



API documentation

June 25, 2020

by midtrans

Contents

1	Getting Started	5
	Overview	5
	Supported Browsers	5
	Mobile	5
	Desktop	7
	Library and Plugin	7
	Libraries	7
	Plugin	7
2	On Board with Snap	9
	Existing Merchant	9
	1. Set Activate Payment Option in Snap Payment List	9
	New Merchant	9
	1. Register to Midtrans Sandbox/Production account	11
	2. Fill in the required information in Merchant Admin Portal (MAP)	12
	3. Take note of your account Access Keys	12
	4. Configure Redirection URL	14
	5. Configure Merchant Info (Name, Logo & Theme Color)	14
	6. Set Activate Payment Option in Snap Payment List	14
3	Backend Integration	19
	Endpoint	19
	Request Headers	19
	Request Body (JSON Parameter)	20
	JSON Objects	24
	Transaction Details Object	24
	Item Details Object	25
	Customer Details Object	26
	Address Object	27

	Credit Card Object	28
	BCA Virtual Account Object	29
	Permata Virtual Account Object	30
	BNI Virtual Account Object	31
	Gopay Object	31
	Cstore Object	32
	Callbacks Object	32
	Expiry Object	32
	Response	33
	Response Success	33
	Response Failed	34
4	Frontend Integration	36
	Overview	36
	Snap.js location	37
	Viewport Meta Tag	37
	Snap JS	37
	pay(snapToken, options)	37
	show()	39
	hide()	39
	JS Callback	40
	Transaction Result	40
	Window Redirection	42
5	Supported Payment Channels	44
	Credit Card	44
	Sample JSON Request Body	44
	BCA Virtual Account	46
	Sample JSON Request Body	46
	Permata Virtual Account	47
	Sample JSON Request Body	48
	BNI Virtual Account	49
	Sample JSON Request Body	49
	Mandiri Bill	50
	Sample JSON Request Body	50
	GoPay	51
	Detailed payment Flow for GoPay on a Desktop PC/Laptop	54
	Detailed Payment Flow for GoPay on a Smartphone	54
	Sample JSON Request Body	55
	KlikBCA	56

Sample JSON Request Body	56
BCA KlikPay	58
Sample JSON Request Body	58
Mandiri Clickpay	59
Sample JSON Request Body	59
CIMB Clicks	60
Sample JSON Request Body	60
Danamon Online Banking	61
Sample JSON Request Body	61
BRI Epay	62
Sample JSON Request Body	63
Indomaret	64
Sample JSON Request Body	64
Alfamart	65
Sample JSON Request Body	65
Akulaku	66
Sample JSON Request Body	66
6 Other Features	68
Two Clicks	68
Initial Transaction	69
Successive Transactions	69
One Click	69
Initial Transaction	70
Successive Transaction	71
Custom Virtual Account Number	71
Pre-Authorization	72
7 Transaction Status	73
Email Notification	73
HTTP(S) POST Notification	73
Merchant Admin Portal(MAP) Transaction Status	76
8 Handling Notifications	77
Challenge Response	77
Signature Key	78
Best Practice to Handle notification	78
Override Notification Url	80
Sample case	80
9 Status Code	82

Code 2xx	82
Code 3xx	84
Code 4xx	84
Code 5xx	85
10 Testing Credentials	86
Credit Card	86
Bank Transfer	86
Direct Debit	87
e-Wallet	87
Convenience Store	88
11 Going Live	89
Minimum Transaction Amount	89
PHP	89
Magento	90
OpenCart	90
Prestashop	90
Woocommerce	90

Getting Started

Overview

1. User performs the checkout operation
2. Merchant server makes an [api request](#) to the snap backend to get the SNAP_TOKEN
3. Snap backend responds to the api call with the SNAP_TOKEN
4. Merchant server constructs the [html](#) page and sends it back to the browser
5. User verifies the details and clicks the pay button. Merchant's javascript code calls [snap.pay\(SNAP_TOKEN, options\)](#). User then fills up the payment details and clicks the confirm button.
6. Snap JS sends the payment details to the snap backend
7. Snap backend processes the details and responds with the charge status. Snap JS then calls the corresponding [callback](#) provided by the merchant's javascript code.
8. Snap backend notifies the merchant server about the charge status

Supported Browsers

We try to support the latest versions of all major browsers. The list below provides the minimum supported version of each browser in Mobile and Desktop platform.

Mobile

- Chrome 32 and above
- Android 5.0 and above

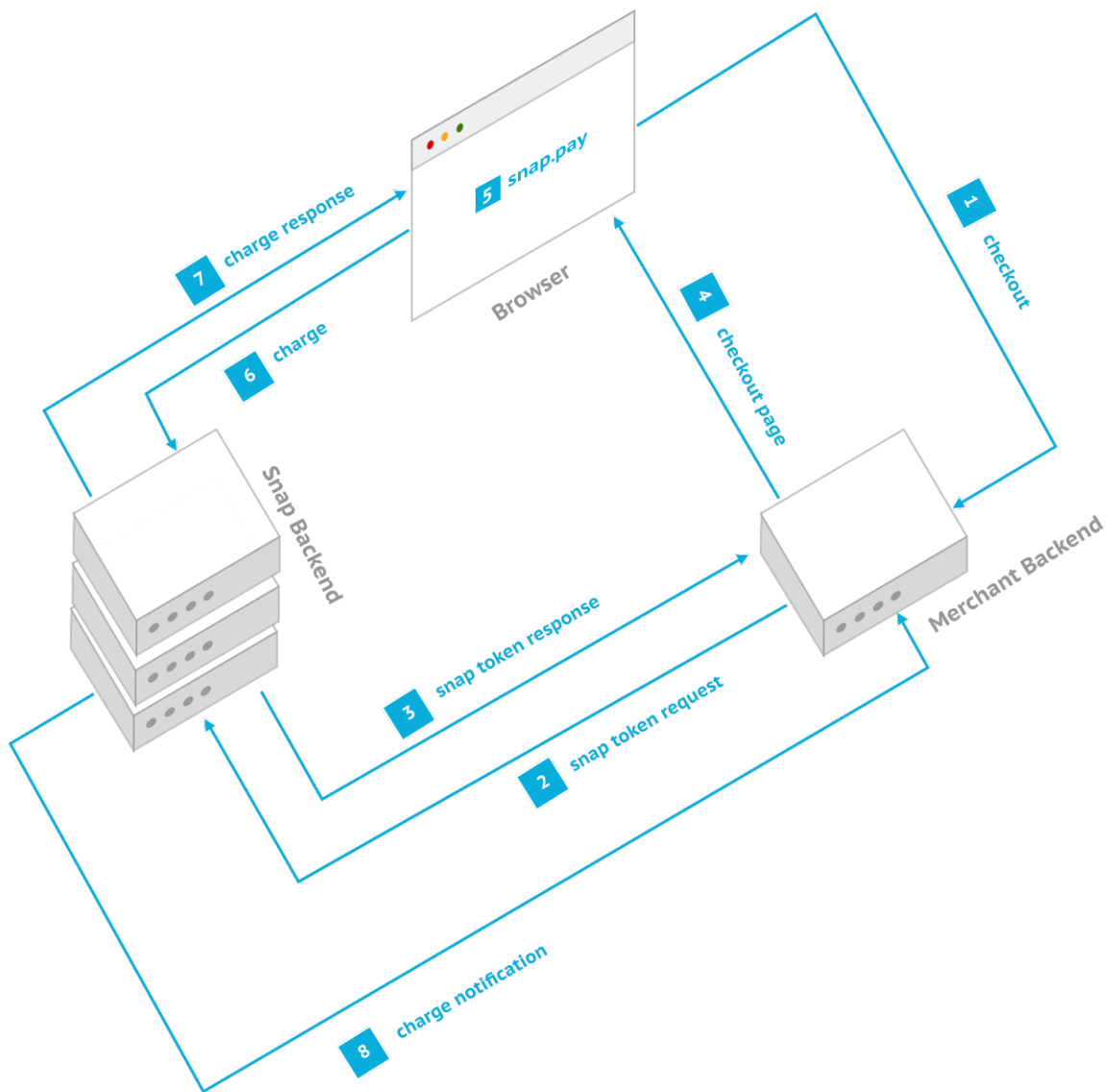


Figure 1.1: payment flow

- Safari 8 and above
- BlackBerry 10 and above

Desktop

- Chrome 26 and above
- Firefox 29 and above
- Internet Explorer 10 and above
- Safari 6 and above

Library and Plugin

We are trying to make the integration process as easy as possible. This section contains a list of plugins and libraries owned by Midtrans. If you write your own plugin or library and would like us to link it, contact support@midtrans.com.

Libraries

Platform	Resources
PHP	Github

Plugin

Platform	Resources
Prestashop	v1.6 Github
Magento	v1.8, v1.9 Github
Opencart	v2.0, v2.1, v2.2 Github
WHMCS	WHMCS v5.3.12 or greater Github

Platform	Resources
Wordpress	Wordpress v3.9.1 - v4.x
Woocommerce	WooCommerce v2.1.11 - v2.5.x Github

On Board with Snap

To start using Snap, you have to register for Midtrans Sandbox/Production account [here](#). Once you have registered, Midtrans will provide all available payment channels in sandbox mode that you can start to integrate with.

To know more about steps to enable GoPay payment please see this [section](#).

Existing Merchant

For existing merchants (merchants already live in Production), only one step need to be taken to enable a specific payment method.

1. Set Activate Payment Option in Snap Payment List

Merchants can pick to activate a payment method through the Merchant Administration Portal (MAP) in menu [Settings -> Snap Preferences Settings](#) on Tab 'Payment Channels'. Merchant can drag the a desired payment channels into which payment channel's order. Consider the following figure.

Just drag a desired payment channels option to the active list of payment list. Through this option, the merchant no longer need to do any code adjustment in their end, just set a desired payment channel on and that payment will be live.

New Merchant

Goal: Merchants complete the pre-requisite before integrating with Midtrans

Settings

General Settings

Access Keys

VT-Web Preferences

Snap Preferences

Configuration

Billing

Email Notifications

VT-Link

Daily Report

Snap Preferences

Set your custom preferences on Snap Payment Page








Theme and Logo

Payment Channels

Bank List

System Settings


Inactive Payment Channels


	Telkomsel Cash Pay through your Telkomsel Cash account	+
	e-Pay BRI Pay using e-Pay BRI account	+
	XL Tunai Pay with XL Tunai	+
	Indomaret Pay from Indomaret	+
	Indosat Dompotku Pay with Indosat Dompotku	+
	Gift Card Indonesia Pay with gift card	+
	Mandiri e-Cash Pay with your Mandiri e-Cash account	+


PREVIEW

Papi Live

SELECT PAYMENT

 **Credit Card**
Pay with Visa or MasterCard

 **ATM/Bank Transfer**
Pay from ATM Bersama, Prima or Alto

 **GO-PAY**
Pay with your GO-PAY wallet

Drag/drop to manually rearrange sorting, or

Apply Recommended Sorting

Figure 2.1: payment list

10

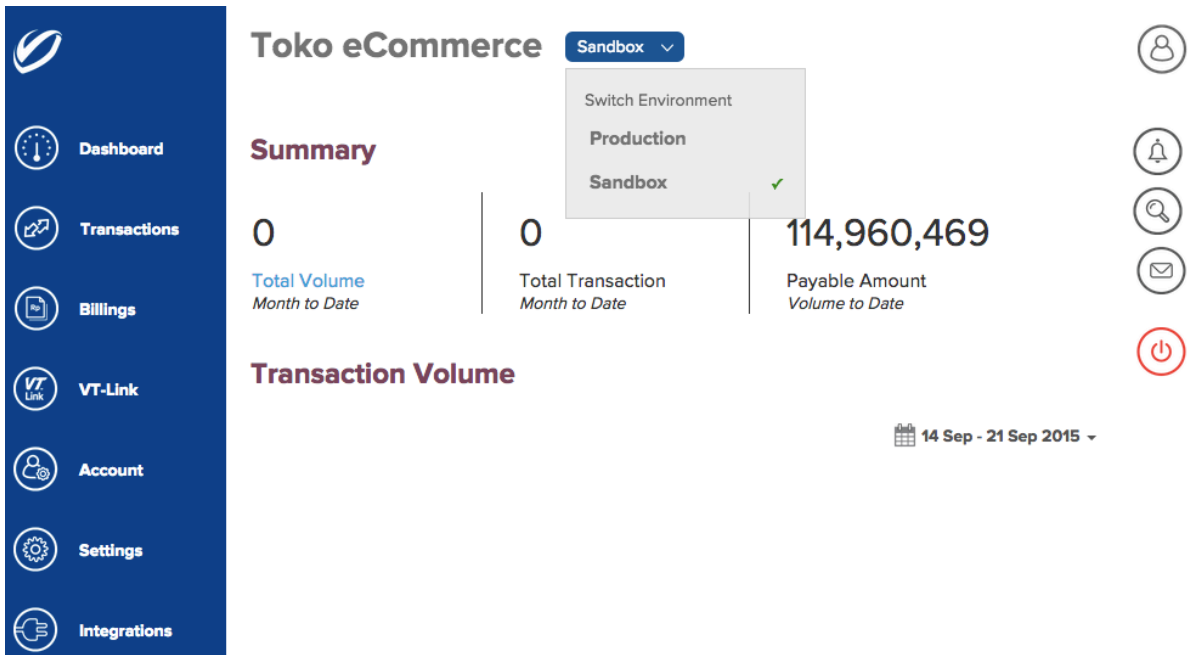
There are a few pre-requisites before integrating with Midtrans:

1. Register to Midtrans Sandbox/Production account

Midtrans has one central login to access both production and sandbox account. Sandbox is utilized for development period while production is utilized when the merchant has completed the integration process and want to go live. Data and transaction made on sandbox account will not trigger an actual purchase while in production account will trigger an actual process. Register for Midtrans Sandbox/Production account [here](#).

Once logged in, there will be a small button on the header of the dashboard that shows you on whether you are in the production or sandbox environment. The color of the navigation sidebar are also set differently between Production (light blue) and Sandbox (dark blue) for further clarity.

The screenshot displays the 'Toko eCommerce' dashboard interface. On the left is a blue sidebar with icons and labels for 'Dashboard', 'Transactions', 'Billings', 'Account', and 'Settings'. The main header area includes the 'Toko eCommerce' logo, a 'Production' status button with a dropdown arrow, and a user profile icon. A green notification bar states 'Signed in successfully'. Below this is a 'Summary' section with two columns: 'Total Volume Month to Date' showing '0' and 'Total Transaction Month to Date' also showing '0'. The 'Transaction Volume' section is partially visible at the bottom. On the right side, there is a vertical stack of icons: a user profile, a bell for notifications, a magnifying glass for search, an envelope for email, and a red power button. The footer shows a calendar icon and the date range '14 Sep - 21 Sep 2015' with a dropdown arrow.



2. Fill in the required information in Merchant Admin Portal (MAP)

The required fields can be found under [Settings - General Settings](#).

Warning!

Merchant Name: DO NOT input symbols.

Valid Merchant URL:

- Lesser than 25 char.
- Root domain lesser than 5 char.
- Accepted symbols are ‘:’, ‘/’, ‘.’.

3. Take note of your account Access Keys

Your account keys can be found under [Settings - Access Keys](#).

Caution!

Be discreet on your ‘Server Key’.

General Settings

Merchant Name	Toko Buah eCommerce
Director Name	Andri
Director Phone	02112345678
NPWP	1000000000000001
URL Merchant	http://www.tokobuah.com
Merchant email	andri@sneekers.com
Timezone	+7h - Jakarta, Sumatera, Java, Western Borneo ▼

Save

Figure 2.2: general settings

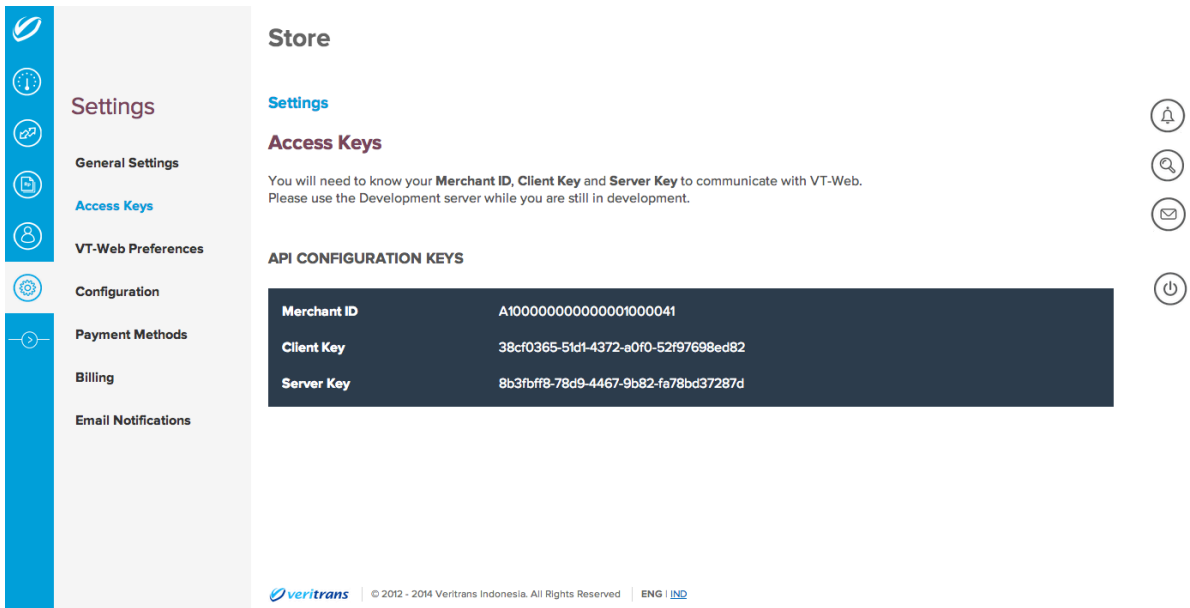


Figure 2.3: access keys

4. Configure Redirection URL

Customer will be returned to your website after payment process is completed.

Go to [Settings - Configuration](#) menu to manage the redirection URL.

Warning!

Please fill the URL with http:// or https://

5. Configure Merchant Info (Name, Logo & Theme Color)

Merchants can configure their name (brand name), logo and theme color that can be matched with the merchant's web theme color through [Settings - Snap Preferences](#) on tab 'Theme and Color' in their dashboard.

6. Set Activate Payment Option in Snap Payment List

Merchants can pick to activate specific payment methods through Merchant Administration Portal (MAP) in menu [Settings - Snap Preferences Settings](#) on Tab 'Payment Channels'. Merchants can drag

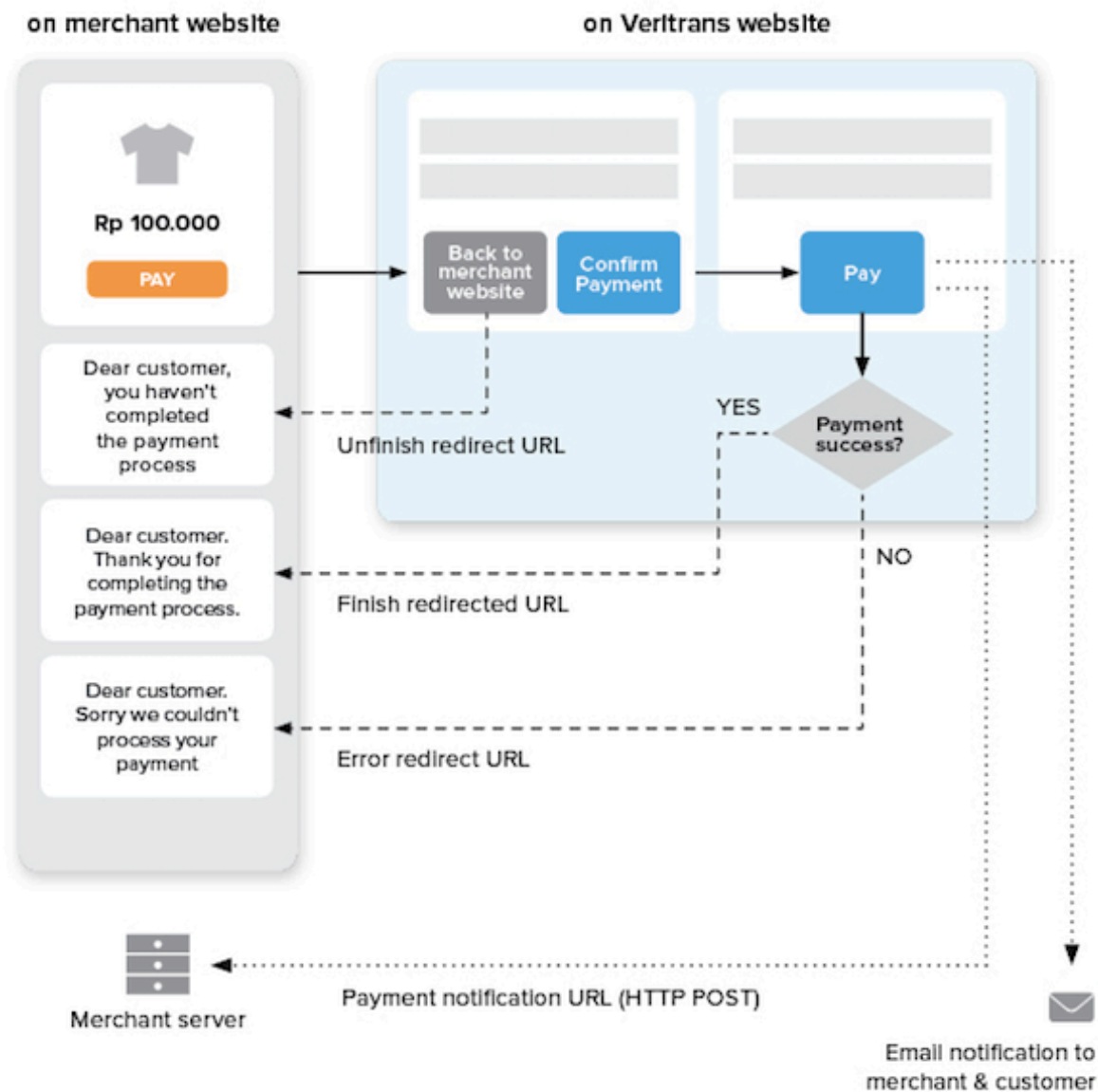
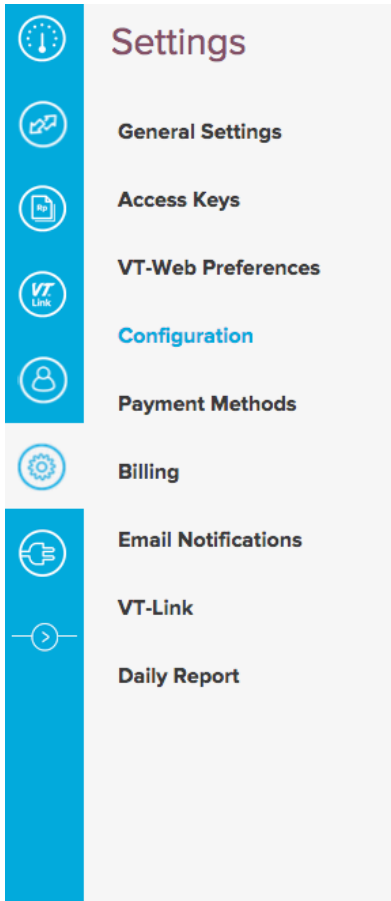


Figure 2.4: redirection url



Settings

Configuration

Veritrans requires the URL endpoints for the following scenarios:

Payment Notification URL*	https://sneekers.com/payment/notification Address where we will send the notification via HTTP Post request. E.g http://yourwebsite.com/notification/handling Send test notification See history
Finish Redirect URL*	https://sneekers.com/payment/finish Customer sent here if payment is successful e.g http://yourwebsite.com/payment/finish
Unfinish Redirect URL*	https://sneekers.com/payment/unfinish Customer sent here if they click on 'Back to Order Website' on VT-Web's payment page e.g http://yourwebsite.com/payment/unfinish
Error Redirect URL*	https://sneekers.com/payment/error Customer sent here if payment is unsuccessful or an error occurs during payment e.g http://yourwebsite.com/payment/error

Figure 2.5: map redirection url

Settings

General Settings

Access Keys

VT-Web Preferences

Snap Preferences

Configuration

Billing

Email Notifications

VT-Link

Daily Report

Papi Live

Sandbox

Hi fauzar

Snap Preferences

Set your custom preferences on Snap Payment Page

Theme and Logo

Payment Channels

Bank List

System Settings

Nama Display

COCO

Logo (Max 1 MB)

Click here to choose file or drag & drop your logo below

Logo Placement Tips

- Logo will be resized to 72 × 240 pixels
- Use PNG image instead of GIF if your logo is transparent
- Max file size is 1 MB
- Uploaded Logo will replace Display Name

Theme Color

Language

English

PREVIEW

COCO

ORDER SUMMARY

amount

Rp5,104,000

order ID

MID-62215736789

order details

shipping details

item(s)	amount (Rp)
Magic Wand× 2	800,000
Large Tin Cauldron ...× 2	1,800,000
Green Frogs	599,000
Cat Whiskers× 12	1,800,000

SELECT YOUR PAYMENT

Figure 2.6: snap themes

Settings








General Settings
Access Keys
VT-Web Preferences
Snap Preferences
Configuration
Billing
Email Notifications
VT-Link
Daily Report

Snap Preferences

Set your custom preferences on Snap Payment Page

Theme and Logo
Payment Channels
Bank List
System Settings


Inactive Payment Channels

	Telkomsel Cash Pay through your Telkomsel Cash account	+
	e-Pay BRI Pay using e-Pay BRI account	+
	XL Tunai Pay with XL Tunai	+
	Indomaret Pay from Indomaret	+
	Indosat Dompotku Pay with Indosat Dompotku	+
	Gift Card Indonesia Pay with gift card	+
	Mandiri e-Cash Pay with your Mandiri e-Cash account	+

PREVIEW

Papi Live


SELECT PAYMENT



Credit Card

Pay with Visa or MasterCard


×



ATM/Bank Transfer

Pay from ATM Bersama, Prima or Alto

×



GO-PAY

Pay with your GO-PAY wallet

×

Drag/drop to manually rearrange sorting, or

Apply Recommended Sorting

Figure 2.7: payment list

the particular payment into which payment channel's order. Such as this figure. Just drag the a payment channel option to the activate the payment method.

Backend Integration

Backend integration goal is to acquire SNAP_TOKEN by providing payment informations. We provide a HTTP API to do this.

Endpoint

HTTP Method: POST

Production: <https://app.midtrans.com/snap/v1/transactions>

Sandbox: <https://app.sandbox.midtrans.com/snap/v1/transactions>

Request Headers

```
SERVER_KEY = "VT-server-Cpo03kYD0c0cNUKgt6hnLkKg"
```

```
AUTH_STRING = Base64("VT-server-Cpo03kYD0c0cNUKgt6hnLkKg:")
```

```
AUTH_STRING = "VlQtc2VydjVybUNwbzAza1lET2MwY05VS2d0NmhuTGtLZzo="
```

To create valid Snap HTTP request, merchant have to add 3 headers:

- Accept: application/json
- Content-Type: application/json
- Authorization: Basic AUTH_STRING

Snap validates HTTP request by using Basic Authentication method. The username is your SERVER_KEY while the password is empty. You can see your SERVER-KEY on [Settings - Access Keys](#).

Authorization header value is represented by AUTH_STRING. AUTH_STRING is base-64 encoded string of your username & password.

AUTH_STRING = Base64(SERVER_KEY + :)

Request Body (JSON Parameter)

Minimum request

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  }
}
```

Complete request

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,
    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456",
    "billing_address": {
```

```

    "first_name": "TEST",
    "last_name": "MIDTRANSE",
    "email": "test@midtrans.com",
    "phone": "081 2233 44-55",
    "address": "Sudirman",
    "city": "Jakarta",
    "postal_code": "12190",
    "country_code": "IDN"
  },
  "shipping_address": {
    "first_name": "TEST",
    "last_name": "MIDTRANSE",
    "email": "test@midtrans.com",
    "phone": "0 8128-75 7-9338",
    "address": "Sudirman",
    "city": "Jakarta",
    "postal_code": "12190",
    "country_code": "IDN"
  }
},
"enabled_payments": ["credit_card", "mandiri_clickpay", "cimb_clicks",
  "bca_klikbca", "bca_klikpay", "bri_epay", "echannel", "permata_va",
  "bca_va", "bni_va", "other_va", "gopay", "indomaret",
  "danamon_online", "akulaku"],
"credit_card": {
  "secure": true,
  "channel": "migs",
  "bank": "bca",
  "installment": {
    "required": false,
    "terms": {
      "bni": [3, 6, 12],
      "mandiri": [3, 6, 12],
      "cimb": [3],
      "bca": [3, 6, 12],
      "offline": [6, 12]
    }
  }
},
"whitelist_bins": [

```

```

        "48111111",
        "41111111"
    ]
},
"bca_va": {
    "va_number": "12345678911",
    "sub_company_code": "00000",
    "free_text": {
        "inquiry": [
            {
                "en": "text in English",
                "id": "text in Bahasa Indonesia"
            }
        ],
        "payment": [
            {
                "en": "text in English",
                "id": "text in Bahasa Indonesia"
            }
        ]
    }
},
"bni_va": {
    "va_number": "12345678"
},
"permata_va": {
    "va_number": "1234567890",
    "recipient_name": "SUDARSONO"
},
"callbacks": {
    "finish": "https://demo.midtrans.com"
},
"expiry": {
    "start_time": "2018-12-13 18:11:08 +0700",
    "unit": "minutes",
    "duration": 1
},
"custom_field1": "custom field 1 content",
"custom_field2": "custom field 2 content",

```

```
"custom_field3": "custom field 3 content"
}
```

Parameter	Description
transaction_details Transaction Details (required)	Specific information regarding the transaction
item_details Array of Item Details (optional)	Shopping item details will be paid by customer
customer_details Customer Details (optional)	Specific information regarding the customer
enabled_payments Array (optional)	<p>List of payment types that should be enabled. If blank, all active payment types are included.</p> <p>Options:</p> <p>credit_card,mandiri_clickpay,cimb_clicks, bca_klikbca,bca_klikpay,bri_epay, telkomsel_cash,echannel,permata_va,other_va, bca_va,bni_va,indomaret,danamon_online, akulaku.</p> <p>An alias refers to a list of payment types. Adding an alias is the equivalent of adding all the the payment types it refers to.</p> <p>Supported aliases:</p> <p>bank_transfer => permata_va, bca_va, bni_va</p> <p>cstore => kioson, indomaret.</p>
credit_card CreditCard (optional)	<p>If you want to use other_va, either permata_va or bni_va because Midtrans handles other bank transfer as either Permata or BNI VA.</p> <p>Credit card payment options</p>
bca_va BCA Virtual Account (optional)	BCA Virtual Account payment options
permata_va Permata Virtual Account (optional)	Permata Virtual Account payment options
bni_va BNI Virtual Account (optional)	BNI Virtual Account payment options

Parameter	Description
callbacks Callbacks (optional)	Redirect URL after transaction is successfully paid (Overridden by JS callback). Can also be set via Snap Settings menu in your dashboard.
expiry Expiry (optional)	Custom transaction lifetime
custom_field1 String(255) (optional)	Custom field 1 for custom parameter from merchant
custom_field2 String(255) (optional)	Custom field 2 for custom parameter from merchant
custom_field3 String(255) (optional)	Custom field 3 for custom parameter from merchant

JSON Objects

Collection of JSON objects that are used in [Create Snap Token](#) parameter.

Note: More JSON objects exists for each specific [payment channels](#).

Transaction Details Object

```
{
  "order_id": "ORDER-101",
  "gross_amount": 10000
}
```

Parameter	Description
transaction_details.order_id String(50) (required)	Unique transaction ID. A single ID could be used only once by a Merchant. NOTE: Allowed Symbols are dash(-), underscore(_), tilde (~), and dot (.)
transaction_details.gross_amount Integer (required)	Amount to be charged

Item Details Object

```
{
  "id": "ITEM1",
  "price": 10000,
  "quantity": 1,
  "name": "Midtrans Bear",
  "brand": "Midtrans",
  "category": "Toys",
  "merchant_name": "Midtrans"
}
```

Parameter	Description
id	Item ID
String(50) (optional)	
price	Price of the item
Integer (required)	NOTE: Don't add decimal
quantity	Quantity of the item
Integer (required)	NOTE: Must be greater than or equal 1
name	Name of the item
String(50) (required)	
brand	Brand of the item
String(50) (optional)	
category	Category of the item
String(50) (optional)	
merchant_name	Merchant selling the item
String(50) (optional)	

Warning!

Subtotal (item price multiplied by quantity) of all the item details needs to be exactly the same as the gross_amount inside transaction_details object

Tips:

You can actually make an item with minus price to be presented as discount

Customer Details Object

```
{
  "first_name": "TEST",
  "last_name": "MIDTRANSER",
  "email": "test@midtrans.com",
  "phone": "+628123456",
  "billing_address": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "081 2233 44-55",
    "address": "Sudirman",
    "city": "Jakarta",
    "postal_code": "12190",
    "country_code": "IDN"
  },
  "shipping_address": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "0 8128-75 7-9338",
    "address": "Sudirman",
    "city": "Jakarta",
    "postal_code": "12190",
    "country_code": "IDN"
  }
}
```

Parameter	Description
first_name	
String(255) (optional)	
last_name	
String(255) (optional)	
email	
String(255) (optional)	
phone	
String(255) (optional)	

Parameter	Description
billing_address	
Address (optional)	
shipping_address	
Address (optional)	

Tips:

You can use these parameters as rules in our Fraud Detection System

Address Object

```
{
  "first_name": "TESTER",
  "last_name": "MIDTRANSER",
  "email": "test@midtrans.com",
  "phone": "081 2233 44-55",
  "address": "Sudirman",
  "city": "Jakarta",
  "postal_code": "12190",
  "country_code": "IDN"
}
```

Parameter	Description
first_name	
String(255) (optional)	
last_name	
String(255) (optional)	
email	
String(255) (optional)	
phone	
String(255) (optional)	
address	
String(200) (optional)	
country_code	
String(10) (optional)	
postal_code	
String(10) (optional)	

Parameter	Description
city	
String(100) (optional)	

Credit Card Object

```
{
  "save_card": true,
  "secure": true,
  "channel": "migs",
  "bank": "maybank",
  "installment": {
    "required": false,
    "terms": {
      "bni": [3, 6, 12],
      "mandiri": [3, 6, 12],
      "cimb": [3],
      "bca": [3, 6, 12],
      "offline": [6, 12]
    }
  },
  "whitelist_bins": [
    "48111111",
    "41111111",
    "bni"
  ]
}
```

Parameter	Description
secure	Use 3D-Secure authentication when using credit card.
Boolean (optional)	Default: false
bank	Acquiring bank. Options: bca, bni, mandiri, cimb,
String (optional)	bri, danamon, maybank, mega
channel	Acquiring channel. Options: migs
String (optional)	
type	Credit card transaction type. Options: authorize,
String (optional)	authorize_capture. Default: "authorize_capture"

Parameter	Description
whitelist_bins Array (optional)	Allowed credit card BIN numbers. The bin value can be either a prefix(upto 8 digits) of card number or the name of a bank, in which case all the cards issued by that bank will be allowed. The supported bank names are bni bca mandiri cimb bri and maybank. Default: allow all cards
installment.required Boolean (optional)	Force installment when using credit card. Default: false
installment.terms Object (optional)	Available installment terms

BCA Virtual Account Object

```
{
  "va_number": "12345678911",
  "sub_company_code": "00000",
  "free_text": {
    "inquiry": [
      {
        "en": "text in English",
        "id": "text in Bahasa Indonesia"
      }
    ],
    "payment": [
      {
        "en": "text in English",
        "id": "text in Bahasa Indonesia"
      }
    ]
  }
}
```

Parameter	Description
va_number String (optional)	Custom virtual account number . Length should be within 1 to 11.

Parameter	Description
sub_company_code String (optional)	BCA sub company code directed for this transactions. NOTE: Default is 00000
free_text FreeText (optional)	

Free Text

Parameter	Description
inquiry Array of FreeTextItem (optional)	Size should not exceed 10.
payment Array of FreeTextItem (optional)	Size should not exceed 10.

Free Text Item

Parameter	Description
en String (required)	Size should not exceed 50 chars.
id String (required)	Size should not exceed 50 chars.

Permata Virtual Account Object

```
{
  "va_number": "1234567890",
  "recipient_name": "SUDARSONO"
}
```

Parameter	Description
va_number String (optional)	Custom virtual account number . Length should be 10. Only supported for b2b transactions.

Parameter	Description
recipient_name String (optional)	Recipient name shown on the on the bank's payment prompt. It is shown as 20 character uppercase string. Anything over 20 character will be truncated. NOTE: Default is merchant name

BNI Virtual Account Object

```
{
  "va_number": "12345678"
}
```

Parameter	Description
va_number String (optional)	Custom virtual account number . Length should be within 1 to 8.

Gopay Object

```
{
  "enable_callback": true,
  "callback_url": "http://gopay.com"
}
```

Parameter	Description
enable_callback Boolean (optional)	Enable redirect back to merchant from GoJek apps. Default: false
callback_url String (optional)	Determine where should customer be redirected from GoJek apps. It supports both HTTP and deeplink. Default: same value as finish url

Cstore Object

```
{  
  "alfamart_free_text_1" : "qwerty",  
  "alfamart_free_text_2" : "asdfg",  
  "alfamart_free_text_3" : "zxcvb"  
}
```

Parameter	Description
alfamart_free_text_1 String(40) (optional)	First row of printed receipt description
alfamart_free_text_2 String(40) (optional)	Second row of printed receipt description
alfamart_free_text_3 String(40) (optional)	Third row of printed receipt description

Callbacks Object

```
{  
  "finish": "https://demo.midtrans.com"  
}
```

Parameter	Description
finish String(255) (optional)	Redirect URL after transaction is successfully paid (Overriden by JS callback). Can also be set via Snap Settings menu in your dashboard.

Expiry Object

Note: If this parameter is not sent, the default expiry for snap token is 24 hours from the time the token was created. Furthermore, if only unit and duration is given, start_time will equal the timestamp of the token creation.

Note: Maximum expiry for the snap token is 7 days. If the expiry sent is more than 7 days after the token is created. The snap token will still expire 7 days after creation.

```
{
  "start_time": "2018-12-13 18:11:08 +0700",
  "unit": "minutes",
  "duration": 1
}
```

Parameter	Description
start_time String(255) (optional)	Timestamp in yyyy-MM-dd HH:mm:ss Z format. If not specified, transaction time will be used as start time (when customer charge)
duration Integer (required for expiry)	Expiry duration
unit String (required for expiry)	Expiry unit. Options: day, hour, minute (plural term also accepted)

Response

Response Success

```
{
  "token": "d379aa71-99eb-4dd1-b9bb-eefe813746e9",
  "redirect_url": "https://app.sandbox.veritrans.co.id/snap/v2/vtweb/d379aa71-99eb-4"
}
```

HTTP status code: 201

Field	Description
token String(36)	Snap token for opening the Snap popup
redirect_url String(75)	URL for redirection

Response Failed

Authentication Failed

```
{
  "error_messages": [
    "Access denied due to unauthorized transaction, please check client or server key",
    "Visit https://snap-docs.midtrans.com/#request-headers for more details"
  ]
}
```

HTTP status code: 401

Field	Description
error_messages Array	Error messages

Validation Error

```
{
  "error_messages": [
    "transaction_details.gross_amount is not equal to the sum of item_details"
  ]
}
```

HTTP status code: 400

Field	Description
error_messages Array	Error messages

Order ID Already Paid and Utilized

```
{
  "error_messages": [
    "transaction_details.order_id has been paid and utilized, please use another order"
  ]
}
```

```
]
}
```

HTTP status code: 400

Field	Description
error_messages	Error messages
Array	

Internal System Error

```
{
  "error_messages": [
    "Sorry, we encountered internal server error. We will fix this soon."
  ]
}
```

HTTP status code: 500

Field	Description
error_messages	Error messages
Array	

Frontend Integration

Overview

```
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <script type="text/javascript"
      src="https://app.sandbox.midtrans.com/snap/snap.js"
      data-client-key="<CLIENT-KEY>"></script>
  </head>
  <body>
    <button id="pay-button">Pay!</button>
    <script type="text/javascript">
      var payButton = document.getElementById('pay-button');
      payButton.addEventListener('click', function () {
        snap.pay(' <SNAP_TOKEN> ');
      });
    </script>
  </body>
</html>
```

Frontend integration goal is to show Snap payment page within your site.

Include `snap.js` into your page so snap module is available. Don't forget to put your CLIENT-KEY as value of `data-client-key` attribute in `snap.js` script tag. You can see your CLIENT-KEY on [Settings - Access Keys](#).

You can start payment process by calling `snap.pay` with `SNAP_TOKEN` acquired from [backend inte-](#)

gration as parameter.

Snap.js location

Production: <https://app.midtrans.com/snap/snap.js>

Sandbox: <https://app.sandbox.midtrans.com/snap/snap.js>

Viewport Meta Tag

To ensure that Snap popup modal is displayed correctly on a mobile device, please include the viewport meta tag inside your <head> element.

The most common implementation is the line of code below:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Snap JS

snap has 3 public functions: pay, show & hide

pay(snapToken, options)

Start Snap payment page.

```
snap.pay('YOUR_SNAP_TOKEN', {
  onSuccess: function(result){console.log('success');console.log(result);},
  onPending: function(result){console.log('pending');console.log(result);},
  onError: function(result){console.log('error');console.log(result);},
  onClose: function(){console.log('customer closed the popup without finishing the p
  )})
```

Parameter:

Name	Description
snapToken String (required)	Snap token acquired from backend integration

Name	Description
enabledPayments Array (optional)	List of payment types to be displayed. This will filter out enabled payments from backend integration.
options.onSuccess Function (optional)	Payment success callback (200 status_code)
options.onPending Function (optional)	Payment pending callback (201 status_code)
options.onError Function (optional)	Payment error callback (4xx or 5xx status_code)
options.onClose Function (optional)	Called if customer has closed the payment popup prematurely without finishing the payment
options.language String (optional)	Sets the language. This will override language setting on Merchant Administration Portal. Supported values are en (English) and id (Bahasa Indonesia). Defaults to id
options.skipOrderSummary Boolean (optional)	Skips the order summary page if set to true. Set to false by default.
options.skipCustomerDetails Boolean (optional)	Make email and phone number optional in CC form if set to true. Set to false by default.
options.autoCloseDelay Integer (optional)	Auto closes the last page of indomaret and bank transfer payments after the specified time delay. The time delay is specified in seconds. Setting it to 0 will disable this feature. Defaults to 0.
options.showOrderId Boolean (optional)	Hide order ID if set to false. Set to true by default.
options.selectedPaymentType String (optional)	Skips order summary and select payment page to directly select a specific payment type. Supported values are credit_card, mandiri_clickpay, cimb_clicks, bca_klikbca, bca_klikpay, bri_epay, telkomsel_cash, echannel, indosat_dompotku, permata_va, other_va, bca_va, bni_va, kioson, indomaret, gci, and danamon_online.
options.gopayMode String (optional)	Choose the UI mode for GoPay . Supported values are deeplink, qr, and auto. Set to auto by default.

onSuccess, onPending, & onError function accept one parameter which is [Transaction Result](#) object.

```

function ajaxGetToken(transactionData, callback){
    var snapToken;
    // Request get token to your server & save result to snapToken variable

    if(snapToken){
        callback(null, snapToken);
    } else {
        callback(new Error('Failed to fetch snap token'),null);
    }
}

payButton.onclick(function(){
    snap.show();
    ajaxGetToken(transactionData, function(error, snapToken){
        if(error){
            snap.hide();
        } else {
            snap.pay(snapToken);
        }
    });
});

```

show()

Show snap loading page. Helper function if you want to show instant loading feedback while getting SNAP_TOKEN using AJAX.

If AJAX success, call `snap.pay` to continue payment process. Else, call `snap.hide` to end loading page.

hide()

Hide active snap page. Complementary function of `snap.show`. Helper function if you want to show instant loading feedback while getting SNAP_TOKEN using AJAX.

JS Callback

Transaction Result

Success credit card result

```
{
  "status_code": "200",
  "status_message": "Success, Credit Card transaction is successful",
  "transaction_id": "6d9677da-45a3-40d0-a0f0-8f0b2f860a64",
  "masked_card": "481111-1114",
  "order_id": "1459499971",
  "gross_amount": "10000.00",
  "payment_type": "credit_card",
  "transaction_time": "2016-04-01 15:39:58",
  "transaction_status": "capture",
  "fraud_status": "accept",
  "approval_code": "100057",
  "bank": "bni",
  "card_type": "credit"
}
```

Pending echannel result

```
{
  "status_code": "201",
  "status_message": "Transaksi sedang diproses",
  "transaction_id": "0ae66c29-e4a6-4e7b-b223-a103564a8d29",
  "order_id": "1459500813",
  "gross_amount": "10000.00",
  "payment_type": "echannel",
  "transaction_time": "2016-04-01 15:54:07",
  "transaction_status": "pending",
  "fraud_status": "accept",
  "bill_key": "001689",
  "biller_code": "70012",
  "pdf_url": "https://app.midtrans.com/snap/v1/transactions/0ae66c29-e4a6-4e7b-b223-a103564a8d29"
}
```

Error result

```
{
  "status_code": "406",
  "status_message": ["transaction has been processed"]
}
```

Object representing transaction result passed to [Snap callback](#).

Name	Description
status_code String	Transaction status code. Possible values: 200, 201, 202, 400, 404, 406, 500
status_message String	Transaction status message
order_id String	Merchant's payment ID
gross_amount String	Processed gross amount
payment_type String	Payment type paid by customer. Possible values: credit_card, bca_klikpay, bca_klikbca, bri_epay, mandiri_clickpay, telkomsel_cash, bank_transfer, echannel, indosat_dompdku, cstore
transaction_time String	Timestamp in yyyy-MM-dd hh:mm:ss format
transaction_status String	Transaction status. Possible values: capture, settlement, pending, cancel, expired
fraud_status String	Fraud status. Possible values: accept, challenge, deny
approval_code String	Bank approval code
masked_card String	Customer's masked card (only in credit_card & mandiri_clickpay)
bank String	Acquiring Bank
permata_va_number String	Permata VA Number (only in bank_transfer)
bill_key String	Customer bill key (only in echannel)
biller_code String	Customer biller code (only in echannel)

Name	Description
redirect_url String	Where customer should be redirected (only in bca_klikbca)
saved_token_id String	TWO_CLICKS_TOKEN value. Only available in credit_card payment type
saved_token_id_expired_at String	Specifies the expiration time of the TWO_CLICKS_TOKEN
card_type String	Type of card used. Possible values: credit, debit
pdf_url String	Link to show the payment instructions. Used for most asynchronous payment channels.

Warning!

For security reason, results from JS callback should only be used for UI feedback to user and should NOT be used to alter transaction status on your database. We provide HTTP Notification for that purpose. You can set your payment HTTP Notification URL in [Settings - Configuration](#)

Window Redirection

There is an alternative technique to load snap UI page instead of calling snap.js, this method called window redirection. Merchants can utilize the response value in `redirect_url` from the [Create Snap Token](#) request. Merchants don't have to load snap.js in their page, as the user is redirected to a page that is hosted by Midtrans.

With this redirection merchant can load snap page in full window mode view like this:

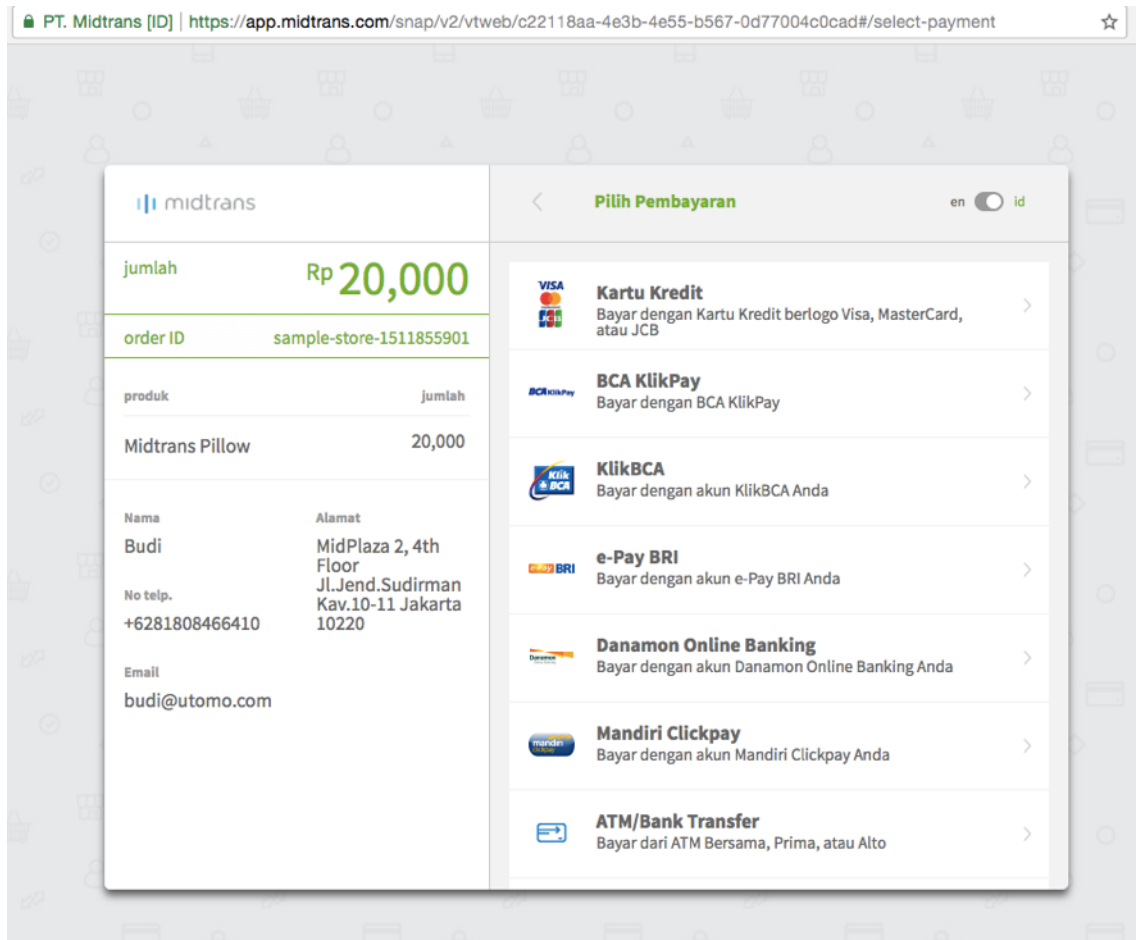


Figure 4.1: web page

Supported Payment Channels

Credit Card

Payment by credit or debit card with VISA, MasterCard, JCB, or Amex logo. We support both local and international Banks.

Note: Please also refer to the [Two Clicks](#), [One Click](#), and [Pre-Authorization](#) section

Sample JSON Request Body

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,
    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
```

```

    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
  "enabled_payments": ["credit_card"],
  "credit_card": {
    "save_card": true,
    "secure": true,
    "channel": "migs",
    "bank": "maybank",
    "installment": {
      "required": false,
      "terms": {
        "bni": [3, 6, 12],
        "mandiri": [3, 6, 12],
        "cimb": [3],
        "bca": [3, 6, 12],
        "offline": [6, 12]
      }
    }
  },
  "whitelist_bins": [
    "48111111",
    "41111111",
    "bni"
  ]
}

```

Parameter	Description
transaction_details	Unique transaction ID
Transaction Details Object (required)	
item_details	Shopping item details will be paid by customer
Item Details Object (optional)	
customer_details	Details of the customer
Customer Details Object (optional)	
enabled_payments	Set Credit Card payment method. Value: credit_card
Array (optional)	
credit_card	Credit card payment options
CreditCard (optional)	

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

BCA Virtual Account

BCA Virtual Account is a virtual payment method offered by Bank BCA. Users can pay using their BCA Bank account. Payment can be made through all of Bank BCA's channels (KlikBCA, m-BCA, and ATM).

Sample JSON Request Body

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,
    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
  "enabled_payments": ["bca_va"],
  "bca_va": {
    "va_number": "12345678911",
    "sub_company_code": "00000",
    "free_text": {
      "inquiry": [
        {
```

```

        "en": "text in English",
        "id": "text in Bahasa Indonesia"
    }
],
"payment": [
    {
        "en": "text in English",
        "id": "text in Bahasa Indonesia"
    }
]
}
}
}
}

```

Parameter	Description
transaction_details	Unique transaction ID
Transaction Details Object (required)	
item_details	Shopping item details will be paid by customer
Item Details Object (optional)	
customer_details	Details of the customer
Customer Details Object (optional)	
enabled_payments	Set BCA Virtual Account payment method. Value:
Array (optional)	bca_va
bca_va	BCA Virtual Account payment options
BCA Virtual Account (optional)	

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section

Note: Please also refer to the [Custom VA Number](#) section

Permata Virtual Account

Permata Virtual Account is a virtual payment method facilitated by Bank Permata. Users can pay using any Indonesian Bank account. Payment can be made through ATM Bersama, Prima, or Alto ATM networks.

Sample JSON Request Body

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,
    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
  "enabled_payments": ["permata_va"],
  "permata_va": {
    "va_number": "1234567890",
    "recipient_name": "SUDARSONO"
  }
}
```

Parameter	Description
transaction_details	Unique transaction ID
Transaction Details Object (required)	
item_details	Shopping item details will be paid by customer
Item Details Object (optional)	
customer_details	Details of the customer
Customer Details Object (optional)	
enabled_payments	Set Permata Virtual Account payment method. Value:
Array (optional)	permata_va

Parameter	Description
permata_va Permata Virtual Account (optional)	Permata Virtual Account payment options

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

Note: Please also refer to the [Custom VA Number](#) section

BNI Virtual Account

BNI Virtual Account is a virtual payment method facilitated by Bank BNI. Users can pay using any Indonesian Bank account. Payment can be made through ATM Bersama, Prima, or Alto ATM networks.

Sample JSON Request Body

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,
    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
}
```

```

"enabled_payments": ["bni_va"],
"bni_va": {
  "va_number": "12345678"
}
}

```

Parameter	Description
transaction_details Transaction Details Object (required)	Unique transaction ID
item_details Item Details Object (optional)	Shopping item details will be paid by customer
customer_details Customer Details Object (optional)	Details of the customer
enabled_payments Array (optional)	Set BNI Virtual Account payment method. Value: bni_va
bni_va BNI Virtual Account (optional)	BNI Virtual Account payment options

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

Note: Please also refer to the [Custom VA Number](#) section

Mandiri Bill

Mandiri Bill is a virtual payment method offered by Bank Mandiri. Users can pay using their Mandiri bank account. Payment can be made through all of Bank Mandiri's channels (Internet Banking, SMS Banking & ATM).

Sample JSON Request Body

```

{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },

```

```

"item_details": [{
  "id": "ITEM1",
  "price": 10000,
  "quantity": 1,
  "name": "Midtrans Bear",
  "brand": "Midtrans",
  "category": "Toys",
  "merchant_name": "Midtrans"
}],
"customer_details": {
  "first_name": "TEST",
  "last_name": "MIDTRANSER",
  "email": "test@midtrans.com",
  "phone": "+628123456"
},
"enabled_payments": ["echannel"]
}

```

Parameter	Description
transaction_details	Unique transaction ID
Transaction Details Object (required)	
item_details	Shopping item details will be paid by customer
Item Details Object (optional)	
customer_details	Details of the customer
Customer Details Object (optional)	
enabled_payments	Set Mandiri Bill payment method. Value: echannel
Array (optional)	

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

GoPay

GoPay is an e-Wallet payment method by GoJek. Users will pay using the GoJek app.

The user flow varies when using a Web Browser (on a computer) compared to a SmartPhone

Note: This assumes that the value of gopayMode in [snap.js options](#) is auto.

When users make a purchase using GoPay on a Web Browser(on a computer)

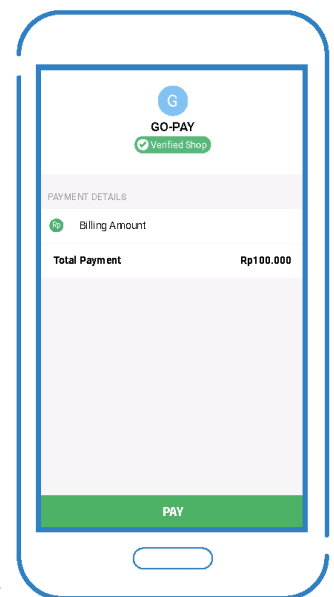
1. Users see a QR code on their Web Browser
2. Users open the GoJek app on their phone

3. Users tap the Pay function on the GoJek app





4. Users point their camera to the QR Code



5. Users check their payment details on the GoJek app and then tap Pay

6. The transaction is complete and the users' GoPay balance is deducted

When users make a purchase on their SmartPhone

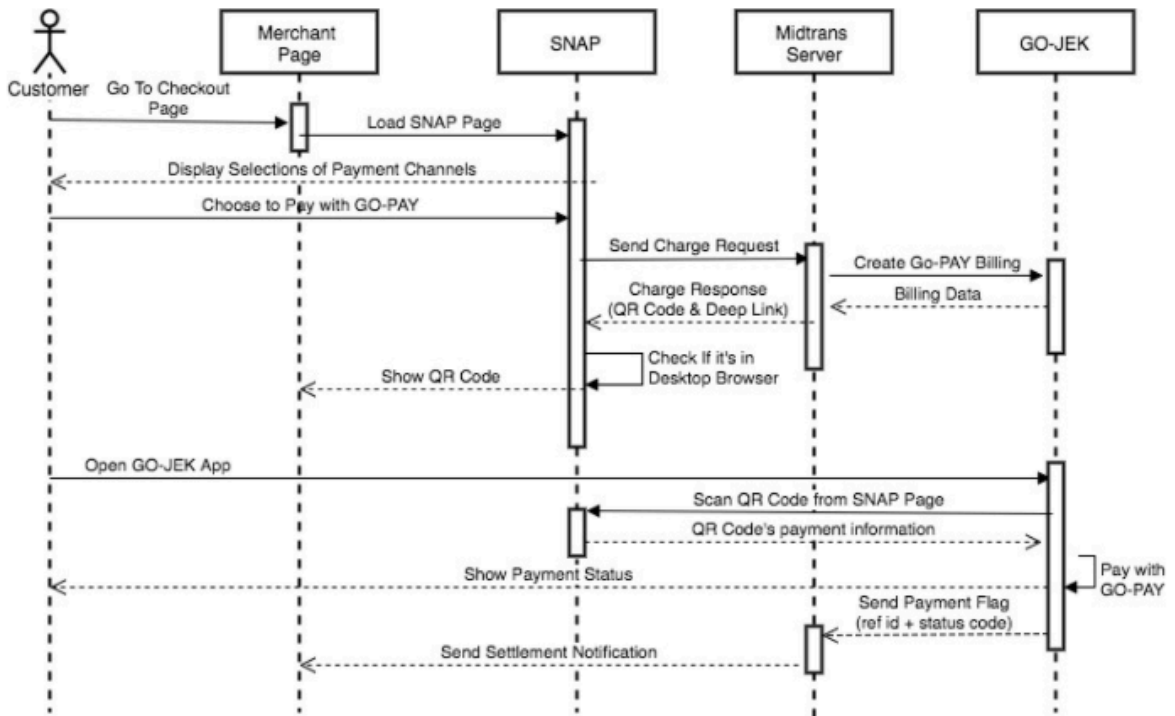


Figure 5.1: gopay table

1. Users are automatically redirected to the GoJek app when making purchases on their Smart-Phone
2. Users finish the payment on the GoJek app
3. The transaction is complete and their GoPay balance is deducted

Note: For a list of failure cases, please refer [here](#)

Detailed payment Flow for GoPay on a Desktop PC/Laptop

This is the payment flow if customers process GoPay Payments on a Desktop PC/Laptop.

Detailed Payment Flow for GoPay on a Smartphone

This is the payment flow if customers process GoPay Payments on a smartphone.

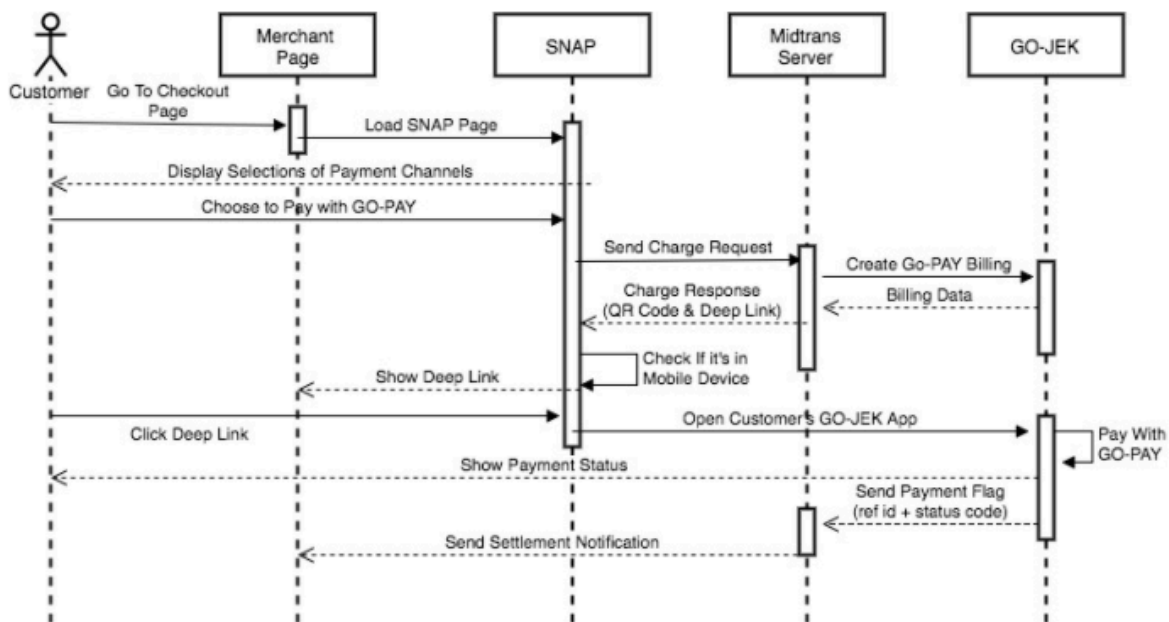


Figure 5.2: gopay table

Sample JSON Request Body

```

{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,
    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
  
```



```

    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
  "enabled_payments": [ "gopay" ],
  "gopay": {
    "enable_callback": true,
    "callback_url": "http://gopay.com"
  }
}

```

Parameter	Description
transaction_details Transaction Details Object (required)	Unique transaction ID
item_details Item Details Object (optional)	Item details will be paid by customer
customer_details Customer Details Object (optional)	Details of the customer
enabled_payments Array (optional)	Set GoPay payment method. Value: gopay
gopay Gopay (optional)	GoPay payment options

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

KlikBCA

Internet banking direct payment method by Bank BCA. User will be redirected to the KlikBCA website for payment.

Sample JSON Request Body

```

{
  "transaction_details": {

```

```

    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,
    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
  "enabled_payments": ["bca_klikbca"]
}

```

Parameter	Description
transaction_details	Unique transaction ID
Transaction Details Object (required)	
item_details	Shopping item details will be paid by customer
Item Details Object (optional)	
customer_details	Details of the customer
Customer Details Object (optional)	
enabled_payments	Set KlikBCA payment method. Value: bca_klikbca
Array (optional)	

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

BCA KlikPay

Internet banking direct payment method by Bank BCA. User will be redirected to the BCA KlikPay website for payment.

Sample JSON Request Body

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,
    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
  "enabled_payments": ["bca_klikpay"]
}
```

Parameter	Description
transaction_details	Unique transaction ID
Transaction Details Object (required)	
item_details	Shopping item details will be paid by customer
Item Details Object (optional)	
customer_details	Details of the customer
Customer Details Object (optional)	

Parameter	Description
enabled_payments Array (optional)	Set BCA KlikPay payment method. Value: bca_klikpay

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

Mandiri Clickpay

Internet banking direct payment method by Bank Mandiri. User will be redirected to the Mandiri Click-pay website for payment.

Sample JSON Request Body

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,
    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
  "enabled_payments": ["mandiri_clickpay"]
}
```

Parameter	Description
transaction_details	Unique transaction ID
Transaction Details Object (required)	
item_details	Shopping item details will be paid by customer
Item Details Object (optional)	
customer_details	Details of the customer
Customer Details Object (optional)	
enabled_payments	Set Mandiri Clickpay payment method. Value:
Array (optional)	mandiri_clickpay

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

CIMB Clicks

Internet banking direct payment method by Bank CIMB. User will be redirected to the CIMB Clicks website for payment.

Sample JSON Request Body

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,
    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
```

```

    "last_name": "MIDTRANSE",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
  "enabled_payments": ["cimb_clicks"]
}

```

Parameter	Description
transaction_details Transaction Details Object (required)	Unique transaction ID
item_details Item Details Object (optional)	Shopping item details will be paid by customer
customer_details Customer Details Object (optional)	Details of the customer
enabled_payments Array (optional)	Set CIMB Clicks payment method. Value: cimb_clicks

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

Danamon Online Banking

Internet banking direct payment method by Bank Danamon. User will be redirected to the Danamon Online Banking website for payment.

Sample JSON Request Body

```

{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,

```

```

    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
  "enabled_payments": [ "danamon_online" ]
}

```

Parameter	Description
transaction_details	Unique transaction ID
Transaction Details Object (required)	
item_details	Shopping item details will be paid by customer
Item Details Object (optional)	
customer_details	Details of the customer
Customer Details Object (optional)	
enabled_payments	Set Danamon Online Banking payment method. Value:
Array (optional)	danamon_online

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

BRI Epay

Internet banking direct payment method by Bank BRI. User will be redirected to the BRI Epay website for payment.

Sample JSON Request Body

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,
    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
  "enabled_payments": ["bri_epay"]
}
```

Parameter	Description
transaction_details	Unique transaction ID
Transaction Details Object (required)	
item_details	Shopping item details will be paid by customer
Item Details Object (optional)	
customer_details	Details of the customer
Customer Details Object (optional)	
enabled_payments	Set BRI Epay payment method. Value: bri_epay
Array (optional)	

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

Indomaret

Convenience store payment method by Indomaret. User will pay through the Indomaret physical convenience store.

Sample JSON Request Body

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,
    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
  "enabled_payments": ["indomaret"]
}
```

Parameter	Description
transaction_details	Unique transaction ID
Transaction Details Object (required)	
item_details	Shopping item details will be paid by customer
Item Details Object (optional)	
customer_details	Details of the customer
Customer Details Object (optional)	

Parameter	Description
enabled_payments Array (optional)	Set Indomaret payment method. Value: indomaret

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

Alfamart

Convenience store payment method by Alfamart. User will pay through the Alfa group convenience stores: Alfamart, Alfamidi, DAN+DAN, and Lawson (coming soon).

Sample JSON Request Body

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,
    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
  "enabled_payments": ["alfamart"],
  "cstore": {
```

```

    "alfamart_free_text_1" : "qwerty",
    "alfamart_free_text_2" : "asdfg",
    "alfamart_free_text_3" : "zxcvb"
  }
}

```

Parameter	Description
transaction_details Transaction Details Object (required)	Unique transaction ID
item_details Item Details Object (optional)	Shopping item details will be paid by customer
customer_details Customer Details Object (optional)	Details of the customer
enabled_payments Array (optional)	Set Alfamart payment method. Value: alfamart
cstore Convenience Store (optional)	Convenience store payment options

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

Akulaku

Customer financing by Akulaku. User will be redirected to the Akulaku website for payment.

Sample JSON Request Body

```

{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "item_details": [{
    "id": "ITEM1",
    "price": 10000,

```

```

    "quantity": 1,
    "name": "Midtrans Bear",
    "brand": "Midtrans",
    "category": "Toys",
    "merchant_name": "Midtrans"
  }],
  "customer_details": {
    "first_name": "TEST",
    "last_name": "MIDTRANSER",
    "email": "test@midtrans.com",
    "phone": "+628123456"
  },
  "enabled_payments": ["akulaku"]
}

```

Parameter	Description
transaction_details	Unique transaction ID
Transaction Details Object (required)	
item_details	Shopping item details will be paid by customer
Item Details Object (optional)	
customer_details	Details of the customer
Customer Details Object (optional)	
enabled_payments	Set Akulaku payment method. Value: akulaku
Array (optional)	

Note: For a full list of request body parameters please refer to the [Request Body \(JSON Parameter\)](#) section.

Other Features

Two Clicks

1. Request `/snap/v1/transactions` with `credit_card.save_card` value set to true

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "credit_card": {
    "secure": true,
    "save_card": true
  }
}
```

2. Save the `saved_token_id` provided in the `OnSuccess` callback or [HTTP Notification](#)

```
snap.pay('YOUR_SNAP_TOKEN', {
  onSuccess: function(result) {
    if (result.saved_token_id) {
      // save customer's TWO_CLICKS_TOKEN in database
    }
  }
})
```

3. Request `/snap/v1/transactions` with `credit_card.card_token` value set to

TWO_CLICKS_TOKEN value saved during initial transaction

```
{
  "transaction_details": {
    "order_id": "ORDER-102",
    "gross_amount": 10000
  },
  "credit_card": {
    "secure": true,
    "card_token": "TWO_CLICKS_TOKEN"
  }
}
```

Two clicks feature allows you to capture customer's card number, expiry date, email and phone number as a TWO_CLICKS_TOKEN. For successive payments by the same customer the TWO_CLICKS_TOKEN can be utilized to pre-fill the details. Customer just needs to fill out the cvv number to finish the payment.

There are two steps required to utilize two clicks feature:

Initial Transaction

- Create SNAP_TOKEN with `credit_card.save_card` value set to `true`
- Customer will be presented with save option when they fill out the credit card details
- If customer chooses the save option, then the [Transaction Result](#) will have two extra fields namely `saved_token_id` and `saved_token_id_expired_at`

Successive Transactions

- Create SNAP_TOKEN with `credit_card.card_token` value set to TWO_CLICKS_TOKEN saved during the initial transaction
- Customer will be presented with pre-filled credit card number, expiry date, email and phone number

One Click

Request SNAP_TOKEN with `user_id` and `credit_card.save_card` value set to `true`

```
{
  "transaction_details": {
    "order_id": "ORDER-101",
    "gross_amount": 10000
  },
  "credit_card": {
    "secure": true,
    "save_card": true
  },
  "user_id": "customer-nR6DC0zqGis"
}
```

In addition to two clicks feature, Snap also supports one click transaction which will also capture card cvv. With this, customer can directly proceed to charge without input any information.

To ease card saving process, Snap provides card token storage feature. So, merchants don't have to store and manage credit card token by themselves. Merchants can easily integrate credit card token storage feature to Snap by providing unique `user_id` that associate with customer account on merchant's system, in addition to enabling `credit_card.save_card` flag.

Snap will then decide to store credit card token as one click token based on two criteria:

1. Merchant has recurring MID enabled
2. Initial transaction is 3DS-enabled

If those two conditions are met, credit card token will be saved as one click token. Otherwise, it will be saved as two clicks token.

Similar to two clicks flow, there are two steps required to utilize credit card token storage feature on Snap:

Initial Transaction

- Create `SNAP_TOKEN` with `user_id` and `credit_card.save_card` set to true
- `user_id` is a unique identifier that represents a customer. Its value can be any string with a maximum length of 255. Avoid using values that could be easily linked to a customer like email, id of records in database etc. Instead use something like `sha512(email/record.id + salt)`. Don't expose the `user_id` to frontends like browser or mobile.
- Customer will be presented with save option when they fill out the credit card details
- If customer chooses the save option, Snap will store credit card token and associate it with `user_id`

- If you want to enable one click feature, make sure to activate recurring MID and set `credit_card.secure` to true. Otherwise, credit card token will be saved as two clicks.

Successive Transaction

- Create `SNAP_TOKEN` with the same `user_id` as in initial transaction
- It's not necessary to have `credit_card.save_card` flag set to true on successive transaction
- Customer can choose between multiple saved cards if they have saved some cards before
- If selected card was saved as one click token, customer will be presented with pre-filled credit card number, expiry date, cvv, email and phone number. Then, customer can directly proceed to charge (this process will bypass 3DS).
- If selected card was saved as two clicks token, customer will need to enter cvv and going through 3DS process when necessary (same as [two clicks flow](#))

Custom Virtual Account Number

bca_va

```
{
  "bca_va": {
    "va_number": "5555555555"
  }
}
```

permata_va

```
{
  "permata_va": {
    "va_number": "1234567890"
  }
}
```

bni_va

```
{
  "bni_va": {
    "va_number": "12345678"
  }
}
```


A virtual account number contains two parts. for example, in {91012}{12435678} , the first part is the company code and the second part is a unique code. The second part can be customized for [bca_va](#), [permata_va](#) and [bni_va](#) payment types.

- Only digits are allowed.
- Different banks have different specs on their custom VA numbers. Please see the documentation on the respective banks.
- If the number provided is already utilized for another order, then a different unique number will be used instead.
- If the number provided is longer than required, then the unnecessary digits in the end will be trimmed.
- If the number provided is shorter than required, then the number will be prefixed with zeros.

Pre-Authorization

Request `/snap/v1/transactions` with `credit_card.type` value set to “authorize”

```
{
  "credit_card": {
    "type": "authorize"
  }
}
```

Pre-authorization feature allows you to check and block customer’s card limit. To use this feature, an additional parameter will be required in credit card object:

Parameter	Description
<code>credit_card.type</code> String (optional)	Credit card transaction type. Options: <code>authorize</code> , <code>authorize_capture</code> . Default: “ <code>authorize_capture</code> ”

Transaction Status

Midtrans provides three means for the merchant to obtain the transaction status.

- Email Notification
- HTTP Notification
- Merchant Administration Portal

Email Notification

You can configure the email notification setting at [Settings - Email Notification](#) in MAP.

Info!

For multiple emails, please use comma as a separator. Example : Andri@myemail.com, litani@myemail.com


HTTP(S) POST Notification

Notification through HTTP(S) POST will be sent to the merchant's server when customer completes the payment process. You can utilize the HTTP(S) POST notification to update a payment status or send the item of a transaction in real time.

Enable Midtrans HTTP(S) POST Notification by setting the Payment Notification URL at [Settings - Configuration](#).

Warning!

Make sure your Payment Notification URL is not the same as your localhost address.



Settings

General Settings

Access Keys

VT-Web Preferences

Configuration

Payment Methods

Billing

Email Notifications

VT-Link

Daily Report

Toko Buah eCommerce

Settings

Email Notifications

Send email to customer

☒

Your customers will receive email notification about transaction

Send email to me

☒

You will receive email notification about every transaction.

andri@myemail.com, litani@myemail.com

This email will be used to receive a notification each time a transaction is done, regardless success or fail. If you have multiple emails, please use (,) / comma to separate emails

Email Support

andri@myemail.com

This email will be used as an email sender in email notification to your customer.

Email Development

litani@myemail.com

This email will be used to receive a notification if there's a validation error.

Email Language

Bahasa

Customer service phone

09922334455

Save Preferences

veritrans

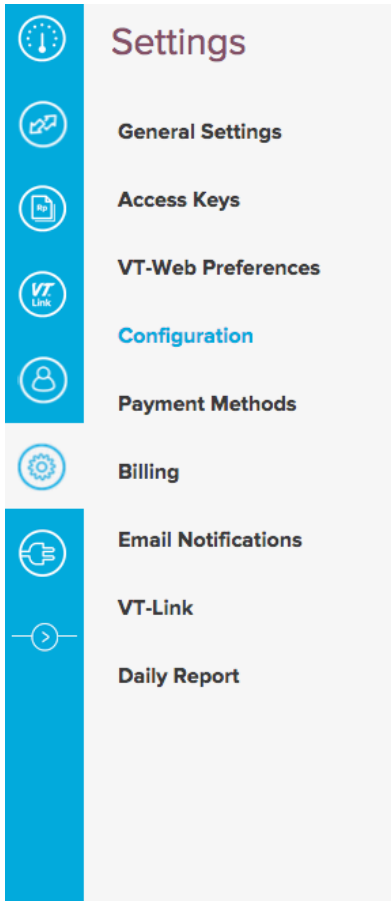
© 2012 - 2015 Veritrans Indonesia. All Rights Reserved

ENG / IND

Hot keys

Figure 7.1: email notification

74



Settings

Configuration

Veritrans requires the URL endpoints for the following scenarios:

Payment Notification URL*	https://sneekers.com/payment/notification Address where we will send the notification via HTTP Post request. E.g http://yourwebsite.com/notification/handling Send test notification See history
----------------------------------	--

Finish Redirect URL*	https://sneekers.com/payment/finish Customer sent here if payment is successful e.g http://yourwebsite.com/payment/finish
-----------------------------	---

Unfinish Redirect URL*	https://sneekers.com/payment/unfinish Customer sent here if they click on 'Back to Order Website' on VT-Web's payment page e.g http://yourwebsite.com/payment/unfinish
-------------------------------	--

Error Redirect URL*	https://sneekers.com/payment/error Customer sent here if payment is unsuccessful or an error occurs during payment e.g http://yourwebsite.com/payment/error
----------------------------	---

Figure 7.2: http notification

Toko eCommerce		Sandbox			
Bank Transfer	Today, 12:00	VTVA001	andri@litani.com	145,000	
Credit Card	Today, 10:59	VTCC01	andri@litani.com	145,000	success
Mandiri Bill	22/10/2015 12:09	vtmanecash02	andri@litani.com	10,000	cancel
Credit Card	16/10/2015 16:16	vt1click2	andri@litani.com	10,000	settlement
Credit Card	16/10/2015 16:15	vt1click01	andri@litani.com	10,000	settlement

Figure 7.3: map transaction

Refer [here](#) for sample json notification for different payment channels.

Merchant Admin Portal(MAP) Transaction Status

Merchant can check the transactions status from [payments](#) menu on MAP.

Example of a successful transaction status for credit card:

Handling Notifications

Signature key logic

```
SHA512(order_id + status_code + gross_amount + serverkey)
```

Sample code generate signature key

```
<?php
$orderId = "1111";
$statusCode = "200";
$grossAmount = "100000.00";
$serverKey = "askvnoibnosifnboseofinbofinfgbiufglnbfg";
$input = $orderId.$statusCode.$grossAmount.$serverKey;
$signature = openssl_digest($input, 'sha512');
echo "INPUT: " , $input."<br/>";
echo "SIGNATURE: " , $signature;
?>
```

In order to increase security aspect, there are several ways to ensure notification received from Mid-trans.

Challenge Response

An additional mechanism we provide to verify the content and the origin of the notification is to challenge. This can be achieved by calling the [get status](#) API. The response is the same as the notification status.

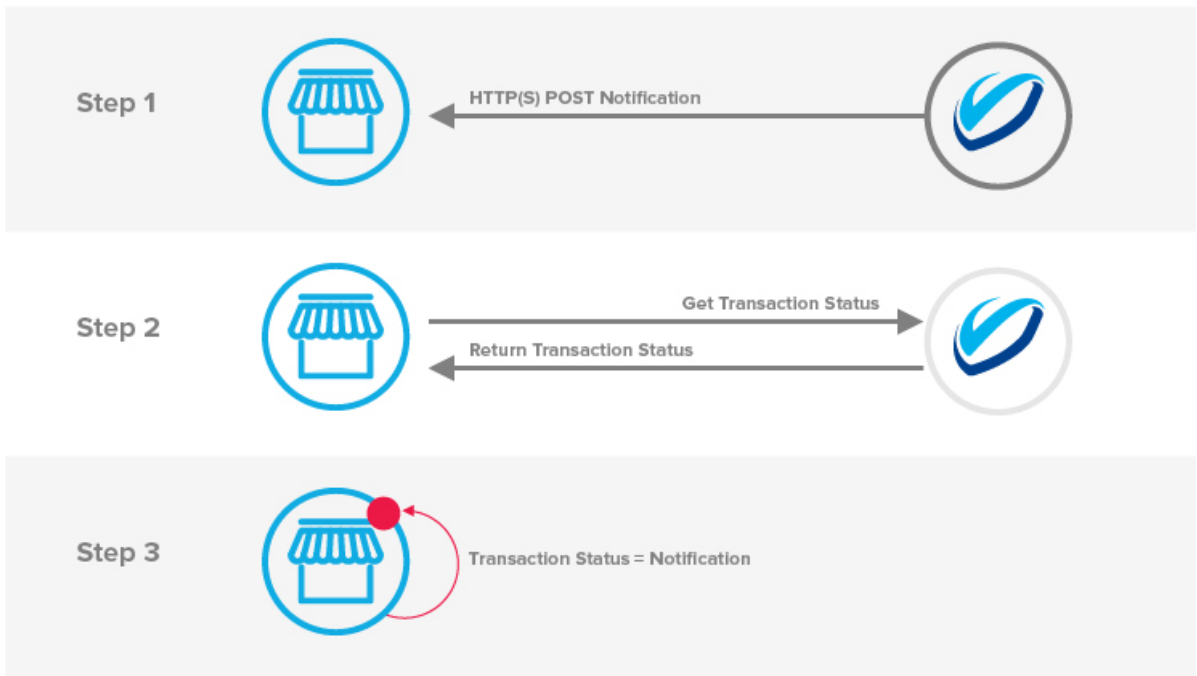


Figure 8.1: challenge response

Signature Key

We add signature key information in our notification. The purpose of this signature key is to validate whether the notification is originated from Midtrans or not. Should the notification is not genuine, merchants can disregard the notification. Please find on the side, the logic of the signature key and the sample code to generate signature key.

Best Practice to Handle notification

- Always use an HTTPS endpoint. It is secure and there cannot be MITM attacks because we validate the certificates match the hosts. Also do not use self signed certificates.
- Always implement notification in an idempotent way, in extremely rare cases, we may send multiple notifications for the same transaction event twice. It should not cause double entries in the merchant end, The simple way of achieving this is to use orderid as the key to track the entries.
- Always check the signature hash of the notification, This will confirm that the notification was actually sent by Midtrans, because we encode the shared secret (server key). Nobody else can build this signature hash.

- Always check all the following three fields for confirming success transaction
 - status code: Should be 200 for successful transactions
 - fraud status: ACCEPT
 - transaction status : settlement/capture
- We strive to send the notification immediately after the transaction has occurred, but in extremely rare cases, it may be delayed because of transaction spikes. If you have not received a notification, please use the [Status API](#) to check for current status of the transaction.
- It is safe to call [Status API](#) to get the latest status of the transaction/order on each notification.
- We set the HTTP timeout to 30 seconds.
Please strive to keep the response time of the the HTTP notifications under 5 seconds.
- In extremely rare cases we may send the HTTP notifications out of order, ie. a “settlement” status for a notification before the notification for “Pending” status. It’s important that such later notifications are ignored. Here’s the state transition diagram that you could use. But again, use our /status API to confirm the actual status.
- We send the notification body as JSON, please parse the JSON with a JSON parser. Always expect new fields will be added to the notification body, so parse it in a non strict format, so if the parser sees new fields, it should not throw exception. It should gracefully ignore the new fields. This allows us to extend our notification system for newer use cases without breaking old clients.
- Always use the right [HTTP Status code](#) for responding to the notification, we handle retry for error cases differently based on the status code
 - for 2xx: No retries, it is considered success
 - for 500: We will retry 1 times in the 1 minute interval
 - for 503: Retry 4 times
 - for 400/404: retry 2 times in 1 minute interval
 - for all other failures: Retry 5 times at 1 minute interval
- Redirection
 - for 307/308: The request will be repeated with the new URL using POST method and the same notification body. A maximum of 5 redirection is allowed.
 - for 301/301/303: The job will be marked as failed without further retries. The merchant will be notified via email. We suggest either to use 307/308 or update the notification endpoint settings in merchant portal.

Following are the standard types of notifications. Note different types of notifications can be added in addition to the below. Also new fields may be added to the existing notification, please confirm with the latest documentation for the exact fields.

Override Notification Url

Merchant can opt to change or add custom notification urls on every transaction. It can be achieved by adding additional HTTP headers into charge request.

There are two headers we provide:

- X-Append-Notification: to add new notification url(s) alongside the settings on dashboard
- X-Override-Notification: to use new notification url(s) disregarding the settings on dashboard

Both header can only receive up to maximum of 2 urls.

Sample case

Append Notification Sample

```
curl -X POST \
  https://app.midtrans.com/snap/v1/transactions \
  -H 'Accept: application/json' \
  -H 'Authorization: Basic NjJiMTVhN2QtZTVmOC00YjNjLTl1YWItY2E4MjdjYTM3ZjU1Og==' \
  -H 'Content-Type: application/json' \
  -H 'X-Append-Notification: https://example.com/test1,https://example.com/test2' \
  -H 'cache-control: no-cache' \
  -d '{
    "transaction_details": {
      "order_id": "1553229166",
      "gross_amount": 10000
    }
  }'
```

Override Notification Sample

```
curl -X POST \
  https://app.midtrans.com/snap/v1/transactions \
  -H 'Accept: application/json' \
  -H 'Authorization: Basic NjJiMTVhN2QtZTVmOC00YjNjLTl1YWItY2E4MjdjYTM3ZjU1Og==' \
  -H 'Content-Type: application/json' \
  -H 'X-Override-Notification: https://example.com/test1,https://example.com/test2' \
  -H 'cache-control: no-cache' \
  -d '{
```

```
"transaction_details": {  
  "order_id": "1553229166",  
  "gross_amount": 10000  
}  
'
```

Assuming merchant has set <https://example.com> as their notification url on the dashboard.

If they set `X-Append-Notification` with values <https://example.com/test1>, <https://example.com/test2>,

Then, every HTTP notification for that specific transaction will be sent to:

- <https://example.com>,
- <https://example.com/test1>, and
- <https://example.com/test2>

Else if they set `X-Override-Notification` with values <https://example.com/test1>, <https://example.com/test2>,

Then, every HTTP notification for that specific transaction will be sent to:

- <https://example.com/test1> and
- <https://example.com/test2>

When both `X-Append-Notification` and `X-Override-Notification` are used together then only override will be used.

Status Code

Goal: Understand all status codes used by API. For more inquiries, please contact us at support@midtrans.com or visit our support web page.

Status Codes used by Midtrans API are categorized into 2xx, 3xx, 4xx dan 5xx.

Code 2xx

Status	Description
200	Credit Card: Success. Request is successful, and transaction is successful (authorize, capture, settlement, cancel, get order, approve challenge transactions), accepted by Midtrans and bank. Other payment methods: Success. Transaction is successful/settlement.

Status	Description
201	<p>Credit Card: Pending. Transaction is successfully made but the 3D secure process has yet to be completed. Transaction will expire within 15 minutes.</p> <p>Challenge. Transaction is successfully sent to bank but the process has not been completed, need manual action from merchant to complete the transaction process. If the merchant does not perform any action until settlement time (H+1 16:00) Midtrans will cancel the transaction.</p> <p>Bank Transfer: Pending. Transaction is successfully sent to bank but the process has not been completed by the customer. By default the transaction will expire within 24 hours.</p> <p>Cimb Clicks: Pending. Transaction successfully sent to bank but the process has not been completed by the customer. By default the transaction will expire within 2 hours.</p> <p>BRI ePay: Pending. Transaction successfully sent to bank but the process has not been completed by the customer. By default the transaction will expire within 2 hours.</p> <p>Klik BCA: Pending. Transaction successfully sent to bank but the process has not been completed by the customer. By default the transaction will expire within 2 hours.</p> <p>BCA Klikpay: Pending. Transaction successfully sent to bank but the process has not been completed by the customer. By default the transaction will expire within 2 hours.</p> <p>Mandiri Bill Payment: Pending. Transaction successfully sent to bank but the process has not been completed by the customer. By default the transaction will expire within 2 hours.</p> <p>Indomaret: Pending. Transaction successfully sent to provider but the process has not been completed by the customer. By default the transaction will expire within 2 hours.</p>

Status	Description
202	Credit Card: Denied. Transaction has been processed but is denied by payment provider or Midtrans' fraud detection system. Other payment methods: Denied. Transaction has been processed but is denied by payment provider.

Code 3xx

Status	Description
300	Move Permanently, current and all future requests should be directed to the new URL

Code 4xx

Status	Description
400	Validation Error, merchant sends bad request data example; validation error, invalid transaction type, invalid credit card format, etc.
401	Access denied due to unauthorized transaction, please check client key or server key
402	Merchant doesn't have access for this payment type
403	The requested resource is only capable of generating content not acceptable according to the accepting headers that sent in the request
404	The requested resource is not found
405	HTTP method is not allowed
406	Duplicate order ID. Order ID has already been utilized previously
407	Expired transaction
408	Merchant sends the wrong data type
409	Merchant has sent too many transactions for the same card number

Status	Description
410	Merchant account is deactivated. Please contact Midtrans support
411	Token id is missing, invalid, or timed out
412	Merchant cannot modify status of the transaction
413	The request cannot be processed due to malformed syntax in the request body

Code 5xx

Status	Description
500	Internal Server Error
501	The feature has not finished yet, it will be available soon
502	Internal Server Error: Bank Connection Problem
503	Internal Server Error
504	Internal Server Error: Fraud detection is unavailable

Testing Credentials

Here is a list of dummy transaction credentials that can be used for transaction in the Sandbox Environment.

Credit Card

Please refer to [Testing Card](#).

Bank Transfer

Payment Methods	Description
Permata Virtual Account	Midtrans will generate a dummy 16 digits Permata Virtual Account Number. To perform a test transaction, use the Permata Virtual Account Simulator
BCA Virtual Account	Midtrans will generate a dummy 11 digits BCA Virtual Account Number. To perform a test transaction, use the BCA Virtual Account Simulator
Mandiri Bill Payment	Midtrans will generate a dummy 12 digits Bill Key to complete payment via Mandiri e-channel (Internet Banking, SMS Banking, Mandiri ATM). To perform a test transaction, use Mandiri Bill Payment Simulator

Payment Methods	Description
BNI Virtual Account	Midtrans will generate a dummy BNI Virtual Account Number. To perform a test transaction, use the BNI Virtual Account Simulator

Direct Debit

Payment Methods	Description
Mandiri Clickpay	Card Number: 4111 1111 1111 1111 Accept Token: 000000 Deny Token: 111111
CIMB Clicks	Midtrans will redirect CIMB Clicks test transaction to a payment simulator. Success Transaction: testuser00 Failure Transaction: testuser01
ePay BRI	Midtrans will redirect ePay BRI test transaction to a payment simulator. Success Transaction: testuser00 Failure Transaction: testuser03
BCA Klikpay	Midtrans will redirect BCA Klikpay test transaction to a payment simulator.
KlikBCA	Midtrans will register user id filled in KlikBCA input. To perform a test transaction, use the KlikBca Simulator

e-Wallet

Payment Methods	Description
Telkomsel Cash	Accept Customer: 0811111111 Deny Customer: 0822222222
Indosat Dompetku	Accept number: 08123456789 Deny number: other than 08123456789
Mandiri E-cash	Accept number: 0987654321 PIN: 12345 OTP: 12123434

Convenience Store

Payment Methods	Description
Indomaret	Midtrans will generate a dummy Indomaret Payment Code. To perform a test transaction, use the Indomaret Simulator
Kioson	Midtrans will generate a dummy Kioson Payment Code. To perform a test transaction, use the Kioson Simulator

Going Live

The tutorial below will assist you in changing the configuration from Sandbox to Production. Please choose the following integration method. If further assistance is required, please contact us at support@midtrans.com or visit our [support page](#).

If you have completed integrating in Sandbox (Testing Environment) and ready to go live in Production Environment, there are a few steps that need to be made:

- Contact Midtrans Support to migrate from Sandbox to Production Environment. Once confirmed, you can now login to Midtrans Production Dashboard at <https://account.midtrans.com/login> with the current credentials.
- Change setting environment in Midtrans plugin from development into production. Here are the list of our lib and plugin setting environment.

Minimum Transaction Amount

Each transaction requires a minimum gross_amount of 10000.

PHP

```
<?php
include_once('Veritrans.php');

// Set our production server key
Veritrans_Config::$serverKey = '<your production server key>';
```

```
// Use production account
Veritrans_Config::$isProduction = true;
...
?>
```

Set Veritrans-PHP library to Production Environment. This can be done by changing the value `Veritrans_Config::$serverKey` with production server key and `Veritrans_Config::$isProduction` into `true`.

Magento

Set magento plugin settings with production client key and server key and change environment to production.

v1.9.x

v2.x.x

OpenCart

Set Midtrans' OpenCart plugin settings with production client key and server key and change environment to production.

Prestashop

Set Midtrans' Prestashop plugin environment settings to production, input your production client key, and production server key.

Woocommerce

Set Midtrans' Woocommerce plugin environment settings to production, input your production client key, and production server key.

Midtrans ver 1.0.0		
Enabled	Yes	[WEBSITE]
Title	Snap	[STORE VIEW]
New Order Status	Pending	[WEBSITE]
Conversion Rate	13000	[STORE VIEW]
	▲ This conversion rate will be used if the IDR currency is not installed in your system.	
Client Key	<<your client key>>	[STORE VIEW]
Server Key	<<your server key>>	[STORE VIEW]
Environment	Production	[STORE VIEW]
	▲ Change to 'Production' once your store is going live	
Enable 3D Secure	Yes	[STORE VIEW]
	▲ You must enable 3D Secure. Please contact us if you wish to disable this feature in the Production environment.	
Enable One Click	Yes	[STORE VIEW]
	▲ You must have oneclick MID to enable oneclick transaction, otherwise transaction will be 2 clicks. Please contact us if you wish to enable this feature in the Production environment.	
Expiry Duration	120,minute	[STORE VIEW]
	▲ Filled with duration,unit. Unit:minute,hour,day. example: 2,minute	

Figure 11.1: magento set production

Midtrans Snap



Enabled [website]	Yes	▼
Client Key [store view]	<<your client key>>	
Server Key [store view]	<<your server key>>	
Title [store view]	Midtrans Snap	
Is Production [website]	Yes	▼
New Order Status [website]	Pending	▼
Enable One click [website]	Yes	▼
Enable 3D Secure [website]	Yes	▼
Custom Expiry [website]	120 minute	

Figure 11.2: magento set production



 Configure Midtrans Payment Gateway	
Status:	<div>Enabled</div>
* Display name:	<div>snap</div>
* Environment	<div>Production</div>
* Client Key	<div><<your client key>></div>
* Server Key	<div><<your server key>></div>
* One click:	<div>Enabled</div>
* Custom Expiry Duration	<div>filled with number</div> <div>minutes</div>

Figure 11.3: magento set production

Configure

Midtrans Pay



This module allows you to accept secure payments by Midtrans.
Please consult your Midtrans Merchant Administration Portal for settings.

BASIC INFORMATION

*** Payment Option Display Text**

Customize payment option title that will be displayed to your customer when they click on the payment option.

Payment Option Description Text

Customize payment option description

*** Environment**

Development

✓ Production

*** Client Key**

Consult to your Merchant Administration Portal for the value of this field.

*** Server Key**

Consult to your Merchant Administration Portal for the value of this field.

*** Enable 3D Secure?**

YES

NO

You must enable 3D Secure. Please contact us if you wish to disable this feature in the future.

Figure 11.4: magento set production

Midtrans

Allows payments using Midtrans.

Enable/Disable

☒ Enable Midtrans Payment

Title

?

Customer Message

This controls the description which the user sees during checkout

Environment

Sandbox

☒ Production

Select the Midtrans Environment

Client Key

*Input your **Production** Midtrans Client Key. Get the key [here](#)*

Server Key

*Input your **Production** Midtrans Server Key. Get the key [here](#)*

Enable 3D Secure

☒ Enable 3D Secure?

You must enable 3D Secure. Please contact us if you wish to disable this feature in the

Figure 11.5: magento set production