Transaction Processing Solutions

Integration Guide

Table of Contents

1. Version Tracking

Version Number	Modification Date	Changes Completed
1.00	09-19-2016	Initial document.
1.10	05-13-2018	Updated Start/End parameters for Express Stats.
2.10	02-10-2018	Added Reactivation Activity report
2.20	05-10-2018	Added Reactivation URL retrieval
3.00	05-13-2019	Reformatting

3.10	06-03-2019	Updated PeriodType options for Dynamic Rebills

2. Introduction

This Integration Guide covers all aspects of integrating your site(s) with the Segpay Transaction Processing System. Each section of the guide is self-contained and includes all the information needed to transmit data to Segpay.

White listing Your Info

Depending on your account configuration, sending live MID tests may get caught in platform velocity checks or fraud scrub. To bypass these you could provide Segpay with any of the following values to be white listed:

- IP address
- Email address
- Credit card number

White listing is not necessary when using the Testing options below.

Testing

There are 2 ways to send test transactions.

1. Demo MID
    a. The Demo MID can be assigned to any ProductCode under your Account.
    b. With the Demo MID, any card number passing the MOD-10 check can be used for testing. The card will never be sent to the banking network with the Demo MID but the system will return XML responses with live EventCodes and ResponseCodes.
    c. It's important to keep track of which ProductCodes are using the Demo MID. When you're ready to process live transactions, your ProductCodes must be updated to use your live MID(s).
2. TestCard in Site Settings
    a. When editing one of your sites in the Segpay Gateway admin there is a **Test Credit Card Number** link that you can click to display a card you can use for any ProductCode under that Site.
    b. Using the site TestCard will only result in a Test response. It will not simulate live EventCodes.

3. XML Gateway

The remote gateway is for clients who use their own Join pages. The raw gateway needs interpretation and should never be exposed directly to your customers.

Input / Output Protocols

The gateway takes XML in, via an HTTP request, and sends back an XML response. Parameters are passed in by sending XML data via an HTTP POST variable called **XMLData.** The resulting page (of type text/xml) is your answer. Following are instructions for how to make the http post requests to perform various tasks on the Segpay system.

4. Credit card payments

Credit Cards

Process credit card transactions via a simple HTTP Post request.

Making the request

Below is an example request to process a credit card payment via an HTTP post to:

https://gateway.segpay.com/cgi-bin/auth.xcgi

```
<data>
<auth AccountID="12345" AccountPassword="mYp2ssw0rd"/>
<authrequest RequestID="1" AccountID=”12345” FirstName="Joe"
LastName="Somebody" Address1="123 Main St" City="Anywhere" State="IL"
Zip="66666" Country="840" Email="test@segpay.com"
CardNumber="444444xxxxxx1111" CVV="123" ExpDate="2020-01-01"
ProductCode="12345-12543-100" Amount="1.00" />
</data>
```

Note that the AccountID and AccountPassword values above are just examples. Please substitute your actual ID and password.

The example XML string above includes the following parameters:

Definition of Parameters

Parameter	*Definition*
AccountID	Your Account number
AccountPassword	Your password for request authentication
RequestID	Value will stay as "1" unless you send multiple data sets within one submission
AccountID	Your Account number (The AccountID where the request will be processed under)
FirstName	Customer's first name
LastName	Customer's last name
Address1	Customer's street address
City	Customer's billing city
State	Customer's billing state or province
Zip	Customer's billing zip or postal code
Country	Customer's billing country code (3 digits; 840 = USA)
Email	Customer's email address
CardNumber	Customer Card Number
CVV	CVV code from customer's card
ExpDate	Date format is YYYY-MM-DD (Since most customers don't have the DD available on their credit card, "01" can be passed in for DD. Ex: 2022-12-01)
ProductCode	Unique ID representing the purchase option associated with this request (in the format of 12345-12345-123)
Amount	Amount of the charge being processed (Required only for certain ProductCode types)

MerchantCode	Your gateway MID for the request (OPTIONAL)
Currency	Specific currency (use 3-character Alpha Code) (OPTIONAL)
User1/User2	Merchant-defined variables the platform will pass through and log (OPTIONAL)
AllowDuplicateSignup="1"	Allows multiple subscription signups to the same site (OPTIONAL)

Resulting XML Dataset (Example)

Below is an example of the type of XML response you could receive after credit card payments:

```
<!DOCTYPE three65api -authorize [
<!ELEMENT data (authresponse*)>
<!ATTLIST authresponse RequestID ID #REQUIRED
                TransactionID CDATA #REQUIRED
                EventCode CDATA #REQUIRED
                SubscriberID CDATA #REQUIRED
                Message CDATA #REQUIRED
                User1 CDATA #IMPLIED
                User2 CDATA #IMPLIED
                VisaCode CDATA #IMPLIED >
]>
<data>
<authresponse RequestID="1" TransactionID="1234567890" EventCode="S"
SubscriberID="123456789" Message="DemoMID Approved " VisaCode=""
AuthCode="12345" AuthResponse="DemoMID Approved " ResponseCode="100"
SubEventTypeID="100" MoreInfo="" AVSCode="Z" CVVCode="M"
CurrencyCode="USD" ThreeDID=""/>
</data>
```

In the example above, **ResponseCode=100** reflects an approval. See **Appendix C** of this document for the full list of Response Codes.

An **EventCode** of **A (for auth)** means the transaction is authorized and will be settled later. To control when settlement happens, ask your Segpay rep to create a "Gateway with Delayed Settlement" product for you, which will require you to submit the settlement request manually (see Section 6). Otherwise, if Segpay is handling the rebilling and you want us to automatically settle 'x' number of days after authorization, then you'll need a "Recurring with Automatic Settlement" product. Please contact gatewaysupport@segpay.com.

5. Other Tasks

Issue a Credit

Credit back a specific transaction.

Making the request

Below is an example credit request via an HTTP post to:

https://gateway.segpay.com/cgi-bin/credit.xcgi

```
<data>
     <auth AccountID="12345" AccountPassword="mYp2sswOrd"/>
     <creditrequest RequestID="1" TransactionID="1234567890"/>
</data>
```

The example XML string above includes the following parameters:

Definition of Parameters

Parameter	*Definition*
AccountID	Your Account number
AccountPassword	Your password for request authentication
RequestID	Value will stay as "1" unless you send multiple data sets within one submission
TransactionID	Unique 10-digit ID associated with the customer's transaction. Used for tracking the initial signup in the system.

Resulting XML Dataset (Example)

The XML response you receive after sending a credit request will be similar to the following example:

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE three65api-credit [
  <!ELEMENT data (creditresponse*)>
  <!ATTLIST creditresponse    RequestID ID #REQUIRED
                              SubscriberID CDATA #REQUIRED
                              TransactionID CDATA #REQUIRED
                              EventCode CDATA #REQUIRED
                              Message CDATA #REQUIRED
                              User1 CDATA #IMPLIED
                              User2 CDATA #IMPLIED >
]>
<data>
<creditresponse RequestID="1" SubscriberID="123456789"
TransactionID="1234567890" EventCode="C" CreditCode="150" Message="OK"
User1="test" User2="" CreditTransactionID="2345678901"/>
</data>
```

Parse for an **Event Code** of "**C**" (Credit), "**V**" (Void) or "**H**" (Pending Credit).

You will only submit the credit request once. Segpay will determine whether the status is "C", "V" or "H" and will return that value back to you in the XML response.

Issue a Partial Credit

Process a partial credit amount on a specific transaction.

Making the request

Below is an example of a partial credit request, made via an HTTP post to:

https://gateway.segpay.com/cgi-bin/credit.xcgi

```
<data>
     <auth AccountID="12345" AccountPassword="mYp2sswOrd"/>
     <creditrequest RequestID="1" RefundAmount="1.00"
TransactionID="123456789"/>
</data>
```

The example XML string above includes the following parameters:

Definition of Parameters

Parameter	*Definition*
AccountID	Your Account number
AccountPassword	Your password for request authentication
RequestID	Value will stay as "1" unless you send multiple data sets within one submission
RefundAmount	Specific amount of the partial credit being issued
TransactionID	Unique 10-digit ID associated with the customer's transaction. Used for tracking the initial signup in the system.

Resulting XML Dataset (Example)

The XML response you receive after sending a partial credit request will be similar to the following example:

```
<data>
        <auth AccountID="20003" AccountPassword="xxxxxxxxx"/>
        <creditrequest RequestID="1" RefundAmount="2.00" TransactionID="1492827341"/>
</data>
```

```
<data>

<creditresponse RequestID="1" SubscriberID="392252021" TransactionID="1492827389"
EventCode="C" CreditCode="150" Message="OK" User1="Descr test" User2=""
RefundAmount="2.00" CreditTransactionID="1492827389"/>

</data>
```

Cancel an Account

Process a rebill cancellation. This is only needed for rebilling Subscribers in order to cancel future Rebill transactions. It's like cancelling a rebill subscription.

Making the request

Below is an example cancellation request, made via an HTTP post to:

https://gateway.segpay.com/cgi-bin/cancel.xcgi

```
<data>
     <auth AccountID="12345" AccountPassword="mYp2sswOrd"/>
     <cancelrequest RequestID="1" SubscriberID="123456789"/>
</data>
```

The example XML string above includes the following parameters:

Definition of Parameters

Parameter	*Definition*
AccountID	Your Account number
AccountPassword	Your password for request authentication
RequestID	Value will stay as "1" unless you send multiple data sets within one submission
SubscriberID	Unique membership ID associated with the customer's original purchase (9 digits)

Resulting XML Dataset (Example)

The XML response you receive after sending a cancellation request will be similar to the following example:

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE three65api-cancel [
  <!ELEMENT data (cancelresponse*)>
  <!ATTLIST cancelresponse    RequestID ID #REQUIRED
                              SubscriberID CDATA #REQUIRED
                              TransactionID CDATA #REQUIRED
                              EventCode CDATA #REQUIRED
                              Message CDATA #REQUIRED
```

```
                                User1 CDATA #IMPLIED
                                User2 CDATA #IMPLIED >
]>
<data>
<cancelresponse RequestID="1" SubscriberID="100219" TransactionID="0"
EventCode="M" Message="OK" User1="" User2=""/>
</data>
```

Parse for the Event Code of "**M**" ("cancelled").

6. Auth and Settle a Transaction

Authorize a specific amount on the customer's credit card and settle the transaction later. This feature requires the "Gateway with Delayed Settlement" product (ProductType 'N').

Making the request

If you don't want to send a straight sale, the 'Auth then Settle' allows you to send the Auth request with the option of settlement later.

You will make two requests, one to Auth and one to Settle. For the Auth, use the same request as described for credit card payments in Section 4. The Event Code for the auth transaction will be "A". Below is an example of the request you will send when you want to **settle** the transaction, made via an HTTP post to:

https://gateway.segpay.com/cgi-bin/settle.xcgi

```
<data>
<auth AccountID="12345" AccountPassword="mYp2ssw0rd"/>
<settlementrequest RequestID="1" Amount="1.00"
TransactionID="123456789" />
</data>
```

The example XML string above includes the following parameters:

Definition of Parameters

Parameter	*Definition*
AccountID	Your Account number
AccountPassword	Your password for request authentication
RequestID	Value will stay as "1" unless you send multiple data sets within one submission
Amount	Amount to be authorized

TransactionID	Unique 10-digit ID associated with the customer's transaction. Used for tracking the initial signup in the system.

Resulting XML Dataset (Example)

The XML response you receive after sending a settle request will be similar to the following example:

```
<!DOCTYPE Segpayapi-settle [
<!ELEMENT data (settlementresponse*)>
<!ATTLIST settlementresponse RequestID ID #REQUIRED
                             TransactionID CDATA #REQUIRED
                             EventCode CDATA #REQUIRED
                             Message CDATA #REQUIRED
                             User1 CDATA #IMPLIED
                             User2 CDATA #IMPLIED >
]>

<data>
<settlementresponse RequestID="1" SubscriberID="123456789"
TransactionID="3456789012" EventCode="S" Message="OK" User1=""
User2=""/>
</data>
```

7. Client-Managed Gateway Rebills

Client-Managed Rebills are on-demand rebills where you keep track of the subscriber's rebill schedule and send the rebill requests when needed.

There are two types of *Gateway Recurring* Products that allow you to manage future rebills for existing customers:

Gateway Service Recurring (ProductType 'O'): Use this product for normal sales.

Gateway Service Recurring w/ Delayed Settlement (ProductType 'W'): Use this product to send an authorization request as the first transaction followed by a settlement request (similar to Section 6).

Use the same initial requests with these Gateway Recurring products as noted in Sections 1-6. When using these Gateway Recurring products, an initial sale response will contain EventCode=I because it's an initial with the possibility of future Rebill events.

Making the request

The additional request below is required for subsequent "rebills" for existing subscribers, made via an HTTP post to:

https://gateway.segpay.com/cgi-bin/rebillauth.xcgi

```
<data>
<auth AccountID="12345" AccountPassword="mYp2ssw0rd"/>
<rebillrequest RequestID="1" SubscriberID="123456789" Amount="9.95" />
</data>
```

Definition of Parameters

Parameter	*Definition*
AccountID	Your Account number
AccountPassword	Your password for request authentication
RequestID	Value will stay as "1" unless you send multiple data sets within one submission
SubscriberID	Unique membership ID associated with the customer's original purchase (9 digits)
Amount	Amount to be authorized.

Resulting XML Dataset (Example)

The XML response you receive after sending a rebill request will be similar to the following example:

```
<data>
<rebillresponse TransactionID="397867211" RequestID="1"
SubscriberID="123456789" EventCode="R" ResponseCode="100"
AuthResponse="Bank specific response"
Message=""/>
</data>
```

The SubscriberID will remain the same for all subsequent Gateway rebills.

8. Dynamic Rebills

Allows you to send the rebill terms of a subscription in the XML request at signup. Segpay will manage all future rebills based on the terms sent in the original request.

Product Type

Dynamic Rebills can be used only with Product Type 'A' (Gateway Service / Recurring / Delay Settlement).

Making the Request

Below is an example of an auth request for dynamic rebills:

<data>

<auth AccountID="20000" AccountPassword="P@ssw0rd"/>

<authrequest RequestID="1" FirstName="John" LastName="Smith" Address1="123 Main St." City="Miami" State="FL" Zip="33333" Country="840" Email="test@abcinc.com" CardNumber="4444111111111111" CVV="123" ExpDate="2018-01-01" AccountID="20000" ProductCode="20000-70232-100" **Amount="1.99" InitialMembership="2" RebillAmount="29.99" MaxRebill="127" PeriodType="INTERVAL" DateValue="30"** User1="Dynamic test" />

</data>

Definition of Parameters

Parameter	*Definition*
AccountID	Your AccountID for request authentication
AccountPassword	Your password for request authentication
RequestID	Value will stay as "1" unless you send multiple data sets within one submission
FirstName	Customer's first name
LastName	Customer's last name
Address1	Customer's street address
City	Customer's billing city
State	Customer's billing state or province
Zip	Customer's billing zip or postal code
Country	Customer's billing country code (3 digits; 840 = USA)
Email	Customer's email address
CardNumber	Customer's Card Number
CVV	CVV code from customer's card

ExpDate	Date format is YYYY-MM-DD (Since most customers don't have the DD available on their credit card, "01" can be passed in for DD. Ex: 2005-12-01)
AccountID	Your Account number (The AccountID the request will be processed under)
ProductCode	Unique ID representing the purchase option associated with this request (in the format of 12345-12345-123)
Amount	Initial Amount (signup) of the auth or initial sale. Required.
InitialMembership	Duration (in days) for the auth or initial sale. Required.
RebillAmount	Amount for all rebills upon conversion. Required.
MaxRebill	Max Number of times the rebill should occur (default is 127). Required.
PeriodType	Defines the frequency at which rebills are charged: INTERVAL – include the number of days between rebills in DateValue (see next row). EVERY_MON EVERY_TUE EVERY_WED EVERY_THU EVERY_FRI EVERY_SAT EVERY_SUN FIRST_MONTH – First day of the month SAME_DAY_MONTH – Rebill the same date of the month FIRST_MON_MONTH – Rebill the first Monday of the month FIRST_FRI_MONTH – Rebill the first Friday of the month LAST_MONTH – Rebill the last day of the month FIRST_QUARTER – Rebill on the 1st of the next quarter LAST_QUARTER – Rebill the last day of the quarter

	SAME_DAY_QUARTER – Rebill the same date of the quarter Required field.
DateValue	Required if the PeriodType value is one of the following: INTERVAL: Include the number of days between each rebill; for ex: DateValue=”30” will rebill every 30 days. SAME_DAY_MONTH: Include the date that you want to rebill each month; for ex: DateValue=”15” will rebill on the 15th day each month. SAME_DAY_QUARTER: Include the date that you want to rebill each quarter; for ex: DateValue=”15” will rebill on the 15th day of the first month each quarter (Jan 15th, April 15th, July 15th, October 15th).
User1	Merchant-defined variable the platform will pass through and log (OPTIONAL)
DelaySettle	If you want the initial signup to be an Auth (EventCode=A) and the first rebill (conversion) to be a settlement, include in the authrequest: **DelaySettle="1"** Without that parameter, the default authrequest will result in a normal initial sale (EventCode=I).

9. Auto Purchase

Traditional Auto Purchase

Auto Purchase allows you to process a new sale by submitting an existing SubscriberID to the XML Gateway.

Making the Request

An Auto Purchase request can be sent with a post to the following HTTP address:

https://gateway.segpay.com/cgi-bin/autopurchase.xcgi

```
<data>
<auth AccountID="12345" AccountPassword="mYp2ssw0rd"/>
```

```
<autopurchaserequest SubscriberID="18061569" NewAccountID="12345"
NewProductCode="12345-17758-100" />
</data>
```

The example XML string above includes the following parameters:

Definition of Parameters

Parameter	*Definition*
AccountID	Your AccountID for request authentication
AccountPassword	Your password for request authentication
SubscriberID	Unique membership ID associated with the customer's original purchase (9 digits)
NewAccountID	The AccountID where the request will be processed under
NewProductCode	Unique ID representing the purchase option associated with this request (in the format of 12345-12345-123)

Resulting XML Dataset (Example)

The XML response you receive after sending an Auto Purchase request will be similar to the following example:

```
<data>
<autopurchaseresponse RequestID="1" TransactionID="73839075"
EventCode="I" SubscriberID="18311302" Message="OK" MoreInfo="" />
</data>
```

Same SubscriberID Auto Purchase

Making the Request

If you want to process an Auto Purchase but keep the same SubscriberID assigned to the customer, use the following HTTP Address:

https://gateway.segpay.com/cgi-bin/ssauth.xcgi

With the above address, use the request data below:

```
<data>
<auth AccountID="12345" AccountPassword="mYp2ssw0rd"/>
```

```
<authrequest RequestID="1" SubscriberID="195504277" AccountID="12345" ProductCode="12345-17758-101" Amount="1.05" />
</data>
```

The example XML string above includes the following parameters:

Definition of Parameters

Parameter	*Definition*
AccountID	Your AccountID for request authentication
AccountPassword	Your password for request authentication
RequestID	Value will stay as "1" unless you send multiple data sets within one submission
SubscriberID	Unique membership ID associated with the customer's original purchase (9 digits)
AccountID	The AccountID the request will be processed under
ProductCode	Unique ID representing the purchase option associated with this request (in the format of 12345-12345-123)
Amount	Amount of the charge being processed (Required only for certain ProductCode types)

Resulting XML Dataset (Example)

The XML response you receive after sending a Same SubscriberID Auto Purchase request will be similar to the following example:

```
<data><authresponse RequestID="1" TransactionID="966054387" EventCode="S" SubscriberID="195504277" Message="DEMO20003 cc_authorizeRESPONSE cc_authorizeAUTH" VisaCode="" User1="demo test" AuthCode="cc_authorizeAUTHCODE" AuthResponse="DEMO20003 cc_authorizeRESPONSE cc_authorizeAUTH" ResponseCode="100" SubEventTypeID="100" MoreInfo="" AVSCode="-" CVVCode="-"/></data>
```

10. Event Notification

Event Notifications are background posts that we can send you after events occur in our system. For example, when an auth, single sale, initial sale, rebill, credit, cancellation, chargeback, termination and decline occur, Event Notifications are queued in the system and sent in the order they are queued.

Event Notification Variables

Below is a list of the variables included in the event notifications you receive.

Variable	*Definition*
EventCode	The customer's most recent transaction type. For a list of event codes, see Appendix A.
FirstName	Customer's first name.
Address1/Address2	Customer's street address.
User1/User2	User-defined variable that can store anything. We will store User1/2 in our database and it can be retrieved using our XML reporting (Express Stats) and the Segpay client interface. URL encoding and Base64 encoding can be used with User1/User2. The following characters are allowed: • Letters • Numbers • @ (at symbol) • = (equal) • , (comma) • & (ampersand) • - (dash) • . (period) • # (pound) • _ (underscore) • : (colon) • % (percentage)
IPAddress	IP of the customer's device.
OriginalTransactionID	ID that associates the customer's original transaction to events, such as Chargebacks and Credits.
City	Customer's city.

State	Customer's state; example, FL.
SiteID	Website where the customer signed up.
PackageID	Returned only when a cross-sale has occurred, the PackageID includes the two product codes joined together – first the primary site, and second the cross-sale; e.g.: 2000-100-1001-115.
OriginalEventCode	The customer's original transaction type. For a list of event codes, see Appendix A.
RefundTransactionID	Identifies the customer's most recent refund transaction.
Zip	Customer's zip code (US).
Last name	Customer's last name.
TransactionID	The ID that identifies the customer's most recent transaction. A unique transaction ID exists for every transaction in our system.
SubAccountID	User-defined variable available to merchants.
Email	Customer's email address.
CardType	Credit card type used for this transaction. One of the following: • V = VISA • M = MasterCard • D = Discover • A = Amex • J = JCB
PaymentType	Payment method used by the customer during signup
ProductCode	Unique ID created for each price point you have in the system
OfferID	User-defined variable available to merchants.
AccountID	The client's Account ID for this particular transaction
Amount	Amount of the customer's transaction. ex: 1.23

SitePassword	Used to store a password that can be passed from the client's site through the pay page and onto the client's notification script.
TimeStamp	The time this transaction occurred. (GMT STANDARD TIME)
Country	Customer's country, sent as a 3-digit code (840 = USA). The complete list of M49 codes can be found at: https://unstats.un.org/unsd/methodology/m49.
SiteUsername	Used to store a username that can be passed from the client's site through the pay page and onto the client's notification script.
SubscriberID	ID associated with the customer throughout every transaction that occurs. This ID is unique per customer and doesn't change throughout the billing cycle.

Event Notification FAQs

When Are Event Notifications Sent?

Event Notifications are queued in the system when an event occurs and sent immediately in the order they are queued. The events include auth, single sale, initial sale, rebill, credit, cancellation, chargeback, termination and decline.

Example Event Notification Response

Here is an example of a credit card approval response posted by the Segpay system:

```
EventCode=S&SubEventTypeID=100&FirstName=John&Address2=&User1=test&IPA
ddress=96.77.61.33&OriginalTransactionID=0&City=Boca%20Raton&State=FL&
SiteID=70862&PackageID=&OriginalEventCode=%3F&RefundTransactionID=0&Zi
p=33432&LastName=Smith&TransactionID=1587023665&SubAccountID=0&Email=j
smith%40gmail.com&CardType=m&PaymentType=cc&ProductCode=20003-70000-
100&OfferID=0&AccountID=20003&User2=&Amount=1.00&SitePassword=NONE&Tim
estamp=2019-04-
12%2014%3A52%3A43&Country=840&SiteUsername=NONE&Address1=980%20N.%20Fe
deral%20Hwy&SubscriberID=445364357
```

What Constitutes a Successful Event Notification?

When your system responds to our event notification with a 200 OK header and any content in the body, such as "success." It is not necessary to respond with body content as long as you don't send back an error such as "400."

The Segpay system will perform a HEAD to see if the Notification URL exists. If your program does NOT respond with a valid Response, then the Notification fails and no attempt is made to post the data.

If your system returns a 200 (or any valid) response to the HEAD, then the post will be made with the transaction parameters and values.

How do I Enable Event Notification?

Please email gatewaysupport@segpay.com with your AccountID, SiteID, Event Notification URL and Failover Email Address. Notifications for the following events will default to enabled:

Auth
Single Sale
Initial Sale
Rebill
Cancellation
Termination
Credit
Chargeback
Decline

What happens when an error occurs?

If Segpay can't post to the specified Notification URL in the admin control panel, we will retry the post up to **3** times. At that point, we will send the post to the **failover email address** listed in the Event Notification settings in the Segpay control panel.

Is Event Notification secure?

The post can be done via SSL and you can check to see where the post is coming from. The gateway IP list is located here: https://gateway.segpay.com/iplist.txt.

11. Express Stats

The Segpay Express Stats program allows E-Commerce providers the ability to request reporting data directly from Segpay through a CGI interface. This document specifies the requests you can make and the responses Segpay will send. Our goal is to provide large companies who have their own development staff the ability to receive the reporting functionality they need to run their business.

Communicating with Express Stats

Requests to Express Stats must be sent as a POST.

You can generate a FORM POST similar to the ones on your web browser for generating Segpay transactions.

The post should consist of the following:

Header:

The format will be the same, although the URL it contains may differ with each post:

```
<form method=POST action=https://gateway.segpay.com/express/index.php>
```

Body:

The body contains identification and request, and will be standard for all posts**:**

```
<input type="hidden" name="account_id" value="Your_Segpay_Account_ID">
<input type="hidden" name="password"
value="Your_Segpay_Account_Password">
```

Your account_id and password are the login values used for making XML requests.

The following fields are required:

```
<input type="hidden" name="start_date" value="YYYY-MM-DD">
<input type="hidden" name="end_date" value="YYYY-MM-DD">
<input type="hidden" name="account_list" value="AccountID, AccountID">
<input type="hidden" name="authorization" value="1" >
<input type="hidden" name="query_mode" value="transaction">
<input type="hidden" name="inputStream" value="1">
```

- Use **start date** and **end date** to specify the date range of your transaction search. Use **account list** to specify the AccountID value(s) you want included in your transaction search.
- Leave the values for **authorization**, **query_mode** and **inputStream** as they appear above: “1”, “transaction” and “1”, respectively.

Footer:

This footer closes the form with a SUBMIT button and is standard for all posts.

```
<input type="submit" value="Give_Me_My_Reports">
```

A standard post to request data with a start date of 01/15/2019 and an end date 01/31/2019 with a max record return of 50 would look like this:

```
<form method=POST action=https://gateway.segpay.com/express/index.php>
<input type="hidden" name="account_id" value="Your_Segpay_Account_ID">
<input type="hidden" name="password"
value="Your_Segpay_Account_Password">
<input type="hidden" name="start_date" value="2019-01-15">
<input type="hidden" name="end_date" value="2019-01-31">
<input type="hidden" name="account_list" value="20003,20004">
<input type="hidden" name="authorization" value="1" >
<input type="hidden" name="query_mode" value="transaction">
<input type="hidden" name="inputStream" value="1">
<input type="hidden" name="limit" value="50">
<input type="submit" name="submit" value="Give_Me_My_Reports">
</form>
```

A standard GET to do the same thing would look like this:

Click Here

In the above examples, you can see the following:

Segpay Express Stats URL = gateway.segpay.com
Segpay Express Stats Program = express/index.php
account_id = Your_Segpay_Account_ID
password = Your_Segpay_Account_Password
start_date = 2019-01-01
end_date = 2019-01-31
account_list = 20003,20004
authorization=1
query_mode=transaction
inputStream=1
limit = 50

When using Express Stats, post the data to us from within your CGI program and wait for our response. Typically, the response should arrive in a couple of seconds. In rare cases, or in cases where you have sent a lot of data, it could take a little longer for us to respond. The response will be in XML format. Your CGI program is responsible for parsing the XML into a format that you can use. There are many XML parsers available on the market for a host of different programming languages.

Variables Explained

Below is a list of variables you can post to Express Stats to retrieve various types of data:

Required variables:

account_id - The AccountID you use for XML requests

password - The password you use for XML requests

start_date - The start date of the records you are trying to retrieve. Example: 2019-01-15 (YYYY-MM-DD)

end_date - The end date of the records you are trying to retrieve. Example: 2019-01-31 (YYYY-MM-DD)

account_list -The list of accounts you would like to query

authorization – Set this value to “1”

query_mode – Set this value to “transaction”

inputStream – Set this value to “1”

These are optional fields that can be posted and are returned in the XML response:

transaction_id - Allows you to specify a single transaction number.

subscriber_id- Allows you to specify a single subscription number.

sub_account_id - Allows you to specify a single affiliate ID, this will only be populated if the signup was sent from an affiliate (0 for non-affiliate).

product_code - The unique ID created for each price point in the Segpay system. A specific product code can be retrieved.

customer_ip - Allows you to specify a single IP address.

limit - The number of records you would like to retrieve. The max number is 4000 records per request.

start_id - Allows you to specify a starting transaction number.

end_id - Allows you to specify an ending transaction number.

event_code - An event ID that is based on the outcome of the transaction. For a list of Event Codes, see Appendix A.

sub_event_code – A more granular description of the transaction type (i.e. Primary sale, cross sale, etc). For a list of SubEventTypeIDs, see Appendix B.

site_name - Can be used to store a username that can be passed from the client's site through the pay page and onto the client's notification script.

site_password - Can be used to store a password that can be passed from the client's site through the pay page and onto the client's notification script.

user1 - Allows you to retrieve the data for User1. Only populated if used during the initial transaction.

user2 - Allows you to retrieve the data for User2. Only populated if used during the initial transaction.

card_type - The credit card used in the transaction. Ex: "v" – VISA

payment_type - The payment method used for that particular transaction. Ex: "cc" – credit card

affiliate_id - Client Affiliate information

affiliate_tag - Affiliate Tag

amount - Amount of the transaction

merchant_code - MerchantCode value of transaction

invoice_number – Invoice Number

XML Response

Below is a sample XML response returned from Express Stats:

EventCode = "R" (rebill transaction)

<transaction Page="1" TransactionID="1580687078" PaymentType="cc" CardType="v" SubscriberID="430463351" EventCode="R" ResponseCode="100" SiteID="70000" ProductCode="20003-70000-109" MerchantCode="TSYX44412" Amount="5.00" ScheduledAction="R" ScheduledDate="2019-05-01 20:45:03" SubEventTypeID="301" User1="test" User2="test2" Timestamp="2019-04-01 20:15:02" ClientAffiliateID="0" AffiliateTag="" InitialTransactionID="1559938605" OriginalTransactionID="1580195500" ExternalTransactionID="909104403200" Response="Approved" FirstName="John" LastName="Smith" Address1="123 Main St." Address2="" City="Boca" State="FL" Zip="33445" Country="840" CardBIN="444433" CardLastFour="1111" ExpDate="2023-12-01" Email="jsmith@jsmith.com" Phone="" IPAddress="rebill.pl" SiteUsername="jsmith" SitePassword="NONE" CurrencyCode="USD" OriginID="0" Status="UNKNOWN" fieldValue="" SubscriberUserFields="CryptFileVersion,User1,User2" ARN="" ReasonCode="" ReasonNote="" ReasonDescription=""/>

12. Appendix A – Event Codes

A	=	Authorization – Card was authorized but funds are not yet settled
S	=	Single – One-time transaction on a Single/One-time product code.
I	=	Initial – Initial transaction on a recurring product code
R	=	Recurring – Recurring transaction on a recurring product code
C	=	Credit – The transaction has been refunded
V	=	Void – The transaction has been voided
H	=	Hold – The credit attempt is on hold, usually because it has been scheduled to be batched. A Credit will usually follow a Hold.
~	=	Pending Batch – The transaction is scheduled for a batch
!	=	Declined – The transaction has been declined
M	=	Cancelled – The subscription has been cancelled
D	=	Terminated – The subscription has expired
X	=	Chargeback – The transaction was charged back
T	=	Test – Test transaction using the system test card from site settings
F	=	Failed – Transaction failed
K	=	Converted – The transaction was converted to a new product code
?	=	Incomplete Transaction – Transaction failed for various reasons
L	=	Reactivated Subscriber- When a recently cancelled subscriber upgrades their membership via quick-convert. The result is a reactivated subscription

13. Appendix B – SubEventTypeID Codes

ID	Description	Sub. State
0	Unspecified SubEventType	Initial
100	Single Sale	Initial
200	Non-Trial Initial	Initial
201	Non-Trial Recurring	Rebilled
202	Non-Trial Converted	Converted
300	Trial - Initial Transaction	Initial
301	Trial Recurring	Rebilled
302	Trial Converted	Converted
600	Imported Data	Initial
601	Imported Data Rebill	rebilled
602	Imported Data Converted	Converted
1000	Quick-convert	Initial
1001	QuickConvert Instant Conversion Rebill	Rebilled
1002	QuickConvert Instant Conversion Converted	Converted
1100	Cancel/Keep Conversion	Initial
1101	Cancel/Keep Conversion Rebill	Rebilled
1102	Cancel/Keep Conversion Converted	Converted
1200	Member Reactivation	Initial
1201	Membership Reactivation Rebill	Rebilled
1202	Membership Reactivation Converted	Converted
1300	Cross Sale, Primary Item	Initial
1301	Cross Sale, Primary Item Rebill	Rebilled
1302	Cross Sale, Primary Item Converted	Converted
1400	Cross Sale, Secondary Item	Initial
1401	Cross Sale, Secondary Item Rebill	Rebilled
1402	Cross Sale, Secondary Item Converted	Converted
1500	Accelerated Conversion	Initial
1501	Accelerated Conversion Rebill	Rebilled
1502	Accelerated Convert, Cross Sale	Converted
1600	Members Plus	Initial
1601	Members Plus Rebill	Rebilled
1602	Members Plus Converted	Converted
4000	Test Transaction	Test

14. Appendix C – Response Codes

ResponseCode	Description
100	Approved
200	Declined
201	Expired Card
204	Invalid method for this card
207	Contact issuing bank
209	Invalid Terminal ID
210	Invalid amount
212	PIN is required
213	Integration or processor does not support 3D Secure
214	Card does not support this type of transaction
217	PIN is incorrect
218	Card is not registered
223	Wrong CAUV
224	Cardholder needs to change PIN before use
225	Card not active
226	Blocked due to invalid PIN
229	Card not operative (error in CVV)
230	Denomination not supported by card issuer
261	Brand not allowed by issuer
280	Card type not accepted by issuer
281	Card has debit restrictions
283	Authentication error : transaction
284	Authentication error
289	Processor blocked because of risk check rule
290	Decline by issuer - no reason given
291	CVV code is missing
299	The CVV2 has more than three digits
300	Refer to card issuer
301	Refer to card issuer, special condition
302	Invalid merchant or service provider
303	Pick up card
304	Do not honor
305	Error
306	Pick up card, special condition
307	Honor with ID
308	VIP approval
309	Invalid transaction
310	Invalid amount
311	Invalid account number
312	No such issuer
313	Re-enter transaction
314	No action taken
315	Unable to locate record in file
316	File is temporarily unavailable
317	No credit account
318	Pick up card (lost card)
319	Pick up card (stolen card)

320 Insufficient funds
321 No checking account
322 No savings account
323 Expired card
324 Incorrect PIN
325 Transaction not permitted to cardholder
326 Transaction not allowed at terminal
327 Activity amount limit exceeded
328 Restricted card
329 Security violation
330 Activity count limit exceeded
331 Response late
332 Allowable number of PIN-entry tries exceeded
333 Unable to locate previous message
334 Message inconsistent with original message
335 Nonexistent account
336 Key exchange validation failed
337 Invalid date
338 PIN crypto error
339 Incorrect CVV
340 Unable to verify PIN
341 Invalid auth life cycle
342 No reason to decline request for account or address verification
343 Issuer or switch inoperative
344 Destination cannot be found for routing
345 Transaction cannot be completed; violation of law.
346 System malfunction
347 Duplicate add, action not performed
348 Transaction rejected AVS is invalid
349 Expired Auth
350 Cash service not available
351 Cash request exceeds issuer limit
352 Decline for CVV2 failure
353 Approved; PVID missing, invalid or expired
354 Declined; PVID missing, invalid or expired
355 Invalid biller information
356 Card authentication failed
357 Forward to issuer
358 Forward to issuer
359 Decline; unable to go online
360 Approved (Express Rewards Program)
361 Service not permitted
362 Invalid effective date
363 Please wait
364 Format error
365 Invalid currency code
366 Reversal Accepted
367 Unable to refund. Transaction has been charged back
368 Missing Required Data
400 Invalid Message Type

401 Merchant ID not found
402 Invalid processing code
403 Debit not supported
404 Private label not supported
405 Invalid card type
406 Unit not active
407 Duplicate ID
408 Manual card entry invalid
409 Invalid track information
410 Unsupported card information
411 Master merchant record not found
412 Invalid card format
413 Invalid tran type
414 Invalid Track 2 field
415 Invalid Entry Mode
416 Invalid amount
417 Invalid Msg Format
418 Invalid ABA number
419 Record not found
420 Merchant ID Error
421 Refund not Allowed
422 Refund Denied
423 Report format not supported
424 No detail records for merchant
425 Group/Provider ID missing
426 Report type not available
427 No total records for merchant
428 Invalid block requested
429 No status record for merchant
430 Invalid receipt requested
431 Bad status
432 Batch totals mismatch
433 Issuer unavailable
434 Terminal ID not found
435 Invalid Response from Processor
436 Message not supported
500 Expired Credit Card
501 Insufficient credit
502 Exceeded credit limit
503 Card has credit restrictions
504 Maximum PIN retries reached
505 Invalid card length
506 Invalid Month
507 Invalid Year
508 Invalid card number
509 Data validation bank error
601 R01 -- insufficient funds to cover amount
602 R02 -- previously active account has been closed
603 R03 -- unable to Locate Account
604 R04 -- unauthorized debit to consumer account using corporate SEC Code

605 R05 -- invalid number failed check digit - invalid number of digits
606 R06 -- originating bank return
607 R07 -- customer revoked authorization -- cancel
608 R08 -- recurring debit has stopped payment
609 R09 -- insufficient funds because of outstanding transactions
610 R10 -- customer has not authorized account
611 R11 -- check safekeeping return (exceeds dollar amount, stale date, etc)
612 R12 -- account at a branch that has been sold to another financial institution
613 R13 -- Bad routing number or unauthorized financial institution
614 R14 -- account holder is dead
616 R16 -- funds unavailable due to specific action by the RDFI or by legal action
617 R17 -- bad input from processor
620 R20 -- non-transaction account prohibited or limited
621 R21 -- bad company id
622 R22 -- invalid CIE entry individual ID is used by the Receiver to identify the account
623 R23 -- receiver refuses credit entry
624 R24 -- has received what appears to be a duplicate entry
629 R29 -- notified by Receiver (nonconsumer) that entry was not authorized
631 R31 -- ODFI agrees to accept a return entry beyond normal return deadline
633 R33 -- returns an XCK entry (code only used for XCK returns)
651 C01 -- customer's account number is Incorrect
652 C02 -- customer's routing number is incorrect
653 C03 -- customer's routing number and DFI account numbers are incorrect
654 C04 -- customer's name is incorrect
655 C05 -- customer's account type (Savings/Checking) is incorrect
656 C06 -- customer's account numb is incorrect & transaction routed to wrong account type
700 Suspicion of Fraud
701 Contact issuing bank; Suspicion of fraud
702 General decline: withhold card
800 Swap down
801 Database error
802 Close fields - comm error
803 Release failed - comm error
804 Transaction timeout
805 Communication Failure
806 Processing temporarily not possible
807 Card issuer temporarily not reachable
900 Scrub Related
901 AVS - No Match
902 AVS - Zip does not match
903 AVS - Address does not match
904 Number of attempts was exceeded for period, CC
905 Number of approvals was exceeded for period, CC
906 Number of declines was exceeded for period, CC
907 Number of refunds was exceeded for period, CC
908 Number of attempts was exceeded for period, ACH Acct
909 Number of approvals was exceeded for period, ACH Acct
910 Number of declines was exceeded for period, ACH Acct
911 Number of refunds was exceeded for period, ACH Acct
912 Number of attempts was exceeded for period, Email

913	Number of approvals was exceeded for period, Email
914	Number of declines was exceeded for period, Email
915	Number of refunds was exceeded for period, Email
916	Number of attempts was exceeded for period, IP
917	Number of approvals was exceeded for period, IP
918	Number of declines was exceeded for period, IP
919	Number of refunds was exceeded for period, IP
920	Reject because BIN is in the negative database, admin
921	Reject because Country is in the negative database
922	Reject because CC is in the negative database
923	Reject because CC is in the negative database, CB
924	Reject because CC is in the negative database, Rule
925	Reject because CC is in the negative database, bank response
926	Reject because ach is in the negative database, admin
927	Reject because ach is in the negative database, CB
928	Reject because ach is in the negative database, Rule
929	Reject because ACH Acct is in the negative db, bank response
930	Reject because email is in the negative db, admin
931	Reject because email is in the negative db, CB
932	Reject because email is in the negative db, Rule
933	Reject because email is in the negative db, bank response
934	Reject because IPAddress is in the negative database
935	Reject because IPAddress is in the negative database, CB
936	Reject because IPAddress is in the negative database, Rule
937	Reject because ipaddress is in the negative database, bank response
938	Person entered junk, repetitive letters, curse words in first/last name, city etc
939	Address validation failure
940	Country does not match IP Country
941	BIN does not match Country BIN
942	ThreatMetrix rejected
943	General failure. Program failed for some internal error.
966	CARDCOUNTER - Card use exceeds allowed limits.
2000	Gateway Off-line
2001	Could not map external response code to system response code
2002	Problems with the arguments passed as parameters
2003	Feature Not Implemented
2004	Missing Information from Previous Transaction
2005	Unable to Parse MerchantFieldData
2006	Unable to Parse TransactionFieldData
2007	Invalid Transaction State
2100	General Connection Failure
2101	Connection Timeout
2102	Connection Refused
2103	Remote Server Error
2150	Remote HTTP 301
2151	Remote HTTP 302
2152	Remote HTTP 401
2153	Remote HTTP 403
2154	Remote HTTP 404
2155	Remote HTTP 500

2156	Remote HTTP 503
2200	General Response Error
2201	Unable to Parse Output
2202	Missing Response
2203	Invalid Response
2204	Un-handled Response
2500	Bad transaction in batch
2501	Duplicate batch sent
2600	Customer on Hold (card previously received ResponseCode 700)
2605	Customer Has Active Rebill
2998	Program crashed or gateway timed-out
2999	Can't execute