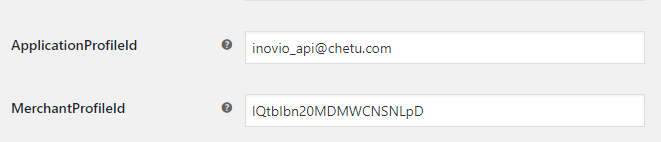
**Nab velocity Woo-Commerce extension manual changes in code and Word-Press Dashboard**

**Note**: This document is depending on Nab velocity extension of Woo-Commerce. So first check that Nab velocity is installed.

1. Open Word-Press dashboard and navigate Woo-Commerce->Settings->Payments->Velocity.
2. Replace ApplicationProfileId with username of inovio
3. Replace MerchantProfileId with username of Password.



1. Open Nab velocity URL configuration file

**File Location**: wordpress/wp-content/plugins/woocommerce-velocity/sdk/configuration.php

1. Change URL From <https://api.cert.nabcommerce.com/REST/2.0.18/> to

http://localhost:8080/nabvwrapper/velocitywrapper?site\_id=22895&

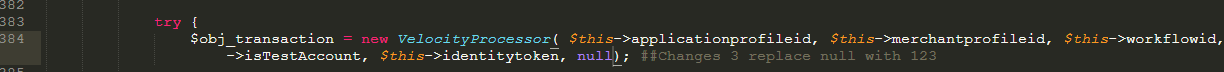
**Note**: [localhost:8080](http://172.16.13.82:8080) hostname and port number.

With

public static $baseurl\_test = 'http://localhost:8080/nabvwrapper/ velocitywrapper?site\_id=22895&';

1. Open Woo-Commerce gateway velocity file

**Location File**: wordpress/wp-content/plugins/ woocommerce-gateway-velocity.php

1. Replace null with 123 in obj\_transection variable in try {} block and because we are bypassing authentication which will use as signOn method.

try {

$obj\_transaction = new VelocityProcessor($this->applicationprofileid, $this->merchantprofileid, $this->workflowid, $this->isTestAccount, $this->identitytoken, 123);

1. Add currency code because there is bug in Nab velocity existing extension and without adding currency code it is showing currency code error on checkout page.

**Location File**: wordpress/wp-content/plugins/ woocommerce-gateway-velocity.php

**In function process\_payment replace below code**

$res\_authandcap = $obj\_transaction->authorizeAndCapture( array(

'amount' => $total,

'avsdata' => $avsData,

'token' => $response['PaymentAccountDataToken'],

'order\_id' => $order\_id,

'entry\_mode' => 'Keyed',

'IndustryType' => 'Ecommerce',

'Reference' => 'xyz',

'EmployeeId' => '11'

)

);

**With**

$res\_authandcap = $obj\_transaction->authorizeAndCapture( array(

'amount' => $total,

'avsdata' => $avsData,

## added currency code is required to make payment

# This is the bug in coadin

'currency\_code' => $order->get\_order\_currency(),

'token' => $response['PaymentAccountDataToken'],

'order\_id' => $order\_id,

'entry\_mode' => 'Keyed',

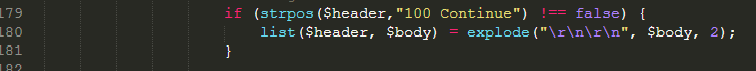
'IndustryType' => 'Ecommerce',

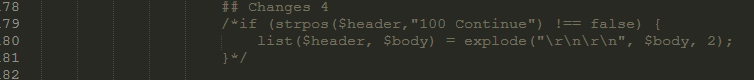
'Reference' => 'xyz',

'EmployeeId' => '11'

)

);

1. Open Nab Velocity Connection file. wp-content/plugins/woocommerce-velocity/sdk/velocity/connection.php
2. Comment below code

with

/\*if (strpos($header,"100 Continue") !== false) {

list($header, $body) = explode("\r\n\r\n", $body, 2);

}\*/

1. Add below code after comment the above code

if((int)$statusCode === 200) {

list($header, $body) = explode("\r\n\r\n", $body, 2);

}

1. Now are able to do the transaction for auth and capture.
2. For refund functionality do the next steps as mentioned.
3. Open Woo-Commerce gateway velocity file

wordpress/wp-content/plugins/ woocommerce-gateway-velocity.php

1. Replace null with 123 in obj\_transection variable in try {} block

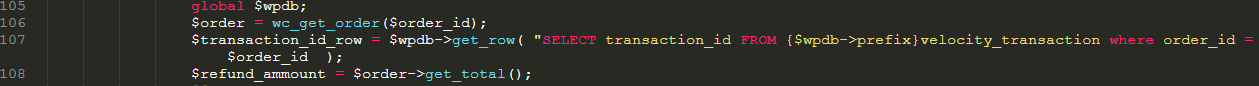
because we are bypassing authentication which will use as signOn method

1. try {

$obj\_transaction = new VelocityProcessor($this->applicationprofileid, $this->merchantprofileid, $this->workflowid, $this->isTestAccount, $this->identitytoken, 123);

1. Add some extra code to implement the refund functionality.

**File Location**: wordpress/wp-content/plugins/ woocommerce-gateway-velocity.php



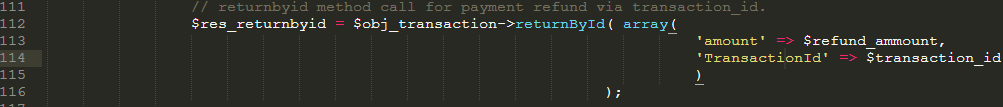
global $wpdb;

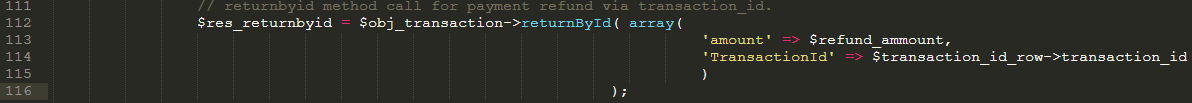
$order = wc\_get\_order($order\_id);

$transaction\_id\_row = $wpdb->get\_row( "SELECT transaction\_id FROM {$wpdb->prefix}velocity\_transaction where order\_id = ".$order\_id );

$refund\_ammount = $order->get\_total();

**Note**: code will add before the commented text // returnbyid method call for payment refund via transaction\_id.

1. Replace below code.

With

$res\_returnbyid = $obj\_transaction->returnById( array(

'amount' => $refund\_ammount,

'TransactionId' => $transaction\_id\_row->transaction\_id

)

);

1. Add below code after $this->description = $this->settings['description'];



add\_action( 'woocommerce\_order\_status\_refunded', array($this,'refund\_payment') );

1. Replace below code

with

function refund\_payment($order\_id, $amount=null, $reason=null) {

1. Replace below code

With

//$this->refund\_payment($refund\_ammount, $transaction\_id, $obj->id );