



DOKU API MERCHANT-HOSTED DOCUMENTATION

Version 1.10

June 2017

Table of Contents

1.0	Introduction.....	3
1.1	Merchant Hosted Integration	3
2.0	Credit Card.....	5
2.1	Advanced Features	10
2.1.1	BIN Filtering	10
2.1.2	2-Click Payment [Tokenization].....	11
2.1.3	1-Click Payment [Recurring]	13
2.1.4	Installment.....	16
2.1.5	Authorize & Capture	18
3.0	DOKU Wallet.....	19
4.0	Virtual Account.....	23
4.1	Bank Transfer.....	23
4.2	Convenience Store.....	27
5.0	Internet Banking.....	30
5.1	Mandiri Clickpay	30
6.0	Appendix.....	35
6.1	Payment Methods.....	35
6.2	Payment Channel Codes.....	35
6.3	Parameters Required.....	36
6.3.1	Payment Request.....	36
6.3.2	Check Status Request.....	37
6.3.2	Check Status Response.....	37
6.2	DOKU Response Codes	38
6.2.1	General response codes.....	38
6.2.2	Credit Card	39
6.2.3	DOKU Wallet.....	43
6.2.4	Virtual Account.....	44
6.2.5	Mandiri Clickpay.....	44

1.0 Introduction

DOKU enables you to accept a wide range of different payment methods. Each payment method has a unique approach to integration. The payment methods can be grouped into five distinct categories:

- ✓ Credit Card
- ✓ DOKU wallet
- ✓ Bank transfer (Virtual Account)
- ✓ Convenience store
- ✓ Internet banking

For more information about the payment methods, please see Appendix 6.1.

The DOKU API supports two methods of payment processing – namely, merchant-hosted and DOKU-hosted. These two services can be tailored to accommodate the separate business needs of merchants from those having advanced IT personnel, to merchants who do not even have a website. The DOKU-hosted API is detailed in a *separate document*. See below for features of each service:

1. **Merchant Hosted**

The payment page and data input is native to the merchant's website, without having to redirect to a DOKU-hosted page. Having the payment form on the merchant page does not compromise the security of the cardholder however, as DOKU is PCI (Level 1) certified, and none of the cardholder data will actually be stored on the merchant's server.

Merchant profile: Tech-savvy merchants who want a seamless transaction process and maintain full control of the payment page branding and experience. IT expertise is required to modify the payment page and use client-side encryption to encrypt card data.

2. **DOKU Hosted**

Instant payment services where the payment input form is located within the DOKU page. The selection of payment methods can be done on the merchant or DOKU page. With this service, the customer will be redirected to a DOKU-hosted page upon checkout to complete the payment.

Merchant profile: Merchants who prefer to let DOKU manage the entire data security management and do not require extensive payment page customization.

1.1 Merchant Hosted Integration

The next section gives an example of how you can integrate with DOKU for the various payment methods. Once you have confirmed to become a DOKU merchant through our Sales process, our integration team to proceed to the technical integration stage will contact you. All new merchants will receive a *shared key* and a *merchant code*. Take note of this information, as you will need to enter them into the API script during integration. The response codes are categorized by payment method, and can be found in the appendix.



The following instructions are divided into standalone APIs for each payment method. Therefore, there may be some repetition. When you do your actual integration, some of the steps may be skipped when adding on different payment methods and features.

2.0 Credit Card

By default all credit card payments processed by DOKU will undergo 3D secure. Non-3D secure payments are available, however would require further assessment by DOKU and the bank.

Credit card integration comprises 3 easy steps:

1. Insert JavaScript
2. Initiate JavaScript parameters
3. Create payment form

To get started on your integration, follow these steps one by one by pasting the template scripts onto your website. The DOKU payment form is saved in HTML format; the template script acts as a placeholder for where the payment form will appear on your page. Prepare your HTML pages as seen in the template, and customize accordingly.

1. Insert the `doku.js`, `fancybox.js` and `fancybox.css` onto your website's payment page, along with your custom style. See example:

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.pack.js"></script>
<link href="https://cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.min.css"
rel="stylesheet">

<script src="http://staging.doku.com/doku-js/assets/js/doku.js"></script>
<link href="http://staging.doku.com/doku-js/assets/css/doku.css" rel="stylesheet">
```

2. Initialize the payment form by creating the words using `doku_library`

```
<?php
require_once(' ../Doku.php');

Doku_Initiate::$sharedKey = '<Put Your Shared Key Here>';
Doku_Initiate::$smallId = '<Put Your Merchant Code Here>';

$invoice = 'invoice_1458123951';

$params = array(
    'amount' => '10000.00',
    'invoice' => $invoice,
    'currency' => '360'
);

$words = doku_Library::doCreateWords($params);
?>
```

...and adding the following example script on to your webpage:

```

<script type="text/javascript">
$(function() {
  var data = new Object();
  data.req_merchant_code = '1';
  data.req_chain_merchant = 'NA';
  data.req_payment_channel = '15'; // '15' = credit card
  data.req_transaction_id = '<?php echo $invoice ?>';
  data.req_currency = '<?php echo $currency ?>';
  data.req_amount = '<?php echo $amount ?>';
  data.req_words = '<?php echo $words ?>';
  data.req_form_type = 'full';

  getForm(data);
});
</script>

```

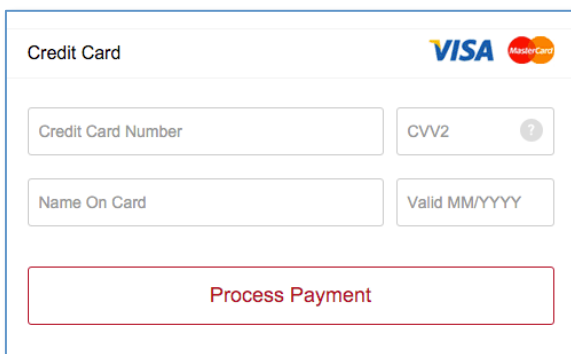
3. Insert the template script where you wish the payment form to appear on your website.

```

<form action="charge.php" method="POST" id="payment-form">
  <div doku-div='form-payment'>
    <input id="doku-token" name="doku-token" type="hidden" />
    <input id="doku-pairing-code" name="doku-pairing-code" type="hidden" />
  </div>
</form>

```

When completed correctly, the payment form should appear as seen below, and you may start receiving payments immediately. The DOKU script only provides the four fields to be filled in by the customer. All other parts of the payment page, including the logo and 'Process Payment' button are customizable to your needs for a completely white label payment flow.



Once your customer has input their card data and clicked the "Process Payment" button, the data will be processed by DOKU. The DOKU server then responds with the card ID, and the browser will post the data to the merchant server, according to the URL set in your action form.

Total Payment (IDR)
100000.00

BCA Verified by **VISA**

An Authorisation Code has been sent to your mobile number +62815xxxx007. Please enter the Authorisation Code to authenticate this payment before session time-out.

Time remaining: 05:08

Merchant Name : DOKU MERCHANT
 Amount : IDR 10.000,00
 Transaction Date : Mon Apr 4 2016
 16:34:09 GMT+0700
 BCA VISA No. : xxxx xxxx xxxx 7324 XXXX
 Authorisation Code :

This information is **Not Shared** with the Merchant.
 Any problem in receiving your Authorisation code? Please contact our **Halo BCA 500 888** or **+ 62 21 500 888** for assistance.

Due to potential fraud, Indonesia is a 3D Secure market, where customer authentication is required to proceed with the transaction. The form of authentication differs from each bank, but typically it involves a One-Time Password sent via SMS. Not all issuing banks in Indonesia have implemented 3D secure, but the majority of them have. If you would like for the 3D secure process to be lifted, please contact DOKU for more information. Further assessment by DOKU and the acquiring bank is required to grant a non-3D secure Merchant ID.

You can retrieve the posted data from the form within your server. Then send the payment request to the DOKU server to be processed using our library. See example:

```

<?php
require_once('../Doku.php');

Doku_Initiate::$sharedKey = '<Put Your Shared Key Here>';
Doku_Initiate::$mallId = '<Put Your Merchant Code Here>';

$token = $_POST['doku-token'];
$pairing_code = $_POST['doku-pairing-code'];
$invoice_no = $_POST['doku-invoice-no'];

$params = array(
    'amount' => '10000.00',
    'invoice' => $invoice_no,
    'currency' => '360',
    'pairing_code' => $pairing_code,
    'token' => $token
);

$words = Doku_Library::doCreateWords($params);

$basket[] = array(
    'name' => 'sayur',
    'amount' => '10000.00',
    'quantity' => '1',
    'subtotal' => '10000.00'
);

$customer = array(
    'name' => 'TEST NAME',
    'data_phone' => '08121111111',
    'data_email' => 'test@test.com',
    'data_address' => 'bojong gede #1 08/01'
);

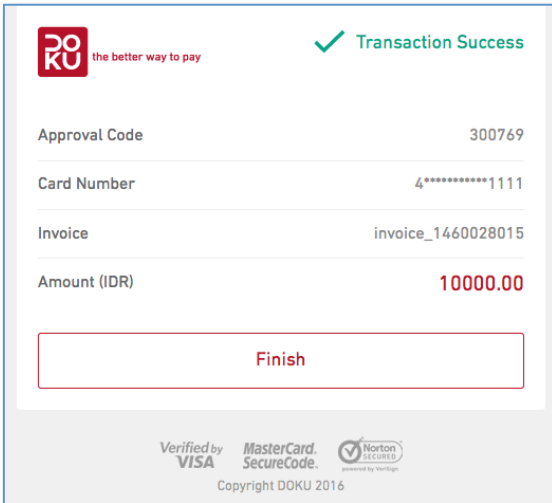
$dataPayment = array(
    'req_mall_id' => Doku_Initiate::$mallId,
    'req_chain_merchant' => 'NA',
    'req_amount' => '10000.00',
    'req_words' => $words,
    'req_purchase_amount' => '10000.00',
    'req_trans_id_merchant' => $invoice_no,
    'req_request_date_time' => date('YmdHis'),
    'req_currency' => '360',
    'req_purchase_currency' => '360',
    'req_session_id' => sha1(date('YmdHis')),
    'req_name' => $customer['name'],
    'req_payment_channel' => 15,
    'req_basket' => $basket,
    'req_email' => $customer['data_email'],
    'req_token_id' => $token
);

$result = Doku_Api::doPayment($dataPayment);

if($result->res_response_code == '0000'){
    echo 'SUCCESS';
    //success
}else{
    echo 'FAILED';
    //failed
}

```

After this is done, you may display the result on your browser for the customer to see. This result page is completely customizable according to the design that you prefer.



then, you can retrieve the JSON data like this. With this JSON data, you can acknowledge for updating the payment status of transaction.

MIPPayment.processRequest ACKNOWLEDGE :

```
{
  "res_approval_code": "681905",
  "res_edu_status": "NA",
  "res_trans_id_merchant": "124",
  "res_amount": "29575.04",
  "res_payment_date_time": "20160812145059",
  "res_verify_score": "-1",
  "res_verify_id": "",
  "res_verify_status": "NA",
  "res_words": "7c3d05dcad40ec7b2d41d2f75e2a896ff3e898d9",
  "res_response_msg": "SUCCESS",
  "res_mcn": "548117*****6802",
  "res_mid": "000002187008751",
  "res_bank": "PT. BANK CIMB NIAGA TBK.",
  "res_response_code": "0000",
  "res_session_id": "b2c981931fcff3c3f6db89f1a7c33656b10b11a6",
  "res_payment_channel": "15"}
}
```

2.1 Advanced Features

2.1.1 BIN Filtering

BIN filtering is a feature that can be used to filter Credit Cards issued by certain banks for the purpose of doing promotions. Each issuer has a unique BIN number, which is made up of the first 6 digits in the Credit Card number. The conditions you set in the filter will specify which BIN numbers that are allowed to make payments on your site. When a card number that has been blocked by the BIN filter is entered, the DOKU server will not be able to process the payment.

In order to apply the BIN filter, you must insert the following conditions into a prepayment request. The prepayment request must be sent before the actual payment request. You can add several conditions by separating with a comma. See example:

```
<?php
require_once('../Doku.php');

Doku_Initiate::$sharedKey = '<Put Your Shared Key Here>';
Doku_Initiate::$mallId = '<Put Your Merchant Code Here>';

$data = array('req_token_id' => $data['doku-token'],
             'req_pairing_code' => $data['doku-pairing-code'],
             'req_bin_filter' => array("42*1", "4?3??3", "411111", "5*"),
             'req_words' => $words
            );

$responsePrePayment = Doku_Api::doPrePayment($data);

if($responsePrePayment->responseCode == '0000'){
    //prepayment success
    // Pre payment success, then you can do payment
}else{
    // Pre payment fail
}
```

Afterwards, you may insert your original payment request form. Refer to the Appendix for prepayment response codes.

2.1.2 2-Click Payment [Tokenization]

2-click payment enables the customer to make a purchase without having to input card details or personal information, apart from the CVV number. This process is typically used by merchants that have repeat customers who will benefit from a faster checkout by reducing the number of fields the customer needs to fill in. If the card issuer requires 3D secure verification process, the customer will still have to complete this to make a purchase. In order for this process to work, the customer enters all of the card information only during the very first time they make a purchase. DOKU stores this data in a secure form and gives the merchant a token, which is paired to the customer's login credentials on the merchant website. After this process, has been completed, each time they make a payment from hereon out, they only have to input the CVV.

Follow these steps to apply 2-click payment to your credit card payment process:

1. Insert the additional script to your server under the payment data.
 2. Generate and save the token during the first payment.
 3. For subsequent payments, retrieve the token from your database and send it to the DOKU server.
1. To initialize 2-click payment, follow the same steps as general credit card processing, but add the additional parameter "data.reg_customer_id" to your javascript under the payment data. The "data.reg_customer_id" parameter may represent the customer ID that you assign to each customer within your database. This ID will be paired with the token that DOKU gives in the status response.

```
<script type="text/javascript">
$(function() {
    var data = new Object();
    data.req_merchant_code = '1';
    data.req_chain_merchant = 'NA';
    data.req_payment_channel = '15';
    data.req_transaction_id = '<?php echo $invoice ?>';
    data.req_currency = '<?php echo $currency ?>';
    data.req_amount = '<?php echo $amount ?>';
    data.req_words = '<?php echo $words ?>';
    data.req_form_type = 'full';
    data.req_customer_id = '12124';
    getForm(data);
});
</script>
```

The javascript with the additional parameter will generate the following payment form, which enables the customer to save their credit card, for faster payment. When a transaction is sent as a 2-Click payment to DOKU, in addition to the 4 fields for credit card data, DOKU will also display on the merchant website a tick box asking the customer's approval to save the card.

2. When the customer has filled in their card details and clicked "Process Payment", the data is sent to the DOKU server. Because the 'data.reg_customer_id' parameter has been added to the payment form, the DOKU server will create a token to pair with the Customer ID. If the customer checks the box next to "Save this Credit Card for faster checkout?" the payment response to the Merchant server will include this token. See example response:

```
{
  "res_approval_code": "844647",
  "res_trans_id_merchant": "1706221359",
  "res_amount": "30000.00",
  "res_payment_date_time": "20160319114638",
  "res_verify_score": "-1",
  "res_verify_id": "",
  "res_verify_status": "NA",
  "res_words": "7553a51a091775a2462eb9150c7135f4a8d58ff161db022ca42e0ef65666ebf0",
  "res_response_msg": "SUCCESS",
  "res_mcn": "4*****1111",
  "res_mid": "000100013000195",
  "res_bank": "JPMORGAN CHASE BANK",
  "res_response_code": "0000",
  "res_session_id": "4cf212f141a1d7fe672db93db75cc069, PRODUCTION",
  "res_payment_channel": "15",
  "res_bundle_token": {
    "res_token_payment": "0bea1c1c653dbc8e1e6c24155c629fe237325a06",
    "res_token_msg": "SUCCESS",
    "res_token_code": "0000"
  }
}
```

When the payment response is received, store it in your database for the next payment using the 2-Click service.

3. After a successful first payment, (assuming that the merchant has been correctly storing the Token data) only a slight modification needs to be made to the javascript. Add the extra parameter 'data.reg_token_payment' as seen below, by using the token value that was obtained during the first payment:

```

<script type="text/javascript">
$(function() {
    var data = new Object();
    data.req_merchant_code = '1';
    data.req_chain_merchant = 'NA';
    data.req_payment_channel = '15';
    data.req_transaction_id = '<?php echo $invoice ?>';
    data.req_currency = '<?php echo $currency ?>';
    data.req_amount = '<?php echo $amount ?>';
    data.req_words = '<?php echo $words ?>';
    data.req_form_type = 'full';
    data.req_customer_id = '12124';
    data.req_token_payment = '0bea1c1c653dbc8e1e6c24155c629fe237325a06';
    getForm(data);
});
</script>

```

The above script will generate the following payment form:

As you can see from the screenshot above, the customer no longer needs to fill out the credit card data apart from the CVV number. When the customer clicks the “Process Payment” button, it will follow the same process as regular card payments.

2.1.3 1-Click Payment [Recurring]

Using the same principles as 2-Click Payment, 1-Click payment takes it a step further and allows the customer to make a purchase with a single click on the website. This means that they can skip the process of inputting their card details, personal information, CVV number *and* 3D secure. The customer will have to enter the card details and complete the 3D secure verification process only during the first time they make a purchase. By eliminating the extra steps, you are able to create a more seamless and easy checkout process, which may lead to a lower drop-off rate. However, please note that this is **subject to DOKU's and the bank's approval due to an increase in fraud risk**. Please contact DOKU if you are interested to implement the 1-Click Payment feature.

Follow these steps to apply 1-click payment to your credit card payment process:

1. Insert the additional script to your server under the payment data.
2. Generate and save the token during the first payment.
3. For subsequent payments, retrieve the token from your database and send it to the DOKU server.
4. Change method of payment in the payment request form.

1. To initialize 1-click payment, follow the same steps as general credit card processing, but add the additional parameter "data.reg_customer_id" to your javascript under the payment data. The "data.reg_customer_id" parameter may represent the customer ID that you assign to each customer within your database. This ID will be paired with the token that DOKU gives in the status response.

```
<script type="text/javascript">
$(function() {
  var data = new Object();
  data.req_merchant_code = '1';
  data.req_chain_merchant = 'NA';
  data.req_payment_channel = '15';
  data.req_transaction_id = '<?php echo $invoice ?>';
  data.req_currency = '<?php echo $currency ?>';
  data.req_amount = '<?php echo $amount ?>';
  data.req_words = '<?php echo $words ?>';
  data.req_form_type = 'full';
  data.req_customer_id = '12124';
  getForm(data);
});
</script>
```

The javascript with the additional parameter will generate the following payment form, which enables the customer to save their credit card.

Credit Card VISA MasterCard

CC CVV

NAME EXP

Save this credit card for faster checkout?

2. When the customer has filled in their card details and clicked "Process Payment", the data is sent to the DOKU server. Because the 'data.reg_customer_id' parameter has been added to the payment form, the DOKU server will create a token to pair with the Customer ID. If the customer checks the box next to "Save this credit card for faster checkout?", the payment response to the Merchant server will include this token. See example response:

```

{
  "res_approval_code": "844647",
  "res_trans_id_merchant": "1706221359",
  "res_amount": "3000.00",
  "res_payment_date_time": "20160319114638",
  "res_verify_score": "-1",
  "res_verify_id": "",
  "res_verify_status": "NA",
  "res_words": "7553a51a091775a2462eb9150c7135f4a8d58ff161db022ca42e0ef65666ebf0",
  "res_response_msg": "SUCCESS",
  "res_mcn": "4*****1111",
  "res_mid": "000100013000195",
  "res_bank": "JPMORGAN CHASE BANK",
  "res_response_code": "0000",
  "res_session_id": "4cf212f141a1d7fe672db93db75cc069, PRODUCTION",
  "res_payment_channel": "15",
  "res_bundle_token": {
    "res_token_payment": "0bea1c1c653dbc8e1e6c24155c629fe237325a06",
    "res_token_msg": "SUCCESS",
    "res_token_code": "0000"
  }
}

```

When the payment response is received, store it in your database for the next payment using the 1-Click service.

- After a successful first payment, (assuming that the merchant has been correctly storing the Token data) only a slight modification needs to be made to the javascript. Add the extra parameter 'data.reg_token_payment' as seen below, by using the token value that was obtained during the first payment:

```

<script type="text/javascript">
$(function() {
  var data = new Object();
  data.req_merchant_code = '1';
  data.req_chain_merchant = 'NA';
  data.req_payment_channel = '15';
  data.req_transaction_id = '<?php echo $invoice ?>';
  data.req_currency = '<?php echo $currency ?>';
  data.req_amount = '<?php echo $amount ?>';
  data.req_words = '<?php echo $words ?>';
  data.req_form_type = 'full';
  data.req_customer_id = '12124';
  data.req_token_payment = '0bea1c1c653dbc8e1e6c24155c629fe237325a06';
  getForm(data);
});
</script>

```

The above script will not generate another payment form, but will instead just display the "Process Payment" button.

- Once your customer has input their card data and clicked the "Process Payment" button, the data will be processed and DOKU will send a response containing the card ID. When you retrieve this data from your server, create the payment request following the instructions for regular credit card payments. However, when you make the payment request, you must change the method of payment to `Doku_Api::doDirectPayment` instead of `Doku_Api::doPayment`. See example:

```

<?php
require_once('../Doku.php');

Doku_Initiate::$sharedKey = '<Put Your Shared Key Here>';
Doku_Initiate::$mallId = '<Put Your Merchant Code Here>';

$params = array(
    'amount' => '100000.00',
    'invoice' => '123456789'
);

$words = Doku_Library::doCreateWords($params);

$customer = array(
    'name' => 'TEST NAME',
    'data_phone' => '08121111111',
    'data_email' => 'test@test.com',
    'data_address' => 'bojong gede #1 08/01'
);
$basket[] = array(
    'name' => 'sayur',
    'amount' => '10000.00',
    'quantity' => '1',
    'subtotal' => '10000.00'
);
$basket[] = array(
    'name' => 'buah',
    'amount' => '10000.00',
    'quantity' => '1',
    'subtotal' => '10000.00'
);
$dataPayment = array(
    'req_mall_id' => Doku_Initiate::$mallId,
    'req_chain_merchant' => 'NA',
    'req_amount' => $params['amount'],
    'req_words' => $words,
    'req_purchase_amount' => $params['amount'],
    'req_trans_id_merchant' => $params['invoice'],
    'req_request_date_time' => date('YmdHis'),
    'req_currency' => '360',
    'req_purchase_currency' => '360',
    'req_session_id' => sha1(date('YmdHis')),
    'req_name' => $customer['name'],
    'req_payment_channel' => '15',
    'req_email' => $customer['data_email'],
    'req_basket' => $basket,
    'req_address' => $customer['data_address'],
    'req_token_payment' =>
'0bealc1c653dbc8e1e6c24155c629fe237325a06',
    'req_customer_id' => '12345'
);
$response = Doku_Api::doDirectPayment($dataPayment);

if($response->res_response_code == '0000'){
    echo 'PAYMENT SUCCESS -- ';
}else{
    echo 'PAYMENT FAILED -- ';
}

?>

```

2.1.4 Installment

Installment allows your customers to pay in plans using their credit cards. To enable this feature, you will need an agreement with the participating banks; please contact DOKU sales representative for assistance. This feature applies to both cards issued by banks that have their own installment product (*on-us*



installment) and those without installment product (*off-us installment*), each with different installment plans and rate. There is only a single integration process that you need to go through to enable both types of installment:

1. Create an installment form on your payment page

Installment integration follows the same steps as a normal credit card payment, only with a few additional parameters. To start, please create an installment selection form on your payment page for your customer to choose their card issuer and preferred payment plan along with the doku.js credit card form. The installment form may look like this:

2. Send Payment Request

There is no difference in the parameters sent from doku.js to your server, please do it as you would for a normal credit card payment. However, pair the credit card payment request parameters together with the installment information as you would need to combine the two when conducting the server-to-server payment charging process.

3. Charge payment to DOKU server along with the installment parameters

Send a payment charge request to DOKU using the usual DOKU library, only with the additional installment parameters. **installment_acquirer** refers to the acquirer bank code, **tenor** is the number of months in the installment plan, **plan_id** is a promotion code the merchant has from the bank, and **installment_off_us** with value 'O' only for off-us transaction.

```
$dataPayment = array(
    'req_mall_id' => Doku_Initiate::$mallId,
    'req_chain_merchant' => 'NA',
    'req_amount' => '10000.00',
    'req_words' => $words,
    'req_purchase_amount' => '10000.00',
    'req_trans_id_merchant' => $invoice_no,
    'req_request_date_time' => date('YmdHis'),
    'req_currency' => '360',
    'req_purchase_currency' => '360',
    'req_session_id' => shal(date('YmdHis')),
    'req_name' => $customer['name'],
    'req_payment_channel' => 15,
    'req_basket' => $basket,
    'req_email' => $customer['data_email'],
    'req_token_id' => $token
    'req_installment_acquirer' => '100' → from merchant
    'req_tenor' => '12', → from merchant
    'req_plan_id' => '001' → from merchant
    'req_installment_off_us' => 'O' → only for off-us transaction
)
```

2.1.5 Authorize & Capture

Authorize & Capture is a feature that allows you to block a certain amount from the customer's credit card limit (*Authorize*), then hold it for a certain period before charging a payment – which can be a different amount from what you block (*Capture*). To enable this feature, you will need an agreement with the acquiring bank; please contact DOKU sales representative for more detail.

The integration process for Authorize & Capture works as follow:

1. Authorize

To send a payment authorization request, follow the exact same steps as a normal credit card payment charging request, with additional parameters that you send during the server-to-server process:

```
$dataPayment = array(
    'req_mall_id' => Doku_Initiate::$mallId,
    'req_chain_merchant' => 'NA',
    'req_amount' => '10000.00',
    'req_words' => $words,
    'req_purchase_amount' => '10000.00',
    'req_trans_id_merchant' => $invoice_no,
    'req_request_date_time' => date('YmdHis'),
    'req_currency' => '360',
    'req_purchase_currency' => '360',
    'req_session_id' => sha1(date('YmdHis')),
    'req_name' => $customer['name'],
    'req_payment_channel' => 15,
    'req_basket' => $basket,
    'req_email' => $customer['data_email'],
    'req_token_id' => $token
    'req_authorize_expiry_date' => '1440' → from merchant,
    per minute (optional)
    'req_payment_type' => 'A',
)
```

req_authorize_expiry_date refers to the time limit (in minutes) that you want to impose on authorizing a certain amount of the card limit; the maximum is 15 days. While the value **'A'** for **req_payment_type** refers to an Authorize request. During this process DOKU will check whether there is still enough limit on the card to Authorize the amount.

After successfully completing the Authorize process, you will receive an **approval_code** parameter. Store and pair it with your **trans_id_merchant** parameter to later complete the Capture process.

2. Capture

To capture, simply send a server-to-server charging request to DOKU as shown by example below. You do not have to follow the full credit card steps since the customer does not have to fill in their card details anymore and getToken process can be bypassed.

```

$dataPayment = array(
    'req_mall_id' => Doku_Initiate::$mallId,
    'req_chain_merchant' => 'NA',
    'req_amount' => '10000.00',
    'req_words' => $words,
    'req_trans_id_merchant' => $invoice_no,
    'req_request_date_time' => date('YmdHis'),
    'req_session_id' => sha1(date('YmdHis')),
    'req_payment_channel' => 15,
    'req_approval_code' => '', → based on
    transaction authorize
)

```

3.0 DOKU Wallet

Doku Wallet integration comprises 3 easy steps:

1. Insert JavaScript
2. Initiate JavaScript parameters
3. Create payment form

To get started on your integration, follow these steps one by one by pasting the template scripts onto your website. The DOKU payment form is saved in HTML format; the template script acts as a placeholder for where the payment form will appear on your page. Prepare your HTML pages as seen in the template, and customize accordingly.

1. Insert the doku.js, fancybox.js and fancybox.css onto your website's payment page, along with your custom style. See example:

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.pack.js"></script>
<link href="https://cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.min.css"
rel="stylesheet">

<script src="http://staging.doku.com/doku-js/assets/js/doku.js"></script>
<link href="http://staging.doku.com/doku-js/assets/css/doku.css" rel="stylesheet">

```

2. Initialize the payment form by creating the words using doku_library...

```

<?php
require_once('../Doku.php');

Doku_Initiate::$sharedKey = '<Put Your Shared Key Here>';
Doku_Initiate::$mallId = '<Put Your Merchant Code Here>';

$invoice = 'invoice_1458123951';

$params = array(
    'amount' => '10000.00',
    'invoice' => $invoice,
    'currency' => '360'
);

$words = doku_Library::doCreateWords($params);
?>

```

...and adding the following example script on to your webpage:

```
<script type="text/javascript">
$(function() {
  var data = new Object();
  data.req_merchant_code = '1';
  data.req_chain_merchant = 'NA';
  data.req_payment_channel = '04';
  data.req_transaction_id = '<?php echo $invoice ?>';
  data.req_currency = '<?php echo $currency ?>';
  data.req_amount = '<?php echo $amount ?>';
  data.req_words = '<?php echo $words ?>';
  data.req_form_type = 'full';

  getForm(data);
});
</script>
```

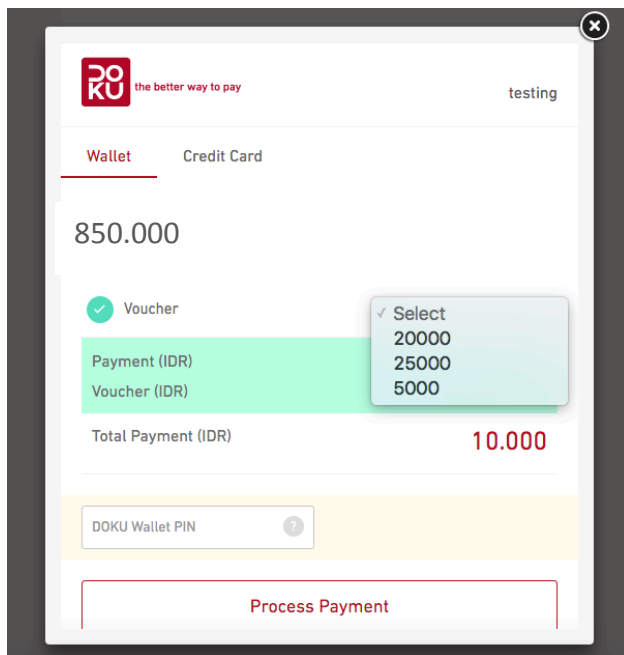
Remember to use **'04'** for the **payment_channel** parameter, which refers to DOKU Wallet.

3. Insert the template script where you wish the payment form to appear on your website.

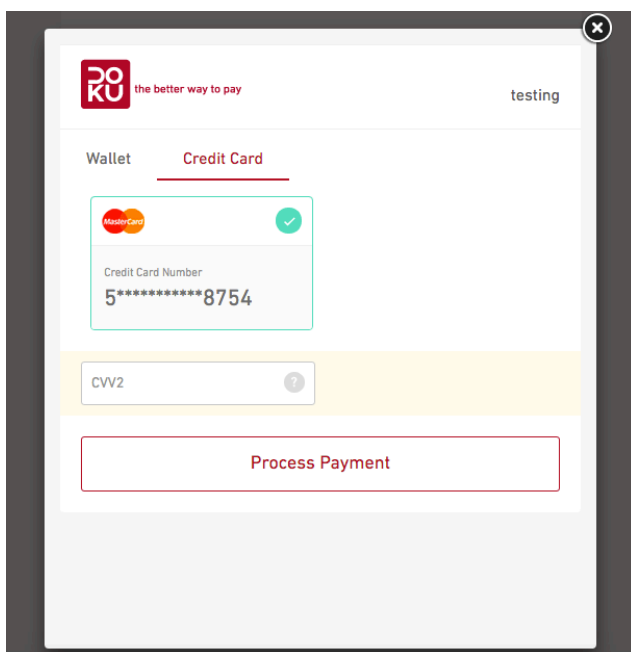
```
<form action="charge.php" method="POST" id="payment-form">
  <div doku-div='form-payment'>
    <input id="doku-token" name="doku-token" type="hidden" />
    <input id="doku-pairing-code" name="doku-pairing-code" type="hidden" />
  </div>
</form>
```

When completed correctly, the payment form should appear as seen below, and you may start receiving payments immediately. The DOKU script only provides the two fields to be filled in by the customer. All other parts of the payment page, including 'Process Payment' button are customizable to your needs.

Once your customer has input their DOKU ID and signed in to their account, their balance and payment options will appear in a pop-up window, which follows the DOKU template, like this:



The customer may choose to pay with their existing cash balance and may apply a Voucher or Promo Code if he/she has a valid one. The customer may also choose to pay with the Credit Card that has been linked to their DOKU Wallet. Note however, that to activate the credit card option, further process is required. If the Credit Card option is not yet activated for you and you would like to do so, please contact DOKU.



When the customer clicks the "Process Payment" button, the DOKU server will process the data and send a response to the browser. The browser will then post the data to the merchant server, according to the URL set in your action form.

You can retrieve the posted data from the form within your server. Then send the payment request to the DOKU server to be processed using our library. See example:

```
<?php
require_once('../Doku.php');

Doku_Initiate::$sharedKey = '<Put Your Shared Key Here>';
Doku_Initiate::$mallId = '<Put Your Merchant Code Here>';

$token = $_POST['doku-token'];
$pairing_code = $_POST['doku-pairing-code'];
$invoice_no = $_POST['doku-invoice-no'];

$params = array(
    'amount' => '10000.00',
    'invoice' => $invoice_no,
    'currency' => '360',
    'pairing_code' => $pairing_code,
    'token' => $token
);

$words = Doku_Library::doCreateWords($params);

$basket[] = array(
    'name' => 'sayur',
    'amount' => '10000.00',
    'quantity' => '1',
    'subtotal' => '10000.00'
);

$customer = array(
    'name' => 'TEST NAME',
    'data_phone' => '08121111111',
    'data_email' => 'test@test.com',
    'data_address' => 'bojong gede #1 08/01'
);

$data = array(
    'req_token_id' => $token,
    'req_pairing_code' => $pairing_code,
    'req_customer' => $customer,
    'req_basket' => $basket,
    'req_words' => $words
);

$responsePrePayment =
Doku_Api::doPrePayment($data);

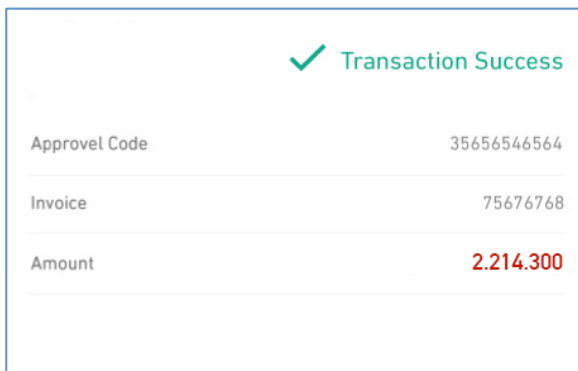
if($responsePrePayment->res_response_code ==
'0000'){
//prepayment success

    $dataPayment = array(
        'req_mall_id' => '1',
        'req_chain_merchant' => 'NA',
        'req_amount' => '10000.00',
        'req_words' => $words,
        'req_purchase_amount' => '10000.00',
        'req_trans_id_merchant' => $invoice_no,
        'req_request_date_time' =>
date('YmdHis'),
        'req_currency' => '360',
        'req_purchase_currency' => '360',
        'req_session_id' => shal(date('YmdHis')),
        'req_name' => $customer['name'],
        'req_payment_channel' => 04,
        'req_basket' => $basket,
        'req_email' => $customer['data_email'],
        'req_token_id' => $token
    );

    $result = Doku_Api::doPayment($dataPayment);

    if($result->res_response_code == '0000'){
        echo 'SUCCESS';
        //success
    }else{
        echo 'FAILED';
        //failed
    }
}
}
//prepayment fail
```

After this, you may display the result on your browser for the customer to see.



4.0 Virtual Account

DOKU Virtual Account aggregates the funds using 3 different entities – Bank Permata, Bank Sinarmas, Alfa Group and Indomaret. When the customer clicks 'Process Payment', DOKU will generate a one-time use, 11 digit payment code which is valid at any Prima, ALTO or Bersama ATM as well as all of Alfa Group's and Indomaret convenience stores. For each of the different acquiring entities, the first 5 digit codes will define where the payment should be made. See codes below:

Bank / Store	First 5 digit code
Permata	89650
Mandiri	88899
Sinarmas	88900
CIMB Niaga	51490
AlfaGroup	88888
Indomaret	88xxx

So a payment code that is valid for payment at an Alfa store would look like this: 88888-39421877483. And a bank transfer with Permata acquiring would look like this: 89650-39421877483.

Integration for ATM transfer and convenient store is practically identical; however, keep in mind that you will have to set the first 5 digits according to the payment method, and the last 11 digits will be queried from the DOKU server.

4.1 Bank Transfer

Follow these simple steps for Bank Transfer integration:

1. Generate Payment Code
2. Display Payment Code in your browser
3. Receive Payment Notification
4. Notify DOKU server that Payment Notification has been received

Notes: For interface, will be show from merchant website

1. By using the DOKU PHP Library, you can make a payment code request with ease. The request process is performed host to host. Examples of the request is seen below:

```
<?php
require_once('../Doku.php');

date_default_timezone_set('Asia/Jakarta');
Doku_Initiate::$sharedKey = <Put Your Shared Key Here>;
Doku_Initiate::$smallId = <Put Your Merchant Code Here>

$params = array(
    'amount' => $_POST['amount'],
    'invoice' => $_POST['trans_id'],
    'currency' => $_POST['currency']
);

$words = Doku_Library::doCreateWords($params);

$customer = array(
    'name' => 'TEST NAME',
    'data_phone' => '08121111111',
    'data_email' => 'test@test.com',
    'data_address' => 'bojong gede #1 08/01'
);

$dataPayment = array(
    'req_mall_id' => $_POST['mall_id'],
    'req_chain_merchant' => $_POST['chain_merchant'],
    'req_amount' => $params['amount'],
    'req_words' => $words,
    'req_trans_id_merchant' => $_POST['trans_id'],
    'req_purchase_amount' => $params['amount'],
    'req_request_date_time' => date('YmdHis'),
    'req_session_id' => sha1(date('YmdHis')),
    'req_email' => $customer['data_email'],
    'req_name' => $customer['name'],
    'req_basket' => 'sayur,10000.00,1,10000.00;',
    'req_address' => 'Plaza Asia Office Park Unit 3 Kav 59',
    'req_mobile_phone' => '081987987999'
    'req_expiry_time' => 'default 360' (optional)
);

$response = Doku_Api::doGeneratePaycode($dataPayment);

if($response->res_response_code == '0000'){
    echo 'GENERATE SUCCESS -- ';
}else{
    echo 'GENERATE FAILED -- ';
}
?>
```

The parameter 'req_expiry_time' refers to the custom expiry window for the payment to be made. Exceeding this time limit will render the payment code invalid. You may set the time limit however you like, in *minute format*. If you do not set the expiry time parameter, DOKU will set it at the default time of 360 minutes (6 hours).

The DOKU server responds in JSON, like this:




```
{
  "res_pay_code": "6270000003",
  "res_pairing_code": "290316110837531987",
  "res_response_msg": "SUCCESS",
  "res_response_code": "0000"
}
```

2. Display the result in your browser however you wish. Remember to add the Permata/Sinarmas code in front of the payment code. If you want to choose on behalf of the merchant, which bank they are transferring to, the result can be displayed like this:

KODE PEMBAYARAN & DETAIL TRANSAKSI ATM

Kode Pembayaran
8965 - 0506 - 0000 - 0012

Cara Membayar di ATM

1. Masukkan PIN
2. Pilih "TRANSFER". Apabila menggunakan ATM BCA, pilih "Transaksi lainnya" lalu "Transfer"
3. Pilih "KE REK BANK LAIN"
4. Masukkan Kode Bank Permata (013) kemudian tekan "Benar"
5. Masukkan Jumlah pembayaran sesuai dengan yang ditagihkan (Jumlah yang ditransfer harus sama persis tidak boleh lebih dan kurang) Jumlah nominal yang tidak sesuai dengan tagihan akan menyebabkan transaksi gagal.
6. Isi Nomor Rekening tujuan dengan 16 digit Nomor Kode Pembayaran anda 8965 - 0506 - 0000 - 0012
7. Muncul Layar Konfirmasi Transfer yang berisi nomor rekening tujuan Bank Permata dan Nama beserta jumlah yang dibayar, jika sudah benar, Tekan "Benar".
8. Selesai.

Cara Membayar Melalui Internet Banking

1. Login ke dalam akun Internet Banking
2. Pilih transfer ke bank lainnya
3. Pilih Bank Permata (013) sebagai rekening tujuan
4. Masukkan jumlah pembayaran sesuai dengan yang di tagihkan
5. Isi nomor rekening tujuan dengan 16 digit kode pembayaran anda 8965 - 0506 - 0000 - 0012
6. Muncul layar konfirmasi Transfer yang berisi nomor rekening tujuan Bank Permata dan Nama beserta jumlah yang dibayar. Jika sudah benar, tekan "Benar".
7. Selesai.
8. *Note: Pembayaran tidak bisa dilakukan melalui BCA Internet Banking*

INVOICE	lyKmGPI2v8kn
AMOUNT	IDR 10,000

Notes: for template, it depends on merchant requested.

Otherwise, you can display all three options like this:

KODE PEMBAYARAN :

	89650 - 00900000417
	88888 - 00900000417
	88220 - 00900000417
	88900 - 00900000417

3. Once the customer has made a payment, DOKU will send a payment notification containing the payment parameters to your server. The notification sent from DOKU will look something like this:

```
PAYMENTDATETIME=20160329110948
PURCHASECURRENCY=360
PAYMENTCHANNEL= FROM DOKU
AMOUNT=10000.00
PAYMENTCODE=00100000029
WORDS=01d9b362d3c1b80ff9196c6a565c49e5d9b03b8a
RESULTMSG=SUCCESS
TRANSIDMERCHANT=ZA912172
BANK=PERMATA
STATUSTYPE=P
APPROVALCODE=068992
RESPONSECODE=0000
SESSIONID=7b6647973dd13211a7fcf42eba79acea68aa69a1
```

4. Notify the DOKU server that you have received the payment notification, using the following example script:

```
<?php
$PAYMENTDATETIME = $_POST['PAYMENTDATETIME'];
$PURCHASECURRENCY = $_POST['PURCHASECURRENCY'];
$PAYMENTCHANNEL = $_POST['PAYMENTCHANNEL'];
$AMOUNT = $_POST['AMOUNT'];
$PAYMENTCODE = $_POST['PAYMENTCODE'];
$WORDS = $_POST['WORDS'];
$RESULTMSG = $_POST['RESULTMSG'];
$TRANSIDMERCHANT = $_POST['TRANSIDMERCHANT'];
$BANK = $_POST['BANK'];
$STATUSTYPE = $_POST['STATUSTYPE'];
$APPROVALCODE = $_POST['APPROVALCODE'];
$RESPONSECODE = $_POST['RESPONSECODE'];
$SESSIONID = $_POST['SESSIONID']

$WORDS_GENERATED = <function to generate words>

if ( $WORDS == $WORDS_GENERATED )
{
    echo "CONTINUE";
}
else
{
    echo "WORDS NOT MATCH";
}
?>
```

4.2 Convenience Store

Follow these simple steps for convenience store payment integration:

1. Generate Payment Code
 2. Display Payment Code in your browser
 3. Receive Payment Notification
 4. Notify DOKU server that Payment Notification has been received
1. By using the DOKU PHP Library, you can make a payment code request with ease. The request process is performed host to host. Examples of the request is seen below:

```
<?php
require_once('../Doku.php');

date_default_timezone_set('Asia/Jakarta');
Doku_Initiate::$sharedKey = '<Put Your Shared Key Here>';
Doku_Initiate::$smallId = '<Put Your Merchant Code Here>';

$params = array(
    'amount' => $_POST['amount'],
    'invoice' => $_POST['trans_id'],
    'currency' => $_POST['currency']
);

$words = Doku_Library::doCreateWords($params);

$customer = array(
    'name' => 'TEST NAME',
    'data_phone' => '08121111111',
    'data_email' => 'test@test.com',
    'data_address' => 'bojong gede #1 08/01'
);

$dataPayment = array(
    'req_mall_id' => Doku_Initiate::$smallId,
    'req_chain_merchant' => 'NA',
    'req_amount' => $params['amount'],
    'req_words' => $words,
    'req_trans_id_merchant' => $_POST['trans_id'],
    'req_purchase_amount' => $params['amount'],
    'req_request_date_time' => date('YmdHis'),
    'req_session_id' => sha1(date('YmdHis')),
    'req_email' => $customer['data_email'],
    'req_name' => $customer['name'],
    'req_basket' => 'sayur,10000.00,1,10000.00;',
    'req_address' => 'Plaza Asia Office Park Unit 3 Kav 59',
    'req_mobile_phone' => '081987987999'
    'req_expiry_time' => '60'
);

$response = Doku_Api::doGeneratePaycode($dataPayment);

if($response->res_response_code == '0000'){
    echo 'GENERATE SUCCESS -- ';
}else{
    echo 'GENERATE FAILED -- ';
}
?>
```

The parameter 'reg_expiry_time' refers to the custom expiry window for the payment to be made. Exceeding this time limit will render the payment code invalid. You may set the time limit however you like, in *minute format*.

The DOKU server responds in JSON, as follows:

```
{
  "res_pay_code": "62700000003",
  "res_pairing_code": "290316110837531987",
  "res_response_msg": "SUCCESS",
  "res_response_code": "0000"
}
```

2. Display the result in your browser however you wish, but we also recommend you include a "How to Pay" section to inform the customer of the steps they need to complete at the cashier. Remember to add the Alfa or Indomaret code in front of the payment code. For example, the result can be displayed like this:

KODE PEMBAYARAN & DETAIL TRANSAKSI MINI MARKET



Kode Pembayaran

8888 - 8506 - 0000 - 0015

1. Catat kode pembayaran di samping dan datang ke gerai Alfamart, Alfa Midi, Alfa Express, Lawson atau DAN+DAN terdekat.
2. Datangi kasir dan beritahukan kasir untuk membuka menu e-transaction terminal. pilih no. '2', no. '5' lalu no. '1' *-DOKU- pembayaran merchant DOKU*.
3. Kasir akan menanyakan kode pembayaran. Berikan kode pembayaran anda 8888 - 8506 - 0000 - 0015
4. Kasir akan menginformasikan nama Merchant dan nominal yang harus dibayarkan.
5. Lakukan pembayaran ke kasir sejumlah nominal yang disebutkan. Pembayaran dapat menggunakan uang tunai atau non tunai. Non tunai antara lain Kartu Debit BCA, Kartu Debit BNI, BCA Flazz, BNI Prepaid dan Mandiri e-money.
6. Terima struk sebagai bukti pembayaran sudah sukses dilakukan. Notifikasi pembayaran akan langsung diterima oleh Merchant.
7. Selesai.


INVOICE

AMOUNT

VL3y2iVwyxpy

IDR 10,000

Bahasa



Payment Code

8822000000000214

Amount

IDR 10000.00

Invoice Number

tl3uA7biqfRc

How to Pay at Indomaret

1. Take note of your payment code and go to your nearest Indomaret store
2. Tell the cashier that you wish to make a DOKU payment
3. If the cashier is unaware of DOKU, provide the instruction to

3. Once the customer has made a payment over the counter, DOKU will send a payment notification containing the payment parameters to your server. The notification sent from DOKU will look something like this:

```
PAYMENTDATETIME=20160329110948
PURCHASECURRENCY=360
PAYMENTCHANNEL= From Channel DoKu
AMOUNT=10000.00
PAYMENTCODE=00100000029
WORDS=01d9b362d3c1b80ff9196c6a565c49e5d9b03b8a
RESULTMSG=SUCCESS
TRANSIDMERCHANT=ZA912172
BANK=ALFA
STATUSTYPE=P
APPROVALCODE=068992
RESPONSECODE=0000
SESSIONID=7b6647973dd13211a7fcf42eba79acea68aa69a1
```

```
PAYMENTDATETIME=20160329110948
PURCHASECURRENCY=360
PAYMENTCHANNEL= From Channel DoKu
AMOUNT=10000.00
PAYMENTCODE=00100000029
WORDS=01d9b362d3c1b80ff9196c6a565c49e5d9b03b8a
RESULTMSG=SUCCESS
TRANSIDMERCHANT=ZA912172
BANK=INDOMARET
STATUSTYPE=P
APPROVALCODE=068992
RESPONSECODE=0000
SESSIONID=7b6647973dd13211a7fcf42eba79acea68aa69a1
```

4. Notify the DOKU server that you have received the payment notification, using the following example script:

```
<?php
$PAYMENTDATETIME = $_POST['PAYMENTDATETIME'];
$PURCHASECURRENCY = $_POST['PURCHASECURRENCY'];
$PAYMENTCHANNEL = $_POST['PAYMENTCHANNEL'];
$AMOUNT = $_POST['AMOUNT'];
$PAYMENTCODE = $_POST['PAYMENTCODE'];
$WORDS = $_POST['WORDS'];
$RESULTMSG = $_POST['RESULTMSG'];
$TRANSIDMERCHANT = $_POST['TRANSIDMERCHANT'];
$BANK = $_POST['BANK'];
$STATUSTYPE = $_POST['STATUSTYPE'];
$APPROVALCODE = $_POST['APPROVALCODE'];
$RESPONSECODE = $_POST['RESPONSECODE'];
$SESSIONID = $_POST['SESSIONID']

$WORDS_GENERATED = <function to generate words>

if ( $WORDS == $WORDS_GENERATED )
{
    echo "CONTINUE";
}
else
{
    echo "WORDS NOT MATCH";
}

?>
```

5.0 Internet Banking

Each bank has its own flow and authentication process for Internet Banking Payments. The majority of Internet Banking is hosted on the respective banks' own webpages where the customer enters his/her credentials and completes the authentication. So even though the initial payment steps will occur on the merchant page, it is redirected to the bank page eventually. Out of the Internet Banking facilities that are supported by DOKU currently, only Mandiri Clickpay allows for a merchant hosted flow.

5.1 Mandiri Clickpay

Mandiri Clickpay integration comprises 3 easy steps:

1. Insert JavaScript
2. Add JavaScript code
3. Create payment form

To get started on your integration, follow these steps one by one by pasting the template scripts onto your website. The DOKU payment form is saved in HTML format; the template script acts as a placeholder for where the payment form will appear on your page. Prepare your HTML pages as seen in the template, and customize accordingly.

The form you create in this step will receive the Mandiri Clickpay input from the customer.

1. Insert the doku.js onto your website's payment page, along with your custom style. See example:

```
<script src="doku.js"></script>

<link href="http://staging.doku.com/doku-js/assets/css/doku.css" rel="stylesheet">
<script src='http://staging.doku.com/doku-js/assets/js/jquery.payment.min.js'></script>
```

2. Initialize the payment form by adding the following example script onto your webpage.

```
<script type="text/javascript">

    jQuery(function($) {
        $(''.cc-number').payment('formatCardNumber');

        var challenge3 = Math.floor(Math.random() * 999999999);
        $('#CHALLENGE_CODE_3').val(challenge3);
    });

    $(function() {
        var data = new Object();
        data.req_cc_field = 'cc_number';
        data.req_challenge_field = 'CHALLENGE_CODE_1';

        dokuMandiriInitiate(data);
    });
</script>
```

3. Insert the template script where you wish the payment form to appear on your website.

```

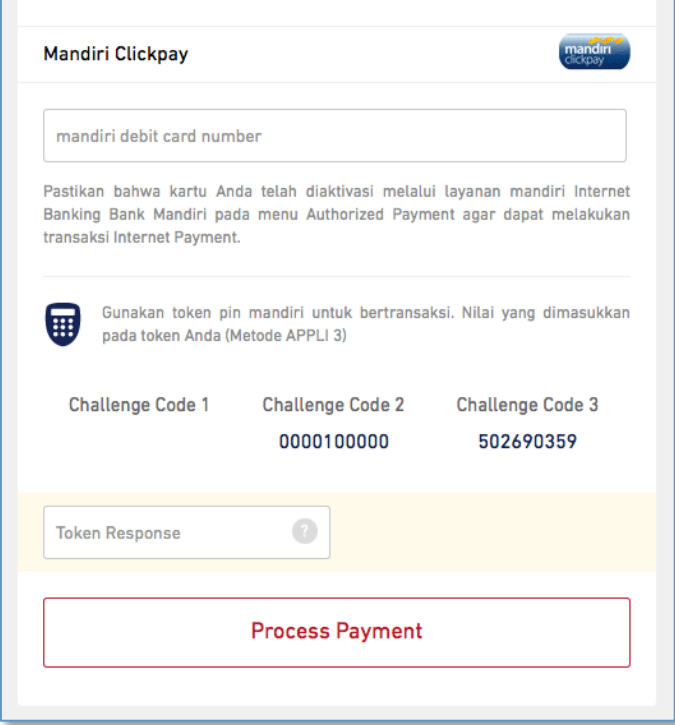
<form method="post" action="../example/mandiri-clickpay-charge.php">
<div id="mandiriclickpay" class="channel">

  <div class="list-chacode">
    <ul>
      <li>
        <div class="text-chacode">Challenge Code 1</div>
        <input type="text" id="CHALLENGE_CODE_1" name="CHALLENGE_CODE_1"
readonly="true" required/>
        <div class="clear"></div>
      </li>
      <li>
        <div class="text-chacode">Challenge Code 2</div>
        <div class="num-chacode">0000100000</div>
        <input type="hidden" name="CHALLENGE_CODE_2" value="0000100000"/>
        <div class="clear"></div>
      </li>
      <li>
        <div class="text-chacode">Challenge Code 3</div>
        <div class="num-chacode" id="challenge_div_3"></div>
        <input type="hidden" name="CHALLENGE_CODE_3"
id="CHALLENGE_CODE_3" value=""/>
        <div class="clear"></div>
      </li>
    </ul>
  </div>
  <div class="validasi">
    <div class="styled-input fleft width50">
      <input type="text" required="" name="response_token">
      <label>Respon Token</label>
    </div>
    <div class="clear"></div>
  </div>
  <input type="hidden" name="invoice_no" value="invoice_1458547984">
  <input type="button" value="Process Payment" class="default-btn"
onclick="this.form.submit();" />
</div>
</form>

```

Mandiri ClickPay WORDS = sha1
[AMOUNT+MALLID+<shared key> +TRANSIDMERCHANT+ CURRENCY]

When completed correctly, the payment form should appear as seen below, and you may start receiving payments using Mandiri Clickpay immediately.



The image shows a screenshot of the Mandiri Clickpay payment form. At the top left, it says "Mandiri Clickpay" and at the top right is the Mandiri Clickpay logo. Below the header is a text input field labeled "mandiri debit card number". Underneath is a paragraph of Indonesian text: "Pastikan bahwa kartu Anda telah diaktivasi melalui layanan mandiri Internet Banking Bank Mandiri pada menu Authorized Payment agar dapat melakukan transaksi Internet Payment." Below this is a section with a pin icon and the text: "Gunakan token pin mandiri untuk bertransaksi. Nilai yang dimasukkan pada token Anda (Metode APPLI 3)". This section contains three challenge codes: "Challenge Code 1", "Challenge Code 2" (0000100000), and "Challenge Code 3" (502690359). Below the challenge codes is a "Token Response" input field with a question mark icon. At the bottom is a large red button labeled "Process Payment".

Once your customer has input their Mandiri Clickpay data and clicked the "Process Payment" button, the data will be processed on DOKU. The DOKU server responds to the browser, and the data will be posted to the merchant server, according to the URL set in your action form.

You can retrieve the posted data from the form within your server. Then send the payment request to the DOKU server to be processed using our library. See example:


```

<?php
require_once('../Doku.php');

Doku_Initiate::$sharedKey = '<Put Your Shared Key Here>';
Doku_Initiate::$smallId = '<Put Your Merchant Code Here>';

$params = array(
    'amount' => '10000.00',
    'invoice' => $_POST['invoice_no']
);

$cc = str_replace(" - ", "", $_POST['cc_number']);
$words = Doku_Library::doCreateWords($params);

$customer = array(
    'name' => 'TEST NAME',
    'data_phone' => '08121111111',
    'data_email' => 'test@test.com',
    'data_address' => 'bojong gede #1 08/01'
);

$basket[] = array(
    'name' => 'sayur',
    'amount' => '10000.00',
    'quantity' => '1',
    'subtotal' => '10000.00'
);

$basket[] = array(
    'name' => 'buah',
    'amount' => '10000.00',
    'quantity' => '1',
    'subtotal' => '10000.00'
);

$dataPayment = array(
    'req_mall_id' => '1',
    'req_chain_merchant' => 'NA',
    'req_amount' => $params['amount'],
    'req_words' => $words,
    'req_purchase_amount' => $params['amount'],
    'req_trans_id_merchant' => $_POST['invoice_no'],
    'req_request_date_time' => date('YmdHis'),
    'req_currency' => '360',
    'req_purchase_currency' => '360',
    'req_session_id' => shal(date('YmdHis')),
    'req_name' => $customer['name'],
    'req_payment_channel' => '02',
    'req_email' => $customer['data_email'],
    'req_card_number' => $cc,
    'req_basket' => $basket,
    'req_challenge_code_1' => $_POST['CHALLENGE_CODE_1'],
    'req_challenge_code_2' => $_POST['CHALLENGE_CODE_2'],
    'req_challenge_code_3' => $_POST['CHALLENGE_CODE_3'],
    'req_response_token' => $_POST['response_token'],
    'req_mobile_phone' => $customer['data_phone'],
    'req_address' => $customer['data_address']
);

$response = Doku_Api::doDirectPayment($dataPayment);

if($response->res_response_code == '0000'){
    echo 'PAYMENT SUCCESS -- ';
}else{
    echo 'PAYMENT FAILED -- ';
}

var_dump($response);

?>

```

The DOKU server will send a response in JSON, like this:

```
{
  "res_response_msg": "SUCCESS",
  "res_transaction_code": "d4efbb8c4ebb9a3597c05aa32f2b341e77f98e63",
  "res_mcn": "4*****1111",
  "res_approval_code": "1234",
  "res_trans_id_merchant": "1706332101",
  "res_payment_date": "20160319184828",
  "res_bank": "MANDIRI CLICK PAY",
  "res_amount": "30000.00",
  "res_message": "PAYMENT APPROVED",
  "res_response_code": "0000",
  "res_session_id": "50d240541f4d8d7565b18cb5ca93a660"
}
```

After this is done, you may display the result on your browser for the customer to see.

6.0 Appendix

6.1 Payment Methods

Payment Type	Description
Credit Card	<ul style="list-style-type: none"> • Visa, Mastercard and AMEX for Overseas Partner. JCB upon request • Direct API available • Features (acquirer dependant): 3D and non 3D Secure, recur, authorize capture, online refund, tokenization, installment, BIN filtering, point rewards
Internet Banking	<ul style="list-style-type: none"> • Available: Mandiri Clickpay, • Each bank has different authentication process through OTP or token • Direct API only available for Mandiri Clickpay. The rest is re-direct only
DOKU Wallet	<ul style="list-style-type: none"> • E-wallet product issued by DOKU • Source of fund: cash balance, linked credit card, linked bank account • Max. balance is Rp1,000,000 for non-KYC and Rp10,000,000 for KYC users • Authenticate with email, password and static PIN that is pre-set by the user • Direct API available
Convenience Store	<ul style="list-style-type: none"> • Accessible in almost 12,000 Alfa group stores (Alfa Express, Alfa Midi, Alfa Mart, Lawson and DAN+DAN) and 14,000 Indomaret stores • Generate 16 digit payment code at checkout, user goes to nearest store and makes payment over the counter with cash or non cash • Max. cash transaction value of Rp5,000,000 • Merchant can set payment code expiry time for every transaction • Direct API available
Bank Transfer	<ul style="list-style-type: none"> • Virtual account housed in Bank Permata, Mandiri or SinarMas but payable from any bank that is connected to ATM Bersama, Prima or Alto networks (over 120 banks in Indonesia) • Generate 16 digit payment code at checkout, user makes payment via ATM or Internet/mobile banking that is connected to 1 of the 3 networks • Merchant can set payment code expiry time for every transaction • Direct API available

6.2 Payment Channel Codes

Code	Description
02	Mandiri ClickPay
04	DOKU Wallet
05	ATM Permata VA LITE
06	BRI e-Pay
07	ATM Permata VA
14	Alfagroup
15	Credit Card Visa/Master Multi Currency
16	Credit Card Tokenization
17	Recurring Payment
18	KlikPayBCA
19	CIMB Clicks
21	Sinarmas VA Full
22	Sinarmas VA Lite
23	MOTO

31	Indomaret
32	CIMB VA
41	Mandiri VA

6.3 Parameters Required

In this section of the Appendix, you will find the list of required parameters for the different APIs.

6.3.1 Payment Request

Credit Card Payment Request Parameter

No	Name	Type	Length	Comments	Mandatory
1	req_mall_id	N		Given by DOKU	Yes
2	req_chain_merchant	N		Given by DOKU, if not using Chain, default value is NA	Yes
3	req_amount	N	12.2	Total amount. Eg:10000.00	Yes
4	req_purchase_amount	N	12.2	Total amount. Eg:10000.00	Yes
5	req_trans_id_merchant	AN	...30	Transaction ID from Merchant	Yes
6	req_words	AN	...200	Hashed key combination encryption (use SHA1 method). The hashed key generated from combining these parameters value in this order : Amount + Mall ID+ <shared key> + Invoice No. + Purchase Currency + Token ID + Pairing Code	Yes
7	req_request_date_time	N	X	YYYYMMDDHHMMSS	Yes
8	req_currency	N	3	ISO3166 , numeric code	Yes
9	req_purchase_currency	N	3	ISO3166 , numeric code	Yes
10	req_session_id	AN	...48		Yes
11	req_name	AN	...50	Travel arranger name / buyer name	Yes
12	req_email	ANS	...100	Customer email	Yes
13	req_basket	ANS	...1024	Show transaction description. Use comma to separate each field and semicolon for each item. Item 1, 1000.00.,2,20000.00;item2,15000.00,2,30000.00	Yes
14	token_id	AN		Sent by DOKU during getToken process	Yes
15	req_shipping_address	ANS	...100	Shipping address contains street and number	Optional
16	req_shipping_city	ANS	...100	City name	Optional
17	req_shipping_state	AN	...100	State / province name	Optional
18	req_shipping_country	A	2	ISO3166 , alpha-2	Optional
19	req_shipping_zip_code	N	...10	Zip Code	Optional
20	req_payment_channel	N	2	See payment channel code list	Optional
21	req_address	ANS	...100	Home address contains street and number	Optional
22	req_city	ANS	...100	City name	Optional
23	req_state	AN	...100	State / province name	Optional
24	req_country	A	2	ISO3166, alpha-2	Optional
25	req_zip_code	N	...10	Zip Code	Optional
26	req_mobile_phone	ANS	...11	Home Phone	Yes
27	req_work_phone	ANS	...13	Work Phone / Office Phone	Optional
28	req_birth_date	N	...8	YYYYMMDD	Optional
INSTALLMENT					

29	req_installment_acquirer	N	3	Acquirer code for installment	Yes for Installment
30	req_tenor	N	2	Number of month to pay the installment	Yes for Installment
31	req_plan_id	N	3	Promotion ID from the bank for the current merchant	Yes for Installment
TOKENIZATION					
32	req_customer_id	AN	...16	Merchant's customer identifier	Yes for Tokenization
33	req_token_payment				Yes for Tokenization (1-click)
Authorize & Capture					
34	req_authorize_expiry_date	N	6	Expiry time of the payment Authorization (in minutes). Maximum is 15 days	Yes for Authorize
35	req_payment_type	N	...10	Please enter 'A' for Authorize	Yes for Authorize
36	req_approval_code			Received from DOKU during Authorize process	Yes for Capture

6.3.2 Check Status Request

Check transaction status for all type of payment channel

No	Name	Type	Length	Comments	Mandatory
1	req_mall_id	N		Given by DOKU	Yes
2	req_chain_merchant	N		Given by DOKU, if not using Chain, default value is NA	Yes
3	req_trans_id_merchant	AN	...30	Transaction ID from merchant	Yes
4	req_session_id	AN	...48		Yes
5	req_words	AN	...200	Hashed key combination encryption (use SHA1 method). The hashed key is generated from combining the parameter values in this order : req_mall_id+<shared key>+ req_trans_id_merchant	Yes

6.3.2 Check Status Response

Response of check status request will be in JSON format. Below is the sample:

```
{
  "res_currency": "360",
  "res_status_type": "P",
  "res_threeed_secure_status": "",
  "res_approval_code": "",
  "res_purchase_currency": "360",
  "res_edu_status": "NA",
  "res_trans_id_merchant": "1494246982571",
  "res_amount": "5000000.00",
  "res_payment_date_time": "20170508194517",
  "res_verify_score": "-1",
  "res_verify_id": "",
  "res_verify_status": "NA",
  "res_words": "08009f077a52d54dfa0443d58f99ec34227ddd49",
  "res_liability": "",
  "res_result_msg": "SUCCESS",
  "res_mcn": "000000000064",
  "res_bank": "DANAMON",
  "res_response_code": "0000",
  "res_payment_code": "8911000000000064",
  "res_session_id": "1494246982571",
  "res_payment_channel": "33",
  "res_brand": "" //for CC
}
```

6.2 DOKU Response Codes

In this section of the Appendix, you will find the list of response codes and their description for the different payment methods.

6.2.1 General response codes

The response codes listed in this section include both prepayment and payment response codes, and mostly apply to all payment methods. These are the most common response codes you will receive from DOKU.

Error Code	Description
0000	Successful approval
5555	Failed decrypt
5501	Payment channel not registered
5502	Merchant is disabled
5503	Maximum attempt 3 times
5504	Words not match
5505	Invalid parameter
5506	Notify failed
5507	Invalid parameter detected / Customer click cancel process
5508	Re-enter transaction
5509	Payment code already expired
5510	Cancel by Customer
5511	Not an error, payment code has not been paid by Customer
5512	Insufficient Parameter
5514	Reject by Fraud System
5515	Duplicate PNR
5516	Transaction Not Found
5517	Error in Authorization process
5518	Error parsing XML
5519	Customer stop at 3D Secure page
5520	Transaction Failed via scheduler
5521	Invalid Merchant
5522	Rates were not found
5523	Failed to get Transaction status
5524	Failed to void transaction
5525	Transaction can not be process
5526	Transaction is voided because timeout to wallet
5527	Transaction will be process as Off Us Instalment
5529	Invalid Merchant
5530	Internal server error
5531	Pairing Code does not exist
5532	Invalid Payment Channel
5533	Failed to inquiry list of fund
5534	Invalid Pairing Code
5535	Invalid Token

5536	Time Out
5537	Invalid Currency
5538	Invalid Purchase Currency
5539	3D Secure Enrolment check failed
5540	3D Secure Authentication failed
5541	Form Type is not valid
5542	Duplicate Transaction ID
5543	Please check 3D Secure result
5544	Failed to delete token
5545	Failed to Void
5547	BIN are not allowed in promo
5548	Invalid Parameter
5553	Failed to tokenize

6.2.2 Credit Card

The response codes in this section only apply to credit card transactions.

Error Code	VISA	MASTERCARD	ORIGIN	ACTIONS
0001	Refer to card issuer	Refer to card issuer	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0002	Refer to card issuer, special condition	-	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0003	Invalid merchant or service provider	Invalid Merchant	VISA/MASTER	Contact DOKU or acquiring bank
0004	Pickup card	Capture card	VISA/MASTER	Should consider blocking the card temporarily or Block login ID
0005	Do Not Honor	Do Not Honor	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0006	Error	-	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0007	Pickup card, special condition (other than lost/stolen card)	-	VISA/MASTER	Should consider blocking the card
0008	-	Honor with ID	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0010	Partial Approval - Private label	-	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0011	VIP Approval	-	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0012	Invalid Transaction	Invalid Transaction	VISA/MASTER	Contact DOKU or ACQUIRING BANK

0013	Invalid amount (currency conversion field overflow. Visa Cash - Invalid load mount)	Invalid Amount	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0014	Invalid account number (no such number)	Invalid Card Number	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0015	No such issuer	Invalid issuer	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0019	Re-enter transaction	-	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0021	No Action taken (unable to back out prior transaction)	-	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0025	Unable to locate record in file, or account number is missing from inquiry	-	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0028	File is temporarily unavailable	-	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0030	-	Format error	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0041	Pickup card (lost card)	Lost Card	VISA/MASTER	Should consider blocking the card temporarily or Block login ID
0043	Pickup card (stolen card)	Stolen Card	VISA/MASTER	Should consider blocking the card temporarily or Block login ID
0051	Insufficient funds	Insufficient Funds/Over Credit limit	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0052	No checking account	-	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0053	non savings account	-	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0054	Expired card	Expired Card	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0055	Incorrect PIN (Visa cash - invalid or missing SI signature)	Invalid PIN	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0057	Transaction not permitted to cardholder (Visa cash - incorrect routing, not a load request)	Transaction not permitted to issuer/cardholder	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0058	Transaction not allowed at terminal	Transaction not permitted to acquirer/terminal	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0061	Activity amount limit exceeded	Exceeds withdrawal amount limit	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.

0062	Restricted card (for example in country exclusion table)	Restricted Card	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0063	Security violation	Security Violation	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0065	Activity count limit exceeded	Exceeds withdrawal count limit	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0075	Allowable number of PIN-entry tries exceeded	Allowable number of PIN tries exceeded	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0076	Unable to locate previous message (no match on Retrieval Reference number)	Invalid/nonexistent "To Account" specified	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0077	Previous message located for a repeat or reversal, but repeat or reversal data are inconsistent with original message	Invalid/nonexistent "From account" specified	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0078	-	Invalid/nonexistent account specified (general)	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0080	invalid date (For use in private label card transactions and check acceptance transactions)	-	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0081	PIN Cryptographic error found (error found by VIC security module during PIN decryption)	-	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0082	Incorrect CW/1CW	-	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0083	Unable to verify PIN	-	VISA/MASTER	Tell Customer to contact the Bank Issuer of the card used.
0084	-	Invalid Authorization Life Cycle	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0085	No reason to decline a request for account number verification or address verification	Not Decline Valid for AVS only, balance inquiry, or SET Cardholder certificate requests [VISA Only]	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0091	Issuer unavailable or switch inoperative (STIP not applicable or available for this transaction)	Authorization System or issuer system inoperative	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0092	Destination cannot be found for routing	Unable to route transaction	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0093	Transaction cannot be completed; violation of law	-	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0094	-	Duplicate transmission detected	VISA/MASTER	Contact DOKU or ACQUIRING BANK
0096	System malfunction / System malfunction or certain field error conditions	System Error	VISA/MASTER	Contact DOKU or ACQUIRING BANK

00NO	Force STIP		VISA/MASTER	Contact DOKU or ACQUIRING BANK
00N3	Cash service not available		VISA/MASTER	Contact DOKU or ACQUIRING BANK
00N4	Cash request exceeds issuer limit		VISA/MASTER	Contact DOKU or ACQUIRING BANK
00N7	Decline for CW2 failure		VISA/MASTER	Contact DOKU or ACQUIRING BANK
00P2	Invalid biller information		VISA/MASTER	Contact DOKU or ACQUIRING BANK
00P5	PIN Change/Unblock request declined		VISA/MASTER	Contact DOKU or ACQUIRING BANK
00P6	Unsafe PIN		VISA/MASTER	Contact DOKU or ACQUIRING BANK
00TO	Timeout / Transaction's response exceed time limit	Timeout / Transaction's response exceed time limit	DOKU	Contact DOKU or ACQUIRING BANK
00UE	Unknown Exception / PosServer not responding	Unknown Exception / PosServer not responding	DOKU	Contact DOKU or ACQUIRING BANK

6.2.3 DOKU Wallet

The response codes in this section only apply to DOKU Wallet transactions.

Error Code	Description
0E01	FAILED GET MERCHANT
0E02	MASTER MERCHANT INACTIVE
0E03	INVALID WORDS FROM MERCHANT
0E04	INVALID MERCHANT
0E05	FAILED TO PROCESS PAYMENT
0E06	PAYMENT METHOD NOT DEFINE
0E07	FAILED EXECUTE PRE AUTH PLUGINS
0E08	FAILED EXECUTE POST AUTH PLUGINS
0E09	INVALID PAY ID
0E10	ERROR PAY ID
0E11	FAILED EXECUTE PRE TRANS MIP PLUGINS
0E12	VERIFY RESPONSE STOP FROM MERCHANT
0E13	FAILED VERIFY TO MERCHANT
0E14	FAILED SEND PAYMENT CASH WALLET
0E15	NOTIFY RESPONSE STOP FROM MERCHANT
0E16	FAILED NOTIFY TO MERCHANT
0E18	FAILED EXECUTE POST TRANS MIP PLUGINS
0E19	NOT ENOUGH CASH BALANCE AND DON'T HAVE CREDIT CARD
0E20	SPENDER NO HAVE LINK TO CREDIT CARD
0E21	ERROR CHECK 3D SECURE CREDIT CARD
0E22	PIN/OTP IS NOT VALID
0E23	PLEASE INPUT CVV2
0E24	INVALID SESSION
0E25	FAILED SEND LINK AUTHENTICATION TO CARD HOLDER
0E26	INSUFFICIENT PARAMS
0E27	FAILED EXECUTE PRE TRANS CIP PLUGINS
0E28	FAILED EXECUTE POST TRANS CIP PLUGINS
0E29	FAILED SEND PAYMENT MIP CREDIT CARD
0E30	YOU DO NOT HAVE PIN
0E31	DUPLICATE INVOICE NO
0E32	URL NOT FOUND
0E33	CUSTOMER NOT FOUND
0E34	VOID PROCESS FAILED
0E35	Failed Send ONE TIME PIN to your email
0E36	Failed Send Link for create PIN to your email
0E37	THIS SPENDER CAN'T TRANSACT IN THIS MERCHANT
0E38	You have reach your DOKU ID Transaction Limit
0E39	Process MIP Transaction Failed
0E99	ERROR SYSTEM

6.2.4 Virtual Account

The response codes in this section only apply to Convenience Store and Bank Transfer transactions.

Error Code	Description
0001	Decline (internal error)
0013	Invalid amount
0014	Bill not found
0066	Decline
0088	Bill already paid

6.2.5 Mandiri Clickpay

The response codes in this section only apply to Mandiri Clickpay transactions.

Error Code	Description
0001	Internal system error: cannot parse message
0002	Internal system error: unmatched signature hash
0003	Internal system error: Cannot process message
0004	Internal system error: Error on field
0005	Internal system error: Transaction not found
0006	Internal system error: Create VPA response error
0101	Internal system error: Create velis-authenticator message
0102	Internal system error: Runtime try/catch error when creating VTCPStream
0103	Internal system error: Cannot connect to velis-authenticator
0104	Internal system error: Send request to velis-authenticator failed
0105	Internal system error: Waiting response from velis-authenticator failed
0106	Internal system error: Read response from velis-authenticator failed
0107	Internal system error: Parse response from velis-authenticator failed
0108	Internal system error: Signature key from velis-authenticator is invalid
1101	User not registered: Channel not register in database (not found)
1102	User not registered: User not active
1103	User not registered: User has deleted
1104	User not registered: User not found
1105	User not registered: Channel for User not active
1106	User not registered: Channel for User has deleted - no access
1107	User not registered: Channel for User not register / not found
1108	User has blocked: User has disabled
1109	User has blocked
1110	User has blocked: Channel for User has disabled
1111	User has blocked: Channel for User has blocked
1112	User already activated: User has invalid status (or already active)
1113	User already activated: Channel for User has invalid status (or already active)
1114	Invalid token: Token of User not active
1115	Invalid token: Token of User has disable
1116	Invalid token: Token of User has deleted

1117	Invalid token: Token of User not found
1118	Invalid token: Method CR not allowed for Token of User
1119	Invalid token: Method RO not allowed for Token of User
1120	Invalid token: Method SG not allowed for Token of User
1121	Invalid token: Device Token Type not valid (only support VS = VASCO Token)
1122	Invalid token response: Code Not Verified
1123	Invalid token response: Code Replay Attempt
1124	Invalid token response: Challenge Too Small
1125	Invalid token response: Challenge Too Long
1126	Invalid token response: Challenge Check Digit Wrong (Host Check Challenge Mode)
1127	Invalid token response: Challenge Character Not Decimal
1128	Invalid token response: Challenge Corrupt (Host Check Challenge Mode)
1129	Invalid token response: Response Length Out of Bounds
1130	Invalid token response: Response Too Small
1131	Invalid token response: Response Too Long
1126	Invalid token response: Challenge Check Digit Wrong (Host Check Challenge Mode)
1127	Invalid token response: Challenge Character Not Decimal
1128	Invalid token response: Challenge Corrupt (Host Check Challenge Mode)
1129	Invalid token response: Response Length Out of Bounds
1130	Invalid token response: Response Too Small
1131	Invalid token response: Response Too Long
1132	Invalid token response: Response Check Digit Wrong
1133	Invalid token response: Response Character Not Decimal
1134	Invalid token response: Response Character Not Hexadecimal
1135	Invalid token response: Token Authentication Failed
1199	Receive error response from VA
0201	Internal system error: Create DSP-ISO message failed
0202	Internal system error: No active DSPSession
0203	Internal system error: Cannot send request to DSP-Silverlake
0204	Internal system error: Waiting response from DSP-Silverlake
0205	Internal system error: Read response from DSP-Silverlake without bit 39
0206	Internal system error: Read response from DSP-Silverlake without bit126
0207	Invalid card number: Card number not belong to this CIF
2101	Invalid card number: Card not found
2102	Not enough balance
2103	Invalid customer account
2104	DSP-Silverlake system error
2199	Receive error response from DSP-Silverlake
0301	Internal system error: Cannot connect to VAM
3101	Invalid XML request: Invalid data XML (tc)
3102	Invalid XML request: Invalid data XML (userid)
3103	Invalid XML request: Invalid data XML (trace number)
3104	Invalid XML request: Invalid data XML (reference number)
3105	Invalid XML request: Invalid data XML (datetime)

3106	Invalid XML request: Invalid data XML (merchantid)
3107	Invalid XML request: Invalid data XML (bankid)
3108	Invalid XML request: Invalid data XML (item detail)
3109	Invalid XML request: Invalid data XML (amount)
3110	Invalid XML request: Invalid data XML (challenge)
3111	Invalid XML request: Invalid data XML (authentication)
3112	Invalid XML request: Invalid data XML (signature)
3113	Invalid XML request: Invalid data XML (aggregator)
3114	Invalid XML request: Error parse XML
3115	Invalid XML request: XML data is null
3116	Invalid XML request: Unmatched signature request
3117	Invalid XML request: Cannot find Aggregator
3118	User already registered: Duplicate UserID
3119	Customer account not found: Cannot find customer account
3120	Not registered UserID
3121	Daily transaction limit is reached
3122	Maximum transaction limit is reached
3123	Transaction payment rejected: Invalid limit configuration
3124	Transaction payment rejected: Cannot find Merchant ID
3125	Transaction payment rejected: Inactive merchant
3126	Transaction payment rejected: Cannot find Bank Commission
3127	Transaction payment rejected: Cannot find Bank Commission Tearing
3128	Transaction payment rejected: Cannot find Aggregator Commission
3129	Transaction payment rejected: Cannot find Aggregator Commission Tearing
3130	Transaction payment rejected: Duplicate Transaction request
3131	Reversal rejected: Cannot find original data for reversal
3132	Reversal rejected: Cannot find merchant account for reversal
3133	Registration failed: Failed add customer channel
3134	Unregistered failed: Failed remove customer channel
3135	Merchant registration failed: Duplicate Merchant
3201	Error init database
3202	Error write to database
4000	No connection to Aggregator
9000	Other error
9013	Unable to send request to bank