



MisterPay Specification

Release 1.36

Last Updated: October 21, 2019
Creation Date: May 04, 2018

Table of contents

Introduction	6
Change Control	7
1. Communicating with MisterPay	8
1.1. MisterPay URLs	8
2. MisterPay Test Environment	9
2.1. Connecting to the MisterPay Test Environment	10
2.2. Testing credentials (User ID, Secret KEY, Processing ID and card numbers).	10
3. General Examples	11
3.1. SALE transaction	11
3.1.1. SALE transaction (IFRAME)	11
3.1.1.1. Request URL	11
3.1.1.2. Initial transaction	11
3.1.1.4. Request parameters.	11
3.1.1.5. Products array parameters.	13
3.1.1.6. Response parameters.	13
3.1.1.7. Redirect customer after payment.	14
3.1.1.8. Callback parameters.	14
3.1.1.9. Signature generation.	16
3.1.1.10. Transaction example	17
3.1.1.11. Callback example	19
3.1.2. SALE transaction (DIRECT for PCI DSS)	22
3.1.2.1. Request URL	22
3.1.2.2. Request parameters	22
3.1.2.3. Transaction example	23
3.1.2.4. Callback example	25
3.1.3. SALE transaction (REDIRECT to secure page)	27
3.1.3.1. Request URL	27
3.1.3.2. Request parameters	27
3.1.3.3. Transaction example	27
3.1.3.4. Callback example	28
3.1.4. SALE transaction (TOKEN)	29
3.1.4.1. Request URL	29
3.1.4.2. Request parameters	29
3.1.4.3. Requesting a token	29
3.1.4.4. Callback with token value	30
3.1.4.5. Request parameters	32
3.1.4.6. Sale transaction with token example	32

3.1.4.7. Callback example	34
3.1.5. SALE transaction (Create INVOICE)	35
3.1.5.1. Request URL	35
3.1.5.2 Create INVOICE request	35
3.1.5.3. Request parameters.	35
3.1.5.4. Products array parameters.	36
3.1.5.5. Response parameters.	36
3.1.5.6. Transaction example	36
3.1.5.7. Callback example	38
3.1.6. SALE transaction (Update INVOICE)	39
3.1.6.1. Request URL	39
3.1.6.2 Update INVOICE request	39
3.1.6.3. Request parameters.	39
3.1.6.4. Products array parameters.	40
3.1.6.5. Response parameters.	40
3.1.6.7. Signature generation.	40
3.1.6.8. Transaction example	41
3.1.6.9. Callback example	42
3.2. VOID transaction	42
3.2.1. Request URL	42
3.2.2. Request parameters	42
3.2.3. Transaction example	42
3.2.4. Response parameters	43
3.2.5. Response example	43
3.3. REFUND transaction	44
3.3.1. Request URL	44
3.3.2. Request parameters	44
3.3.3. Transaction example	44
3.3.4. Response parameters	44
3.3.5. Response example	45
3.4. STATE transaction	46
3.4.1. Request URL	46
3.4.2. Request parameters	46
3.4.3. Transaction example	46
3.4.4. Response parameters	46
3.4.5. Response example	47
3.5. AUTHORIZATION transaction (HOLD)	48
3.5.1. Request URL	48

3.5.2. Request parameters	48
3.5.3. Transaction example	48
3.5.4. Callback example	49
3.6. CAPTURE transaction (HOLD confirmation)	52
3.6.1. Request URL	52
3.6.2. Request parameters	52
3.6.3. Transaction example	52
3.6.4. Response parameters	52
3.6.5. Response example	53
3.7. A2C transaction (account to card)	54
3.7.1. Request URL	54
3.7.2. Request parameters	54
3.7.3. Initial transaction example	54
3.7.4. A2C request parameters	56
3.7.5. A2C transaction with consumer card number	56
3.7.6. A2C transaction with token instead card number	57
3.7.6.1. Sale request example to issue card token	57
3.7.6.2. Callback with token value	58
3.7.6.4. Initial A2C request to get transactionID	59
3.7.6.5. A2C transaction example with token	61
3.7.7. Callback example	61
3.8. Account balance	64
3.8.1. Request URL	64
3.8.2. Request example	64

THIS PAGE LEFT BLANK INTENTIONALLY



Introduction

The MisterPay interface provides a flexible and easy to use interface for processing transactions with the MisterPay system. Through one interface, a merchant can perform credit card, debit card, ACH, electronic money, self-service cash terminal and mobile transactions across multiple industries and countries.

This document is for merchants or third-party developers who will be generating their payment requests to MisterPay. This document does assume that the reader is familiar with JSON generation and parsing, using the https protocol, and the rudiments of credit card processing.

You will be assigned a ProcessingID, MisterPay key (or UserID) and Secret key that you will use for testing and in production. You will also be required to supply an Order, Customer and Cart details for each transaction. Please verify that each transaction is sent with the all of the necessary data elements required by the card issuer. Response codes should be reviewed to verify that your system interprets and handles the response codes correctly.

Note:

All XML examples are purely EXAMPLES ONLY.
Please DO NOT copy examples for coding purposes.

Disclaimer: The information in this guide is current as of the date of printing. However, card acceptance and processing procedures are subject to change. MisterPay is not responsible for your use of the information contained in this guide, including errors, omissions, inaccuracy or non-timeliness of any kind. MisterPay will publish updates as needed.

Any questions or problems with this document should be sent to:

- MisterPay at email address: support@misterpay.com

Change Control

Date	Version	Description
10/21/2019	1.36	Sale token request updated ForceAuthorization option added
07/29/2019	1.35	A2C account balance request added
07/03/2019	1.34	A2C transfer added
06/25/2019	1.33	CardToken payment option added
05/17/2019	1.32	AllowedProcessings array added to Sale/Redirect, Create/Invoice and Update/Invoice requests
05/09/2019	1.31	Create/Update invoice requests updated
04/24/2019	1.30	Create & Update INVOICE request added
04/15/2019	1.22	Redirect Sale request added
03/03/2019	1.21	OCT request updated
30/01/2019	1.20	Authorization request added
12/12/2018	1.10	OCT requests added
05/04/2018	1.00	Sale request example updated

1. Communicating with MisterPay

The JSON is sent to MisterPay inside of an HTTPS POST message. Each post can contain only one message. The JSON itself must contain only valid UTF-8 characters. Transactions that use the non-ASCII portions of character sets such as ISO-8859-1, ISO-8859-15 or Windows-1252 will be rejected. For best results, it is strongly recommended that merchants restrict themselves to using only the UTF-8-character set.

Our test server is signed with a DomainSSL certificate; DomainSSL has been part of every major vendor's SSL certificate pack for several years. For example, it has been part of Microsoft's trusted root CA's since XP Service pack 3.

To be able to resolve the certificate chain for our server certificate, our certificate is provided by COMODO CA Limited. We use the COMODO RSA Domain Validation Secure Server CA. If you navigate to <https://api.misterpay.com> via your web browser, the certificate details should be available there as well.

1.1. MisterPay URLs

The Internet connection to MisterPay can be found at the URL listed below.

[https://api.misterpay.com /api/payment](https://api.misterpay.com/api/payment)

Stress testing is not permitted over the Internet connections without the specific consent of MisterPay Corp. Any entity discovered to be stress testing over the Internet without consent of MisterPay Corp. will be categorized as a Denial of Service attacker and be prohibited from using the MisterPay system without notice.

These URL's are subject to change, please contact your account representative to get the current URL's before starting testing.

2. MisterPay Test Environment

The MisterPay Test Environment allows developers to simulate the submission and subsequent approval (or denial) of financial transactions. Merchants can system-test their financial transaction programs, building and submitting their MisterPay transactions in a safe environment. The Test Environment operates in a non-production environment; any transaction submitted to the Test Environment does not actually enter the world of live financial transactions. Using the Test Environment, merchants can debug their custom programs without worrying about bogus test transactions being processed as live transactions.

The Test Environment uses virtually the same MisterPay software as the production environment. Use the Test Environment with the knowledge that it will operate in the same manner as the production system. The minor differences between the Test Environment and the production system are:

- The back-end simulators.
- Enhancements being introduced for test purposes prior to deployment in production.

The Test Environment is disconnected from the financial institutions. Instead of communicating with financial institutions, the MisterPay Test Environment accesses a suite of back-end simulators that send artificial responses back to MisterPay, thereby simulating the connections with Visa, MasterCard, American Express, and numerous other financial organizations. Few organizations provide any system for simulating real-time responses for financial transactions, but the MisterPay Test Environment does.

Due to ongoing development, the newest MisterPay features will always appear on the Test Environment before their deployment into production. The Test Environment serves as the test bed for the newest features. By deploying new features on the Test Environment, this also allows MisterPay to rigorously test the backward compatibility of those enhancements prior to deployment.

The Test Environment has features that allow merchants to test both approved transactions and declined transactions. Methods for configuring such tests are explained in this document.

This document lists the various features of the Test Environment. This document assumes that the reader is capable of posting a JSON transaction to MisterPay. This document concentrates on relating the testing methodology available to developers through the Test Environment.

2.1. Connecting to the MisterPay Test Environment

For XML development, testing XML-generating software, or for testing applications, the Internet connection to the MisterPay Test Environment is at the following URL:

<https://api.misterpay.com/api/payment>

MisterPay prohibits stress testing over its Internet connections without explicit prior consent. Any entity discovered to be stress testing over the Internet without consent of MisterPay will be categorized as a Denial of Service operator and barred from using MisterPay URLs without notice.

2.2. Testing credentials (User ID, Secret KEY, Processing ID and card numbers).

The MisterPay Test Environment recognizes:

- **AssetPaymentsKey (UserID)** = “03e5515f-7cd8-49ce-9284-5d78ff1390d9”
- **SecretKey** = “b6e6b617-88b0-4102-8f8f-704327fa6d9f”
- **ProcessingID** = 1

- **Successful payment**
Card number – 5235-2329-9769-9603
Expiration date – any MM/YY or MM/YYYY value
CVV – any 3- or 4-digits value

- **Failed Payment**
Card number – 5383-3226-0185-6622
Expiration date – any MM/YY or MM/YYYY value
CVV – any 3- or 4-digits value

- **Payment with RedirectGet method**
Card number – 5200-1799-6623-7744
Expiration date – any MM/YY or MM/YYYY value
CVV – any 3- or 4-digits value

- **Payment with RedirectPost method**
Card number – 5244-6163-6045-9976
Expiration date – any MM/YY or MM/YYYY value
CVV – any 3- or 4-digits value

3. General Examples

The following examples illustrate MisterPay transaction messages, showing a sample request message with a corresponding sample response message.

3.1. SALE transaction

A SALE transaction, MisterPay accepts the transaction block containing the Credit Card information, Operation mode, ProcessingID, Transaction Amount, Currency, Shopping Cart Items, Delivery and Customerdetails. Depending on PCI DSS compliance status you can choose IFRAME (payment with card details entering on the MisterPay side) or DIRECT (payment with card details entering on the merchant side) implementation of SALE transaction below.

3.1.1. SALE transaction (IFRAME)

An IFRAME implementation of SALE transaction is a 'two-steps' operation, completing both Sale initial request and Credit Card Data Retrieval Information request using only a single submission.

3.1.1.1. Request URL

<https://api.misterpay.com/api/payment/create>

3.1.1.2. Initial transaction

A SALE INITIAL transaction, MisterPay accepts the JSON transaction block containing the encrypted block of UserID, Processing ID, Transaction Amount and Currency, Shopping Cart Items, Delivery and Customerdetails and responds back to the original sender with the IFRAME FORM (a credit card data retrieval form). MisterPay accepts transaction block containing credit card information, retrieves an authorization response from the card issuer, and responds back to the original sender with that resulting authorization information.

An IFRAME FORM should be shown to the Customer to request credit card number, expiration date and CVV to complete Sale transaction.

3.1.1.4. Request parameters.

AssetPaymentsKey	A personal UserID generated after registration.	Required
TransactionType	Allowed value = 'Sale' .	Required
ProcessingId	A personal processingID generated after registration	Required
OperationMode	Allowed value = 'iframe'.	Required

MisterPay Specification

TemplateId	Preferred payment form template ID (Reserved for future use).	Optional
Amount	Purchase price to be transacted, including taxes and fees. Allowed value format = '0.00' Two (2) digits after decimal point!	Required
Currency	The currency type assumed for the amount. For example: 'USD', 'EUR', 'RUB', 'UAH' etc. Three (3) capital characters!	Required
AltAmount	User-defined field to show to the customer specific Order Amount on Credit Card Form not allowed for direct authorization (KZT, for example). The actual processing amount should be sent as Amount value. Allowed value format = '0.00' Two (2) digits after decimal point!	Optional
AltCurrency	User-defined field to show to the customer specific Currency on Credit Card Form not allowed for direct authorization (KZT, for example). The actual processing currency should be sent as Currency value. For example: 'USD', 'EUR', 'RUB', 'UAH' etc. Three (3) capital characters!	Optional
MerchantInternalOrderId	User-defined field, Customers Purchase Order number	Required
MerchantInternalUserId	User-defined field, Customers Unique Identification Number	Optional
FirstName	Customer First name	Optional
LastName	Customer Last name	Optional
Email	Customer Email address	Required
Phone	Customer Phone number	Optional
Address	Customer full Address (Number, Street name etc.)	Optional
City	Customer city	Optional
Region	Customer region name	Optional
State	Customer state name	Optional
ZIP	Customer ZIP code	Optional
Zone	Customer zone	Optional
CountryIso	Customer country. ISO-2 or ISO-3 format!	Required
ConvertText	Converts any text to Latin characters. (Reserved for future use).	Optional
DynamicDescriptor	The text that your customer sees on a bank account statement	Optional
CustomMerchantInfo	User-defined field, available for reporting purposes. MerchantCustomInfo will be added to RedirectURL.	Optional
StatusUrl	MisterPay will send the transaction result object to this page after purchase is complete. Value must start with 'http' or 'https'	Required

MisterPay Specification

ReturnUrl	MisterPay will redirect Customer back to this page after purchase is complete. Value must start with 'http' or 'https'	Required
IpAddress	Customers IP address Allowed value format = '10.10.10.10'	Optional
SendCallback	Disable callback sending to StatusURL Allowed values = true/false	Optional
CreateToken	Requests a non-sensitive equivalent of card details Allowed values = true/false	Optional
Products	Shopping cart items array. Refer Section 3.1.1.5. Products Array Definition.	Optional

3.1.1.5. Products array parameters.

It is strongly recommended to include Shopping Cart Items, Discounts or Delivery price in a Product Array separately.

ProductName	User-defined Product Name.	Required
ProductSku	A unique product identificatory, like SCU.	Optional
ImageUrl	Product related Image URL path.	Optional
ProductItemsNum	Quantity.	Required
ProductPrice	Product price. Allowed value format = '0.00' Two (2) digits after decimal point!	Required
Tax	Tax amount. Allowed fixed value = '0.00' Two (2) digits after decimal point! Allowed percentage value = '00'	Optional
TaxType	Tax type or percentage. Allowed values: fixed value = '1', percentage = '0'	Optional

3.1.1.6. Response parameters.

PaymentToken	A non-sensitive equivalent of credit card details
PaymentTokenExpireDate	Token expiration date
HtmliframeForm	A unique URL to get Credit Card Form.
RedirectUrl3DS	A unique URL to get 3DS page using RedirectGet method.
TransactionState	Transaction states: Pending – Waiting for credit card details Processing – Waiting for 3DS confirmation page Approved – Payment successful Declined – Payment failed Authorized – Payment amount was successfully blocked Refunded – Payment was refunded successfully Voided – Payment was reversed successfully

MisterPay Specification

TransactionStateCode	Transaction state codes: Pending = 0 Processing = 3 Approved = 1 Declined = 2 Authorized = 4 Refunded = 5 Voided = 6
TransactionId	MisterPay transaction number
ErrorCode	Error Code
ResponseMessage	Definition of error

3.1.1.7. Redirect customer after payment.

OrderId	Equal to MerchantInternalPaymentId
CustomMerchantInfo	Equal to MerchantCustomInfo
Result	Transaction state

Redirect Response example	
<pre>http://www.site.com/return.php? OrderId=12345&CustomMerchantInfo=Order%201bear&Result=success</pre>	

3.1.1.8. Callback parameters.

Payment {} array	
- OperationType	Transaction types: Sale – Purchase transaction Token - Purchase using token Void – Reverse payment Refund – Refund payment Authorization – Block payment amount at credit card Capture – Capture authorized payment State – Transaction state Invoice – Issue invoice Report – Request transaction report Recurrent – Recurrent payment P2P – Payment to card
- OperationTypeCode	Transaction codes: Sale = 0 Token = 1 Void = 2 Refund = 3 Authorization = 4 Capture = 5 State = 6

MisterPay Specification

	Invoice = 7 Report = 8 Recurrent = 9 P2P = 10
- TransactionId	MisterPay transactionID
- PaymentSystemId	Equal to processingID
- TransactionDate	Date and time of transaction
- TerminalId	Bank terminalID
- BankOrderId	Bank transactionID
- Eci	Electronic commerce indicator
- Rrn	Reference Retrieval Number
- Signature	The signature is used to protect the data exchanged between your server with MisterPay server
- Status	Payment status: Pending – Waiting for credit card details Processing – Waiting for 3DS confirmation page Approved – Successful payment Declined – Payment failed Authorized – Payment amount was successfully blocked Refunded – Payment was refunded successfully Voided – Payment was reversed successfully
- StatusCode	Transaction state codes: Pending = 0 Processing = 3 Approved = 1 Declined = 2 Authorized = 4 Refunded = 5 Voided = 6
- ErrorCode	Error Code
- ErrorDescription	Definition of error
- IsRedirect3DS	3DS secure transaction (true/false)
Order {} array	
- OrderId	Equal to MerchantInternalPaymentId
- CustomInfo	Equal to MerchantCustomInfo
- Amount	Payment amount
- Currency	The currency type assumed for the amount
- AltAmount	User-defined field to show to the customer specific Order Amount on Credit Card Form not allowed for direct authorization (KZT, for example)
- AltCurrency	User-defined field to show to the customer specific Currency on Credit Card Form not allowed for direct authorization (KZT, for example)
- Fee	Fee amount
- CreditAmount	Transaction amount subtract a fee
Card {} array	
- NameOnCard	Name on card
- CardBin	Card bank identification number

MisterPay Specification

- CardBrand	Credit card brand name
- CardType	Credit card type (debit, credit etc.)
- CardLevel	Credit card type (customer, business etc.)
- CardBank	Bank issuer
- CardCountry	Credit card country
- PaymentToken	A non-sensitive equivalent of credit card details
- CardMask	Masked credit card number
Customer {} array	
- CustomerId	Equal to MerchantInternalUserId
- FirstName	First name
- LastName	Last name
- Phone	Phone number
- Email	Email
Address {} array	
- Address	Delivery address
- City	City
- Zone	Zone
- Country	Country (ISO-3 format)
- Zip	Zip code
- IpAddress	Customer IP address

3.1.1.9. Signature generation.

The signature is not mandatory for IFRAME Sale transaction, but should be calculated to validate callback request. The signature should be generated using the MD5 hash the concatenated string of the parameters:

AssetPaymentsKey	A personal UserID generated after registration. Lowercase characters allowed!
SecretKEY	A personal SecretKEY generated after registration Lowercase characters allowed
TransactionId	MisterPay transactionID

Signature generation example

```
<?php
    $key = AssetPaymentsKey
    $secret = SecretKEY
    $transactionId = TransactionId

    $requestSign = $key.'.'. $transactionId.'.'.strtoupper($secret);
    $Sign = hash_hmac('md5',$requestSign,$secret);
?>
```


3.1.1.10. Transaction example

Request URL
<code>https://api.misterpay.com/api/payment/create</code>
Request
<pre>{ "ProcessingId": 1, "TemplateId": 0, "OperationMode": "Iframe", "TransactionType": "Sale", "MerchantInternalUserId": "67890", "MerchantInternalOrderId": "12345", "FirstName": "Mike", "LastName": "Wazowsky", "Phone": "+12703456677", "Email": "customer@site.com", "Address": "125, Rodeo Drive", "Zone": "NY", "City": "New York", "Region": "NY", "State": "NY", "ZIP": "12345", "CountryIso": "US", "ConvertText": false, "StatusUrl": "https://www.site.com/status", "ReturnUrl": "https://www.site.com/return", "DynamicDescriptor": "Purchase at site.com", "CustomMerchantInfo": "Order 12345", "Products": [{ "ProductSku": "SCU-12345", "ProductName": "Teddy bear", "ImageUrl": "https://www.site.com/bear.jpg", "ProductItemsNum": 1.00, "ProductPrice": 80.00, "Tax": 5.00, "TaxType": 1, }, { "ProductSku": "SCU-67890", "ProductName": "Small mouse", "ImageUrl": "https://www.site.com/mouse.jpg", "ProductItemsNum": 1.00, "ProductPrice": 10.00, "Tax": 5.00, "TaxType": 1, }] }</pre>

MisterPay Specification

```
}
],
"Amount": 100.00,
"AltAmount": 0.00,
"Currency": "USD",
"AltCurrency": "",
"AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9",
"IpAddress": "10.10.10.10"
}
```

Response with IFRAME form

```
{
  "PaymentToken": null,
  "PaymentTokenExpireDate": null,
  "HtmlIframeForm": "<script> \r\nvar head = document.getElementsByTagName('head')[0];\r\nvar s
= document.createElement('script');\r\ns.type = 'text/javascript';\r\ns.src =
'https://forms.misterpay.com/js/quick_form.js';\r\nhead.appendChild(s);\r\ns.onload = function ()
{\r\nQuickForm.initialize('03e5515f-7cd8-49ce-9284-5d78ff1390d9');\r\n}\r\n</script>\r\n<iframe
\r\nid='quick_external_payment_form'\r\nsource='https://forms.
misterpay.com/api/forms/PaymentForm/?param=MTg2ZDVkOTMtMhMWZHNjc5&fdata=lu9CzGDm
4YS%2fN5QDqhK%2fkfuTDTBs1O%2fK7ZcsCSi4wjg%3d&ldata=ehcbQCQO7Cr4BXj%2f5bySBPVegYzt
ELxT%2fJUNFTgpbI8%3d'\r\nstyle='z-index: 9999; margin-top: 60px; display: block; border: 0px none
transparent; overflow-x: hidden; overflow-y: auto; visibility: visible; margin: 0px; padding: 0px; -
webkit-tap-highlight-color: transparent; position: fixed; left: 0px; top: 0px; width: 100%; height:
100%; background: transparent;'\r\n>\r\n</iframe>",
  "RedirectUrl3DS": null,
  "TransactionState": "Pending",
  "TransactionStateCode": 0,
  "TransactionId": 3412613,
  "ErrorCode": 0,
  "ResponseMessage": null
}
```

Showing the form

```
<?php
  $externalForm = $result['htmlIframeForm'];
  echo $externalForm;
?>
```

Form example

```
<html>
<head>
  <script>
    var head = document.getElementsByTagName('head')[0];
    var s = document.createElement('script');
    s.type = 'text/javascript';
    s.src = 'https://forms.misterpay.com/js/quick_form.js';
```

```
head.appendChild(s);
s.onload = function () {
    QuickForm.initialize('03e5515f-7cd8-49ce-9284-5d78ff1390d9');
}
</script>
<script type="text/javascript" src="https://forms.misterpay.com/js/quick_form.js"></script>
</head>
<body>
<iframe id="quick_external_payment_form"
source="https://forms.misterpay.com/api/forms/PaymentForm/?param=MGYxZTJkOTctNWUyZi00NmFkLWlZyZQtY2E5ZDg0YjdlY2Y5&amp;fdata=lu9CzGDm4YS%2fN5QDqhK%2fKfuTDTBs1O%2fK7ZcsCSi4wjg%3d&amp;ldata=ehcbQCQO7Cr4BXj%2f5bySBPVegYztELxT%2fJUNFTgpbI8%3d" style="z-index: 9999; margin-top: 60px; display: block; border: 0px none transparent; overflow-x: hidden; overflow-y: auto; visibility: visible; margin: 0px; padding: 0px; -webkit-tap-highlight-color: transparent; position: fixed; left: 0px; top: 0px; width: 100%; height: 100%; background: transparent;"
src="https://forms.misterpay.com/api/forms/PaymentForm/?param=MGYxZTJkOTctNWUyZi00NmFkLWlZyZQtY2E5ZDg0YjdlY2Y5&amp;fdata=lu9CzGDm4YS%2fN5QDqhK%2fKfuTDTBs1O%2fK7ZcsCSi4wjg%3d&amp;ldata=ehcbQCQO7Cr4BXj%2f5bySBPVegYztELxT%2fJUNFTgpbI8%3d">
</iframe>
</body>
</html>
```

3.1.1.11. Callback example

The MisterPay System uses JSON POST to send transaction callback response.

Note: You may use `php://input` to read raw POST data. For example:
`$json = json_decode(file_get_contents('php://input'), true);`
`$signature = $json['Payment']['Signature'];`

Callback Response Body

```
{
  "Payment": {
    "OperationType": "Sale",
    "OperationTypeCode": 0,
    "TransactionId": 3412659,
    "PaymentSystemId": 1,
    "TransactionDate": "2019-01-26T14:19:06.1850034",
    "TerminalId": null,
    "BankOrderId": null,
    "Eci": null,
    "Rrn": null,
    "Signature": "dbe8539bc7be53eee345860662cb4be2",
    "Status": "Approved",
    "StatusCode": 1,
  }
}
```

```
"ErrorCode":0,
"ErrorDescription":null,
"IsRedirect3DS":false
},
"Order": {
  "OrderId":"12345",
  "CustomInfo":"Order 12345",
  "Amount":100.00,
  "Currency":"USD",
  "AltAmount":0.00,
  "AltCurrency":null,
  "Fee":0.00,
  "CreditAmount":100.00
},
"Card": {
  "NameOnCard":null,
  "CardBin":null,
  "CardBrand":null,
  "CardType":null,
  "CardLevel":null,
  "CardBank":null,
  "CardCountry":null,
  "PaymentToken":null,
  "CardMask":"523523*****9603"
},
"Customer": {
  "CustomerId":"67890",
  "FirstName":"Mike",
  "LastName":"Wazowsky",
  "Phone":"+12703456677",
  "Email":customer@site.com
},
"Address": {
  "Address":"125, Rodeo Drive",
  "City":"New York",
  "Zone":null,
  "Country":"US",
  "Zip":"12345"
},
"IpAddress":"185.177.242.87"
}
```

Signature check example

```
"Signature":"d8e8539bc7be53eee345860662cb4be2"
```

```
<?php
```

```
$key = '03e5515f-7cd8-49ce-9284-5d78ff1390d9';
$secret = 'b6e6b617-88b0-4102-8f8f-704327fa6d9f';
```

```
$transactionId = '3412659';  
$Signature = 'dbe8539bc7be53eee345860662cb4be2';  
  
$requestSign = $key.'.'. $transactionId.'.'.strtoupper($secret);  
$Sign = hash_hmac('md5', $requestSign, $secret);
```

```
?>
```

3.1.2. SALE transaction (DIRECT for PCI DSS)

DIRECT implementation of SALE transaction is a 'one-step' operation, completing Sale request using only a single submission. PCIDSS compliance is required!

3.1.2.1. Request URL

<https://api.misterpay.com/api/payment/sale>

3.1.2.2. Request parameters

A SALE transaction, MisterPay accepts the JSON transaction block containing the encrypted block of UserID, Processing ID, Transaction Amount and Currency, Card details, Shopping Cart Items, Delivery and Customer details and responds back to the original sender with that resulting authorization information.

AssetPaymentsKey	A personal UserID generated after registration.	Required
TransactionType	Allowed value = 'Direct' .	Required
ProcessingId	A personal processingID generated after registration	Required
OperationMode	Allowed value = 'Sale'.	Required
TemplateId	Preferred payment form template ID (Reserved for future use).	Optional
Amount	Purchase price to be transacted, including taxes and fees. Allowed value format = '0.00' Two (2) digits after decimal point!	Required
Currency	The currency type assumed for the amount. For example: 'USD', 'EUR', 'RUB', 'UAH' etc. Three (3) capital characters!	Required
AltAmount	User-defined field to show to the customer specific Order Amount on Credit Card Form not allowed for direct authorization (KZT, for example). The actual processing amount should be sent as Amount value. Allowed value format = '0.00' Two (2) digits after decimal point!	Required
AltCurrency	User-defined field to show to the customer specific Currency on Credit Card Form not allowed for direct authorization (KZT, for example). The actual processing currency should be sent as Currency value. For example: 'USD', 'EUR', 'RUB', 'UAH' etc. Three (3) capital characters!	Optional
MerchantInternalOrderId	User-defined field, Customers Purchase Order number	Required
MerchantInternalUserId	User-defined field, Customers Unique Identification Number	Optional

MisterPay Specification

FirstName	Customer First name	Optional
LastName	Customer Last name	Optional
Email	Customer Email address	Required
Phone	Customer Phone number	Optional
Address	Customer full Address (Number, Street name etc.)	Optional
City	Customer city	Optional
Region	Customer region name	Optional
State	Customer state name	Optional
ZIP	Customer ZIP code	Optional
Zone	Customer zone	Optional
CountryIso	Customer country. ISO-2 or ISO-3 format!	Required
ConvertText	Converts any text to Latin characters. (Reserved for future use).	Optional
DynamicDescriptor	The text that your customer sees on a bank account statement	Optional
CustomMerchantInfo	User-defined field, available for reporting purposes. MerchantCustomInfo will be added to RedirectURL.	Optional
StatusUrl	MisterPay will send the transaction result object to this page after purchase is complete. Value must start with 'http' or 'https'	Required
ReturnUrl	MisterPay will redirect Customer back to this page after purchase is complete. Value must start with 'http' or 'https'	Required
IpAddress	Customers IP address Allowed value format = '10.10.10.10'	Optional
SendCallback	Disable callback sending to StatusURL Allowed values = true/false	Optional
Products	Shopping cart items array. Refer Section 3.1.1.5. Products Array Definition.	Optional
CardNumber	Credit card number	Required
Cvv	The CVV code is either the four digits on the front of your American Express card, or the three digits on the back of your card for Visa, MasterCard and Discover	Required
ExpiredMonth	Expiration month Allowed values = '01-12'	Required
ExpiredYear	Expiration year Two allowed	Required
NameOnCard	Card owner first and last name	Required

3.1.2.3. Transaction example

Request URL
https://api.misterpay.com/api/payment/sale
Request

```
{
  "ProcessingId": 1,
  "TemplateId": 0,
  "OperationMode": "Direct",
  "TransactionType": "Sale",
  "MerchantInternalUserId": "67890",
  "MerchantInternalOrderId": "12345",
  "FirstName": "Mike",
  "LastName": "Wazowsky",
  "Phone": "+12703456677",
  "Email": "customer@site.com",
  "Address": "125, Rodeo Drive",
  "Zone": "NY",
  "City": "New York",
  "Region": "NY",
  "State": "NY",
  "ZIP": "12345",
  "CountryIso": "US",
  "ConvertText": false,
  "StatusUrl": "https://www.site.com/status",
  "ReturnUrl": "https://www.site.com/return",
  "DynamicDescriptor": "Purchase at site.com",
  "CustomMerchantInfo": "Order 12345",

  "CardNumber" : "5235232997699603",
  "Cvv" : "254",
  "ExpiredMonth" : "04",
  "ExpiredYear" : "20",
  "NameOnCard" : "Mike Wazovsky",

  "Products": [
    {
      "ProductSku": "SCU-12345",
      "ProductName": "Teddy bear",
      "ImageUrl": "https://www.site.com/bear.jpg",
      "ProductItemsNum": 1.00,
      "ProductPrice": 80.00,
      "Tax": 5.00,
      "TaxType": 1,
    },
    {
      "ProductSku": "SCU-67890",
      "ProductName": "Small mouse",
      "ImageUrl": "https://www.site.com/mouse.jpg",
      "ProductItemsNum": 1.00,
      "ProductPrice": 10.00,
      "Tax": 5.00,
      "TaxType": 1,
    }
  ]
}
```



```
],  
"Amount": 100.00,  
"AltAmount": 0.00,  
"Currency": "USD",  
"AltCurrency": "",  
"AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9",  
"IpAddress": "10.10.10.10"  
}
```

3.1.2.4. Callback example

The MisterPay System uses JSON POST to send transaction callback response.

Callback Response Body

```
{  
  "Payment": {  
    "OperationType": "Sale",  
    "OperationTypeCode": 0,  
    "TransactionId": 3412659,  
    "PaymentSystemId": 1,  
    "TransactionDate": "2019-01-26T14:19:06.1850034",  
    "TerminalId": null,  
    "BankOrderId": null,  
    "Eci": null,  
    "Rrn": null,  
    "Signature": "dbe8539bc7be53eee345860662cb4be2",  
    "Status": "Approved",  
    "StatusCode": 1,  
    "ErrorCode": 0,  
    "ErrorDescription": null,  
    "IsRedirect3DS": false  
  },  
  "Order": {  
    "OrderId": "12345",  
    "CustomInfo": "Order 12345",  
    "Amount": 100.00,  
    "Currency": "USD",  
    "AltAmount": 0.00,  
    "AltCurrency": null,  
    "Fee": 0.00,  
    "CreditAmount": 100.00  
  },  
  "Card": {  
    "NameOnCard": null,  
    "CardBin": null,  
    "CardBrand": null,  
  }  
}
```

```
"CardType":null,
"CardLevel":null,
"CardBank":null,
"CardCountry":null,
"PaymentToken":null,
"CardMask":"523523*****9603"
},
"Customer": {
  "CustomerId":"67890",
  "FirstName":"Mike",
  "LastName":"Wazowsky",
  "Phone":"+12703456677",
  "Email":customer@site.com
},
"Address": {
  "Address":"125, Rodeo Drive",
  "City":"New York",
  "Zone":null,
  "Country":"US",
  "Zip":"12345"
},
"IpAddress":"185.177.242.87"
}
```

Signature check example

```
"Signature":"dbe8539bc7be53eee345860662cb4be2"
```

```
<?php
```

```
$key = '03e5515f-7cd8-49ce-9284-5d78ff1390d9';
$secret = 'b6e6b617-88b0-4102-8f8f-704327fa6d9f';
$transactionId = '3412659';
$Signature = 'dbe8539bc7be53eee345860662cb4be2';
```

```
$requestSign = $key.'.'.$transactionId.'.'.strtoupper($secret);
$Sign = hash_hmac('md5',$requestSign,$secret);
```

```
?>
```

3.1.3. SALE transaction (REDIRECT to secure page)

REDIRECT implementation of SALE transaction is a 'one-step' operation, completing Sale request using only a single submission. In this mode the customer will be redirected from the merchant checkout to MisterPay secure page to complete payment.

3.1.3.1. Request URL

<https://portal.misterpay.com/checkout/pay>

3.1.3.2. Request parameters

A SALE transaction, MisterPay accepts a POST request with Base64 encoded JSON containing the MisterPay ID, Processing ID, Transaction Amount, Currency, Shopping Cart Items, Delivery and Customer details and responds back to the original sender with that resulting authorization information.

AllowedProcessings	An array of payment methods should be shown to a customers on a payment page. You can specify a list of methods and view order. Example = [1, 2, 3]	Optional
---------------------------	--	----------

Refer the section 3.1.1.4 to see request parameters.

3.1.3.3. Transaction example

Request URL
https://portal.misterpay.com/checkout/pay
HTML form & Script
<pre> <form method="POST" id="checkout" action="https://portal.misterpay.com/checkout/pay" accept-charset="utf-8"> <input type="hidden" name="data" id="data" value="" /> <input type="text" name="Email" id="Email" placeholder="Email" required /> <input type="text" name="Amount" id="Amount" placeholder="Amount" required /> <input type="text" name="Currency" id="Currency" placeholder="Currency" required /> <input type="button" value="Click me" onclick="process()"> </form> <script> function process() { function b64EncodeUnicode(str) { return btoa(encodeURIComponent(str).replace(/%[[0-9A-F]]{2}/g, </pre>

```
function toSolidBytes(match, p1) {
  return String.fromCharCode('0x' + p1);
});
}

var processingIDs = [];
var send_data = {
  'TemplateId': '19',
  'CountryISO': 'UA',
  'MerchantInternalOrderId': '12345',
  'StatusURL': 'https://site.com/status',
  'ReturnURL': 'https://site.com/return',
  'Email': document.getElementById('Email').value,
  'Amount': document.getElementById('Amount').value,
  'Currency': document.getElementById('Currency').value,
  'AssetPaymentsKey': '03e5515f-7cd8-49ce-9284-5d78ff1390d9',
  'AllowedProcessings': processingIDs,
};
var data = b64EncodeUnicode(JSON.stringify(send_data));
document.getElementById('data').value = data;
document.getElementById('checkout').submit();
}
</script>
```

3.1.3.4. Callback example

Refer the section 3.1.1.11 to see callback example.

3.1.4. SALE transaction (TOKEN)

MisterPay tokens are data that can be used in the place of card information when performing transactions. Because the tokens are not card holder information, merchants using them do not have to protect them in the same way they have to protect card holder information.

The token takes the place of the card number and the expiration date. No other card holder information is associated with the token. Tokens are valid for 12 months after the last use of the token, after which a new token must be generated. Additional charges may or may not be assessed to the merchant when tokens are used. This would be indicated in the merchant agreement.

A TOKEN Sale is a 'two-steps' operation. You have to send an initial Sale request to get transactionID and calculate signature before sending token request.

3.1.4.1. Request URL

<https://api.misterpay.com/api/payment/token>

3.1.4.2. Request parameters

The first step to using the tokens is getting a token that will map to a card number. This can be done using the CreateToken attribute to return a token as part of SALE transaction.

CreateToken	Map a token to a credit card number. Allowed values = true/false	Optional
--------------------	---	----------

3.1.4.3. Requesting a token

The first step to using the tokens is requesting a token that will map to a card number. This can be done using the CreateToken attribute to return a token as part of SALE transaction.

Request URL
https://api.misterpay.com/api/payment/create
Request
<pre>{ "ProcessingId": 1, "TemplateId": 0,</pre>

```
"OperationMode": "Iframe",
"TransactionType": "Sale",
"MerchantInternalUserId": "67890",
"MerchantInternalOrderId": "12345",
"FirstName": "Mike",
"LastName": "Wazowsky",
"Phone": "+12703456677",
"Email": "customer@site.com",
"Address": "125, Rodeo Drive",
"Zone": "NY",
"City": "New York",
"Region": "NY",
"State": "NY",
"ZIP": "12345",
"CountryIso": "US",
"ConvertText": false,
"StatusUrl": "https://www.site.com/status",
"ReturnUrl": "https://www.site.com/return",
"DynamicDescriptor": "Purchase at site.com",
"CustomMerchantInfo": "Order 12345",

" CreateToken" : true,

"Products": [
  {
    "ProductSku": "SCU-12345",
    "ProductName": "Teddy bear",
    "ImageUrl": "https://www.site.com/bear.jpg",
    "ProductItemsNum": 1.00,
    "ProductPrice": 80.00,
    "Tax": 5.00,
    "TaxType": 1,
  }
],
"Amount": 100.00,
"AltAmount": 0.00,
"Currency": "USD",
"AltCurrency": "",
"AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9",
"IpAddress": "10.10.10.10"
}
```

3.1.4.4. Callback with token value

The MisterPay System uses JSON POST to send transaction callback response.

Callback Response with token value

```
{
  "Payment": {
```

```
"OperationType":"Sale",
"OperationTypeCode":0,
"TransactionId":3412659,
"PaymentSystemId":1,
"TransactionDate":"2019-01-26T14:19:06.1850034",
"TerminalId":null,
"BankOrderId":null,
"Eci":null,
"Rrn":null,
"Signature":"dbe8539bc7be53eee345860662cb4be2",

"Status":"Approved", // or "Status":"Authorized" if ForceAuthorization option enabled

"StatusCode":1,
"ErrorCode":0,
"ErrorDescription":null,
"IsRedirect3DS":false
},
"Order": {
  "OrderId":"12345",
  "CustomInfo":"Order 12345",
  "Amount":100.00,
  "Currency":"USD",
  "AltAmount":0.00,
  "AltCurrency":null,
  "Fee":0.00,
  "CreditAmount":100.00
},
"Card": {
  "NameOnCard":null,
  "CardBin":null,
  "CardBrand":null,
  "CardType":null,
  "CardLevel":null,
  "CardBank":null,
  "CardCountry":null,

  "PaymentToken": "e34eda6b-a314-461b-b4d4-939ff24686ec",

  "CardMask":"523523*****9603"
},
"Customer": {
  "CustomerId":"67890",
  "FirstName":"Mike",
  "LastName":"Wazowsky",
  "Phone":"+12703456677",
  "Email":customer@site.com
},
"Address": {
```

```

"Address": "125, Rodeo Drive",
"City": "New York",
"Zone": null,
"Country": "US",
"Zip": "12345"
},
"IpAddress": "185.177.242.87"
}

```

Signature check example

```
"Signature": "dbe8539bc7be53eee345860662cb4be2"
```

```
<?php
```

```

$key = '03e5515f-7cd8-49ce-9284-5d78ff1390d9';
$secret = 'b6e6b617-88b0-4102-8f8f-704327fa6d9f';
$transactionId = '3412659';
$Signature = 'dbe8539bc7be53eee345860662cb4be2';

```

```

$requestSign = $key.'.'.$transactionId.'.'.strtoupper($secret);
$Sign = hash_hmac('md5', $requestSign, $secret);

```

```
?>
```

3.1.4.5. Request parameters

Parameter	Description	Required
TransactionType	Allowed value = "Token"	Required
TransactionId	A unique transaction number, generated by MisterPay server.	Required
Signature	Signature is the result of a cryptographic process that used to authenticate a sequence of data and sender.	Required
CardToken	A non-sensitive equivalent of credit card details	Required
ForceAuthorization	Reserve an order amount for future settlement. Refer the section 3.6 to capture authorized funds. Allowed values = true/false	Optional

3.1.4.6. Sale transaction with token example

Initial request URL
https://api.misterpay.com/api/payment/create
Requesting transactionID
{


```
"ProcessingId": 1,
"TemplateId": 0,
"OperationMode": "Iframe",
"TransactionType": "Token",
"MerchantInternalUserId": "67890",
"MerchantInternalOrderId": "12345",
"FirstName": "Mike",
"LastName": "Wazowsky",
"Phone": "+12703456677",
"Email": "customer@site.com",
"Address": "125, Rodeo Drive",
"Zone": "NY",
"City": "New York",
"Region": "NY",
"State": "NY",
"ZIP": "12345",
"CountryIso": "US",
"ConvertText": false,
"StatusUrl": "https://www.site.com/status",
"ReturnUrl": "https://www.site.com/return",
"DynamicDescriptor": "Purchase at site.com",
"CustomMerchantInfo": "Order 12345",
"Products": [
  {
    "ProductSku": "SCU-12345",
    "ProductName": "Teddy bear",
    "ImageUrl": "https://www.site.com/bear.jpg",
    "ProductItemsNum": 1.00,
    "ProductPrice": 80.00,
    "Tax": 5.00,
    "TaxType": 1,
  }
],
"Amount": 100.00,
"AltAmount": 0.00,
"Currency": "USD",
"AltCurrency": "",
"AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9",
"IpAddress": "10.10.10.10"
}
```

Response with transactionID

```
{
  "PaymentToken": null,
  "PaymentTokenExpireDate": null,
  "HtmlIframeForm": null,
  "RedirectUrl3DS": null,
  "TransactionState": "Pending",
}
```

<pre>"TransactionStateCode": 0, "TransactionId": '3412659', "ErrorCode": 0, "ResponseMessage": null }</pre>
Signature calculation
<pre><?php \$key = '03e5515f-7cd8-49ce-9284-5d78ff1390d9'; \$secret = 'b6e6b617-88b0-4102-8f8f-704327fa6d9f'; \$transactionId = '3412659'; \$signature = 'dbe8539bc7be53eee345860662cb4be2'; \$requestSign = \$key.'.'. \$transactionId.'.'.strtoupper(\$secret); \$sign = hash_hmac('md5', \$requestSign, \$secret); ?></pre>
Token sale request URL
<pre>https://api.misterpay.com/api/payment/token</pre>
Completing token request
<pre>{ "AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9", "TransactionId": "3412659", "CardToken": "e34eda6b-a314-461b-b4d4-939ff24686ec", "Signature": "dbe8539bc7be53eee345860662cb4be2" }</pre>

3.1.4.7. Callback example

Refer the section 3.1.4.4 to see response example.

3.1.5. SALE transaction (Create INVOICE)

An INVOICE request generates a link to secure processing page you send to collect money owed to you for services you've provided or items you've sold.

3.1.5.1. Request URL

<https://api.misterpay.com/api/merchant/invoice/create>

3.1.5.2 Create INVOICE request

A Create INVOICE request, MisterPay accepts the JSON transaction block containing the encrypted block of UserID, Template ID, Expiration date, Amount, Currency, Shopping Cart Items, Delivery and Customer details and responds back to the original sender with that resulting information.

3.1.5.3. Request parameters.

AssetPaymentsKey	A personal UserID generated after registration.	Required
TemplateId	Preferred invoice template ID	Optional
ExpirationDate	The date until invoice could be paid. Allowed format = 'YYYY-MM-DD' ('2019-10-31') or 'YYYY-MM-DDTHH:MM:SSZ' (2019-10-31T18:00:00Z)	Optional
Amount	Purchase price to be transacted, including taxes and fees. (*Please read the note below) Allowed value format = '0.00' Two (2) digits after decimal point!	Optional*
Currency	The currency type assumed for the amount. For example: 'USD', 'EUR', 'RUB', 'UAH' etc. Three (3) capital characters!	Required
MerchantInternalOrderId	User-defined field, Customers Purchase Order number	Required
CustomerNotification	Disable payment link sending to customer email Allowed values = true/false	Optional
CustomerName	Customer Name	Optional
CustomerEmail	Customer Email address	Required
CustomerPhone	Customer Phone number	Optional
CustomerAddress	Customer full Address (Number, Street name etc.)	Optional
CompanyName	Company Name	Optional
CompanyEmail	Company Email address	Optional
CompanyPhone	Company Phone number	Optional
CompanyAddress	Company full Address (Number, Street name etc.)	Optional

MisterPay Specification

StatusUrl	MisterPay will send the transaction result object to this page after payment is complete. Value must start with 'http' or 'https'	Optional
Products	Shopping cart items array. Refer Section 3.1.5.4. Products Array Definition.	Optional*
AllowedProcessings	An array of payment methods should be shown to a customer on a payment page. You can specify a list of methods and view order. Example = [1, 2, 3]	Optional

Note:*

Please, pay attention on correct invoice amount. Both variables (Amount and Products) are optional but you have to send at least one of them. In case of empty Amount value MisterPay system try to calculate prices of every product.

3.1.5.4. Products array parameters.

It is strongly recommended to include Shopping Cart Items, Discounts or Delivery price in a Product Array separately.

ProductName	User-defined Product Name.	Required
ProductSku	A unique product identificatory, like SCU.	Optional
ImageUrl	Product related Image URL path.	Optional
ProductItemsNum	Quantity.	Required
ProductPrice	Product price. Allowed value format = '0.00' Two (2) digits after decimal point!	Required
Tax	Tax amount. Allowed fixed value = '0.00' Two (2) digits after decimal point! Allowed percentage value = '00'	Optional
TaxType	Tax type or percentage. Allowed values: fixed value = '1', percentage = '0'	Optional

3.1.5.5. Response parameters.

invoicePaymentLink	A link to a secure page to accept payment
invoiceId	MisterPay invoice number
errorCode	Error Code
responseMessage	Definition of error

3.1.5.6. Transaction example

MisterPay Specification

Request URL

<https://api.misterpay.com/api/merchant/invoice/create>

Request

```
{
  "AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9",
  "TemplateId": 19,
  "ExpirationDate": "2020-05-01",
  "Currency": "USD",
  "MerchantInternalOrderId": "12345",
  "CustomerNotification": true,
  "CustomerName": "Mike Wazowsky",
  "CustomerPhone": "+12703456677",
  "CustomerEmail": "customer@site.com",
  "CustomerAddress": "125, Rodeo Drive, NY, NY, US, 11223",
  "CompanyName": "Toy store",
  "CompanyPhone": "+17583456656",
  "CompanyEmail": "shop@site.com",
  "CompanyAddress": "46, Union Sq., NY, 11225, US",
  "StatusUrl": "https://www.site.com/status",
  "AllowedProcessings": [ ],
  "Products": [
    {
      "ProductSku": "SCU-12345",
      "ProductName": "Teddy bear",
      "ImageUrl": "https://www.site.com/bear.jpg",
      "ProductItemsNum": 1.00,
      "ProductPrice": 80.00,
      "Tax": 5.00,
      "TaxType": 1,
    },
    {
      "ProductSku": "SCU-67890",
      "ProductName": "Small mouse",
      "ImageUrl": "https://www.site.com/mouse.jpg",
      "ProductItemsNum": 1.00,
      "ProductPrice": 10.00,
      "Tax": 5.00,
      "TaxType": 1,
    }
  ]
}
```

Response

```
{
  "invoicePaymentLink": "https://assetpayments.us/invoice/pay/?data=tMlBaQEI72qxVuUCFsaDbDPFK7K%2fqfbdnMDSgXji4OD9bS4Qloofhfe7BHCEazJ",
  "invoiceId": 286,
  "errorCode": 0,
  "responseMessage": null
}
```

3.1.5.7. Callback example

Refer the section 3.1.1.11 to see callback parameters.



3.1.6. SALE transaction (Update INVOICE)

A created invoice could be easily updated.

3.1.6.1. Request URL

<https://api.misterpay.com/api/merchant/invoice/update>

3.1.6.2 Update INVOICE request

An Update INVOICE request, MisterPay accepts the JSON transaction block containing the encrypted block of UserID, Template ID, Expiration date, Currency, Shopping Cart Items, Delivery and Customer details, InvoiceID and Signature and responds back to the original sender with that resulting information.

3.1.6.3. Request parameters.

AssetPaymentsKey	A personal UserID generated after registration.	Required
InvoiceId	An ID of created invoice	Required
Signature	The signature is used to protect the data exchanged between your server with MisterPay server	Required
TemplateId	Preferred invoice template ID	Optional
ExpirationDate	The date until invoice could be paid. Allowed format = 'YYYY-MM-DD' ('2019-10-31') or 'YYYY-MM-DDTHH:MM:SSZ' (2019-10-31T18:00:00Z)	Optional
Amount	Purchase price to be transacted, including taxes and fees. (*Please ready the note below) Allowed value format = '0.00' Two (2) digits after decimal point!	Optional
Currency	The currency type assumed for the amount. For example: 'USD', 'EUR', 'RUB', 'UAH' etc. Three (3) capital characters!	Required
MerchantInternalOrderId	User-defined field, Customers Purchase Order number	Required
CustomerNotification	Disable payment link sending to customer email Allowed values = true/false	Optional
CustomerName	Customer Name	Optional
CustomerEmail	Customer Email address	Required
CustomerPhone	Customer Phone number	Optional
CustomerAddress	Customer full Address (Number, Street name etc.)	Optional
CompanyName	Company Name	Optional
CompanyEmail	Company Email address	Optional
CompanyPhone	Company Phone number	Optional
CompanyAddress	Company full Address (Number, Street name etc.)	Optional

MisterPay Specification

StatusUrl	MisterPay will send the transaction result object to this page after payment is complete. Value must start with 'http' or 'https'	Optional
Products	Shopping cart items array. Refer Section 3.1.5.4. Products Array Definition.	Optional
AllowedProcessings	An array of payment methods should be shown to a customer on a payment page. You can specify a list of methods and view order. Example = [1, 2, 3]	Optional

3.1.6.4. Products array parameters.

Refer 3.1.5.4 section for Product parameters

3.1.6.5. Response parameters.

invoicePaymentLink	A link to a secure page to accept payment
invoiceId	MisterPay invoice number
errorCode	Error Code
responseMessage	Definition of error

3.1.6.7. Signature generation.

The signature should be generated using the MD5 hash the concatenated string of the parameters:

AssetPaymentsKey	A personal UserID generated after registration. Lowercase characters allowed!
SecretKEY	A personal SecretKEY generated after registration Lowercase characters allowed
InvoiceId	MisterPay invoiceID

Signature generation example
<pre><?php \$key = AssetPaymentsKey \$secret = SecretKEY \$invoiceId = TransactionId \$requestSign =\$key.'.'.\$invoiceId.'.'.strtoupper(\$secret); \$Sign = hash_hmac('md5',\$requestSign,\$secret); ?></pre>

3.1.6.8. Transaction example

Request URL
https://api.misterpay.com/api/merchant/invoice/update
Request
<pre>{ "AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9", "TemplateId": 19, "ExpirationDate": "2020-05-01", "Currency": "USD", "MerchantInternalOrderId": "12345", "CustomerNotification": true, "CustomerName": "Mike Wazowsky", "CustomerPhone": "+12703456677", "CustomerEmail": "customer@site.com", "CustomerAddress": "125, Rodeo Drive, NY, NY, US, 11223", "CompanyName": "Toy store", "CompanyPhone": "+17583456656", "CompanyEmail": "shop@site.com", "CompanyAddress": "46, Union Sq., NY, 11225, US", "StatusUrl": "https://www.site.com/status", "InvoiceId": 279, "Signature": "d6e0f9899dfabfbaaa270111df107ba4", "AllowedProcessings": [], "Products": [{ "ProductSku": "SCU-12345", "ProductName": "Teddy bear", "ImageUrl": "https://www.site.com/bear.jpg", "ProductItemsNum": 1.00, "ProductPrice": 80.00, "Tax": 5.00, "TaxType": 1, }, { "ProductSku": "SCU-67890", "ProductName": "Small mouse", "ImageUrl": "https://www.site.com/mouse.jpg", "ProductItemsNum": 1.00, "ProductPrice": 10.00, "Tax": 5.00, "TaxType": 1, }] }</pre>
Response
<pre>{</pre>

```

"invoicePaymentLink": "https://assetpayments.us/invoice/pay/?data=tMImBaQEI72qxVuUCFsaDbDP
FK7K%2fqfbdnMDSgXji4OD9bS4Qloofhfe7BHCEazJ",
"invoiceId": 279,
"errorCode": 0,
"responseMessage": null
}
    
```

3.1.6.9. Callback example

Refer the section 3.1.1.11 to see callback parameters.

3.2. VOID transaction

Note: A VOID transaction maybe declined due to bank processing rules. In this case you can use a REFUND request.

A VOID transaction is a debit or credit card purchase that is canceled after it has been authorized but before it has been settled. A void transaction does not appear on the customer's account statement, though it might appear in a list of pending transactions when the customer checks their account online.

A VOID transaction should be sent until 00:00. In other case refer section 3.3 (REFUND transaction).

3.2.1. Request URL

<https://api.misterpay.com/api/payment/void>

3.2.2. Request parameters

Amount	Void amount. Allowed value format = '0.00' Two (2) digits after decimal point!	Required
AssetPaymentsKey	A personal UserID generated after registration.	Required
TransactionId	Initial transaction number that should be voided.	Required
Signature	Signature is the result of a cryptographic process that used to authenticate a sequence of data and sender.	Required

3.2.3. Transaction example

Request URL

https://api.misterpay.com/api/payment/void

Request

```
{
  "AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9",
  "TransactionId": "3412659",
  "Amount": 100.00,
  "Signature": "dbe8539bc7be53eee345860662cb4be2"
}
```

3.2.4. Response parameters

TransactionState	Operation type. Successful result value = 'Voided'
TransactionStateCode	Reserved value.
ErrorCode	Successful result value = 0
ResponseMessage	Error description in case of failed operation

3.2.5. Response example

The MisterPay System uses JSON POST to send transaction callback response.

Response Body

```
{
  "PaymentToken": null,
  "PaymentTokenExpireDate": null,
  "HtmlIframeForm": null,
  "RedirectUrl3DS": null,
  "TransactionState": "Voided",
  "TransactionStateCode": 5,
  "TransactionId": '3412659',
  "ErrorCode": 0,
  "ResponseMessage": null
}
```

3.3. REFUND transaction

A REFUND transaction is a debit or credit card purchase that is canceled after it has been settled. A refund transaction appears on the customer's account statement. It is allowed to send partial refund amount request.

3.3.1. Request URL

<https://api.misterpay.com/api/payment/refund>

3.3.2. Request parameters

Amount	Refund amount. Allowed value format = '0.00' Two (2) digits after decimal point!	Required
AssetPaymentsKey	A personal UserID generated after registration.	Required
TransactionId	Initial transaction number that should be voided.	Required
Signature	Signature is the result of a cryptographic process that used to authenticate a sequence of data and sender.	Required

3.3.3. Transaction example

Request URL
https://api.misterpay.com/api/payment/refund
Request
<pre>{ "AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9", "TransactionId": "3412659", "Amount": 100.00, "Signature": "dbe8539bc7be53eee345860662cb4be2" }</pre>

3.3.4. Response parameters

TransactionState	Operation type. Successful result value = 'Refunded'
TransactionStateCode	Reserved value.
ErrorCode	Successful result value = 0
ResponseMessage	Error description in case of failed operation

3.3.5. Response example

The MisterPay System uses JSON POST to send transaction callback response.

Response Body

```
{
  "PaymentToken": null,
  "PaymentTokenExpireDate": null,
  "HtmlIframeForm": null,
  "RedirectUrl3DS": null,
  "TransactionState": "Refunded",
  "TransactionStateCode": 5,
  "TransactionId": '3412659',
  "ErrorCode": 0,
  "ResponseMessage": null
}
```

3.4. STATE transaction

A STATE transaction gives you information such as the error code, description and status of the transaction.

3.4.1. Request URL

<https://api.misterpay.com/api/payment/state>

3.4.2. Request parameters

AssetPaymentsKey	A personal UserID generated after registration.	Required
TransactionId	Initial transaction number that should be voided.	Required
SendCallback	Repeat callback sending to StatusURL Allowed values = true/false	Optional
Signature	Signature is the result of a cryptographic process that used to authenticate a sequence of data and sender.	Required

3.4.3. Transaction example

Request URL
https://api.misterpay.com/api/payment/state
Request
<pre>{ "AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9", "TransactionId": "3412659", "Signature": "dbe8539bc7be53eee345860662cb4be2" }</pre>

3.4.4. Response parameters

TransactionState	Transaction states: Pending – Waiting for credit card details Processing – Waiting for 3DS confirmation page Approved – Payment successful Declined – Payment failed Authorized – Payment amount was successfully blocked Refunded – Payment was refunded successfully Voided – Payment was reversed successfully
TransactionStateCode	Reserved value.

ErrorCode	Successful result value = 0
ResponseMessage	Error description in case of failed operation

3.4.5. Response example

The MisterPay System uses JSON POST to send transaction callback response.

Response body
<pre>{ "PaymentToken": null, "PaymentTokenExpireDate": null, "HtmlIframeForm": null, "RedirectUrl3DS": null, "TransactionState": "Approved", "TransactionStateCode": 1, "TransactionId": '3412659', "ErrorCode": 0, "ResponseMessage": null }</pre>

3.5. AUTHORIZATION transaction (HOLD)

An Authorization is the process of confirming whether your customer's credit card has sufficient credit to purchase goods or services from you, and also confirms whether the card is valid. Successful Authorization means that the amount you specified will be reserved from the cardholder's available credit limit for future settlement.

An Authorization is a 'two-steps' operation. You have to send an authorization request first to reserve purchase amount and specify settlement amount sending a CAPTURE request.

3.5.1. Request URL

<https://api.misterpay.com/api/payment/authorization>

3.5.2. Request parameters

The first step is authorization of a purchase amount. This can be done using the TransactionType = "Authorization" attribute.

TransactionType	Allowed value = 'Authorization'	Required
------------------------	---------------------------------	----------

3.5.3. Transaction example

Request URL
https://api.misterpay.com/api/payment/authorization
Request
<pre>{ "ProcessingId": 1, "TemplateId": 0, "OperationMode": "Iframe", "TransactionType": "Authorization", "MerchantInternalUserId": "67890", "MerchantInternalOrderId": "12345", "FirstName": "Mike", "LastName": "Wazowsky", "Phone": "+12703456677", "Email": "customer@site.com", "Address": "125, Rodeo Drive", "Zone": "NY", "City": "New York",</pre>


```
"Region": "NY",
"State": "NY",
"ZIP": "12345",
"CountryIso": "US",
"ConvertText": false,
"StatusUrl": "https://www.site.com/status",
"ReturnUrl": "https://www.site.com/return",
"DynamicDescriptor": "Purchase at site.com",
"CustomMerchantInfo": "Order 12345",
"Products": [
  {
    "ProductSku": "SCU-12345",
    "ProductName": "Teddy bear",
    "ImageUrl": "https://www.site.com/bear.jpg",
    "ProductItemsNum": 1.00,
    "ProductPrice": 80.00,
    "Tax": 5.00,
    "TaxType": 1,
  }
],
"Amount": 100.00,
"AltAmount": 0.00,
"Currency": "USD",
"AltCurrency": "",
"AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9",
"IpAddress": "10.10.10.10"
}
```

Response body

```
{
  "PaymentToken": null,
  "PaymentTokenExpireDate": null,
  "HtmlIframeForm": null,
  "RedirectUrl3DS": null,
  "TransactionState": "Authorized",
  "TransactionStateCode": 4,
  "TransactionId": '3412659',
  "ErrorCode": 0,
  "ResponseMessage": null
}
```

3.5.4. Callback example

The MisterPay System uses JSON POST to send transaction callback response.

Callback body

```
{
```

```
"Payment": {
  "OperationType": "Authorization",
  "OperationTypeCode": 0,
  "TransactionId": 3412659,
  "PaymentSystemId": 1,
  "TransactionDate": "2019-01-26T14:19:06.1850034",
  "TerminalId": null,
  "BankOrderId": null,
  "Eci": null,
  "Rrn": null,
  "Signature": "dbe8539bc7be53eee345860662cb4be2",
  "Status": "Authorized",
  "StatusCode": 1,
  "ErrorCode": 0,
  "ErrorDescription": null,
  "IsRedirect3DS": false
},
"Order": {
  "OrderId": "12345",
  "CustomInfo": "Order 12345",
  "Amount": 100.00,
  "Currency": "USD",
  "AltAmount": 0.00,
  "AltCurrency": null,
  "Fee": 0.00,
  "CreditAmount": 100.00
},
"Card": {
  "NameOnCard": null,
  "CardBin": null,
  "CardBrand": null,
  "CardType": null,
  "CardLevel": null,
  "CardBank": null,
  "CardCountry": null,
  "PaymentToken": "e34eda6b-a314-461b-b4d4-939ff24686ec",
  "CardMask": "523523*****9603"
},
"Customer": {
  "CustomerId": "67890",
  "FirstName": "Mike",
  "LastName": "Wazowsky",
  "Phone": "+12703456677",
  "Email": "customer@site.com"
},
}
```

```
"Address": {  
  "Address": "125, Rodeo Drive",  
  "City": "New York",  
  "Zone": null,  
  "Country": "US",  
  "Zip": "12345"  
},  
"IpAddress": "185.177.242.87"  
}
```

Signature check example

```
"Signature": "dbe8539bc7be53eee345860662cb4be2"
```

```
<?php
```

```
$key = '03e5515f-7cd8-49ce-9284-5d78ff1390d9';  
$secret = 'b6e6b617-88b0-4102-8f8f-704327fa6d9f';  
$transactionId = '3412659';  
$Signature = 'dbe8539bc7be53eee345860662cb4be2';
```

```
$requestSign = $key.'.'.$transactionId.'.'.strtoupper($secret);  
$Sign = hash_hmac('md5', $requestSign, $secret);
```

```
?>
```

3.6. CAPTURE transaction (HOLD confirmation)

All credit card payment providers have an authorization period to allow a charge to be made to a customer's credit card. The authorization period will end after a certain amount of time. The length of the authorization period depends on your credit card payment provider. You need to capture a payment within the authorization period to collect money for your order using a CAPTURE transaction. It is allowed to make a partial capture instead of authorized amount.

3.6.1. Request URL

<https://api.misterpay.com/api/payment/complete>

3.6.2. Request parameters

Amount	Refund amount. Allowed value format = '0.00' Two (2) digits after decimal point!	Required
AssetPaymentsKey	A personal UserID generated after registration.	Required
TransactionId	Authorized transaction number.	Required
Signature	Signature is the result of a cryptographic process that used to authenticate a sequence of data and sender.	Required

3.6.3. Transaction example

Request URL
https://api.misterpay.com/api/payment/complete
Request
<pre>{ "AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9", "TransactionId": "3412659", "Amount": 100.00, "Signature": "dbe8539bc7be53eee345860662cb4be2" }</pre>

3.6.4. Response parameters

TransactionState	Operation type. Successful result value = 'Approved'
TransactionStateCode	Reserved value.

ErrorCode	Successful result value = 0
ResponseMessage	Error description in case of failed operation

3.6.5. Response example

The MisterPay System uses JSON POST to send transaction callback response.

Response Body
<pre>{ "PaymentToken": null, "PaymentTokenExpireDate": null, "HtmlIframeForm": null, "RedirectUrl3DS": null, "TransactionState": "Approved", "TransactionStateCode": 5, "TransactionId": '3412659', "ErrorCode": 0, "ResponseMessage": null }</pre>

3.7. A2C transaction (account to card)

Account-to-person payments (A2C) is an online technology that allows customers to transfer funds from account balance to consumer card account.

3.7.1. Request URL

<https://api.misterpay.com/api/payment/p2p>

3.7.2. Request parameters

An A2C transaction is a 'two-steps' operation. You have to send an Initial request to get transactionID and calculate signature before sending A2C transfer request. This can be done using the TransactionType = "P2P" attribute.

TransactionType	Allowed value = 'P2P'	Required
------------------------	-----------------------	----------

3.7.3. Initial transaction example

Initial request URL
https://api.misterpay.com/api/payment/create
Initial request body
<pre>{ "ProcessingId": 1, "TemplateId": 0, "OperationMode": "Iframe", "TransactionType": "P2P", "MerchantInternalUserId": "67890", "MerchantInternalOrderId": "12345", "FirstName": "Mike", "LastName": "Wazowsky", "Phone": "+12703456677", "Email": "customer@site.com", "Address": "125, Rodeo Drive", "Zone": "NY", "City": "New York", "Region": "NY", "State": "NY", "ZIP": "12345", "CountryIso": "US", "ConvertText": false, "StatusUrl": "https://www.site.com/status",</pre>

```
"ReturnUrl": "https://www.site.com/return",
"DynamicDescriptor": "Purchase at site.com",
"CustomMerchantInfo": "Order 12345",
"Products": [
  {
    "ProductSku": "SCU-12345",
    "ProductName": "Teddy bear",
    "ImageUrl": "https://www.site.com/bear.jpg",
    "ProductItemsNum": 1.00,
    "ProductPrice": 80.00,
    "Tax": 5.00,
    "TaxType": 1,
  }
],
"Amount": 100.00,
"AltAmount": 0.00,
"Currency": "USD",
"AltCurrency": "",
"AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9",
"IpAddress": "10.10.10.10"
}
```

Response

```
{
  "PaymentToken": "",
  "PaymentTokenExpireDate": null,
  "HtmlIframeForm": null,
  "RedirectUrl3DS": null,
  "TransactionState": "Approved",
  "TransactionStateCode": 1,

  "TransactionId": '3412659',

  "ErrorCode": 0,
  "ResponseMessage": null
}
```

Signature calculation

```
<?php

$key = '03e5515f-7cd8-49ce-9284-5d78ff1390d9';
$secret = 'b6e6b617-88b0-4102-8f8f-704327fa6d9f';
$transactionId = '3412659';

$requestSign = $key.'.'.$transactionId.'.'.strtoupper($secret);
$signature = hash_hmac('md5',$requestSign,$secret);

?>
```

3.7.4. A2C request parameters

AssetPaymentsKey	A personal UserID generated after registration.	Required
TransactionId	Transaction number from Initial Sale request.	Required
P2PCardNumber	Receiver's card number.	Required
P2PCardToken	Credit card token. You can use it instead of a P2PCardNumber parameter.	Optional
PayFee	Add transaction fee to the transfer amount Allowed values = true/false	Optional
Signature	Signature is the result of a cryptographic process that used to authenticate a sequence of data and sender.	Required

3.7.5. A2C transaction with consumer card number

Request URL
https://api.misterpay.com/api/payment/p2p
Request
<pre>{ "AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9", "TransactionId": "3412659", "P2PCardNumber" : "1111222233334444", "PayFee" : false, "Signature": "dbe8539bc7be53eee345860662cb4be2" }</pre>
Response
<pre>{ "PaymentToken": null, "PaymentTokenExpireDate": null, "HtmlIframeForm": null, "RedirectUrl3DS": null, "TransactionState": "Approved", "TransactionStateCode": 1, "TransactionId": 3412659, "ErrorCode": 0, "ResponseMessage": null }</pre>

3.7.6. A2C transaction with token instead card number

To issue a card token you should make payment with full card details and CreateToken=true parameter. If a payment will be successful MisterPay system responds back with token value.

3.7.6.1. Sale request example to issue card token

Request URL
https://api.misterpay.com/api/payment/create
Sale request
<pre>{ "ProcessingId": 1, "TemplateId": 0, "OperationMode": "Iframe", "TransactionType": "Sale", "MerchantInternalUserId": "67890", "MerchantInternalOrderId": "12345", "FirstName": "Mike", "LastName": "Wazowsky", "Phone": "+12703456677", "Email": "customer@site.com", "Address": "125, Rodeo Drive", "Zone": "NY", "City": "New York", "Region": "NY", "State": "NY", "ZIP": "12345", "CountryIso": "US", "ConvertText": false, "StatusUrl": "https://www.site.com/status", "ReturnUrl": "https://www.site.com/return", "DynamicDescriptor": "Purchase at site.com", "CustomMerchantInfo": "Order 12345", "CreateToken" : true, "Products": [{ "ProductSku": "SCU-12345", "ProductName": "Teddy bear", "ImageUrl": "https://www.site.com/bear.jpg", "ProductItemsNum": 1.00, "ProductPrice": 80.00, "Tax": 5.00, "TaxType": 1, }], }</pre>

```
"Amount": 100.00,  
"AltAmount": 0.00,  
"Currency": "USD",  
"AltCurrency": "",  
"AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9",  
"IpAddress": "10.10.10.10"  
}
```

3.7.6.2. Callback with token value

Callback Response with token value

```
{  
  "Payment": {  
    "OperationType": "Sale",  
    "OperationTypeCode": 0,  
    "TransactionId": 3412659,  
    "PaymentSystemId": 1,  
    "TransactionDate": "2019-01-26T14:19:06.1850034",  
    "TerminalId": null,  
    "BankOrderId": null,  
    "Eci": null,  
    "Rrn": null,  
    "Signature": "dbe8539bc7be53eee345860662cb4be2",  
    "Status": "Approved",  
    "StatusCode": 1,  
    "ErrorCode": 0,  
    "ErrorDescription": null,  
    "IsRedirect3DS": false  
  },  
  "Order": {  
    "OrderId": "12345",  
    "CustomInfo": "Order 12345",  
    "Amount": 100.00,  
    "Currency": "USD",  
    "AltAmount": 0.00,  
    "AltCurrency": null,  
    "Fee": 0.00,  
    "CreditAmount": 100.00  
  },  
  "Card": {  
    "NameOnCard": null,  
    "CardBin": null,  
    "CardBrand": null,  
    "CardType": null,  
    "CardLevel": null,  
    "CardBank": null,  
    "CardCountry": null,  
  }  
}
```

```

"PaymentToken": "e34eda6b-a314-461b-b4d4-939ff24686ec",

"CardMask": "523523*****9603"
},
"Customer": {
  "CustomerId": "67890",
  "FirstName": "Mike",
  "LastName": "Wazowsky",
  "Phone": "+12703456677",
  "Email": "customer@site.com"
},
"Address": {
  "Address": "125, Rodeo Drive",
  "City": "New York",
  "Zone": null,
  "Country": "US",
  "Zip": "12345"
},
"IpAddress": "185.177.242.87"
}

```

3.7.6.4. Initial A2C request to get transactionID

Initial request URL
https://api.misterpay.com/api/payment/create
Initial request body
<pre> { "ProcessingId": 1, "TemplateId": 0, "OperationMode": "Iframe", "TransactionType": "P2P", "MerchantInternalUserId": "67890", "MerchantInternalOrderId": "12345", "FirstName": "Mike", "LastName": "Wazowsky", "Phone": "+12703456677", "Email": "customer@site.com", "Address": "125, Rodeo Drive", "Zone": "NY", "City": "New York", "Region": "NY", "State": "NY", </pre>

```
"ZIP": "12345",
"CountryIso": "US",
"ConvertText": false,
"StatusUrl": "https://www.site.com/status",
"ReturnUrl": "https://www.site.com/return",
"DynamicDescriptor": "Purchase at site.com",
"CustomMerchantInfo": "Order 12345",
"Products": [
  {
    "ProductSku": "SCU-12345",
    "ProductName": "Teddy bear",
    "ImageUrl": "https://www.site.com/bear.jpg",
    "ProductItemsNum": 1.00,
    "ProductPrice": 80.00,
    "Tax": 5.00,
    "TaxType": 1,
  }
],
"Amount": 100.00,
"AltAmount": 0.00,
"Currency": "USD",
"AltCurrency": "",
"AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9",
"IpAddress": "10.10.10.10"
}
```

Response with TransactionID

```
{
  "PaymentToken": "",
  "PaymentTokenExpireDate": null,
  "HtmlIframeForm": null,
  "RedirectUrl3DS": null,
  "TransactionState": "Approved",
  "TransactionStateCode": 1,

  "TransactionId": '3412659',

  "ErrorCode": 0,
  "ResponseMessage": null
}
```

Signature calculation

```
<?php
$key = '03e5515f-7cd8-49ce-9284-5d78ff1390d9';
$secret = 'b6e6b617-88b0-4102-8f8f-704327fa6d9f';

$transactionId = '3412659';
```

```
$requestSign = $key.'!'.$transactionId.'!'.strtoupper($secret);  
$signature = hash_hmac('md5',$requestSign,$secret);  
?>
```

3.7.6.5. A2C transaction example with token

Request URL
https://api.misterpay.com/api/payment/p2p
Request
<pre>{ "AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9", "TransactionId": "3412659", "P2PCardToken" : "e34eda6b-a314-461b-b4d4-939ff24686ec", "PayFee" : false, "Signature": "dbe8539bc7be53eee345860662cb4be2" }</pre>
Response
<pre>{ "PaymentToken": null, "PaymentTokenExpireDate": null, "HtmlIframeForm": null, "RedirectUrl3DS": null, "TransactionState": "Approved", "TransactionStateCode": 1, "TransactionId": 3412659, "ErrorCode": 0, "ResponseMessage": null }</pre>

3.7.7. Callback example

The MisterPay System uses JSON POST to send transaction callback response.

Callback body
<pre>{ "Payment": { "OperationType": "P2P",</pre>

```
"OperationTypeCode":0,
"TransactionId":3412659,
"PaymentSystemId":1,
"TransactionDate":"2019-01-26T14:19:06.1850034",
"TerminalId":null,
"BankOrderId":null,
"Eci":null,
"Rrn":null,
"Signature":"d8e8539bc7be53eee345860662cb4be2",
"Status":"Approved",
"StatusCode":1,
"ErrorCode":0,
"ErrorDescription":null,
"IsRedirect3DS":false
},
"Order": {
  "OrderId":"12345",
  "CustomInfo":"Order 12345",
  "Amount":100.00,
  "Currency":"USD",
  "AltAmount":0.00,
  "AltCurrency":null,
  "Fee":0.00,
  "CreditAmount":100.00
},
"Card": {
  "NameOnCard":"Mike Wazovsky",
  "CardBin":"111122",
  "CardBrand":null,
  "CardType":null,
  "CardLevel":null,
  "CardBank":null,
  "CardCountry":null,
  "PaymentToken": "e34eda6b-a314-461b-b4d4-939ff24686ec",
  "CardMask":"111122*****4444"
},
"Customer": {
  "CustomerId":"67890",
  "FirstName":"Mike",
  "LastName":"Wazowsky",
  "Phone":"+12703456677",
  "Email":customer@site.com
},
"Address": {
  "Address":"125, Rodeo Drive",
  "City":"New York",
  "Zone":null,
  "Country":"US",
  "Zip":"12345"
```

```
}  
"IpAddress":"185.177.242.87"  
}
```

3.8. Account balance

You can check current account balance available for A2C transfers sending a Balance request.

Note: The refill method depends on the terms of merchant agreement.

3.8.1. Request URL

<https://api.misterpay.com/api/Merchant/balance>

3.8.2. Request example

Request URL
<code>https://api.misterpay.com/api/Merchant/balance</code>
Request
<pre>{ "ProcessingId": 1, "AssetPaymentsKey": "03e5515f-7cd8-49ce-9284-5d78ff1390d9" }</pre>
Response
<pre>{ "currency": "UAH", "amount": 1.96, "errorCode": 0, "responseMessage": null }</pre>