



API documentation

November 27, 2017

by midtrans

Contents

1	Langkah Awal	4
	Gambaran Umum	4
	Browser yang didukung	5
	Mobile	6
	Desktop	6
	Persiapan Awal	6
	1. Daftar ke akun Sandbox/Production Midtrans	6
	2. Lengkapi informasi yang diperlukan di Merchant Admin Portal (MAP)	7
	3. Catat Access Keys akun Anda	8
	4. Konfigurasi Pengalihan URL	9
	Library dan Plugin	11
	Libraries	11
	Plugin	12
2	Integrasi Backend	13
	Endpoint	13
	Request Headers	13
	JSON Parameter (Request Body)	14
	JSON Object	18
	Response	24
	Request Berhasil	24
	Request Gagal	24
3	Integrasi Frontend	26
	Gambaran Umum	26
	Lokasi Snap.js	27
	Viewport Meta Tag	27
	Snap JS	27
	pay(snapToken, options)	27

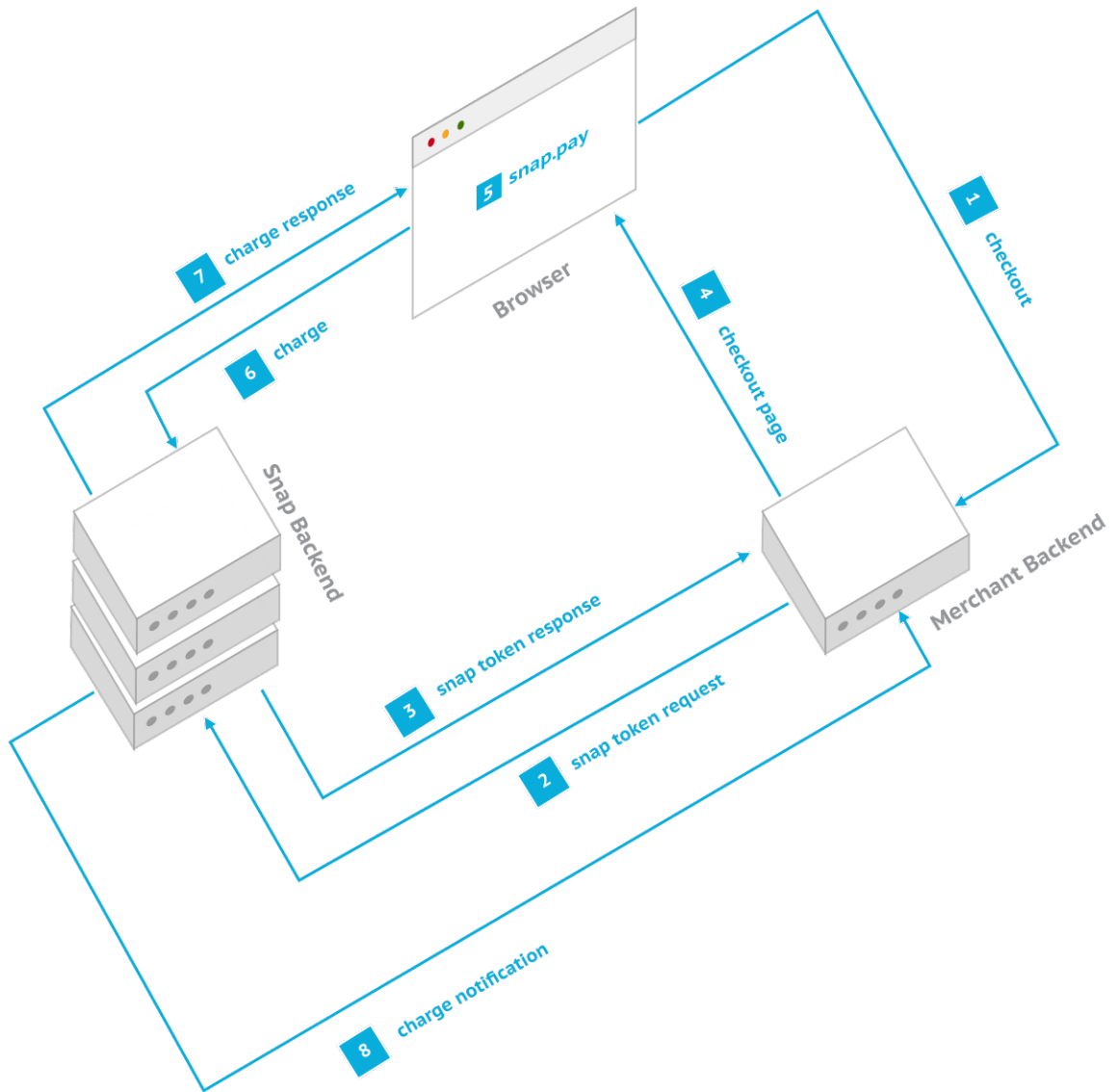
show()	29
hide()	29
JS Callback	29
Transaction Result	29
4 Fitur Lainnya	33
Two Clicks	33
Transaksi Awal	34
Transaksi Selanjutnya	34
One Click	34
Transaksi Awal	35
Transaksi Selanjutnya	36
Nomor Virtual Account Kustom	36
Pre-Authorization	37
5 Status Transaksi	38
Notifikasi email	38
Notifikasi HTTP(S) POST	39
Merchant Admin Portal (MAP) Status Transaksi	40
Membandingkan Response	41
Signature Key	42
Best Practice Menggunakan Notifikasi	42
6 Status Kode	44
Kode 2xx	44
Kode 3xx	46
Kode 4xx	46
Kode 5xx	47
7 Kredensial Testing Pembayaran	48
Kartu Kredit	48
Kartu Umum	48
Kartu Spesifik per Bank	49
Expiry Date dan CVV	50
Bank Transfer	51
Direct Debit	51
e-Wallet	52
Convenience Store	52
8 Going Live	53
Nilai Transaksi Minimum	53

PHP	53
Magento	54
OpenCart	56
Prestashop	57
Woocommerce	58

Langkah Awal

Gambaran Umum

1. User melakukan pembayaran
2. Server Merchant melakukan [api request](#) ke server Snap berisikan data transaksi
3. Server Snap menyimpan data transaksi dan memberikan SNAP_TOKEN
4. Server Merchant menampilkan [halaman](#) ringkasan pembelian dengan SNAP_TOKEN di browser
5. User memverifikasi rincian pembelian dan mengklik tombol bayar. Merchant javascript mengeksekusi [snap.pay \(SNAP_TOKEN, opsi\)](#). Pengguna kemudian mengisi data pembayaran dan mengklik tombol konfirmasi.
6. Snap JS mengirimkan data pembayaran ke server Snap
7. Server Snap memproses data dan meresponse dengan status pembayaran. Snap JS kemudian mengeksekusi [callback](#) sesuai dengan status transaksi
8. Server Snap akan memberikan notifikasi ke server Merchant tentang status pembayaran



payment flow

Browser yang didukung

Kami mencoba untuk mendukung versi terbaru dari semua browser utama. Berikut adalah versi minimum yang didukung untuk masing-masing browser di perangkat Mobile dan Desktop:

Mobile

- Chrome **32** ke atas
- Android **4.4** ke atas
- Safari **8** ke atas
- BlackBerry **10** ke atas

Desktop

- Chrome **26** ke atas
- Firefox **29** ke atas
- Internet Explorer **10** ke atas
- Safari **6** ke atas

Persiapan Awal


Tujuan: Merchants melengkapi syarat yang diperlukan sebelum berintegrasi dengan midtrans

Ada beberapa syarat yang harus dilengkapi sebelum berintegrasi dengan midtrans:

1. Daftar ke akun Sandbox/Production Midtrans

Saat ini midtrans memiliki satu akun login untuk mengakses akun sandbox dan production. Akun sandbox digunakan pada periode development/testing sedangkan akun production digunakan ketika merchant sudah selesai melakukan integrasi dan siap untuk go live. Data dan transaksi yang dibuat di akun sandbox tidak akan memicu terjadinya pembelian sungguhan tetapi pada akun production akan mengakibatkan transaksi sebenarnya. Silakan melakukan pendaftaran akun midtrans [di sini](#).

Setelah masuk, akan ada tombol kecil di bagian atas dari dashboard yang menampilkan anda sedang di dalam mode sandbox atau production. Warna pada menu navigasi samping juga dibedakan antara production (biru muda) dan sandbox (biru tua) untuk lebih memperjelas.



Dashboard

Transactions

Billings

Account

Settings

Toko eCommerce

Production

Signed in successfully

Summary

0


Total Volume
Month to Date

0

Total Transaction
Month to Date

Transaction Volume

14 Sep - 21 Sep 2015



Dashboard

Transactions

Billings

VT-Link

Account

Settings

Integrations

Toko eCommerce

Sandbox

Switch Environment

Production

Sandbox

Summary

0

Total Volume
Month to Date

0

Total Transaction
Month to Date

114,960,469

Payable Amount
Volume to Date

Transaction Volume

14 Sep - 21 Sep 2015

2. Lengkapi informasi yang diperlukan di Merchant Admin Portal (MAP)

Kolom yang perlu diisi bisa dilihat di [Settings - General Settings](#).

7

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

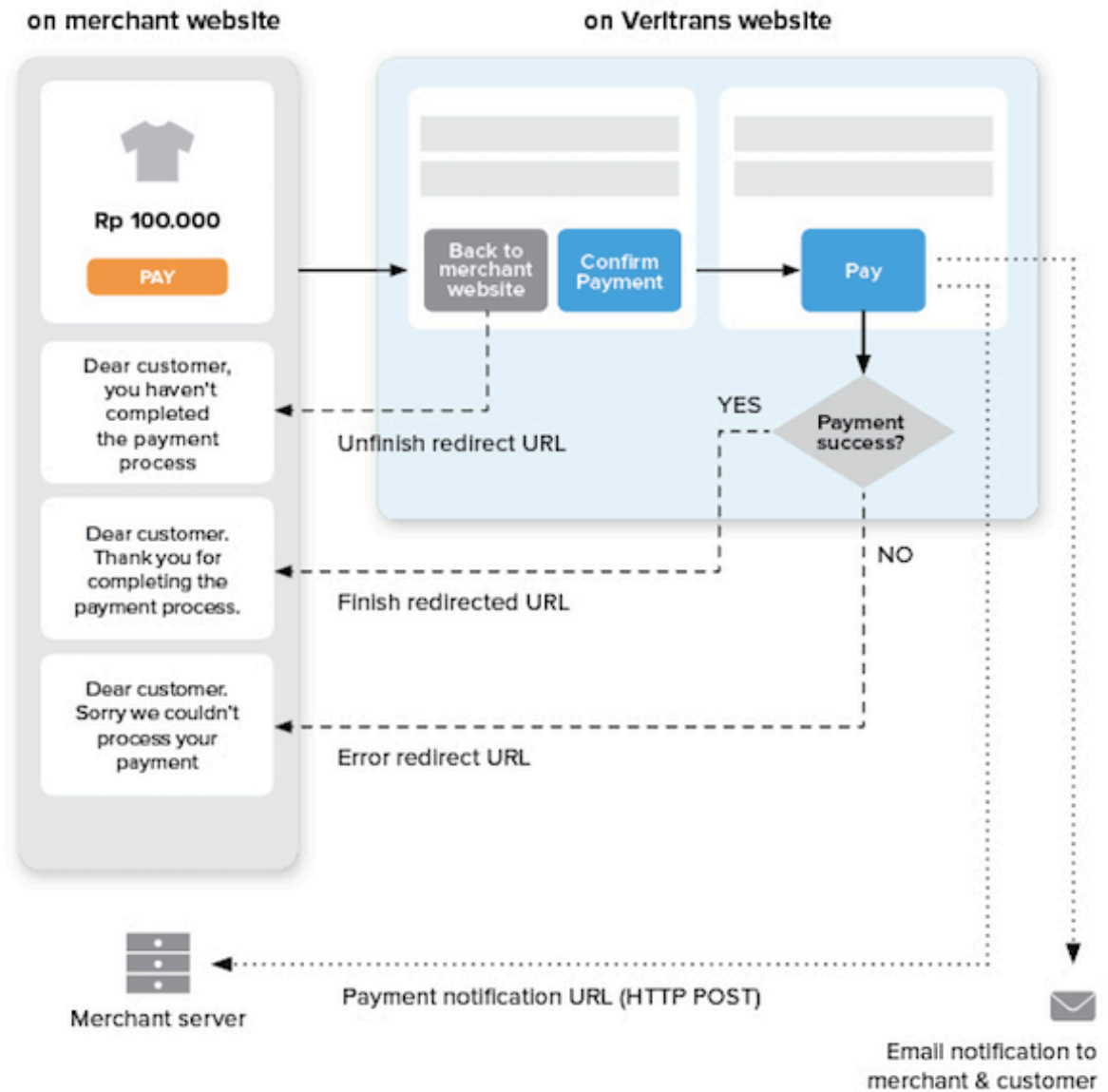
general settings

Peringatan!

Merchant Name: JANGAN masukkan simbol. Merchant URL yang Valid:

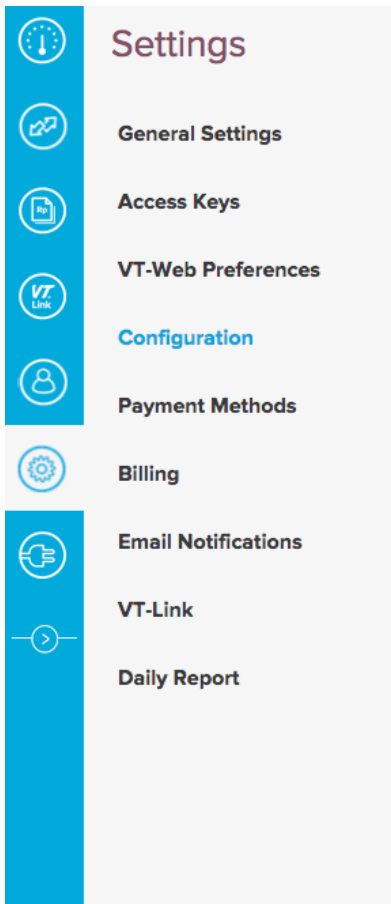
3. Catat Access Keys akun Anda

Access Keys akun Anda dapat dilihat di [Settings - Access Keys](#).



redirection url

Masuk ke [Settings - Configuration](#) menu untuk mengatur pengalihan 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

map redirection url

Peringatan!

Mohon lengkapi URL dengan <http://> atau <https://>

Library dan Plugin

Kami berusaha membuat proses integrasi menjadi semudah mungkin untuk anda. Bagian ini berisi daftar plugin dan library midtrans, jika anda memiliki plugin atau library sendiri dan ingin terhubung dengan plugin dan library kami, hubungi 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 ke atas Github
Wordpress	Wordpress v3.9.1 - v4.x
Woocommerce	WooCommerce v2.1.11 - v2.5.x Github

Integrasi Backend

Tujuan dari integrasi backend adalah memperoleh SNAP_TOKEN dengan mengirimkan informasi pembayaran. Kami menyediakan HTTP API untuk melakukan hal ini.

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 = "VlQtc2VydjVybWVzAza1lET2MwY05VS2d0NmhuTGtLZzo="
```

Untuk melakukan HTTP request yang valid, merchant harus menambahkan 3 *header*:

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

Snap melakukan validasi HTTP request menggunakan Basic Authentication. Pada kasus ini, *username* Anda adalah SERVER_KEY dengan *password* kosong. Anda dapat mengetahui SERVER-KEY Anda dengan melihat di [Settings - Access Keys](#).

Nilai dari header Authorization direpresentasikan sebagai AUTH_STRING. AUTH_STRING adalah base-64 dari *username* & *password* Anda.

AUTH_STRING = Base64(SERVER_KEY + :)

JSON Parameter (Request Body)

Request sederhana

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

Request lengkap

```
{
  "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"
  }],
  "enabled_payments": ["credit_card", "mandiri_clickpay", "cimb_clicks",
    "bca_klikbca", "bca_klikpay", "bri_epay", "telkomsel_cash", "echannel",
    "indosat_dompetku", "mandiri_ecash", "permata_va", "bca_va",
    "bni_va", "other_va", "kioson", "indomaret", "gci", "danamon_online"],
  "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"
],
"customer_details": {
  "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"
  },
  "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	Deskripsi
transaction_details.order_id <i>String(50) (required)</i>	ID Transaksi. Setiap merchant dapat menggunakan 1 Order ID sekali. CATATAN: Simbol yang Diperbolehkan adalah tanda hubung (-), garis bawah (_), tilde (~), dan titik (.)
transaction_details.gross_amount <i>Integer (required)</i>	Jumlah yang akan ditagih ke <i>customer</i>
item_details <i>Array Item Details (optional)</i>	Rincian barang yang akan dibayar

Parameter	Deskripsi
<p>enabled_payments Array (optional)</p>	<p>Daftar tipe pembayaran yang dapat dipilih <i>customer</i>. Jika kosong, semua tipe pembayaran yang aktif akan digunakan.</p> <p>Pilihan: credit_card, mandiri_clickpay, cimb_clicks, bca_klikbca, bca_klikpay, bri_epay, telkomsel_cash, echannel, indosat_dompetku, mandiri_ecash, permata_va, other_va, bca_va, bni_va, kioson, indomaret, gci, danamon_online.</p> <p>Alias berelasi dengan beberapa tipe pembayaran. Menambahkan alias berarti menambahkan tipe pembayaran yang berelasi.</p> <p>Daftar Alias: bank_transfer => permata_va, bca_va cstore => kioson, indomaret.</p> <p>Jika Anda ingin mengaktifkan other_va, jangan lupa untuk turut mengaktifkan permata_va karena Midtrans menggunakan Permata untuk Bank Transfer lainnya</p>
customer_details.first_name String(255) (optional)	
customer_details.last_name String(255) (optional)	
customer_details.email String(255) (optional)	
customer_details.phone String(255) (optional)	
customer_details.billing_address Address (optional)	
customer_details.shipping_address Address (optional)	
credit_card CreditCard (optional)	Kustomisasi pembayaran kartu kredit
bca_va BCA Virtual Account (optional)	Kustomisasi pembayaran BCA virtual account

Parameter	Deskripsi
permata_va <i>Permata Virtual Account (optional)</i>	Kustomisasi pembayaran Permata virtual account
bni_va <i>BNI Virtual Account (optional)</i>	Kustomisasi pembayaran BNI virtual account
callbacks.finish <i>String(255) (optional)</i>	Redirect URL jika pembayaran berhasil diselesaikan (di-override oleh JS callback)
expiry <i>Expiry (optional)</i>	kustomisasi umur transaksi
custom_field1 <i>String(255) (optional)</i>	Custom field 1 untuk kustom parameter dari merchant
custom_field2 <i>String(255) (optional)</i>	Custom field 2 untuk kustom parameter dari merchant
custom_field3 <i>String(255) (optional)</i>	Custom field 3 untuk kustom parameter dari merchant

JSON Object

Kumpulan objek JSON yang digunakan dalam parameter [Create Snap Token](#).

Item Details

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

Parameter	Deskripsi
id <i>String(50) (optional)</i>	ID barang

Parameter	Deskripsi
price <i>Integer (required)</i>	Harga barang NOTE: Jangan menggunakan desimal sebagai harga barang
quantity <i>Integer (required)</i>	Jumlah barang yang dibeli
name <i>String(50) (required)</i>	Nama barang
brand <i>String(50) (optional)</i>	Merk barang
category <i>String(50) (optional)</i>	Kategori barang
merchant_name <i>String(50) (optional)</i>	Nama merchant yang menjual barang

Peringatan!

Subtotal (Harga barang dikalikan dengan jumlah barang) dari semua rincian barang harus sama dengan gross_amount dalam parameter transaction_details

Tips:

Anda dapat membuat item dengan harga negatif untuk ditampilkan sebagai diskon

Address

```
{
  "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	Deskripsi
first_name	
<i>String(255) (optional)</i>	
last_name	
<i>String(255) (optional)</i>	
email	
<i>String(255) (optional)</i>	
phone	
<i>String(255) (optional)</i>	
address	
<i>String(200) (optional)</i>	
country_code	
<i>String(10) (optional)</i>	
postal_code	
<i>String(10) (optional)</i>	
city	
<i>String(100) (optional)</i>	

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	Deskripsi
secure <i>Boolean (optional)</i>	Gunakan autentikasi 3D Secure. Default: false
bank <i>String (optional)</i>	Acquiring bank. Pilihan: bca, bni, mandiri, cimb, bri, danamon, maybank, mega
channel <i>String (optional)</i>	Acquiring channel. Pilihan: migs
type <i>String (optional)</i>	Tipe transaksi kartu kredit. Pilihan: authorize, authorize_capture. Default: "authorize_capture"
whitelist_bins <i>Array (optional)</i>	Nomor BIN kartu kredit yang diperbolehkan. Nilainya dapat berupa 8 digit BIN ataupun nama bank (semua nomor kartu yang dikeluarkan oleh bank tersebut akan diperbolehkan). Nama bank yang didukung yaitu bni bca mandiri cimb bri and maybank. Default: perbolehkan semua nomor kartu
installment.required <i>Boolean (optional)</i>	Transaksi harus menggunakan cicilan. Default: false
installment.terms <i>Object (optional)</i>	Pilihan bank dan durasi cicilan

BCA Virtual Account

```

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

```

```

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

```

Parameter	Deskripsi
va_number <i>String (optional)</i>	Nomor virtual account kustom. Panjang antara 1-11 digit.
free_text <i>FreeText (optional)</i>	

Free Text

Parameter	Deskripsi
inquiry <i>Array of FreeTextItem (optional)</i>	Jumlah item tidak boleh lebih dari 10.
payment <i>Array of FreeTextItem (optional)</i>	Jumlah item tidak boleh lebih dari 10.

Free Text Item

Parameter	Deskripsi
en <i>String (required)</i>	Panjang tidak boleh lebih dari 50 karakter.
id <i>String (required)</i>	Panjang tidak boleh lebih dari 50 karakter.

Permata Virtual Account

```

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

```

```
}
```

Parameter	Deskripsi
va_number <i>String (optional)</i>	Nomor virtual account kustom . Panjang 10 digit. Hanya untuk transaksi B2B.
recipient_name <i>String (optional)</i>	Nama penerima yang ditampilkan pada rincian pembayaran di bank. Nama akan ditampilkan dengan huruf kapital, maksimal 20 karakter. CATATAN: Default adalah nama merchant

BNI Virtual Account

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

Parameter	Deskripsi
va_number <i>String (optional)</i>	Nomor virtual account kustom . Panjang antara 1-8 digit.

Expiry

Note: Jika parameter ini tidak dikirim, secara default snap token akan kadaluarsa dalam 24 jam. Selain itu, jika hanya parameter unit dan duration yang dikirim, start_time akan diisi dengan timestamp dari snap token tersebut.

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

Parameter	Deskripsi
start_time <i>String(255) (optional)</i>	<i>Timestamp</i> dalam format yyyy-MM-dd HH:mm:ss Z. Jika tidak diset, waktu transaksi akan digunakan sebagai start_time (saat customer charge)

Parameter	Deskripsi
duration	Expiry duration
<i>Integer (optional)</i>	
unit	Expiry unit. Pilihan: day, hour, minute (Bentuk plural dapat digunakan)
<i>String (optional)</i>	

Response

Request Berhasil

```
{
  "token": "d379aa71-99eb-4dd1-b9bb-ee813746e9"
}
```

HTTP status code: 201

Field	Deskripsi
token	Snap token untuk integrasi frontend
<i>String(36)</i>	

Request Gagal

Autentikasi Gagal

```
{
  "error_messages": [
    "Access denied, please check client or server key"
  ]
}
```

HTTP status code: 401

Field	Deskripsi
error_messages	Pesan error
Array	

Validasi Gagal

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

HTTP status code: 400

Field	Deskripsi
error_messages	Pesan error
Array	

Kesalahan Sistem

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

HTTP status code: 500

Field	Deskripsi
error_messages	Pesan error
Array	

Integrasi Frontend

Gambaran Umum

```
<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>
```

Tujuan dari integrasi frontend adalah menampilkan pop up Snap di halaman Anda.

Tambahkan `snap.js` ke dalam halaman Anda sehingga modul snap siap digunakan. Jangan lupa untuk menaruh informasi `CLIENT-KEY` Anda sebagai nilai dari atribut `data-client-key` di tag script `snap.js`. Anda dapat mengetahui `CLIENT-KEY` Anda dengan melihat di [Settings - Access Keys](#).

Anda dapat memulai proses pembayaran dengan memanggil fungsi `snap.pay` dan menggunakan

SNAP_TOKEN yang didapat dari [integrasi backend](#) sebagai parameter.

Lokasi Snap.js

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

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

Viewport Meta Tag

Untuk memastikan popup modal Snap ditampilkan secara sesuai oleh browser mobile, jangan lupa untuk menaruh tag viewport meta tag di dalam element <head> Anda.

Pada umumnya, implementasi tag viewport meta dilakukan seperti berikut:

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

Snap JS

snap memiliki 3 fungsi utama: pay, show, & hide

pay(snapToken, options)

Mulai pembayaran di halaman Snap.

```
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:

Nama	Deskripsi
snapToken <i>String (required)</i>	Snap token yang diperoleh saat integrasi backend

Nama	Deskripsi
<code>enabledPayments</code> <i>Array (optional)</i>	Daftar pilihan pembayaran yang akan ditampilkan. Akan memfilter daftar pilihan pembayaran dari integrasi backend.
<code>options.onSuccess</code> <i>Function (optional)</i>	Callback pembayaran sukses (200 status_code)
<code>options.onPending</code> <i>Function (optional)</i>	Callback pembayaran <i>pending</i> (201 status_code)
<code>options.onError</code> <i>Function (optional)</i>	Callback pembayaran <i>error</i> (4xx or 5xx status_code)
<code>options.onClose</code> <i>Function (optional)</i>	Callback saat <i>customer</i> menutup halaman pembayaran (transaksi belum dilakukan)
<code>options.language</code> <i>String (optional)</i>	Pengaturan bahasa. Akan menimpa pengaturan bahasa di Merchant Administration Portal. Bahasa yang didukung yaitu <code>en</code> (Inggris) dan <code>id</code> (Bahasa). Default: <code>id</code>
<code>options.skipOrderSummary</code> <i>Boolean (optional)</i>	Lewati halaman rangkuman jika di-set <code>true</code> . Default: <code>false</code>
<code>options.skipCustomerDetails</code> <i>Boolean (optional)</i>	Membuat email dan no telepon opsional dalam form CC jika di-set to <code>true</code> . Default: <code>false</code>
<code>options.autoCloseDelay</code> <i>Integer (optional)</i>	Tutup secara otomatis halaman pembayaran terakhir indomaret dan bank transfer setelah waktu yang ditentukan. Jeda waktu dinyatakan dalam satuan detik. Atur jeda waktu dengan nilai <code>0</code> untuk menonaktifkan fitur ini. Default: <code>0</code>
<code>options.showOrderId</code> <i>Boolean (optional)</i>	Menampilkan informasi order ID jika di-set <code>true</code> . Default: <code>false</code>

Fungsi `onSuccess`, `onPending`, & `onError` menerima satu parameter, yaitu: objek [Transaction Result](#).

```
function ajaxGetToken(transactionData, callback){
    var snapToken;

    // Request SNAP_TOKEN melalui backend Anda & simpan hasilnya ke variabel snapToken

    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()

Tampilkan halaman *loading* Snap. Fungsi ini berguna jika Anda ingin menampilkan animasi *loading* ketika *request* SNAP_TOKEN sedang berlangsung menggunakan AJAX.

Jika AJAX success, gunakan `snap.pay` untuk melanjutkan proses pembayaran. Anda juga dapat menggunakan `snap.hide` untuk menyembunyikan halaman *loading*.

hide()

Sembunyikan halaman Snap. Merupakan komplemen dari fungsi `snap.show`. Fungsi ini berguna jika anda ingin menyembunyikan halaman Snap yang sedang aktif.

JS Callback

Transaction Result

Kartu kredit - transaksi sukses

```

{
  "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"
}

```

Echannel - transaksi pending

```

{
  "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"
}

```

Transaksi error

```

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

```

Objek yang merepresentasikan hasil transaksi. Merupakan parameter dari [Snap callback](#).

Nama	Deskripsi
status_code <i>String</i>	Kode status dari transaksi. Nilai: 200, 201, 202, 400, 404, 406, 500

Nama	Deskripsi
status_message <i>String</i>	Detail status
order_id <i>String</i>	ID transaksi oleh merchant
gross_amount <i>String</i>	Jumlah yang dibayar oleh <i>customer</i>
payment_type <i>String</i>	Tipe pembayaran yang digunakan oleh customer. Nilai: credit_card, bca_klikpay, bca_klikbca, bri_epay, mandiri_clickpay, telkomsel_cash, bank_transfer, echannel, indosat_dompotku, mandiri_ecash, cstore
transaction_time <i>String</i>	<i>Timestamp</i> dalam format yyyy-MM-dd hh:mm:ss
transaction_status <i>String</i>	Status transaksi. Nilai: capture, settlement, pending, cancel, expired
fraud_status <i>String</i>	Status <i>fraud</i> . Nilai: accept, challenge, deny
approval_code <i>String</i>	Kode <i>approval</i> dari bank
masked_card <i>String</i>	6 digit depan & 4 digit belakang kartu (hanya di credit_card & mandiri_clickpay)
bank <i>String</i>	Acquiring Bank (hanya di credit_card & mandiri_clickpay)
permata_va_number <i>String</i>	<i>Virtual account</i> Permata (hanya di bank_transfer)
bill_key <i>String</i>	Kode <i>bill</i> yang harus dibayar <i>customer</i> (hanya di echannel)
biller_code <i>String</i>	Kode <i>biller</i> merchant (hanya di echannel)
redirect_url <i>String</i>	URL yang harus dikunjungi <i>customer</i> (hanya di bca_klikbca, bri_epay, & cimb_clicks)
saved_token_id <i>String</i>	TWO_CLICKS_TOKEN (hanya di credit_card)
saved_token_id_expired_at <i>String</i>	Waktu kadaluarsa TWO_CLICKS_TOKEN

Peringatan!

Demi alasan keamanan, status transaksi dari JS callback sebaiknya hanya digunakan untuk memberikan feedback UI ke user dan sebaiknya tidak digunakan untuk mengubah status transaksi di database Anda. Kami telah menyediakan notifikasi HTTP untuk itu. Anda dapat mengatur URL notifikasi HTTP transaksi Anda di [Settings - Configuration](#)

Fitur Lainnya

Two Clicks

1. Request `/snap/v1/transactions` dengan `credit_card.save_card` bernilai true

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

2. Simpan nilai `saved_token_id` yang tersedia di dalam callback `OnSuccess` atau [Http Notification](#)

```
snap.pay('YOUR_SNAP_TOKEN', {
  onSuccess: function(result) {
    if (result.saved_token_id) {
      // simpan TWO_CLICKS_TOKEN dari customer ke dalam database
    }
  }
})
```

3. Request `/snap/v1/transactions` dengan `credit_card.card_token` bernilai

TWO_CLICKS_TOKEN yang telah Anda simpan pada transaksi pertama

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

Fitur two clicks memungkinkan Anda untuk menyimpan data nomor kartu kredit, tanggal kadaluarsa kartu kredit, email, dan ponsel customer dalam bentuk TWO_CLICKS_TOKEN. Pada pembayaran selanjutnya, TWO_CLICKS_TOKEN dapat digunakan untuk mengisi detail pembayaran secara otomatis. Customer hanya perlu melengkapi CVV kartu kredit untuk menyelesaikan pembayaran.

Ada 2 langkah yang perlu dilakukan untuk menggunakan fitur two clicks:

Transaksi Awal

- Request SNAP_TOKEN dengan `credit_card.save_card` bernilai `true`
- Snap akan menawarkan *customer* pilihan untuk menyimpan data
- Jika customer memilih simpan data, maka [Transaction Result](#) akan memiliki dua *field* tambahan, yaitu `saved_token_id` dan `saved_token_id_expired_at`

Transaksi Selanjutnya

- Request SNAP_TOKEN dengan `credit_card.card_token` bernilai TWO_CLICKS_TOKEN yang telah disimpan pada transaksi pertama
- Field pembayaran kartu kredit (nomor kartu, tanggal kadaluarsa, email, & ponsel) akan otomatis terisi

One Click

Request SNAP_TOKEN dengan `user_id` dan set `credit_card.save_card` ke `true`

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

Sebagai tambahan dari fitur two clicks, SNAP juga mendukung transaksi one click dimana informasi CVV juga akan disimpan. Dengan ini, customer dapat langsung melanjutkan proses pembayaran tanpa harus memasukkan informasi tambahan.

Untuk menyederhanakan alur penyimpanan data kartu oleh merchant, SNAP menyediakan fitur Card Token Storage. Jadi, merchant tidak perlu menyimpan dan mengatur Card Token sendiri. Merchant dapat dengan mudah mengintegrasikan fitur Card Token Storage ke SNAP dengan menambahkan parameter `user_id` yang berasosiasi dengan akun customer di sistem merchant, selain mengaktifkan flag `credit_card.save_card`, ketika mengirimkan request SNAP Token.

Kemudian, SNAP akan menentukan informasi kartu nantinya akan disimpan sebagai one click berdasarkan kriteria berikut:

1. Merchant memiliki MID recurring yang aktif
2. Transaksi awal adalah 3DS

Jika kedua kondisi di atas terpenuhi, Card Token akan disimpan sebagai one click. Selain itu, maka akan disimpan sebagai two clicks.

Mirip dengan alur two clicks, ada dua langkah yang diperlukan untuk menggunakan fitur Card Token Storage di SNAP:

Transaksi Awal

- Request SNAP_TOKEN dengan parameter `user_id` dan atur `credit_card.save_card` ke `true`
- `user_id` adalah sebuah id unik yang merepresentasikan seorang customer. Nilainya dapat berupa string dengan panjang maksimal 255 karakter. Hindari id yang dapat dengan mudah dihubungkan dengan customer seperti email, id dari database, dsb. Gunakanlah id seperti

sha512(email/record.id + salt). Jangan gunakan user_id di sisi frontend seperti browser atau aplikasi mobile.

- SNAP akan menawarkan customer pilihan untuk menyimpan data
- Jika customer memilih simpan, SNAP akan menyimpan Card Token dan mengasosiasikannya dengan user_id
- Jika Anda ingin menggunakan fitur one click, pastikan untuk mengaktifkan MID recurring dan mengatur credit_card.secure ke true. Jika tidak, Card Token akan disimpan sebagai two clicks.

Transaksi Selanjutnya

- Request SNAP_TOKEN dengan user_id yang sama seperti pada transaksi awal
- Parameter credit_card.save_card tidak harus di set ke true pada langkah ini
- Customer dapat memilih di antara beberapa kartu jika sebelumnya mereka telah menyimpan beberapa kartu
- Jika kartu yang dipilih tersimpan sebagai one click, maka field nomor kartu, tanggal kadaluarsa, cvv, email, dan nomor telepon akan otomatis terisi. Customer dapat langsung melanjutkan proses charge (proses ini juga akan melewati 3DS).
- Jika kartu yang dipilih tersimpan sebagai two clicks, customer harus memasukkan informasi cvv terlebih dahulu dan melalui proses 3DS jika perlu (sama seperti [alur two clicks](#))

Nomor Virtual Account Kustom

bca_va

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

permata_va

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

bni_va

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

Nomor virtual account terdiri dari dua bagian. Contohnya {91012}{12435678}, bagian pertama adalah kode perusahaan dan bagian kedua adalah kode unik. Bagian kedua dapat dikustomisasi untuk tipe pembayaran [bca_va](#), [permata_va](#) dan [bni_va](#).

- Hanya menerima digit
- Masing-masing bank memiliki spesifikasi yang berbeda. Harap periksa dokumentasi VA kustom bagi masing-masing bank.
- Jika nomor sudah pernah digunakan sebelumnya pada order yang lain, maka akan dibuatkan nomor unik yang lain secara random
- Jika nomor terlalu panjang, digit yang tidak terpakai akan dipotong dari belakang
- Jika nomor terlalu pendek, akan diberi awalan nol

Pre-Authorization

Request /snap/v1/transactions dengan `credit_card.type` di-set ke “authorize”

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

Fitur pre-authorization memungkinkan Anda untuk mem-blok limit kartu kredit customer. Untuk menggunakan fitur ini, dibutuhkan parameter tambahan di objek kartu kredit:

Parameter	Deskripsi
<code>credit_card.type</code> <i>String (opsional)</i>	Tipe transaksi kartu kredit. Pilihan: <code>authorize</code> , <code>authorize_capture</code> . Default: “ <code>authorize_capture</code> ”


Status Transaksi

midtrans menyediakan tiga sarana bagi merchant untuk memperoleh status transaksi.

- Notifikasi Email
- Notifikasi HTTP
- Merchant Administration Portal

Notifikasi email

Anda dapat mengkonfigurasi pengaturan notifikasi email di [Settings - Email Notification](#) di MAP.



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.

Email Notification

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

email notification

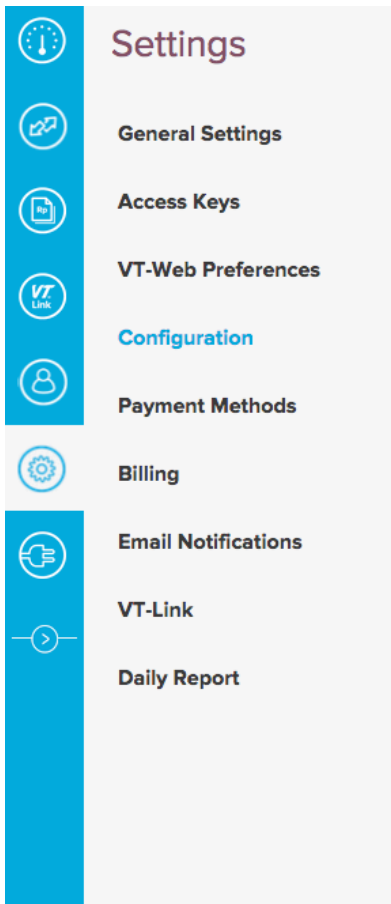
Info!

Untuk menambah beberapa email, silakan gunakan koma sebagai pemisah. contoh: an-dri@myemail.com, litani@myemail.com

Notifikasi HTTP(S) POST

Notifikasi melalui HTTP(S) POST akan dikirim ke merchant server bila pelanggan telah menyelesaikan proses pembayaran. Anda dapat memanfaatkan notifikasi HTTP(S) POST untuk memperbarui status pembayaran atau mengirim barang pembeli secara real time.

Mengaktifkan midtrans notifikasi HTTP(S) POST dengan menetapkan Payment Notification URL di [Settings - Configuration](#).



Settings

Configuration

Veritrans requires the URL endpoints for the following scenarios:

Payment Notification URL*	<input type="text" value="https://sneekers.com/payment/notification"/> <small>Address where we will send the notification via HTTP Post request. E.g http://yourwebsite.com/notification/handling</small> <input type="button" value="Send test notification"/> <input type="button" value="See history"/>
----------------------------------	--

Finish Redirect URL*	<input type="text" value="https://sneekers.com/payment/finish"/> <small>Customer sent here if payment is successful e.g http://yourwebsite.com/payment/finish</small>
-----------------------------	--

Unfinish Redirect URL*	<input type="text" value="https://sneekers.com/payment/unfinish"/> <small>Customer sent here if they click on 'Back to Order Website' on VT-Web's payment page e.g http://yourwebsite.com/payment/unfinish</small>
-------------------------------	---

Error Redirect URL*	<input type="text" value="https://sneekers.com/payment/error"/> <small>Customer sent here if payment is unsuccessful or an error occurs during payment e.g http://yourwebsite.com/payment/error</small>
----------------------------	--

http notification

Peringatan!

Pastikan Payment Notification URL Anda tidak sama dengan alamat lokal komputer Anda.

Lihat [disini](#) untuk contoh json notifikasi setiap channels pembayaran.

Merchant Admin Portal (MAP) Status Transaksi

Merchant dapat memeriksa status transaksi dari [payments](#) menu di MAP.

Contoh status transaksi yang berhasil untuk kartu kredit:

Toko eCommerce					
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

Menggunakan Notifikasi

Logika signature key

SHA512(order_id + status_code + gross_amount + serverkey)

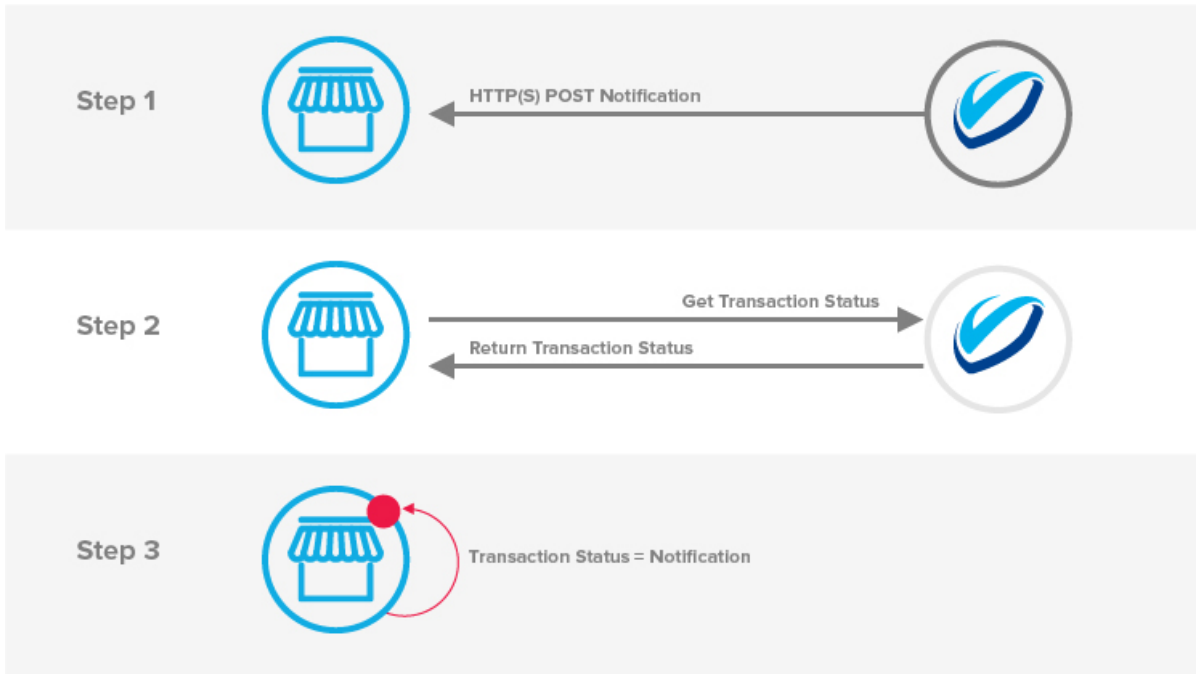
Contoh kode menghasilkan signature key

```
<?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;
?>
```

Dalam rangka meningkatkan aspek keamanan, ada beberapa cara untuk memastikan notifikasi yang diterima dari midtrans.

Membandingkan Response

Mekanisme tambahan yang kami sediakan untuk memverifikasi isi dan asal pengirim notifikasi adalah dengan membandingkan respon yg didapat dari [get status](#) API. Respon akan sama dengan status notifikasi.



challenge response

Signature Key

Kami menambahkan informasi signature key dalam notifikasi kami. Tujuan utama signature key ini adalah untuk memvalidasi apakah notifikasi tersebut berasal dari midtrans atau bukan. Jika notifikasi tidak asli, merchant dapat mengabaikan notifikasi. Silakan merujuk sisi kode di samping untuk melihat logika signature key dan contoh kode untuk menghasilkan signature key.

Best Practice Menggunakan Notifikasi

- Selalu gunakan HTTPS endpoint. Hal ini akan memberikan keamanan dan tidak mungkin ada serangan MITM karena kami memvalidasi https certificates sesuai dengan host. Dan juga jangan menggunakan self signed certificates.
- Selalu gunakan notifikasi dengan cara idempoten, dalam kasus yang sangat jarang, kami dapat mengirimkan beberapa notifikasi untuk transaksi yang sama dua kali. Hal ini seharusnya tidak menyebabkan entri ganda di merchant, Cara sederhana untuk mencapai ini adalah dengan menggunakan orderid sebagai kunci untuk melacak entri.

- Selalu periksa hash notifikasi signature, ini akan mengkonfirmasi bahwa notifikasi itu benar-benar dikirim oleh midtrans, karena kita menandatangani kunci rahasia (server key). Tidak ada orang lain dapat membuat hash signature ini.
- Selalu periksa semua berikut tiga bidang untuk mengkonfirmasi transaksi sukses
- status code: 200 untuk transaksi sukses
- farud status: ACCEPT
- transaction status: settlement/capture
- Kami berusaha untuk mengirim notifikasi segera setelah transaksi terjadi, tetapi dalam kasus yang sangat jarang, notifikasi mungkin tertunda karena banyaknya transaksi. Jika Anda belum menerima notifikasi, silakan gunakan [Status API](#) untuk mengecek status transaksi.
- Gunkana [Status API](#) untuk mendapatkan status terbaru dari transaksi/order pada setiap notifikasi.
- Kami menetapkan HTTP timeout selama 30 detik.
Harap berusaha untuk menjaga waktu respon dari notifikasi HTTP dibawah 5 detik.
- Dalam kasus yang sangat jarang kami dapat mengirimkan notifikasi tidak berurutan. Misalnya status notifikasi “settlement” dikirim sebelum status notifikasi “Pending”. Sangat penting bahwa notifikasi seperti ini diabaikan. Berikut diagram urutan transisi yang dapat Anda gunakan. Tapi sekali lagi, gunakan /status kita API untuk mengkonfirmasi status yang sebenarnya.
- Kami mengirim data notifikasi sebagai JSON, silakan ekstrak JSON dengan JSON parser. Data notifikasi mungkin ada penambahan informasi baru, sehingga gunakanlah format yang lebih adaptif terhadap data baru, jadi jika parser melihat informasi baru, seharusnya tidak membuat program throw exception. Hal ini memungkinkan kita untuk mengembangkan sistem notifikasi untuk menambah fitur baru tanpa membuat error di pengguna lama.
- Selalu gunakan [HTTP Status code](#) untuk menggunakan pemberitahuan, kami mencoba mengirim ulang notifikasi untuk kasus-kasus error berdasarkan kode status
- Untuk 2xx: Tidak dikirim ulang, itu dianggap sukses
- Untuk 500: Kami akan mencoba lagi 1 kali dalam interval 1 menit
- Untuk 503: Coba lagi 4 kali
- Untuk 400 / 404: coba lagi 2 kali dalam interval 1 menit
- Untuk semua kegagalan lainnya: Coba lagi 5 kali pada interval 1 menit

Berikut ini adalah jenis-jenis notifikasi. Catatan, notifikasi jenis baru dapat bertambah. Dan juga informasi baru dalam notifikasi yang telah ada dapat bertambah, silahkan konfirmasi dengan dokumentasi terbaru untuk informasi yang tepat.

Status Kode

Tujuan: Anda memahami kode status yang digunakan oleh midtrans. Apabila butuh bantuan, silahkan hubungi kami di support@midtrans.com atau kunjungi halaman support kami.

Status code di midtrans API terbagi menjadi beberapa bagian, yaitu 2xx, 3xx, 4xx dan 5xx.

Kode 2xx

Status	Keterangan
200	Kartu Kredit: Sukses. Request dan transaksi berhasil (authorize, capture, cancel, get status, approve transaksi), telah diterima oleh midtrans dan bank. Metode pembayaran lainnya: Sukses. Transaksi berhasil/settlement.

Status	Keterangan
201	<p>Kartu Kredit: Challenge. Transaksi berhasil namun belum selesai diproses, diperlukan tindakan manual merchant untuk menyelesaikan proses transaksi. Jika merchant tidak melakukan action apapun hingga waktu settlement (H+1 pkl 16.00 WIB), midtrans akan membatalkan transaksi tersebut.</p> <p>Bank Transfer: Pending. Transaksi berhasil namun proses pembayaran belum selesai dilakukan oleh customer. Secara default transaksi akan kadaluarsa dalam waktu 24 jam.</p> <p>Cimb Clicks: Pending. Transaksi berhasil namun proses pembayaran belum selesai dilakukan oleh customer. Secara default transaksi akan kadaluarsa dalam waktu 2 jam.</p> <p>BRI ePay: Pending. Transaksi berhasil namun proses pembayaran belum selesai dilakukan oleh customer. Secara default transaksi akan kadaluarsa dalam waktu 2 jam.</p> <p>Klik BCA: Pending . Transaksi berhasil namun proses pembayaran belum selesai dilakukan oleh customer. Secara default transaksi akan kadaluarsa dalam waktu 2 jam.</p> <p>BCA Klikpay: Pending . Transaksi berhasil namun proses pembayaran belum selesai dilakukan oleh customer. Secara default transaksi akan kadaluarsa dalam waktu 2 jam.</p> <p>Mandiri Bill Payment: Pending . Transaksi berhasil namun proses pembayaran belum selesai dilakukan oleh customer. Secara default transaksi akan kadaluarsa dalam waktu 2 jam.</p> <p>Indomaret: Pending. Transaksi berhasil namun proses pembayaran belum selesai dilakukan oleh customer. Secara default transaksi akan kadaluarsa dalam waktu 2 jam.</p>

Status	Keterangan
202	Kartu Kredit: Gagal. Request berhasil tapi transaksi ditolak oleh penyedia pembayaran atau Fraud Detection System midtrans. Metode pembayaran lainnya: Gagal. Request berhasil tapi transaksi ditolak penyedia pembayaran.

Kode 3xx

Status	Keterangan
300	Pindah permanen. Semua request, baik yang ada saat ini maupun yang akan datang, harus dialihkan ke URL baru.

Kode 4xx

Status	Keterangan
400	Validasi error. Anda mengirim contoh request data yang salah; validasi error, kesalahan tipe transaksi, kesalahan format kartu kredit, dll.
401	Akses ditolak karena transaksi tidak sah, harap periksa client key atau server key.
402	Anda tidak memiliki akses untuk metode pembayaran ini.
403	Sumber yang diminta hanya mampu memberikan konten yang sesuai dengan accept header yang dikirim di HTTP request.
404	Sumber yang diminta tidak dapat ditemukan.
405	Metode HTTP tidak diizinkan.
406	Order ID sama. Order ID sudah digunakan sebelumnya.
407	Transaksi telah lewat dari masa berlaku.
408	Merchant mengirimkan tipe data yang salah.
409	Merchant sudah mengirimkan terlalu banyak transaksi dengan nomor kartu yang sama.

Status	Keterangan
410	Akun Anda sudah dinonaktifkan. Silahkan hubungi midtrans Support.
411	Token ID tidak ada, salah, atau expired.
412	Merchant tidak bisa memodifikasi status transaksi.
413	Request tidak bisa diproses dikarenakan kesalahan syntax pada request body.

Kode 5xx

Status	Keterangan
500	Server Internal Error.
501	Fitur akan segera tersedia.
502	Server Internal Error: Masalah Koneksi Bank
503	Server Internal Error
504	Server Internal Error: Deteksi Fraud tidak tersedia

Kredensial Testing Pembayaran

Berikut adalah daftar testing pembayaran yang dapat digunakan di Sandbox Environment.

Kartu Kredit

Kartu Umum

Transaksi Normal

VISA	Description
Tidak ada Authentikasi <i>Merchant nonaktifkan 3DS</i>	Transaksi Berhasil: 4011 1111 1111 1112 Transaksi Challenge oleh FDS: 4111 1111 1111 1111 Transaksi Gagal oleh FDS: 4211 1111 1111 1110 Transaksi Gagal oleh Bank: 4311 1111 1111 1119
MASTERCARD	Description
Tidak ada Authentikasi <i>Merchant nonaktifkan 3DS</i>	Transaksi Berhasil: 5481 1611 1111 1081 Transaksi Challenge oleh FDS: 5110 1111 1111 1119 Transaksi Gagal oleh FDS: 5210 1111 1111 1118 Transaksi Gagal oleh Bank: 5310 1111 1111 1117

Transaksi 3D Secure

VISA	Description
Authentikasi Penuh	Transaksi Berhasil: 4811 1111 1111 1114
<i>Pemegang kartu terdaftar 3DS</i>	Transaksi Gagal oleh Bank: 4911 1111 1111 1113
Authentikasi Sebagian	Transaksi Berhasil: 4411 1111 1111 1118
<i>Pemegang kartu tidak terdaftar 3DS</i>	Transaksi Challenge oleh FDS: 4511 1111 1111 1117
	Transaksi Gagal oleh FDS: 4611 1111 1111 1116
	Transaksi Gagal oleh Bank: 4711 1111 1111 1115

MASTERCARD	Description
Authentikasi Penuh	Transaksi Berhasil: 5211 1111 1111 1117
<i>Pemegang kartu terdaftar 3DS</i>	Transaksi Gagal oleh Bank: 5111 1111 1111 1118
Authentikasi Sebagian	Transaksi Berhasil: 5410 1111 1111 1116
<i>Pemegang kartu tidak terdaftar 3DS</i>	Transaksi Challenge oleh FDS: 5510 1111 1111 1115
	Transaksi Gagal oleh FDS: 5411 1111 1111 1115
	Transaksi Gagal oleh Bank: 5511 1111 1111 1114

Kartu Spesifik per Bank

Transaksi Berhasil

Bank	Nomor Kartu
Mandiri	4617 0030 8177 8145 5573 3899 1384 3648
Mandiri Debit	4097 6658 2970 8136
CIMB	4599 2043 5516 9092 5481 1620 3830 7406
BNI	4105 0568 6475 0672 5264 2282 3919 2765
BNI Private Label	1946 4159 8148 7684
BCA	4773 7781 0290 6680 5229 9028 2076 4661
BRI	4365 0299 9362 0368 5520 0219 0920 3008
Maybank	4055 7796 2846 0474 5520 0883 1465 3770

Transaksi Berhasil dengan 3D Secure

Bank	Nomor Kartu	
Mandiri		
Full Authentication	4617 0069 5974 6656	5573 3810 7219 6900
Attempted Authentication	4617 0017 4194 2101	5573 3819 9982 5417
CIMB		
Full Authentication	4599 2078 8712 2414	5481 1698 1883 2479
Attempted Authentication	4599 2039 9705 2898	5481 1671 2103 2563
BNI		
Full Authentication	4105 0586 8948 1467	5264 2210 3887 4659
Attempted Authentication	4105 0525 4151 2148	5264 2249 7176 1016
BCA		
Full Authentication	4773 7760 5705 1650	5229 9031 3685 3172
Attempted Authentication	4773 7738 1098 1190	5229 9073 6430 3610
BRI		
Full Authentication	4365 0263 3573 7199	5520 0298 7089 9100
Attempted Authentication	4365 0278 6723 2690	5520 0254 8646 8439
Maybank		
Full Authentication	4055 7720 2603 6004	5520 0867 5210 2334
Attempted Authentication	4055 7713 3514 4012	5520 0867 7490 8452

Transaksi Gagal

Bank	Nomor Kartu	
Mandiri	4617 0085 6083 1760	5573 3840 4322 4447
Mandiri Debit	4097 6676 7217 8631	
CIMB	4599 2060 0973 3090	5481 1691 9178 2739
BNI	4105 0541 4854 1363	5264 2235 3013 1711
BNI Private Label	1946 4102 7193 1269	
BCA	4773 7752 0201 1809	5229 9034 0542 3830
BRI	4365 0286 6251 2583	5520 0219 0920 3008
Maybank	4055 7796 2846 0474	5520 0883 1465 3770

Expiry Date dan CVV

Input	Nilai
Expiry Month	01

Input	Nilai
Expiry Year	2020
CVV	123

Bank Transfer

Metode Pembayaran	Deskripsi
Permata Virtual Account	Midtrans akan membuat 16 digit nomor testing Virtual Account Permata. Untuk melakukan tes transaksi, gunakan Permata Virtual Account Simulator
BCA Virtual Account	Midtrans akan membuat 11 digit nomor testing Virtual Account BCA. Untuk melakukan tes transaksi, gunakan BCA Virtual Account Simulator
Mandiri Bill Payment	Midtrans akan membuat 12 digit Kode Pembayaran untuk pembayaran via Mandiri e-channel (Internet Banking, SMS Banking, Mandiri ATM). Untuk melakukan tes transaksi, gunakan Mandiri Bill Payment Simulator

Direct Debit

Metode Pembayaran	Deskripsi
Mandiri Clickpay	Nomor kartu: 4111 1111 1111 1111 Token Berhasil: 000000 Token Gagal: 111111
CIMB Clicks	midtrans akan mengalihkan tes transaksi CIMB Clicks ke simulasi pembayaran. Transaksi Sukses: testuser00 Transaksi Gagal: testuser01
ePay BRI	midtrans akan mengalihkan tes transaksi ePay BRI ke simulasi pembayaran. Transaksi Sukses: testuser00 Transaksi Gagal: testuser03
BCA Klikpay	midtrans akan mengalihkan tes transaksi BCA Klikpay ke simulasi pembayaran

Metode Pembayaran	Deskripsi
KlikBCA	midtrans akan mendaftarkan user id terisi di KlikBCA input. Untuk melakukan tes transaksi, gunakan KlikBca Simulator

e-Wallet

Metode Pembayaran	Deskripsi
Telkomsel Cash	Customer Berhasil: 0811111111 Customer Gagal: 0822222222
Indosat Dompetku	Nomor Berhasil: 08123456789 Nomor Gagal: other than 08123456789
Mandiri E-cash	Nomor Berhasil: 0987654321 PIN: 12345 OTP: 12123434

Convenience Store

Metode Pembayaran	Deskripsi
Indomaret	midtrans akan membuat kode pembayaran Indomaret testing. Untuk melakukan tes transaksi, gunakan Indomaret Simulator
Kioson	midtrans akan membuat kode pembayaran Kioson testing. Untuk melakukan tes transaksi, gunakan Kioson Simulator

Going Live

Tutorial berikut akan membantu anda melakukan perubahan konfigurasi dari Sandbox ke mode Production. Silahkan pilih metode integrasi yang digunakan. Apabila butuh bantuan, silahkan hubungi kami di support@midtrans.com atau kunjungi [halaman support](#) kami.

Jika Anda sudah selesai melakukan testing dan sudah siap untuk go live di Production Environment, ada beberapa hal yang perlu dipastikan:

- Hubungi Team Support midtrans untuk migrasi dari Sandbox ke Production Environment. Setelah mendapat konfirmasi, anda dapat login ke Dashboard Production midtrans di <https://account.midtrans.com/login> dengan kredensial yang sama.
- Mengubah setting environment di plugin midtrans dari development menjadi production. Berikut daftar cara mengubah setting lib dan plugin kami.

Nilai Transaksi Minimum

Setiap transaksi memiliki nilai gross_amount minimum yaitu 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;
...
?>
```

Mengubah Veritrans-PHP library ke Production Environment. Ini dapat dilakukan dengan mengubah nilai `Veritrans_Config::$serverKey` dengan production server key dan `Veritrans_Config::$isProduction` menjadi `true`.

Magento

Mengubah pengaturan plugin magento dengan production client key dan server key dan mengubah environment menjadi production.

v1.9.x

Midtrans ver 1.0.0

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

magento set production

v2.x.x

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	

magento set production

OpenCart

Ubah pengaturan plugin OpenCart dengan production client key dan server key dan ubah environment menjadi 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>


magento set production

Prestashop

Ubah pengaturan plugin Prestashop Midtrans dengan production client key dan server key dan ubah environment menjadi 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

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.

magento set production

Woocommerce

Ubah pengaturan plugin Woocommerce Midtrans dengan production client key dan server key dan ubah environment menjadi 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

magento set production