GatewayService

Programmers Guide

Last Updated: 2-Feb-2018

GatewayService Package

Package GatewayService

This class communicates with servers within the RocketGate network to perform standard credit card transactions, i.e. auth-only, ticket, void, etc.

As shown in the example below, details of the transaction to be performed are embedded in a request structure and submitted to a GatewayService object for processing using one of its member functions. The success/failure results of the transaction and any banking details are returned to the calling application through a simple response object.

```
include (“GatewayService.php”);

//
//    Create the objects we need for the example.
//
$request = new GatewayRequest();
$response = new GatewayResponse();
$service = new GatewayService();

//
//    Fill in the fields of the Merchant request.
//
$request->Set(GatewayRequest::MERCHANT_ID(), "21");
$request->Set(GatewayRequest::MERCHANT_PASSWORD(), “password”);
$request->Set(GatewayRequest::MERCHANT_ACCOUNT(), "1");

$request->Set(GatewayRequest::AMOUNT(), "10.99");
$request->Set(GatewayRequest::CARDNO(), "378200000000005");
$request->Set(GatewayRequest::EXPIRE_MONTH(), "08");
$request->Set(GatewayRequest::EXPIRE_YEAR(), "2007");

//
//    If the request succeeded, output the results.
//
if ($service->PerformAuthOnly($request, $response)) {
    print "Transaction succeeded";
    print "Response Code:”. response->Get(GatewayResponse::RESPONSE_CODE());
    print "Auth No:” . $response->Get(GatewayResponse::AUTH_NO());
    print “TransactID:” .  $response->Get(GatewayResponse.TRANSACT_ID());
} else {

//
//     If the request failed, output the reason.
//
    print "Transaction failed";
    print "Response Code:” . $response->Get(GatewayResponse::RESPONSE_CODE());
    print “Reason Code:” . $response->Get(GatewayResponse::REASON_CODE())”;
}
```

Example 1

Example two, shown below, demonstrates the use of the GatewayService to issue a credit for a previous transaction. Again, details of the transaction to be performed are embedded in a request structure and submitted to a GatewayService object for processing using one of its member functions. The success/failure results of the transaction and any banking details are returned to the calling application through a simple response object.

```
include (“GatewayService.php”);

//
//    Create the objects we need for the example.
//
$request = new GatewayRequest();
$response = new GatewayResponse();
$service = new GatewayService();

//
//    Fill in the fields of the Merchant request.
//
$request->Set(GatewayRequest::MERCHANT_ID(), "21");
$request->Set(GatewayRequest::MERCHANT_PASSWORD(), “password”);
$request->Set(GatewayRequest::TRANSACT_ID(), "110000131A56145D");

//
//    If the request succeeded, output the results.
//
if ($service->PerformCredit($request, $response)) {
     print "Transaction succeeded";
     print "Response Code:” . response->Get(GatewayResponse::RESPONSE_CODE());
     print “TransactID:” . $response->Get(GatewayResponse.TRANSACT_ID());
} else {

//
//    If the request failed, output the reason.
//
     print "Transaction failed";
     print "Response Code:” . $response->Get(GatewayResponse::RESPONSE_CODE());
     print “Reason Code:” . $response->Get(GatewayResponse::REASON_CODE());
}
```

Example 2

Method Summary

GenerateXsell(request, response) Submit an auth-only transaction to the RocketGate system.
PerformAuthOnly(request, response) Submit an auth-only transaction to the RocketGate system.
PerformTicket(request, response) Submit a ticket transaction to settle a previous auth-only transaction.
PerformPurchase(request, response)

Submit a complete auth-capture transaction to the RocketGate system.
PerformCredit(request, response) Submit a credit transaction to the RocketGate system.
PerformVoid(request, response) Submit a void transaction for a previous auth-only, ticket, purchase, or credit transaction.
PerformCardScrub(request, response) Perform scrubbing or a transaction without submitting to the bank.
PerformRebillCancel(request, response) Schedule (or unscheduled) cancellation of a recurring billing.
PerformRebillUpdate(request, response) Update price, frequency, cancellation date etc., of a recurring billing.
SetReadTimeout(seconds) Set timeout for RocketGate server response.
SetTestMode(flag) Turn on/off testing mode. In testing mode, transactions are sent to RocketGate test servers.

Method Details

GenerateXsell

GenerateXsell(request, response)

The GenerateXsell() function does not perform xsell transactions directly. Instead, the xsell requests are placed in a queue where they are then processed asynchronously, similar to xsells generated through check-boxes on RocketGate's hosted pages. Similar to the check-boxes, merchants involved in the xsell operation are sent a postback when the asynchronous xsell is complete.

The GenerateXsell() function accepts the same arguments as the hosted page check-boxes. The arguments are simply submitted using the appropriate gateway parameters. The GenerateXsell() function recognizes the following gateway API parameters. All parameters are optional unless stated otherwise.

- MERCHANT_ID – Merchant initiating the xsell – REQUIRED
- MERCHANT_CUSTOMER_ID – Customer ID of original customer, i.e. belonging to MERCHANT_ID. If omitted, value is obtained from XSELL_REFERENCE_XACT.
- AMOUNT – Transaction amount – REQUIRED
- CURRENCY
- XSELL_REFERENCE_XACT – Transaction ID associated with MERCHANT_CUSTOMER_ID – OPTIONAL
- XSELL_MERCHANT_ID – Destination/Partner merchant ID. If omitted, MERCHANT_ID is used.
- XSELL_CUSTOMER_ID – Destination/Partner customer ID. If omitted, MERCHANT_CUSTOMER_ID is used.
- MERCHANT_SITE_ID
- MERCHANT_PRODUCT_ID

- MERCHANT_INVOICE_ID
- UDF01
- CUSTOMER_PASSWORD
- REBILL_FREQUENCY
- REBILL_START
- REBILL_COUNT
- REBILL_AMOUNT

Parameters:

request – GatewayRequest object describing the transaction to be performed, i.e. card number, expiration, amount, etc

response – GatewayResponse object describing the results of the transaction request, i.e. success/failure codes, transaction ID, auth-code, etc.

Returns:

Boolean value indicating whether the Xsell creation was uploaded successfully. If a "true" value is returned, the Xsell uploaded successfully, no transaction was performed though so any transaction id should be ignored. If a "false" value is returned, the upload failed and *response* contains an explanation of the failure.

PerformAuthOnly

PerformAuthOnly(request, response)

Submit an auth-only transaction to the RocketGate system.

Parameters:

request – GatewayRequest object describing the transaction to be performed, i.e. card number, expiration, amount, etc

response – GatewayResponse object describing the results of the transaction request, i.e. success/failure codes, transaction ID, auth-code, etc.

Returns:

Boolean value indicating whether the transaction was performed successfully and accepted by the banking institution. If a "true" value is returned, the transaction completed successfully and *response* contains details include the transaction ID, auth-code, etc. If a "false" value is returned, the transaction failed and *response* contains an explanation of the failure.

PerformTicket

PerformTicket(request, response)

Submit a ticket transaction to settle a previous auth-only transaction.

Parameters:

request – Request object describing the transaction to be performed. At a minimum, the *request* object must contain a reference to the original auth-only transaction in the GatewayRequest::TRANSACT_ID element. (See GatewayRequest documentation.)

response – GatewayResponse object describing the results of the transaction request, i.e. success/failure codes, transaction ID, auth-code, etc.

Returns:

Boolean value indicating whether the transaction was performed successfully and accepted by the banking institution. If a "true" value is returned, the transaction completed successfully and *response* contains details include the transaction ID, auth-code, etc. If a "false" value is returned, the transaction failed and *response* contains an explanation of the failure.

PerformPurchase

PerformPurchase(request, response)

Submit a complete auth-capture transaction to the RocketGate system. In effect, this method performs a combined auth-only and ticket operation.

Parameters:

request – GatewayRequest object describing the transaction to be performed, i.e. card number, expiration, amount, etc.

response – GatewayResponse object describing the results of the transaction request, i.e. success/failure codes, transaction ID, auth-code, etc.

Returns:

Boolean value indicating whether the transaction was performed successfully and accepted by the banking institution. If a "true" value is returned, the transaction completed successfully and *response* contains details include the transaction ID, auth-code, etc. If a "false" value is returned, the transaction failed and *response* contains an explanation of the failure.

PerformCredit

PerformCredit(request, response)

Submit a credit transaction to the RocketGate system.

Parameters:

request – Request object describing the transaction to be performed, i.e. card number, expiration, amount, etc. Alternatively, the *request* object may contain a reference to a previous auth-only, ticket, or purchase transaction in the GatewayRequest::TRANSACT_ID element. (See GatewayRequest documentation.)
response – GatewayResponse object describing the results of the transaction request, i.e. success/failure codes, transaction ID, etc.

Returns:

Boolean value indicating whether the transaction was performed successfully and accepted by the banking institution. If a "true" value is returned, the transaction completed successfully and *response* contains details include the transaction ID, etc. If a "false" value is returned, the transaction failed and *response* contains an explanation of the failure.

PerformVoid

PerformVoid(request, response)

Submit a void transaction for a previous auth-only, ticket, purchase, or credit transaction.

Parameters:

request – Request object describing the transaction to be performed. At a minimum, the *request* object must contain a reference to an original auth-only, ticket, or purchase transaction in the GatewayRequest::TRANSACT_ID element. (See GatewayRequest documentation.)
response – Response object describing the results of the transaction request, i.e. success/failure codes, transaction ID, etc.

Returns:

Boolean value indicating whether the transaction was performed successfully and accepted by the banking institution. If a "true" value is returned, the transaction completed successfully and *response* contains details include the transaction ID, etc. If a "false" value is returned, the transaction failed and *response* contains an explanation of the failure.

PerformCardScrub

PerformCardScrub(request, response)

Submit purchase information to the RocketGate system for validation and fraud analysis. The information is not transmitted to the bank. However, fraud scores and card

information, such as the country of origin, are returned. In effect, this method performs a fraud scrub independent of a financial transaction.

Parameters:

request – Request object describing a proposed credit card transaction, i.e. card number, customer billing and email addresses, customer IP address, etc.. All fields except the credit card number are optional. However, providing more information in the request will provide more detailed and accurate results.

Response – Response object describing the results of the scrub request, i.e. scrub scores, country of origin for the credit card, etc.

Returns:

Boolean value indicating whether the card scrub was performed successfully. If a "true" value is returned, the scrub completed successfully and *response* contains details about the card and customer. If a "false" value is returned, the scrub failed and *response* contains an explanation of the failure.

PerformRebillCancel

PerformRebillCancel(request, response)

Schedules the termination of a recurring billing. The selected subscription/recurring billing will be scheduled to terminate at the next rebilling date.

Additionally, when called with REBILL_END_DATE=CLEAR this will clear an existing cancel date that has not yet occurred.

Parameters:

request – Request object describing the subscription/recurring billing to be terminated. At a minimum, the GatewayRequest::MERCHANT_CUSTOMER_ID element is required. The GatewayRequest::MERCHANT_INVOICE_ID can be used to select a specific subscription if a customer has more than one. Use of the GatewayRequest::MERCHANT_INVOICE_ID element is optional. If it is omitted, all subscriptions for the specified customer will be scheduled for termination. (See GatewayRequest documentation.)
GatewayRequest::REBILL_END_DATE element is optional and is submitted in order to clear a an existing cancel date that has not yet occurred.

Response – Response object describing the success or failure of the cancellation operation.

Returns:

Boolean value indicating whether the cancellation was performed successfully. If a "true" value is returned, the cancellation was successful and the specified subscription has been scheduled for termination. If a "false" value is returned, the cancellation failed and *response* contains an explanation of the failure.

PerformRebillUpdate

PerformRebillUpdate(request, response)

Modifies the scheduling and rates of a recurring billing. Optionally performs a prorated billing as part of the modification.

Parameters:

request – Request object identifying the subscription/recurring billing to be modified and the changes that are to be applied to the subscription. The MERCHANT_CUSTOMER_ID and MERCHANT_INVOICE_ID elements of the GatewayRequest must be used to identify the subscription that is to be modified. The REBILL_AMOUNT, REBILL_FREQUENCY, REBILL_START, and REBILL_END_DATE elements can be used to modify specific terms of the recurring billing operation. The AMOUNT element can be used to request a prorated charge as part of the subscription modification. See GatewayRequest documentation Appendix A for details and examples.

Response – Response object describing the success or failure of the update. If a prorated billing was included in the update, the response object also carries the details of the billing operation, e.g. auth-code, transaction ID, etc.

Returns:

Boolean value indicating whether the update was performed successfully. If a "true" value is returned, the update was successful and the specified changes have been applied to the subscription. If the request included a prorated charge, the response object carries the details of the billing, i.e. auth-code, etc.. If a "false" value is returned, the update failed and *response* contains an explanation of the failure.

SetReadTimeout

SetReadTimeout(seconds)

Sets the timeout for transaction responses from the RocketGate servers. By default, transactions timeout after 60 seconds.

Parameters:

seconds – Number of seconds to wait for a response from the RocketGate servers.

SetTestMode

SetTestMode(flag)

Turn testing mode on or off. By default, testing mode is off.

In testing mode, transactions are automatically routed to RocketGate test servers. Transactions that are sent to the test servers are not actually settled with financial institutions.

In testing mode, a merchant account number 9999 is automatically made available for testing. All test transaction requests should be coded to use this merchant account.

Note that in test mode, communications with the RocketGate servers is not encrypted. Therefore, only test card numbers provided by RocketGate should be submitted.

Parameters:

flag – Boolean value indicating whether the GatewayService instance should operate in test mode. If *flag* is true, the object will operate in test mode and will route all transactions to RocketGate test servers for processing. If *flag* is false, the object will route all transactions to live servers.

Changes

Version	Description
2.1	Added PerformCardScrub, PerformRebillCancel, and PerformRebillUpdate
2.2	Added REBILL_END_DATE=CLEAR for PeformRebillCancel()
2.3	Added ®
6.1.0	Added GenerateXSell function