GoEmerchant, LLC.
GoEmerchant.com

Gateway API Documentation
With CVV2 Support

White Paper Version 3.0

GoEmerchant, LLC.
523 25 Station Ave
Haddon Heights, NJ 08035

Purpose

The purpose of this document is to outline the technology and processes that drive the GoEmerchant.com gateway API. The API provides an interface into the GoEmerchant.com transaction processing network via secure sockets. The end user, after reading this document, should have sufficient knowledge based on the documentation and working samples provided to successfully integrate their ecommerce application into the gateway API.

What is the Gateway API

The GoEmerchant.com gateway API is a programming interface that resides on transaction servers which communicate directly to credit card processing networks. The programming interface requires that the merchant, or their web programming staff, be sufficiently knowledgeable in programming skills in any programming or object oriented scripting language. Several programming samples in a variety of languages are provided for reference to coding into the Gateway API.

The Gateway API accepts the card purchasers information, including credit card information, billing address, total charge amount and order id and produces an authorization or decline directly from the merchant bank. The information is passed via 128bit SSL https post. That post occurs in the background from the merchant's server. Thus the purchaser never leaves the merchant's website. The authorization information is returned in a single text string separated by pipes (|).

For merchant's running Windows NT, the only requirement is that IE 5.01 be installed on the system. IE 5.01 contains the object required to deliver the transaction data via secure ssl post. Alternatively, several other programming languages such as Java, C++, Perl, Visual Basic etc., provide libraries into the Open SSL/Crypt functions.

Unix examples in Perl are available in this documentation.

Authorizations

An authorization is either an accepted "success" or declined "failed" transaction on a purchasers credit card or check. The authorization is produced by the card/check issuing bank and passed to the GoEmerchant.com transaction servers in typically 5 seconds or less.

To utilize the gateway API and obtain a credit card authorization, you must have obtained a merchant credit card processing account through GoEmerchant.com at one of their accepted processing networks. Your Web server or transaction server MUST be capable of performing an https (SSL) silent post to GoEmerchant.com's transaction servers and receive that data, not in the form of a URL but as a text string that can be parsed for results.

Operational process:

1. The merchant's website produces a total amount to be authorized, captures the customer's credit card information and performs an https (silent post) to GoEmerchant.com's transaction servers.

2. GoEmerchant.com's transaction servers send the authorization request to the card processing network's authorization servers.

3. The processing network's authorization servers return to GoEmerchant.com's transaction servers an auth or decline.

4. GoEmerchant.com's transactions servers send a text string to the merchant's web or transaction servers indicating a success or failure.

5. The merchant parses the text string and notifies the customer of a successful authorization or failure.

Secure posting to GoEmerchant.com Credit Card transaction servers:

The URL used in the https post is:

https://www.goemerchant4.com/cgi-bin/gateway/gateway.cgi

The post (key = value) pairs are listed below:

*NOTE – All keys **MUST** be named exactly as below.

Key	Value
merchant	(Merchant name assigned by GoEmerchant.com)
password	(Merchant password assigned by GoEmerchant.com)
operation_type	auth
orderid	(MUST BE UNIQUE ASSIGNED BY MERCHANT WEB SERVER)
total	(In US Dollars. Example 59.99 No $ signs.)
cardname	(Visa, Amex, Discover or MasterCard)
cardnum1	First 4 Numbers on Card
cardnum2	Second 4 Numbers on Card
cardnum3	Third 4 Numbers on Card
cardnum4	Last 4 Numbers on Card - 3 digits for Amex
cardexpm	Card Expiration Month - Format MM
cardexpy	Card Expiration Year - Format YY DO NOT SEND XXXX (example 2001)
CVV2	CVV2 – 3 digits Visa/Mastercard 4 digits American Express
nameoncard	Card Holders Name
cardstreet	Card Holders Billing Street
cardcity	Card Holders Billing City
cardstate	Card Holders Billing State ***(2 character abbreviation)**
cardzip	Card Holders Billing Zip
cardcountry	Card Holders Billing Country ***(2 character abbreviation)**

Example post key=value pairs:

merchant=mymerchantname&password=123test&operation_type=auth&orderid=123456&
total=1.00&cardname=Visa&cardnum1=4111&cardnum2=1111&cardnum3=11113&cardnum4
=1111&cardexpm=01&cardexpy=02&CVV2=123&nameoncard=John Doe&cardstreet
=123 Any Street&cardcity=Any City&cardstate=NJ&cardzip=08000&cardcountry=US

Return values from the secure post:

Upon successful post to the GoEmerchant.com transaction servers, your code will receive a text string with the following variables separated by pipes (|).

Success Value|Authorization Code|Authorization Response|AVS Code|Order ID

String Value	Return Codes
Success Value	1 - indicates Authorization Approved. 0 - Authorization Declined.
Authorization Code	Generally a five or six character code sent by the bank.
Authorization Response	The Successful or Declined message from the bank.
AVS Code	Address Verification Code signifying matches in the card holders address.
Order ID	Echoed back from the server that was provided in the secure post.

Example Successful Response:

1|411111|Authorization Approved|Y|123456-1

Example Declined Response:

0|ND|Invalid Card||123456-1 *Note – that no AVS field on declines.

Sample code for secure posting authorizations written in Perl for NT and Unix, or ASP in Visual Basic Script are available in the section titled Sample Authorization Code.

Sample Code Authorizations

The sample code is provided as is, and has been tested on the platforms in the comments. The code can be cut and pasted directly into your applications based on the variables being titled exactly as they appear in the samples.

Windows NT/2000 Perl 5.6 – IE 5.01 Object Method

```
#####################################################################
# Cut N Paste - Perl5.6 Windows NT Gateway module
#
# You must have Activestate Perl 5.6 or above installed and Windows
# NT 4 or 2000 Server with IE 5+ installed for this version to
# function properly.
#
# Note - All variables listed below in the Transaction Section MUST
# be titled the same in your code for the gateway to function properly.
#
# This code creates a secure https post in the background to
# GoEmerchant.com's secure transaction servers and retrieves an
# authorization placing the results in the variables listed in the
# results comments.
```

```
#
#
# Uncomment Transaction Variables for Billing and provide data for tests
#
# $merchantname - (Merchant name assigned by GoEmerchant.com)
# $password - (Merchant password assigned by GoEmerchant.com)
# $operation_type - (value = auth)
# $orderid - (MUST BE UNIQUE - NON UNIQUE WILL RETURN AN ERROR)
# $total - (In US Dollars.  Example 59.99  No $ signs.)
#
# For Credit Card Transactions
# $cardname - (Visa, Amex, Discover or MasterCard)
# $cardnum1 - First 4 Numbers on Card
# $cardnum2 - Second 4 Numbers on Card
# $cardnum3 - Third 4 Numbers on Card
# $cardnum4 - Last 4 Numbers on Card - 3 digits for Amex
# $cardexpm - Card Expiration Month - Format MM
# $cardexpy - Card Expiration Year - Format YY DO NOT SEND XXXX (example 2001) only 01
# $CVV2 – CVV2 value from card.
# $nameoncard - Card Holders Name
# $cardstreet - Card Holders Billing Street
# $cardcity - Card Holders Billing City
# $cardstate - Card Holders Billing State
# $cardzip- Card Holders Billing Zip
# $cardcountry - Card Holders Billing Country
#
#
# Authorization Result Variables Returned
#
# $success - (1 indicates authorization successful, 0 indicates a failure)
# $authcode - The authorization code
# $authresponse - The bank response
# $avscode - The avs value returned
#
#
# The section below creates the post string of Transaction Variables to send to the gateway
# and sends via https.  The response is parsed and the Authorization Result variables are
# set.
#

############ Use these postvals for a credit card transaction ###############
$Postvals = "merchant=$merchantname&password=$password&operation_type=$operation_type&";
$Postvals .= "orderid=$orderid&total=$total&cardname=$cardname&";
$Postvals .= "cardnum1=$cardnum1&cardnum2=$cardnum2&cardnum3=$cardnum3&cardnum4=$cardnum4&";
$Postvals .= "cardexpm=$cardexpm&cardexpy=$cardexpy&CVV2=$CVV2&nameoncard=$nameoncard&cardstreet=$cardstreet&";
$Postvals .= "cardcity=$cardcity&cardstate=$cardstate&cardzip=$cardzip&cardcountry=$cardcountry";

use WIN32::OLE;

$SendObject=Win32::OLE->new('microsoft.XMLhttp');
$SendObject->open("POST", "https://www.goemerchant4.com/cgi-bin/gateway/gateway.cgi", "false");
$SendObject->setRequestHeader("Content-type", "text/plain");
$SendObject->send ($Postvals);
$Result = $SendObject->responseText;

($success, $authcode, $authresponse, $avscode, $oid) = split(/\|/, $Result);
```

Unix/Windows NT/2000 Perl 5.6 – Crypt Module Installed

```
#!/usr/local/bin/perl
#################################################################
# Cut N Paste - Perl5.6 Unix Gateway module - NOTE this will also
# work on Windows NT with the below listed modules installed.
#
# You must have Perl 5.6 or above installed with the Crypt::SSLeay
# and LWP module installed for this version to function properly.
#
# Note - All variables listed below in the Transaction Section MUST
```

```
# be titled the same in your code for the gateway to function properly.
#
# This code creates a secure https post in the background to
# GoEmerchant.com's secure transaction servers and retrieves an
# authorization placing the results in the variables listed in the
# results comments.
#
#
# Uncomment Transaction Variables for Billing and provide data for tests
#
# $merchantname - (Merchant name assigned by GoEmerchant.com)
# $password - (Merchant password assigned by GoEmerchant.com)
# $operation_type - (value = auth)
# $orderid - (MUST BE UNIQUE - NON UNIQUE WILL RETURN AN ERROR)
# $total - (In US Dollars.  Example 59.99  No $ signs.)
#
# For Credit Card Transactions
# $cardname - (Visa, Amex, Discover or MasterCard)
# $cardnum1 - First 4 Numbers on Card
# $cardnum2 - Second 4 Numbers on Card
# $cardnum3 - Third 4 Numbers on Card
# $cardnum4 - Last 4 Numbers on Card - 3 digits for Amex
# $cardexpm - Card Expiration Month - Format MM
# $cardexpy - Card Expiration Year - Format YY DO NOT SEND XXXX (example 2001) only 01
# $CVV2 – CVV2 value from card.
# $nameoncard - Card Holders Name
# $cardstreet - Card Holders Billing Street
# $cardcity - Card Holders Billing City
# $cardstate - Card Holders Billing State
# $cardzip- Card Holders Billing Zip
# $cardcountry - Card Holders Billing Country
#
#
# Authorization Result Variables Returned
#
# $success - (1 indicates authorization successful, 0 indicates a failure)
# $authcode - The authorization code
# $authresponse - The bank response
# $avscode - The avs value returned
#
# The section below creates the post string of Transaction Variables to send to the gateway
# and sends via https.  The response is parsed and the Authorization Result variables are
# set.

############ Use these postvals for a credit card transaction ###############
$Postvals = "merchant=$merchantname&password=$password&operation_type=$operation_type&";
$Postvals .= "orderid=$orderid&total=$total&cardname=$cardname&";
$Postvals .= "cardnum1=$cardnum1&cardnum2=$cardnum2&cardnum3=$cardnum3&cardnum4=$cardnum4&";
$Postvals .= "cardexpm=$cardexpm&cardexpy=$cardexpy&CVV2=$CVV2&nameoncard=$nameoncard&cardstreet=$cardstreet&";
$Postvals .= "cardcity=$cardcity&cardstate=$cardstate&cardzip=$cardzip&cardcountry=$cardcountry";

use LWP::UserAgent;
use Crypt::SSLeay;

$ua = new LWP::UserAgent;
$ua->agent("SSL/0.1");

my $req = new HTTP::Request('POST', 'https://www.goemerchant4.com/cgi-bin/gateway/gateway.cgi');
$req->content_type('application/x-www-form-urlencoded');
$req->content($Postvals);

my $res = $ua->request($req);
$Result = $res->content;

($success, $authcode, $authresponse, $avscode, $oid) = split(/\|/, $Result);
```

Windows NT/2000 VBScript ASP Object Method

```
' Cut N Paste - VBScript/ASP Windows NT Gateway module
'
' You must have IIS 4 or above installed and Windows
' NT 4 or 2000 Server with IE 5+ installed for this version to
' function properly.
'
' Note - All variables listed below in the Transaction Section MUST
' be titled the same in your code for the gateway to function properly.
'
' This code creates a secure https post in the background to
' GoEmerchant.com's secure transaction servers and retrieves an
' authorization placing the results in the variables listed in the
' results comments.
'
'
' Transaction Variables for Billing
'
' merchantname - (Merchant name assigned by GoEmerchant.com)
' password - (Merchant password assigned by GoEmerchant.com)
' operation_type - (value = auth)
' orderid - (MUST BE UNIQUE - NON UNIQUE WILL RETURN AN ERROR)
' total - (In US Dollars.  Example 59.99  No $ signs.)
'
' For Credit Card Transactions
' cardname - (Visa, Amex, Discover or MasterCard)
' cardnum1 - First 4 Numbers on Card
' cardnum2 - Second 4 Numbers on Card
' cardnum3 - Third 4 Numbers on Card
' cardnum4 - Last 4 Numbers on Card - 3 digits for Amex
' cardexpm - Card Expiration Month - Format MM
' cardexpy - Card Expiration Year - Format YY DO NOT SEND XXXX (example 2001) only 01
' CVV2 – CVV2 value from card.
' nameoncard - Card Holders Name
' cardstreet - Card Holders Billing Street
' cardcity - Card Holders Billing City
' cardstate - Card Holders Billing State
' cardzip- Card Holders Billing Zip
' cardcountry - Card Holders Billing Country
''
' Authorization Result Variables Returned
'
' success - (1 indicates authorization successful, 0 indicates a failure)
' authcode - The authorization code
' authresponse - The bank response
' avscode - The avs value returned
'
'
' The section below creates the post string of Transaction Variables to send to the gateway
' and sends via https.  The response is parsed and the Authorization Result variables are
' set.

<%

Dim SendObject, Postvals, merchantname, password, orderid, total, cardname
Dim cardnum1, cardnum2, cardnum3, cardnum4, CVV2
Dim cardexpm, cardexpy, nameoncard, cardstreet, cardcity, cardstate, cardzip, cardcountry, operation_type
Dim success, authcode, authresponse, avscode, oid, Result, SplitArray
Dim first, last, street, city, state, zip, phone, ABA, checknumber, ACC, memo, sign, email

Postvals = "merchant=" & merchantname & "&password=" & password & "&operation_type=" & operation_type
Postvals = Postvals & "&orderid=" & orderid & "&total=" & total & "&cardname=" & cardname
Postvals = Postvals & "&cardnum1=" & cardnum1 & "&cardnum2=" & cardnum2 & "&cardnum3=" & cardnum3
Postvals = Postvals & "&cardnum4=" & cardnum4 & "&cardexpm=" & cardexpm & "&cardexpy=" & cardexpy & "&CVV2=" & CVV2
Postvals = Postvals & "&nameoncard=" & nameoncard & "&cardstreet=" & cardstreet & "&cardcity=" & cardcity
Postvals = Postvals & "&cardstate=" & cardstate & "&cardzip=" & cardzip & "&cardcountry=" & cardcountry
```

```
Set SendObject = Server.CreateObject("Microsoft.XMLhttp")
SendObject.open "POST","https://www.goemerchant4.com/cgi-bin/gateway/gateway.cgi",false
SendObject.setRequestHeader "Content-type", "text/plain"
SendObject.send Postvals
Result = SendObject.responseText

SplitArray = split(Result, "|", -1, 1)

success = SplitArray(0)
authcode = SplitArray(1)
authresponse = SplitArray(2)
avscode = SplitArray(3)
oid = SplitArray(4)

%>
```

Conclusions

The GoEmerchant.com Gateway API is an intuitive interface to the transaction servers that produce credit card transactions. The gateway is capable of being programmed into by any programming language, from any platform as long as an SSL https silent post can be performed. If you experience any problems or require technical assistance, you may call the Customer Support Center at GoEmerchant.com at 1-888-638-7867.