Mistertango Payment Service API v1.0

Quick Start

Just add this code:

```
<script src="https://payment.mistertango.com/resources/scripts/mt.collect.js?v=01"
type="text/javascript"></script>
<script type="text/javascript">
    //Main loader (mandatory)
    document.addEventListener('DOMContentLoaded', function() {mrTangoCollect.load();}, false);

    //Set recipient (must be registered as Mistertango merchant) (mandatory)
    mrTangoCollect.set.recipient('recipient@merchant.com');

    mrTangoCollect.set.lang('en');
    mrTangoCollect.set.payer('buyer@buyer_email.com');
</script>
<button onclick="mrTangoCollect.submit('1.52', 'EUR', 'Order 1123'); return false;">Pay 1.52 EUR</button>
```

API Reference

Methods

Method	**Short description**
`mrTangoCollect.load()`	Main loader. Mandatory
`mrTangoCollect.set.recipient(recipient)`	Email address of recipient. Mandatory
`mrTangoCollect.set.payer(payer)`	Email address of payer (buyer). Mandatory
`mrTangoCollect.set.lang(lang)`	Language code. ISO-639-1. E.g.: `mrTangoCollect.set.lang('en')` Available language codes: "lt", "en", "lv", "et", "ru", "fi", "fr", "nl", "it", "es", "uk", "hu", "ro", "bg", "cs", "sk", "de"
`mrTangoCollect.set.amount(amount)`	Full amount. Float or string. E.g.: '12345.67' - OK 12345.67 - OK '12,345.67' - wrong
`mrTangoCollect.set.currency(currency)`	Only 'EUR' is available.
`mrTangoCollect.set.description(description)`	Order identification code or other short description by which you can identify the order. Don't use special symbols like @#$-_`~ (max length: 200 symbols)

`mrTangoCollect.set.payment_type_forced (payment_type)`	Opens widget with specific payment type. Set of values: ‘bank_link’, ‘bank_transfer’, ‘bitcoin’, ‘credit_card’.
`mrTangoCollect.set.payment_type_forced_params(params)`	Payment-specific parameters in JSON format

`mrTangoCollect.submit(amount, currency, description)`	All params are optional if you use the setters described above.
	
	`amount - see mrTangoCollect.set.amount(amount)`
	`currency - see mrTangoCollect.set.currency(currency)`
	`description - see mrTangoCollect.set.description(description)`

`mrTangoCollect.custom={"market":"LV"};`	Change market from default to custom: available markets: “LT”, “LV”, “EE”, “FI”, "FR", "NL", "IT", "ES", "SK", "DE". Also available: "UA", "HU", "RO", "BG", "CZ" but please note we don’t accept any currency than EUR.
`mrTangoCollect.custom={"callback":"<encrypted callback url>"};`	Change default callback url. Callback url should be encrypted (**Important:** do it in server side only!). Encryption function - click here

Real Time Push Notifications (websocket events)

By default all events do nothing. You can override all them.
Important! Use websocket event for UI only. For decision to update basket status (paid, not paid etc) use server-side POST notifications.

E.g.

```
mrTangoCollect.onSuccess = function(response){
    console.log('Success response');
    console.log(response);

    $('#pay-button').text('Paid!');
    $('#pay-button').prop('disabled', true);
};
```

Event Types

Event	**Short description**
`mrTangoCollect.onOpened(response)`	Fire when widget is successful opened
`mrTangoCollect.onClosed(response)`	Fire when widget is successful closed
`mrTangoCollect.onOffLinePayment(response)`	Fire when customer chose to pay via bank transfer (wired payment). It's mean nothing - the customer can change the decision!.
`mrTangoCollect.onSuccess(response)`	Fire when payment is successful.

Event Responses

```
{
    "type": "WINDOW_STATUS",
    "status": "OPENED",
```

```
    "order": {
        "invoice": "a2151cdb-5d18-11e5-aab7-0203788e2242",
        "ws_id": "ln5v2DLxzHDNLGO6ZC98",
        "amount": "1.52",
        "currency": "EUR",
        "description": "Order 1123"
    },
    "payment": [],
    "hash":
"AtB4zbTlSmjXheVRpxKRCy39xqtgjyTVRz4kf6n8C7q6u3UvL8nYCojChjR8j7RMdlIyZTluz0kddsHG77hajQwvitemWu+LpDgZ0NW5NP
POE054c6phydmf511BHUgpdry2Cda8ac4egXjGMvuWqwaa+Qkqu0phzIpLJOStBTHZNhvdjQJgZZLf92FmPowZLgwt7yFe8y++ffezd5nko
U1BNDFHyf0hCiNWAK0USlONapLNGmyZ5k7FE7+y4hd7scTkDTSkIq28CdGURMNgXBQu7YaJa6SLzdtpLd3S6GM="
}
```


Response field	**Short description**
`type`	Response message type. (WINDOW_STATUS | PAYMENT)
`status`	Mistertango widget status: ● OPENED - widget was opened (return with type WINDOW_STATUS only) ● CLOSED - widget was closed (return with type WINDOW_STATUS only) ● OFFLINE - Client chose to pay via bank transfer (not paid!) (return with type PAYMENT only) ● PAID - Payment successful (return with type PAYMENT only)
`order.invoice`	Current order’s invoice ID
`order.ws_id`	Websocket ID

`order.amount`	Order's sum amount (see `mrTangoCollect.set.amount(amount)`)
`order.currency`	Order's currency (see `mrTangoCollect.set.currency(currency)`)
`order.description`	Order's description (see `mrTangoCollect.set.description(description)`)
`payment.type`	Return only when: ● type = PAYMENT ● status = PAID Available values: ● MISTERTANGO - paid via Mistertango wallet ● BITCOIN - paid with Bitcoin e-currency ● BANK_LINK - paid via fast payment ● CREDIT_CARD - paid with credit card ● BANK_TRANSFER - paid via bank transfer (wired payment)
`payment.amount`	Return only when: ● type = PAYMENT ● status = PAID Customer's payment amount in EUR
`payment.currency`	Return only when: ● type = PAYMENT ● status = PAID Customer's payment currency (EUR)
`contact.email`	Return only when: ● type = PAYMENT

	Customer’s email address
`contact.contact_details`	Return only when: ● type = PAYMENT Contact details for advance merchant. Skip
`contact.shipping_details`	Return only when: ● type = PAYMENT Shipping details for advance merchant. Skip
`hash`	Event’s signature. See Server-side post notifications (signature)

Server-side Push Notifications (POST callback)

When payment is successful Mistertango send server-side callback. Before doing any decision you must:

- Check `callback_uuid` is not duplicated. We always send the same UUID for the same callback.
- Check signature (`hash`). If you can’t decrypt - skip this callback (maybe it’s fake callback)
- Check amount (`custom.data.amount`) - merchant must to do decision what to do if received amount is not equal of order amount. Be careful!

If callback is successful proceed - return “OK” (without quote)

Message’s Structure

E.g.

Header

```
Array
(
    [callback_uuid] => cf0a34a7-4bea-11e5-aab7-0203788e2242
    [order_type] =>
    [details] =>
    [order_uuid] =>
    [amount] =>
    [currency] =>
    [uid] =>
    [status] =>
    [custom] =>
{"invoice":"a57b7953-4bea-11e5-aab7-0203788e2242","status":"paid","data":{"amount":"25.23","currency":"EUR"
,"description":"MTX1440590730 Order 1123","paid_partly":false},"type":"BANK_LINK","description":"Order
1123","contact":{"email":"buyer@buyer_email.com","contact_details":[],"shipping_details":[]}}
    [hash] => HUsCut6v31IxAmUlI6YEqyI156oAgVqZJaVr3fMAdd7t45Y3mtWx8JnIPhJI4q...xdrcNdnJAL7p4BZ75jTFnxLKZ
)
```

Body (Json decoded custom field (from header))

```
{
    "invoice": "a57b7953-4bea-11e5-aab7-0203788e2242",
    "status": "paid",
    "data": {
        "amount": "25.23",
        "currency": "EUR",
        "description": "MTX1440590730 Order 1123",
        "paid_partly": false
    },
    "type": "BANK_LINK",
```

```
    "description": "Order 1123",
    "contact": {
        "email": "buyer@buyer_email.com",
        "contact_details": [],
        "shipping_details": []
    }
}
```

Header	
Field	**Short description**
`callback_uuid`	Every callback has unique UUID. If you got the same UUID - just return success message (“OK”)
`order_type`	Reserved for Mistertango wallet callback. Skip
`details`	Reserved for Mistertango wallet callback. Skip
`order_uuid`	Reserved for Mistertango wallet callback. Skip
`amount`	Reserved for Mistertango wallet callback. Skip
`currency`	Reserved for Mistertango wallet callback. Skip
`uid`	Reserved for Mistertango wallet callback. Skip
`status`	Reserved for Mistertango wallet callback. Skip
`custom`	The main callback body. JSON encoded.

`hash`	Encrypted all information described above.
Body (decoded custom field)	
Field	**Short description**
`invoice`	Order invoice ID
`status`	Always “Paid”
`data.amount`	Customer’s payment amount in EUR. Check this amount with order amount!
`data.currency`	Payment currency. “EUR”
`data.description`	Description received from bank and other payment providers. This description is not order description!
`data.paid_partly`	Flag if order was paid partly. Default: false
`data.status`	Use this status only when you are enabled double callbacks (ask Mistertango support to enable it, if you need) Incoming payment status: ● UNCONFIRMED - Payment initiated, but still a not received (there is a risk that the transaction may be canceled if the payment will not be received). ● CONFIRMED - Payment is received.
`type`	Payment type: ● MISTERTANGO - paid via Mistertango wallet ● BITCOIN - paid with Bitcoin e-currency ● BANK_LINK - paid via fast payment ● CREDIT_CARD - paid with credit card ● BANK_TRANSFER - paid via bank transfer (wired payment)

`description`	Order's description (see `mrTangoCollect.set.description(description)`)
`contact.email`	Email address of payer (buyer) (see `mrTangoCollect.set.payer(payer)`)
`contact.contact_details`	Contact details for advance merchant. Skip
`contact.shipping_details`	Shipping details for advance merchant. Skip

Message's Signature

All messages has a field `hash` with encrypted information. Just decrypt this hash and use decrypted data in your system.
All information are encrypted with `MCRYPT_RIJNDAEL_128 CBC MODE`.

PHP decryption function

```php
public function decrypt($encoded_text, $key)
{
  $key = str_pad($key, 32, "\0");

  $encoded_text = trim( $encoded_text );
  $ciphertext_dec = base64_decode($encoded_text);

  $iv_size = mcrypt_get_iv_size(MCRYPT_RIJNDAEL_128, MCRYPT_MODE_CBC);

  # retrieves the IV, iv_size should be created using mcrypt_get_iv_size()
  $iv_dec = substr($ciphertext_dec, 0, $iv_size);
```

```
  # retrieves the cipher text (everything except the $iv_size in the front)
  $ciphertext_dec = substr($ciphertext_dec, $iv_size);

  # may remove 00h valued characters from end of plain text
  $sResult = @mcrypt_decrypt(MCRYPT_RIJNDAEL_128, $key, $ciphertext_dec, MCRYPT_MODE_CBC, $iv_dec);
  return trim( $sResult );
}
```

NodeJS decryption method

Please check https://github.com/insidewarehouse/mistertango or https://www.npmjs.com/package/mistertango

C#/.NET decryption method

Please check: https://github.com/mistertango-dev/payment-service-utility.net/blob/master/mt_decrypt.cs

UNCONFIRMED and CONFIRMED callback status

By default Mistertango sends only ‘UNCONFIRMED’ as the status in the callback. This means, that the payment was initiated but has not yet been received. There is a risk that the transaction may be canceled if the payment will not be received.

Is possible to enable to send:

- UNCONFIRMED only callbacks (default. Instant callback)
- CONFIRMED only callbacks (you never receive unconfirmed callbacks. May be send with high delay)
- Both above. Merchant should support double callbacks

Ask to Mistertango support if you need to change default configuration.

Callback PHP example

```php
$aCallbackData = $_POST;
$aCallbackHeader = @json_decode(decrypt($aCallbackData['hash'], 'SecretKey'), true);

if (empty($aCallbackData))
   die();

if (isDuplicate($aCallbackHeader['callback_uuid']))
   die('OK');

$aCallbackBody = @json_decode($aCallbackHeader['custom'], true);

if (!empty($aCallbackBody))
{
   //TODO: find your order $aCallbackBody['description']
   //TODO: check amount $aCallbackBody['data']['amount']
   //TODO: change your order status

   //Callback processed
   die('OK');
}
```

Other

Helpers

PHP encryption function

```
function encrypt($plain_text, $key)
{
  $key = str_pad($key, 32, "\0");

  $plain_text = trim( $plain_text );
  # create a random IV to use with CBC encoding
  $iv_size = mcrypt_get_iv_size(MCRYPT_RIJNDAEL_128, MCRYPT_MODE_CBC);
  $iv = mcrypt_create_iv($iv_size, MCRYPT_RAND);

  # creates a cipher text compatible with AES (Rijndael block size = 128)
  # to keep the text confidential
  # only suitable for encoded input that never ends with value 00h (because of default zero padding)
  $ciphertext = mcrypt_encrypt(MCRYPT_RIJNDAEL_128, $key,
     $plain_text, MCRYPT_MODE_CBC, $iv);

  # prepend the IV for it to be available for decryption
  $ciphertext = $iv . $ciphertext;

  # encode the resulting cipher text so it can be represented by a string
  $sResult = base64_encode($ciphertext);
  return trim( $sResult );
}
```

Server-side Push Notifications (POST callback) test utility

Download: https://github.com/mistertango-dev/callback-test-utility

Test Mode

To enable “test mode" you should go to Mistertango collection payment settings and choose one of “Test Mode” options:

- Disabled - “test mode” is disabled (default)
- Enabled - “test mode” always is enabled. Don’t use in production environment
- Manual - “test mode” by default is disabled, but you can to provide param to enable it.

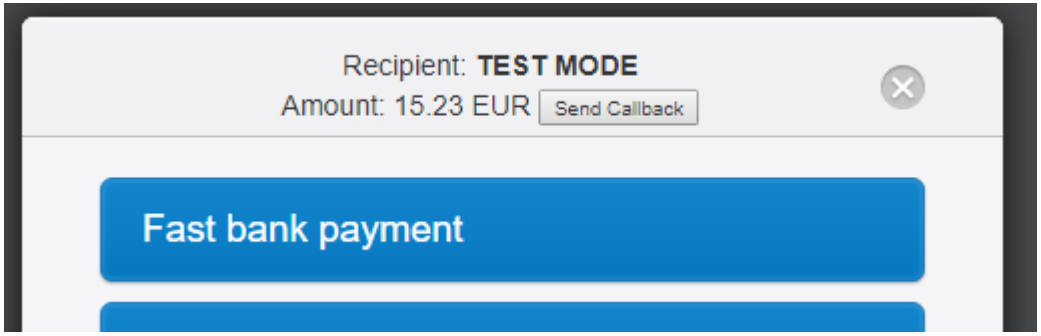
How to enable manual “test_mode”

```
mrTangoCollect.custom={"add_conf":"<encrypted data>"};
```

`<encrypted data>` - should be encrypted json `{"test_mode":true}`

Important: for security reason, please encrypt all data in server side only!. Encryption function - click here

How to use



When “Test mode” is enabled you should see “Recipient: TEST MODE” and “Send Callback” button (please check printscreen).
Just choose payment method (e.g. Credit Card) and click “Send Callback”. The widget will show that the payment has been received (without making a real payment) and will send a callback to your specified callback url address.