s_status</code> parameters (in that order), where <code>secret_code</code> is your sms:bank secret code.



SMS-payments and services

Payment result handler

Payment result processor should accept the following parameters:

Parameter	Type	Description
<code>s_purse</code>	int	Your sms:bank ID.
<code>s_order_id</code>	int	Operation ID (any number; used primarily in order to compare initiated transaction with its' result).
<code>s_amount</code>	float	The exact payment amount, with fee calculations applied (see below), USD.
<code>s_clear_amount</code>	int(1)	End user fee calculation algorithm. If 0, then <code>s_amount</code> is the end user price, otherwise the <code>s_amount</code> is your profit from the operation; the price user pays is varied due to rate issues in different countries.
<code>s_inv</code>	int	Order number of an operation.
<code>s_phone</code>	char(32)	Phone number of a subscriber in question.
<code>s_sign_v2</code>	char(32)	MD5 hash lines conjoined through double (":") <code>secret_code</code> , <code>s_purse</code> , <code>s_order_id</code> , <code>s_amount</code> , <code>s_clear_amount</code> , <code>s_inv</code> and <code>s_phone</code> parameters (in that order), where <code>secret_code</code> is the secret code of your sms:bank service.



SMS-payments and services

Samples and source code

You can find an example of the service [here](#).

Download source code in [PHP](#), [Perl](#), [ASP.NET](#), [Python](#).

Code example for rate scale in XML and JSON format can be downloaded [here](#).

Please note: code samples are purely illustrative, therefore their use in production environments without substantial revision is highly unadvised.



SMS-payments and services

FAQ

When I click the address specified in service settings, I receive an error message.

You do not need to click the address specified in service settings; instead, this address must be specified in parameter of the action form which initiates the transaction. Code which generates the form is stored in code examples (see above).

There is no single country in the list on the payment page.

Please make sure that the cost entered corresponds the rates available in the countries which you've chosen (SMS cost cannot be higher then maximum rate available in each and every country).

How can I add additional fields to transaction initialization form?

This option is already integrated. Fields shown as an example in transaction initialization form are necessary. You can add your own fields to the form, which will be transferred to your server in reply. If your order number contains alphabetical characters or exceeds *int*, please use additional field to transfer it. The total limit of all pairs "field-value" consists of 155 characters.

I'm in the process of connecting sms:bank, and I don't understand which addresses are to be entered in Success URL and Result URL fields.

Fail URL is an address on your website user will get to from our gateway in case of payment failure;

Success URL is an address on your website user will get to from our gateway following successful payment process;

Result URL is an address of the handler on your website, where the parameters of each SMS-message received for your sms:bank are transferred from our server. Unlike Fail URL and Success URL, user do not reach this address, and script should be placed here.



SMS-payments and services

Contacts

Tech support:

ICQ: **429853174**

E-mail: **support@smscoin.com**

Sales and partnership:

ICQ: **570342641**

E-mail: **sales@smscoin.com**

Phones:

in Moscow:	+7 (499) 5044449
in St. Petersburg:	+7 (812) 3090048
in Israel:	(+972) 49111010
in Lithuania:	(+370) 52111429
in Latvia, Riga:	(+371) 67881520
in Ukraine, Kiev:	(+380) 445945742
in United Kingdom, London:	(+44) 2033550074
in Canada, Toronto:	(+1) 4168001115
in USA, New York:	(+1) 7187667853
in Cyprus, Limassol:	(+357) 25030230
in Greece, Athens:	(+30) 2111981540

Fax: (+972) 48408497

Office: Israel, Haifa
Yochanan HaSandlar St 11

Postal address:

P.O. Box 121
Tirat Carmel, 39100
Israel

<http://smscoin.net>