Chargent API Documentation

Document Version 1.7
June 24, 2020

Chargent

PAYMENT PROCESSING FOR SALESFORCE

Chargent API Documentation

Overview

Chargent's JSON API receives inputs that describe the payment instrument and payment options, then processes the specified payment on the defined payment gateway without storing any sensitive account number or card holder details at any time within the Salesforce database.

> Chargent's API is an extension of the **Salesforce SOAP API,** built 100% on the Salesforce platform, so it uses the same underlying data model and standard objects as those in the Salesforce API, in addition to Chargent's custom objects.

When a response is received from the Gateway the details are stored on the Transaction object. If a token is received as part of the response, that token is also written to the parent of the transaction (Chargent Order, or legacy Opportunity / Case packages).

Please note that development with or use of the Chargent API requires an active subscription to Chargent's Platform Edition or above.

API Methods

Class

```
tChargentOperations
```

Methods

```
webservice static TChargentResult AuthorizeOrder_ClickJSON

webservice static TChargentResult ChargeOrder_ClickJSON

webservice static TChargentResult ChargeAuthorizedTransaction_ClickJSON

webservice static TChargentResult RefundTransaction_ClickJSON
```

Input

Each method above receives a JSON string of key value pairs

```
{
	Parameter: value
}
```

AuthorizeOrder_ClickJSON

Note: Most gateways do not allow authorization on a Payment Type of Check

Parameter	Type	Max Length	Payment Type	Required for Type	Description
ObjectId	String (Id)	18	All	YES	The Id of the Chargent Order to associate the transaction to
CardMonth	String	2	Credit Card	YES	Card Expiration Month
CardYear	String	4	Credit Card	YES	2 of 4 digit Card Expiration Year
PaymentMethod	String	11	ALL	YES	One of the following: Credit Card Check **Must be set on the Chargent order**
CardType	String	10	Credit Card	YES	One of the available card types. Same values as the Card Type on the Chargent Order
CardSecurity	String	4	CreditCard	NO	The CVC / CVV code for the credit card. Requirement depend on Gateway
CardNumber	String	16	CreditCard	YES	The credit card

					number being authorized
ChargeAmount	Decimal	14.2	All	YES	The amount to authorize
`TokenOnly	Boolean	N/A	All	NO	Defaults to False. If true only register the payment instrument with the gateway and obtain a token. Does not authorize any amount
BillingFirst	String	255	All	NO	Billing First Name
BillingLast	String	255	All	NO	Billing Last Name
BillingAddress	String	255	All	NO	Billing Street
BillingCity	String	64	All	NO	Billing City
BillingState	String	Var	All	NO	The full Billing State or the Two Letter Abbreviation depending on gateway requirements
BillingZip	String	10	All	NO	Billing Zip / Postal Code
BillingCountry	String	64	All	NO	Country

Authorize Credit Card Example

```
{
        "ObjectId" : "0061400001FviaG",
        "CardMonth" : "02",
        "CardYear" : "17",
        "PaymentMethod" : "Credit Card",
        "CardType" : "Visa",
        "CardNumber" : "4111111111111111",
        "ChargeAmount" : 11.33
}
```

Tokenize Credit Card Example

```
{
        "ObjectId" : "0061400001FviaG",
        "CardMonth" : "02",
        "CardYear" : "17",
        "PaymentMethod" : "Credit Card",
        "CardType" : "Visa",
        "CardNumber" : "4111111111111111",
        "ChargeAmount" : 0.00,
        "RegisterTokenOnly" : true
}
```

Tokenize Bank Account Example

Note: Bank account fields only valid for tokenization

```
{
        "ObjectId" : "0061400001FviaG",
        "PaymentMethod" : "Check",
        "BankRoutingNumber" : "044000039",
        "BankAccountNumber" : "123456789",
        "BankAccountType" : "Checking",
        "BankName" : "Chase",
        "BankAccountName" : "Fred Wilson",
        "ChargeAmount" : 0.00,
        "RegisterTokenOnly" : true
}
```

Note: Tokenization must be enabled on the Gateway Record for this to successfully complete
Note: Bank Account example is for bank accounts in the US on the ACH network.
Note: The Charge Amount passed to the Chargent API must be represented as a decimal in dollars and cents. For example: 3.00 for $3.00.
Realex Note: Realex will return an error if you perform an authorization once you have a token on the Chargent Order record.

ChargeOrder_ClickJSON

Parameter	Type	Max Length	Payment Type	Required for Type	Description
ObjectId	String (Id)	18	All	YES	The Id of the Chargent Order to associate the transaction to
CardMonth	String	2	Credit Card	YES	Card Expiration Month
CardYear	String	4	Credit Card	YES	2 of 4 digit Card Expiration Year
PaymentMethod	String	11	ALL	YES	One of the following: Credit Card Check
CardType	String	10	Credit Card	YES	One of the available card types. Same values as the Card Type on the Chargent Order
CardSecurity	String	4	CreditCard	NO	The CVC / CVV code for the credit card. Requirement depends on Gateway
CardNumber	String	16	CreditCard	YES	The credit card number being authorized
ChargeAmount	Decimal	14.2	All	YES	The amount to authorize
BillingFirst	String	255	All	NO	Billing First Name
BillingLast	String	255	All	NO	Billing Last Name
BillingAddress	String	255	All	NO	Billing Street
BillingCity	String	64	All	NO	Billing City

BillingState	String	Var	All	NO	The full Billing State or the Two Letter Abbreviation depending on gateway requirements
BillingZip	String	10	All	NO	Billing Zip / Postal Code
BillingCountry	String	64	All	NO	Country
BankRoutingNumber	String	11	Check	YES	Routing Number
BankAccountNumber	String	24	Check	YES	Account Number
BankAccountType	String	Var	Check	YES	One of the following: Checking Savings Business Checking
BankName	String	129	Check	YES	Bank Name
BankAccountName	String	50	Check	YES	Account Holder Name

Charge Credit Card Example

```
{
	"ObjectId" : "0061400001FviaG",
	"BillingAddress" : "123 Main street",
	"BillingCity" : "Columbus",
	"BillingState" : "Ohio",
	"BillingZip" : "41234",
	"BillingCountry" : "USA",
	"CardMonth" : "02",
	"CardYear" : "17",
	"PaymentMethod" : "Credit Card",
	"CardType" : "Visa",
	"CardNumber" : "4111111111111111",
	"CardSecurity" : "123",
	"ChargeAmount" : 11.33
}
```

Charge Check Example

```
{
	"ObjectId" : "0061400001FviaG",
	"BankRoutingNumber" : "044000039",
	"BankAccountNumber" : "123456789",
	"BankAccountType" : "Checking",
	"BankName" : "Chase",
	"BankAccountName" : "Fred Wilson",
	"BillingAddress" : "123 Main street",
	"BillingCity" : "Columbus",
	"BillingState" : "Ohio",
	"BillingZip" : "41234",
	"BillingCountry" : "USA",
	"PaymentMethod" : "Check",
	"ChargeAmount" : 11.33
}
```

Charge Record with Existing Token Example

- Note, the record needs to be populated with the correct Charge Amount prior to making this call

```
{
	"ObjectId" : "a0F1400000TcHym"
}
```

ChargeAuthorizedTransaction_ClickJSON

Parameter	Type	Max Length	Required	Description
ObjectId	String (Id)	18	YES	The Id of the transaction to Capture. Transaction must be an approved authorization
ChargeAmount	Decimal	14.2	NO	The amount to capture if different from the authorized amount. Must be equal to or less than the authorized amount

Charge Authorized Example

```
{
        "ObjectId" : "a0F1400000TcHym"
}
```

RefundTransaction_ClickJSON

Parameter	Type	Max Length	Required	Description
ObjectId	String (Id)	18	YES	The Id of the transaction to Capture. Transaction must be an approved authorization
Amount	Decimal	14.2	NO	The amount to refund. Must be equal to or less than the transaction amount
CustomTransactionType	String	N/A	NO	If set, must be set to “PartialRefund” This will process a refund to the card on the related Chargent Order which may be different than the card used on the transaction. Usually referred to as an *Unlinked Refund*

Refund / Partial Transaction Example

```
{
        "ObjectId" : "a0F1400000TcHym",
        "Amount" : 10.00
}
```

Unlinked Refund Transaction Example

```
{
        "ObjectId" : "a0F1400000TcHym",
        "Amount" : 10.00,
        "CustomTransactionType" : "PartialRefund"
}
```

Unlinked Credit Example (Bank Accounts Only)

```
{
  "ObjectId" : "006000000000000000",
        "ChargeAmount" : 3.45,
        "BankRoutingNumber" : "123456",
        "BankAccountNumber" : "123456789",
        "BankAccountType" : "Checking",
        "BankName" : "Chase",
        "BankAccountName" : "Fred Wilson",
        "CustomTransactionType" : "ACH_Credit"
}
```

Output

The results are output as a TChargentResult object with the following format

Parameter	Type	Description
DebugString	String	If debug is enabled this will be the entire response from the gateway
Message (Deprecated)	String	**Use Response Data** ~~General variable multi-line message about the results. In the case of a successful transaction will display~~ ~~**Transaction Created (TRX-xxxxxxxx)**~~ ~~**Gateway returns the status: "Approved"**~~ ~~**Gateway Message: ""**~~
Reload	Boolean	Internal Use
Status	String	Status of the callout attempt. Either 'OK' or 'FAILURE'. **Note:** Does not indicate the approval status of the transaction
TransactID (Deprecated)	String (Id)	**Use Response Data** ~~The Salesforce ID for the transaction record.~~
transaction_JSON	String	Internal Use Only
order_JSON	String	Internal Use Only

responseData	String (JSON Object)	JSON String of the response data object which is a standardized way in which we present the response information - **This should be used to obtain information returned from the gateway**

Example Output

```
{
        Message: 'Transaction Created (TRX-xxxxxxxx)
                  Gateway returns the status: "Approved"
                  Gateway Message: ""',
        Reload: true,
        Status: 'OK',
        TransactID: 'a0F1400000TcHym',
        order_JSON: null,
        transaction_JSON: null,
        responseData:
                          {
                                  "transactionId":"SFTRANSACTIONID",
                                  "responseStatus":"Approved",
                                  "responseMessage":"This transaction has been approved",
                                  "responseCode":"1",
                                  "reasonText":null,
                                  "objectId":"SFOBJECTID",
                                  "gatewayId":"1234567",
                                  "chargentMessage":"Any messages if presented",
                                  "avsCode":"123"
                          }
}
```

Response data format is structured as follows:

Parameter	Type	Description
transactionId	String (Id)	The Id of the created Chargent Transaction
responseStatus	String	The status returned from the gateway
responseMessage	String	Any messages returned from the gateway
responseCode	String	The response code returned from the gateway
objectId	String (Id)	The Salesforce ID for the record that initiated the transaction
gatewayId	String	The gateway id for this transaction
chargentMessage	String	Any messages Chargent may need to present. Typically if there was an error
avsCode	String	The AVS code returned from the gateway

isApproved	Boolean	This field will return True if charge was successful for any gateway and it should return False for any other condition, including valid gateway response reason such as Declined or if there is an API system error.

Void Transaction

When a transaction needs to be voided (same day only), an API method that only requires the Id of the transaction record to act upon.

Methods

```
webservice static TChargentResult VoidTransaction_Click({TransactionId})
```

Testing Callouts

When testing the callouts that Chargent makes, you can't use the **HttpCalloutMock** class in Salesforce because it has to be done from within the Chargent managed package/namespace. To create sample responses and make sure that your code behaves appropriately, here is a workaround:

Wrap the actually call to Chargent Webservices in a

```
if(!test.isRunnungTest()){

     code that makes call out here

 }else{

   //create a transaction record manually or set what ever values need for
your test to continue

}
```

Uncommitted Work Errors

Purpose

When building a custom visualforce page that will be using Chargent Webservice Methods to process a charge there is a potential that you may receive an error from Salesforce when executing the Chargent Webservice Method as follows:

> ***You have uncommitted work pending. Please commit or rollback before calling out***

Background

One of the framework requirements when developing on the Salesforce platform is that you cannot perform any DML prior to making a callout to any external service in the same transaction. Chargent Webservices, when executed, makes a callout to the appropriate gateway to process the charge, thus you cannot perform any DML prior to executing a method from Chargent Webservices.
Solution

Within the lifecycle of a Visualforce page, each execution of a controller method from the Visualforce page itself is a separate transaction. The key is that individual execution of the method must come from the page and not from the controller.
The following is an example of a minimalistic page and controller. It is intended only to show the typical pattern that causes the issue.

Error Example:

Page

```
<apex:page standardController="Orders" extensions="example_Controller">

  <apex:pagemessage id="msgs"/>

  <apex:form>
    <apex:commandbutton action="{!update_and_charge}" rerender="msgs"/>
  <apex:form>

<apex:page>
```

Controller

```
public class example_Controller{

  public Orders o;

  public error_example(ApexPAges.standardController con){}
  }

public void update_and_charge(){
    update o;
    charge();
  }

public void charge(){
    ChargentOrders.TChargentOperations.TChargentResult result = ChargentOrders.tChargentOperations.ChargentOrders_Click(o.id);

  }
}
```

When the command button is clicked it executes the **update_and_charge()** method. That method then updates the Chargent Order record and then attempts to execute the **charge()** method. When the **charge()** method executes the ChargentWebservices **Orders_Click** method and attempts to make the callout to the gateway you will receive the error message.

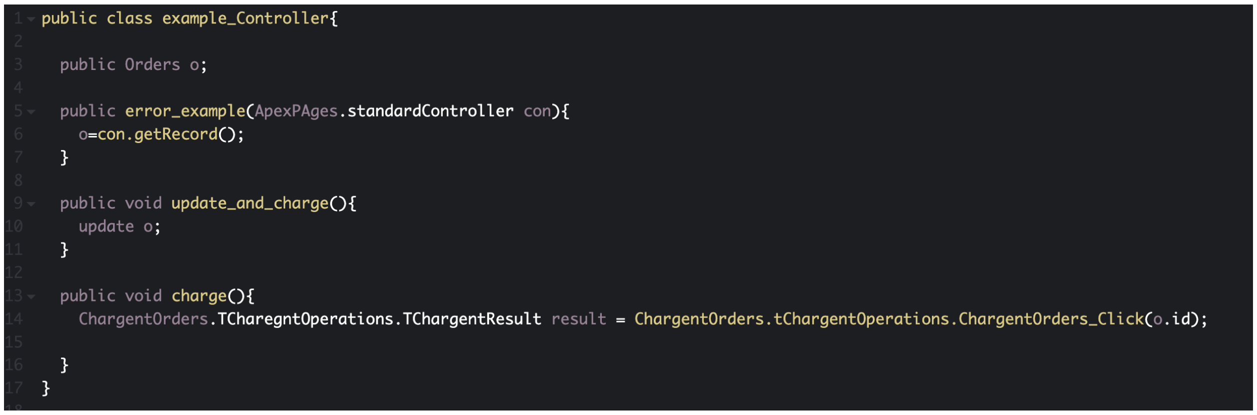
The solution is to break up the calls using the **oncomplete** tag on the command button in conjunction with an **apex:actionFunction** to call the **charge()** method. This solution follows:

Working Example

Page

```
<apex:page standardController="Orders" extensions="example_Controller">

  <apex:pagemessage id="msgs"/>

  <apex:form>

    <apex:actionFunction name="charge_it" action="{charge}" rerender="msgs"/>
    <apex:commandbutton action="{!update_and_charge}" onComplete="charge_it();" rerender="msgs"/>

  </apex:form>

</apex:page>
```

Controller



```
public class example_Controller{

  public Orders o;

  public error_example(ApexPAges.standardController con){
    o=con.getRecord();
  }

  public void update_and_charge(){
    update o;
  }

  public void charge(){
    ChargentOrders.TCharegntOperations.TChargentResult result = ChargentOrders.tChargentOperations.ChargentOrders_Click(o.id);

  }
}
```

Notice that on the page we added an apex:actionFunction to call the **charge()** method in the controller on completion of the **update_and_charge()** method. Also, we removed the call to the **charge()** method from the controller that was on line 11 in the error example.

In doing this, on click of the command button the **update_and_charge()** method is executed which updates the Chargent Order and then the transaction finishes and returns control to the page. The page then calls the **actionFunction** via the oncomplete tag and then executes the **charge()** method in the controller in a separate transaction without producing any errors.

Frequently Asked Questions (FAQ)

What are the Chargent API endpoints?

The endpoints are available in the WSDL you can download from Salesforce. Chargent is built 100% on Salesforce and our API is an extension of the Salesforce API. Chargent does not have its own endpoints.
https://developer.salesforce.com/docs/atlas.en-us.api.meta/api/sforce_api_partner.htm

Is Chargent's API SOAP or REST?

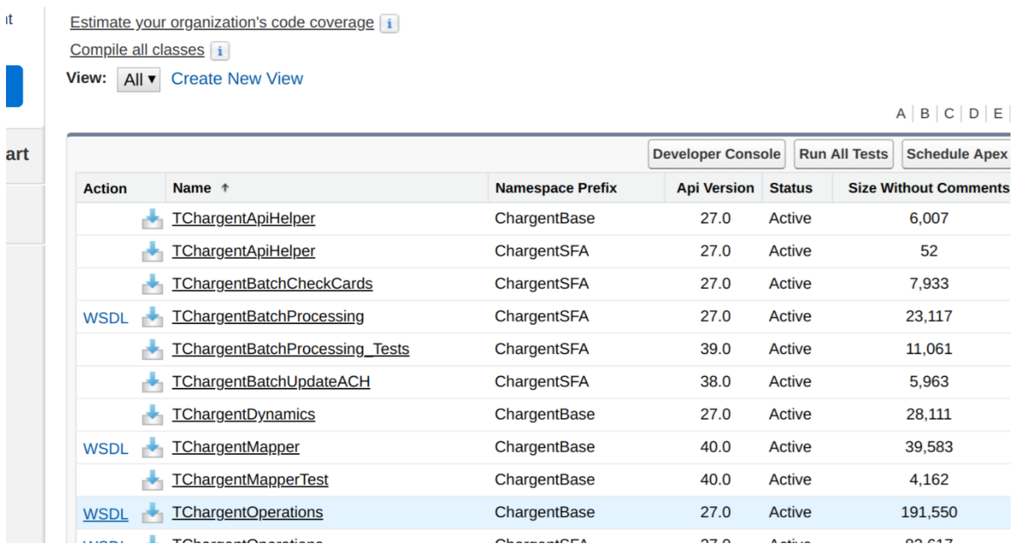
Chargent's API is an extension of Salesforce's SOAP API.
https://developer.salesforce.com/docs/atlas.en-us.212.0.api.meta/api/sforce_api_quickstart_intro.htm

"You have uncommitted work pending. Please commit or rollback before calling out."

One of the framework requirements when developing on the Salesforce platform is that you cannot perform any DML prior to making a callout to any external service in the same transaction. Chargent Webservices, when executed, makes a callout to the appropriate gateway to process the charge, thus you cannot perform any DML prior to executing a method from Chargent Webservices. If you experience this error, please see this for more information:
https://www.appfrontier.com/developers.html#uncommitted**Where can I get the Chargent WSDL?**

To see the Chargent specific methods in a WSDL follow this click path:
Setup | Develop | Apex Classes
then click the WSDL link on the TChargentOperations for ChargentBase.

We suggest using the ChargentBase namespace. Note that there is a TChargentOperations class for each of the Chargent transaction packages as well. All of them do the same thing. Experienced Chargent developers agree that sticking with the base class is simplest.


Estimate your organization's code coverage
Compile all classes
View: All Create New View

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments
	TChargentApiHelper	ChargentBase	27.0	Active	6,007
	TChargentApiHelper	ChargentSFA	27.0	Active	52
	TChargentBatchCheckCards	ChargentSFA	27.0	Active	7,933
WSDL	TChargentBatchProcessing	ChargentSFA	27.0	Active	23,117
	TChargentBatchProcessing_Tests	ChargentSFA	39.0	Active	11,061
	TChargentBatchUpdateACH	ChargentSFA	38.0	Active	5,963
	TChargentDynamics	ChargentBase	27.0	Active	28,111
WSDL	TChargentMapper	ChargentBase	40.0	Active	39,583
	TChargentMapperTest	ChargentBase	40.0	Active	4,162
WSDL	TChargentOperations	ChargentBase	27.0	Active	191,550

The Chargent WSDL defines how code can interact with the Chargent methods. Confused about what a WSDL is? [Read this page.]()

What if I am not using the Chargent Orders package?

If you are using Chargent's legacy Opportunities or Cases package (installed before 2018), then replace the word "Order" with "Opportunity" or Case"

1. AuthorizeOrder_ClickJSON would become
   a. AuthorizeOpportunity_ClickJSON or
   b. AuthorizeCase_ClickJSON

2. ChargeOrder_ClickJSON would become
   a. ChargeOpportunity_ClickJSON or
   b. ChargeCase_ClickJSON

3. The API methods for Transactions are unchanged, as all 3 Chargent packages have an object named Transaction.

4. In all cases where a record ID is required, instead of referring to the Chargent Order, you would refer to the Opportunity or Case (depending on which package you are running)

What is required to create a Transaction record via code?

If you are manually inserting a transaction record, versus having it created by calling one of the methods above, here are the minimal values required.

1. Type
    a. Charge, Void, Authorization, Refund
2. Response Status
    a. Approved, Declined, Rejected, etc
3. Order
    a. Id of the related Chargent Order

All other values on the transaction will depend on the reasons why you are manually creating a transaction. If the above 3 are populated then the transaction will be inserted without error so long as there is no other code acting on the record with additional requirements.

Note that Type=Charge and Response Status=Approved will cause a Transaction to be summed in Chargent's rollup summary fields such as Transaction Total.

Which fields on the Chargent Order get updated from a JSON API call?

The following fields on the Chargent Order record are updated by ClickJSON methods:

- Card_Year__c
- Card_Month__c
- Card_Name__c
- Bank_Routing_Number__c
- Bank_Account_Type__c

Why are my transactions are appearing as cents rather than dollars?

The Charge Amount passed to the Chargent API must be represented as a decimal in dollars and cents. For example: 3.00 for $3.00. If you send only the number 3, this can be interpreted as 3 cents. If you are seeing transactions go through as more or less than the amount that you want to send by two decimal points, make sure you have a decimal specified in your Charge Amount, eg. `"ChargeAmount" : 3.00`.

What if I have more than one active Gateway in Salesforce?

If you have more than one active gateway record set up in Chargent, you will need to populate the “Gateway” lookup field on each Chargent Order record after you create it, and prior to the API callout for a credit card or ACH transaction.

If you have more than one active gateway, and do not define on every record which one to use, you will receive the following error:

There is more than one active gateway configured. Please ask your administrator to make sure only one gateway is active.

This will be shown in the Chargent Transaction record created after the callout attempt.

I am receiving an error trying to Void a transaction after a Charge.

The Void of the transaction will need to be done asynchronously to avoid a loop or other errors. There are several ways to do it:

1. Trigger that evaluates the record after insert and if it needs to be Voided, you would need to call a custom @Future method that would then execute the Chargent tChargentOperations.voidTransaction_Click({Transaction Record Id})

2. You can run a batch process that would have a scope of one that would query for the records to be Voided and in the execute method you would call the same operation.