1stPayGateway.Net

Title: Authorize.NET Emulator Gateway API Specification

PRODUCT: AE Gateway

Version 1.1.0

Document Control/Revisions

Document Modification	Version	Author/Editor	Date
Original Draft AE Gateway API	1.0.0	RMB	1/26/2011
Added x_duplicate_window variable functionality	1.0.1	RMB	5/6/2011
Partial Billing Support	1.0.2	RJT	5/16/2011
Every other month recurring billing support	1.0.3	RJT	10/31/2011
Quick Integration Guide	1.1.0	RJT	1/7/2012
Note about test account	1.1.1	RJT	3/12/2012

TABLE OF CONTENTS

1. Introduction

The purpose of this document is to outline the technology and processes that drive the AE Gateway API. The API provides an interface into the gateway 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 AE Gateway API.

1.1. Scope

This specification will walk the reader through the each type of transaction supported by the AE gateway. A detailed description of each acceptable input message will be documented along with each possible response. This document will not cover language specific integrations, although there will be sample code provided in section 6.

1.2. Product Description

The AE 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 AE Gateway API.

The AE 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 AE Gateway API provides methods to perform the following operations: AUTH, SALE. The information is passed via 128bit SSL https post in delimited string format. 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 csv string format with full error trapping and reporting to indicate the success or failure of the transaction.

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.

URL for secure https post:
https://secure.1stpaygateway.net/secure/gateway/aegateway.aspx

Operational process:

1. The merchant’s website produces the message with all the required information depending on the operation type and performs an https (silent POST) to gateway transaction servers.
2. The gateway’s transaction servers determine the operation type and execute it.
3. Once the correct operation is carried out, the gateway assembles the response in a string.
4. The gateway’s transactions servers send the string response to the merchant’s web or transaction servers indicating a success, failure or error.
5. The merchant parses the string and notifies the customer or user of a successful request, failure or error.

1.3. *Intended Audience*

This document is written exclusively for the use of approved merchants of 1stPayGateway.Net. It is intended to assist merchants with the task of integrating their ecommerce/retail solution with products offered or approved for integration with 1stPayGateway.Net's secure gateway processing networks.

1.4. *Quick Integration Guide*

To start using this gateway, simply make the following changes to your working code that currently uses the Authorize.net gateway:

Set the POST URL to **https://secure.1stpaygateway.net/secure/gateway/aegateway.aspx**

Use the Transaction Center ID provided for the **x_login** parameter
Use the Gateway ID provided for the **x_tran_key** parameter.

The Gateway ID can be retrieved and changed under the Security Settings/Gateway Options menu in the Transaction Center.

The other pages in the documentation can be used as a reference. It is recommended that you read the full documentation to avoid any issues and to take advantage of features which may not be available in the current API you are using.

2. Definitions

2.1. *Operation Types*

Operation Type	Decription
AUTH	An AUTH operation is nothing more than a transaction that reserves the funds on a customer's credit card. The AUTH transaction must be accompanied by a SETTLE transaction in order for the merchant to receive funds.
SALE	A SALE operation produces a transaction that authorizes and captures the funds of the customer's credit card.

2.2. *Field Definitions*

Field Name	Field Type	Field Length	Description
Transaction Header Section			
x_login	Int	9	Unique identifier assigned by gateway. This is your unique Transaction Center number.
x_tran_key	String	Uniqueidentifier	Unique identifier assigned by gateway. Can be found and or reset via the Options Tab in the Transaction Center.
x_type	String	Varchar(20)	String specifying operation attempting to be run. Must be one of the supported operation types:

			AUTH, SALE
Transaction Details Section			
x_invoice_num	String	Varchar(50)	Unique order id or invoice number. Cannot contain “insert”, “update” or “delete”.
x_amount	Numeric	(9,2)	Amount in US dollars. No dollar signs ($) or commas allowed.
Credit Card Data Section			
x_card_name	String	Varchar(30)	Type of card. Visa, mastercard, amex, discover etc. “no” or “yes” can also be used. If “no” or “yes” is used then the system will attempt to determine the card_name based on the card_number passed in.
x_card_num	Numeric	(19,0)	Credit card account number
x_card_exp	Numeric	(4,0)	Credit card expiration date. MMYY format.
x_card_code	Numeric	(4,0)	Credit card security code, cvv2, cvc, cid
Cardholder Billing Address Section			
x_first_name	String	Varchar (75)	Card holders first name
x_last_name	String	Varchar(75)	Card holders last name
x_company	String	Varchar(75)	Card holders company name
x_address	String	Varchar (250)	Billing street address
x_address2	String	Varchar (250)	Billing street address 2
x_city	String	Varchar (100)	Billing city
x_state	String	Varchar (100)	Billing state
x_zip	String	Varchar (20)	Billing zipcode
x_country	String	Varchar (200)	Billing country
x_email	String	Varchar (300)	Billing email address
x_phone	String	Varchar (25)	Billing phone number
x_fax	String	Varchar(25)	Billing fax number
Recurring Billing Section			
x_recurring	Boolean	Bit	0 or 1 – indicates if transaction is a recurring transaction. When this flag is turned on (“1”) , a valid recurring_type must also accompany the transaction.
x_recurring_type	String	Varchar(12)	Indicates the recurring interval for this transaction to repeat. Will be ignored if the recurring flag is not “1”. Accepted values: **daily, weekly, biweekly, monthly, bimonthly, quarterly, semiannually, annually**
Partial Billing Section			
x_split_tender_id	Numeric	(20,0)	This numeric value is returned for partially approved transactions. It is used to tie subsequent transactions together to make up the larger transaction.

x_allow_partial_auth	Boolean	Bit	0 or 1 – indifcates is a partial approval is allowed for this transaction. If this is set to 0 then a partially approved transaction will be rejected. This is 0 by default.
Customer IP Address Section			
x_remote _ip_address	String	Varchar(16)	Ip address of the customer contacting the merchant's site or application.
Purchase Card Level II Data Section			
x_purchase_card	Boolean	Bit	0 or 1 – indicates if the credit card used is a purchase card or not
x_customer_reference_number	String	Varchar(17)	Unique alpha-numeric value used to identify the card holder as a customer
x_local_tax_flag	Int	1	Flag set to 0, 1 or 2 that indicates the tax handling of the transaction being submitted. **0 - tax not provided** **1 - tax included** **2 - non-taxable transaction**
x_tax	Numeric	(9,2)	Amount in US dollars. No dollar signs ($) or commas allowed.
Shipping Address Data Section			
x_ship_to_first_name	String	Varchar(100)	Ship to recipient first name
x_ship_to_last_name	String	Varchar(100)	Ship to recipient last name
x_ship_to_company	String	Varchar(100)	Ship to company name
x_ship_to_address	String	Varchar(250)	Ship to street address
x_ship_to_address2	String	Varchar(250)	Ship to street address 2
x_ship_to_city	String	Varchar(100)	Ship to city
x_ship_to_state	String	Varchar(100)	Ship to state
x_ship_to_zip	String	Varchar(20)	Ship to zipcode
x_ship_to_country	String	Varchar(200)	Ship to country
Additional Fields			
x_test_request	Boolean	Bit	Indicates if the transaction should be processed as a test transaction. If set to TRUE, the payment gateway will return an approved transaction. The transaction will not be entered into the transaction center.
x_email_customer	Boolean	Bit	Indicates whether an email receipt should be sent to the customer. If set to TRUE, the payment gateway will send an email to the customer after the transaction is processed using the customer email address submitted with the transaction. If FALSE or not passed, no email is sent to the customer.

x_delim_char	String	Char(1)	The character that is used to separate fields in the transaction response. Example: ,(comma) | (pipe) “ (double quotes) ‘ (single quote) : (colon) ; (semicolon) / (forward slash) \ (back slash) - (dash) * (star)
x_encap_char	String	Char(1)	The character that is used to encapsulate the fields in the transaction response. This is only necessary if it is possible that your delimiting character could be included in any field values. Example: (comma) | (pipe) “ (double quotes) ‘ (single quote) : (colon) ; (semicolon) / (forward slash) \ (back slash) - (dash) * (star)
x_header_email_receipt	String	Varchar(1000)	Thist ext will appear as the header on the email receipt sent to the customer.
x_footer_email_receipt	String	Varchar(1000)	This text will appear as the footer on the email receipt sent to the customer.
x_merchant_email	String	Varchar(50)	Email address to which the merchant’s copy of the customer confirmation email should be sent. If a value is submitted, an email will be sent to this address.
x_version	String	Varchar(20)	The merchant’s transaction version
x_duplicate_window	Int	5	Indicates in seconds the window of time after a transaction is submitted during which the payment gateway will check for a duplicate transaction. The maximum time allows is 8 hours (28800 seconds). If a value less than 0 is sent, the

			payment gateway will default to 0. If a value greater than 28800 is sent, the payment gateway will default to 28800. If no value is sent the payment gateway will default to 2 minutes (120 seconds).

3. API Specifications

3.1. *Requirements for AEGateway Utilization*

1. Processing Platforms Supported.
    a. FDR (Omaha Platform)
    b. Elavon (Formerly Nova)
    c. Paymentech (Tampa Host Capture Platform)
    d. TSYS (formerly Vital/VisaNet)
2. Transaction Center ID
    a. Numeric access key assigned by 1stPayGateway.Net
    b. Unique to merchant
    c. Cannot be changed
    d. Required for API integration
3. Gateway ID (Tran Key)
    a. Alpha-numeric passphrase assigned by 1stPayGateway.Net
    b. Can be changed via the gateway options area in the Transaction Center
    c. Should be changed every 90 days to ensure security
    d. Required for API integration
    e. Retrieved from the options tab in the Transaction Center https://secure.1stpaygateway.net/secure/transcenter/
4. Secure URL
   https://secure.1stpaygateway.net/secure/gateway/aegateway.aspx
5. Development/Testing
    a. Test account is available
    b. Full test account information is documented in section 5

3.2. *Auth/Sale Credit Card Operations*

The AUTH/SALE Credit Card group of operations is the basic gateway function and the most commonly used function. These operation types are responsible for processing authorization only and sales on a customer's credit card.

Successful AUTH/SALE transactions will contain a reference_number in the generated response.

3.2.1. Auth/Sale Credit Card E-commerce

Field Name	**Field Value**	**Required**
Transaction Header Section		
x_login	Int	X

x_tran_key	Unique ID	X
x_type	AUTH/SALE	X
Transaction Details Section		
x_invoice_num	String	X
x_amount	Numeric	X
Credit Card Data Section		
x_card_name	Visa, Mastercard, Amex, Disc, Yes	
x_card_num	Numeric	X
x_card_exp	Numeric	X
x_card_code	Numeric	X
Cardholder Billing Address Section		
x_first_name	String	X
x_last_name	String	X
x_company	String	
x_address	String	X
x_address2	String	
x_city	String	X
x_state	String	X
x_zip	String	X
x_country	String	X
x_email	String	
x_phone	String	
x_fax	String	
Recurring Billing Section		
x_recurring	0, 1	
x_recurring_type	daily, weekly, biweekly, monthly, bimonthly, quarterly, semiannually, annually	
Customer IP Address Section		
x_remote _ip_address	String	X
Purchase Card Level II Data Section		
x_purchase_card	0 - regular card, 1 - b2b purchase card	
x_customer_reference_number	String	
x_local_tax_flag	Int	
x_tax	Numeric	
Shipping Address Data Section		
x_ship_to_first_name	String	
x_ship_to_last_name	String	
x_ship_to_company	String	

x_ship_to_address	String	
x_ship_to_address2	String	
x_ship_to_city	String	
x_ship_to_state	String	
x_ship_to_zip	String	
x_ship_to_country	String	
Additional Fields		
x_test_request	True, False, T, F, Yes, No, Y, N, 1, 0	
x_email_customer	True, False, T, F, Yes, No, Y, N, 1, 0	
x_delim_char	String	
x_encap_char	String	
x_header_email_receipt	String	
x_footer_email_receipt	String	
x_merchant_email	String	
x_version	String	

3.2.1.1. Auth/Sale

(Auth and Capture) transaction

x_version=1.0&x_amount=1.00&x_card_num=4111111111111111&x_exp_date=1015&x_login=1264&x_tran_key=a91c38c3-7d7f-4d29-acc7-927b4dca0dbe&x_address=123 testrd&x_zip=19036&x_type=sale&x_invoice_num=456789123&x_card_code=123&x_city=cityville&x_first_name=bob&x_last_name=tester&x_email=test@1stPayGateway.Net&x_phone=5552226598&x_state=pa&x_country=usa&x_ship_to_address=123 testrd&x_ship_to_zip=19036&x_tax=&x_test_request=0&x_email_customer=y&x_ship_to_first_name=Bobby&x_merchant_email=test@1stPayGateway.Net&x_description=this+is+a+description

Auth (Authorization only) transaction

x_version=1.0&x_amount=1.00&x_card_num=4111111111111111&x_exp_date=1015&x_login=1264&x_tran_key=a91c38c3-7d7f-4d29-acc7-927b4dca0dbe&x_address=123 testrd&x_zip=19036&x_type=auth&x_invoice_num=456789123&x_card_code=123&x_city=cityville&x_first_name=bob&x_last_name=tester&x_email=test@1stPayGateway.Net&x_phone=5552226598&x_state=pa&x_country=usa&x_ship_to_address=123 testrd&x_ship_to_zip=19036&x_tax=&x_test_request=0&x_email_customer=y&x_ship_to_first_name=Bobby&x_merchant_email=test@1stPayGateway.Net&x_description=this+is+a+description

3.2.2. Auth/Sale Responses

Response for Credit Card Transaction

status, status, status, auth_response, auth_code, avs_code, order_id,reference_number, f09, total, method, f13, f14, f15, f16, f17, f18, f19, f20, f21, f22, f23, f24, f25, f26, f27, f28, f29, f30, f31, f32, f33, f34, f35, f36, f37, md5_hash, cvv2_code,,,,,,,,,,,,,,Split Tender Id,Orig Total,Avail Balance

Example:

1,1,1,APPROVED,123456,Y,6541236,789456123,,1.00,CC,,,,,,,,,,,,,,,,,,,,,,,,,MD5HASH,Y, ,,,,,,,,,,,,,,,,

STATUS – 0 - response:
status: 1-success 2-declined 3-error
authresponse: response from bank. May be same as response
auth_code: character code sent by the bank
avs_code: avs code from the bank
cvv2_code: cvv2 code from the bank
invoice_num: echoed back from original post
reference_number: returned for use with credits/voids/settles in other gateways
total – amount transaction was approved for. Will be same as what was passed unless partially approved
md5_hash – MD5 hash which can be used to confirm result validity
Split Tender ID – only present when transaction was partially approved or part of a partial transaction.
Orig Total – the total amount that was originally passed in. Only present on partial approvals
Avail Balance – remaining balance on card if available. Generally found when a Gift card was used
f: will be empty but may be used in future versions for additional information.

4. Error Responses

Malformed fields or bad messages in all cases will be returned in the response defined for the operation type sent in with the status = 3. The auth_response field will contain the error message.

Example:

```
3,,,Merchant: not authorized.,,,,testaegw1,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
```

5. Test Authorization Account Information

In order to test your implementation of the AE Gateway, you may supply test transaction data that will either produce an authorization or decline or error. By using the test data, you may ensure that you are properly communicating with the transaction gateway and also properly parsing the return values.

NOTE: Transactions submitted using the x_test_request parameter will not be visiable in your transction center. These transactions are processed under a shared test account.

To test a successful response from the gateway, please provide the values for the following variables:

x_test_request = True or T or Yes or Y or 1

To get an approved response for Credit Card, use the following information:

Visa: 4111111111111111	CVV2 = 123
Mastercard: 5000300020003003	CVC2 = 123
Discover: 6011111111111117	CVV2 = 123
Amex: 374255312721002	CVV2 = 1234

Test Address information:
Address: 123 Test St
Zip: 12345-6789

The remaining field values you can make whatever you want to test all the various aspects of the API, just make sure to fulfill the defined criteria listed above for the different operation types.

* As this is a generic test account, try to use a formula that will generate unique order_id's based on your merchant name.

To test a decline Credit Card response from the gateway, please provide the values for the following variables:

Use above information and any other card number other than those listed above. Legitimate card numbers will decline since this is a test account.

The remaining field values you can make whatever you want to test all the various aspects of the API, just make sure to fulfill the defined criteria listed above for the different operation types.

To test an error response from the gateway, please provide the values for the following variables:

To generate an error, use the above account information and provide data in the fields and an error response will be generated.

The remaining field values you can make whatever you want to test all the various aspects of the API, just make sure to fulfill the defined criteria listed above for the different operation types.

NOTE – It is strongly recommended that all procedures be tested to ensure that your gateway integration is complete and correct.

6. Sample Code

The sample code is provided as is. The sample provides an interface to each of the operation_types in one process. This code can be cut out and saved on any windows platform with perl 5 installed and the Win32::OLE module installed. Only invoice_num and reference fields need necessarily be changed. The sample uses the test gateway account.

6.1.1. Perl 5 Sample for E-commerce

```
######      aegatewaytest.cgi   ############################################
#
#This program is used to connect to the ae gateway api
#########  Explanation Over   ############################################
#!c:\perl5\bin\perl.exe

use strict;
use warnings;

$|=1;

# Perl library modules
use WIN32::OLE;

print "Content-type:text/html\n\n";

my $xml =
"x_version=1.0&x_amount=1.00&x_card_num=4111111111111111&x_exp_date=1010&x_login=1264&x_password=password&x_addr
ess=123 test
rd&x_zip=19036&x_type=sale&x_invoice_num=testaegw1&x_card_code=123&x_city=cityville&x_first_name=bob&x_last_name=tester
&x_email=test\@testt.com&x_phone=7144844949&x_state=pa&x_country=usa&x_ship_to_address=123 test
rd&x_ship_to_zip=19036&x_tax=";
my $SendObject = Win32::OLE->new('microsoft.XMLhttp');

$SendObject->open("POST", "https://secure.1stpaygateway.net/secure/gateway/aegateway.aspx", "false");
```

```
$SendObject->setRequestHeader("Content-type", "text/xml");
$SendObject->send($xml);
my $response = $SendObject->responseText;

print "<html><head><title>testing ae  gateway</title><head><body>";
print "GET RESPONSE: $response";


print "</body></html>";


#EOF
```