

APTTUS[®]

Billing Management on Salesforce Spring 2018 API Reference Guide

16 July 2018

Table of Contents

About this Guide	3
Overview	4
API Supported Packages.....	4
API Standards and Development Platforms	5
Standards	5
Development Platforms	5
Field Types	5
Billing APIs	7
Billing Services.....	7
Creating Credit Memo Documents.....	7
Creating Invoice Documents.....	8
Creating Invoices	10
Creating Invoices for Orders.....	13
Previewing Pending Usage Inputs.....	18
Processing Pending Usage Inputs	19
Processing Rated Usage Inputs.....	21
Updating Tax Calculations And Breakups on Credit Memos	21
Updating Tax Calculations and Breakups on Invoice.....	22
REST Services	23
Creating Invoices for Orders - REST Service	23
Creating Invoices - REST Service.....	28
Processing Pending Usage Inputs - REST Service.....	31
WSDL Services	35
Applying Credit Memos to Invoices.....	35
Applying Payments to Invoices	39
Adding a A/R Transaction - Deprecated	43
Adding Multiple A/R Transactions - Deprecated.....	45
Apttus Copyright Disclaimer	47

About this Guide

Apttus provides Application Programming Interfaces (API) for you to extend the features offered by Apttus Billing Management. These extensions add more functionality to the features available through configuration on Salesforce.

The Apttus Billing Management API Reference Guide describes the APIs provided to work with payments, usage inputs, credit memos, and invoice calculations.

Overview

- [API Supported Packages](#)
The package names and version numbers required for the APIs to work seamlessly.
- [API Standards and Development Platforms](#)
Apttus APIs are based on Salesforce APIs and use the same standards and platforms.
- [Field Types](#)
Apttus APIs use a subset of the supported data and field types on Salesforce.

API Supported Packages

The following packages and dependent packages are required for Billing Management APIs.

Package	Latest Certified Version (Name Number)
Apttus Billing Management	4.6.0120 4.120
Apttus Quote/Proposal-Configuration Integration	10.1.0254 10.254
Apttus Configuration & Pricing	10.5.1372.20 10.1372.20
Apttus CPQ API	9.4.0067 9.67
Apttus Quote/Proposal-Asset Integration	6.5.0014 6.14
Apttus Proposal Management	8.3.0160 8.160
Apttus Contract Management	8.9.0392 8.392

API Standards and Development Platforms

Apttus APIs are based on Salesforce APIs and use the same standards and platforms.

Standards

Name	Reference
Simple Object Access Protocol (SOAP) 1.1	http://www.w3.org/TR/2000/NOTE-SOAP-20000508
Web Service Description Language (WSDL) 1.1	http://www.w3.org/TR/2001/NOTE-wsdl-20010315
WS-I Basic Profile 1.1	http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

Development Platforms

Apttus SOAP API works with standard SOAP development environments. For a list of compatible development platforms, see [Salesforce Developer Force API](#) details.

Field Types

Apttus APIs use a subset of the supported data and field types on Salesforce.

The following table lists the field types that Apttus supports. For a comprehensive list of all field types supported by Salesforce, see [Salesforce Data Types](#).

Type	Description
Boolean	The Boolean field has a true(or 1) or false(or 0) value.
Data object	The Data Object field is an ID type and is represented by CPQ.nnDO in this document.

Type	Description
Date	The Date field contains date values only and do not contain relevant time values. Time in a date field is always set to midnight in the UTC time zone. If you want a timestamp you must use a dateTime field.
Decimal	The Decimal field provides an exact numeric value and you can arbitrarily size the precision and scale of the value.
ID	<p>The ID field is an alphanumeric field that acts as the primary key for a specific record associated with an object. The ID value includes a three-character code that identifies which object the record is associated with. The ID for a specific record does not change.</p> <p>For some objects, this field may also be a reference type value, which contains the ID value for a related record. They are identified by field names ending in 'Id', such as priceListId. The ID field acts like foreign keys and their values can be changed using an update() call.</p>
Integer	The Integer field contains whole numbers only. There are no digits after the decimal.
List	The List field includes a fixed set of values from which you must select a single value. Picklists are available as drop-down lists. If a picklist is unrestricted, the API does not limit entries to only currently active values.
String	The String field contains text and may have different length restrictions based on the data you store in the specific field. For instance, City may be limited to 50 characters, while AddressLine1 is limited to 255 characters.

Billing APIs

Apttus Billing Management APIs are categorized into:

Internal APIs

- [Billing Services](#)

External APIs

- [REST Services](#)
- [WSDL Services](#)

Billing Services

Apttus Billing APIs are based on Salesforce APIs and use the same standards and platforms.

You can invoke Apttus Billing APIs from the following command:

```
Apttus_Billing.BillingService.<Name of the Function>  
where the name of the function is API Name and its parameters.
```

Here is a list of APIs along with their parameters:

- [Creating Credit Memo Documents](#)
- [Creating Invoice Documents](#)
- [Creating Invoices](#)
- [Creating Invoices for Orders](#)
- [Previewing Pending Usage Inputs](#)
- [Processing Pending Usage Inputs](#)
- [Processing Rated Usage Inputs](#)
- [Updating Tax Calculations And Breakups on Credit Memos](#)
- [Updating Tax Calculations and Breakups on Invoice](#)

Creating Credit Memo Documents

createCreditMemoDocuments API is used to generate credit memo documents.

```
createCreditMemoDocuments(List<Id> creditMemolds, String creditMemoTemplateName)
```

It generates credit memo documents for the given list of credit memo IDs.

You can generate credit memo documents in the following formats:

- PDF
- DOCX

- DOC
- RTF

It accepts a list of Credit Memo IDs and a valid credit memo template name as input. If you do not specify a valid template name or pass **null** as a value, Billing Management system uses the default template provided at the Account or the Account Location.

If you call this API from a batch or a scheduled job, it can process only one CreditMemo ID. Otherwise, if you call this API from a non-batch or a non-scheduled job, it can process up to 10 CreditMemos.

Request			
Field	Type	Required?	Description
creditMemolds	List <Id>	Yes	IDs of credit memo
creditMemoTemplateName	String	No	Name of the credit memo template

Code Sample

```
Account testAccount = new Account(Name = 'Test Account');
insert testAccount;
CreditMemo__c testCreditMemo = new CreditMemo__c(BillToAccountId__c = testAccount.Id,
  CreditAmount__c = 40.0,
  Status__c = CreditMemo.STATUS_DRAFT);
insert testCreditMemo;
Apttus__APTS_Template__c cmTemplate = new Apttus__APTS_Template__c(Name = 'Default',
  Apttus__IsActive__c = true,
  Apttus__Type__c = 'Credit Memo');
insert cmTemplate;

List<Id> cmIdList = new List<Id> {testCreditMemo.Id};
//create Credit Memo Documents
Apttus_Billing.BillingService.createCreditMemoDocuments(cmIdList, cmTemplate.Name);
```

Creating Invoice Documents

createInvoiceDocuments API creates invoice documents for a given list of Invoice. It accepts a list of invoice IDs as input and produces invoice documents for each corresponding invoice ID. It also accepts Invoice Template Name as a parameter if you want to create Invoice Documents using a specific template.

You can generate invoice documents in the following formats:

- PDF
- DOCX
- DOC
- RTF

The document is generated in the format set at the **Billing Preference** in *Invoice Output Format* field.

If you call this API from a batch or a scheduled job, it can process only one Invoice ID. Otherwise, if you call this API from a non-batch or a non-scheduled job, it can process up to 10 Invoices.

There are two ways of creating Invoice Documents:

- Using the default invoice template
- Using a specific invoice template

createInvoiceDocuments(invoiceIDs)

This API uses the default Invoice Template specified on the Account or Account Location.

Request			
Field	Type	Required?	Description
invoiceIds	List <Id>	Yes	List of Invoice Ids

Code Sample

```
List<ID> invoiceIDs = new List<ID>();
//invoiceIDs.add('a4t1I000000H5hM');
Apttus_Billing.BillingService.createInvoiceDocuments(invoiceIDs);
```

createInvoiceDocuments(invoiceIDs, invoiceTemplateName)

This API is used to create invoice documents using a specific invoice template.

If you do not specify a valid template name or pass null as a value, Billing Management System uses the default template specified on the Account or Account Location.

Request			
Field	Type	Required?	Description
invoiceIds	List <Id>	Yes	List of Invoice Ids
invoiceTemplateName	String	Yes	Invoice Template Name

Code Sample

```
List<ID> invoiceIDs = new List<ID>();
//invoiceIDs.add('a4t1I000000H5hM');
//String invoiceTemplateName = 'myTestTemplate';
Apttus_Billing.BillingService.createInvoiceDocuments(invoiceIDs, invoiceTemplateName);
```

Creating Invoices

createInvoices API is used to create Invoices automatically. It accepts a list of Account IDs and creates Invoices for each AccountID.

createInvoices(billToAccountIDs, invoiceDate, targetDateTime)

This API is used to create invoices for given Account IDs. It accepts a Set of Account IDs, invoiceDate, and targetDateTime as input parameters. It creates invoices for all orders with billing schedules having status as pending billing and end date less than targetDateTime. All the new invoices are created with Invoice Date as the value mentioned in invoiceDate.

Request			
Field	Type	Required?	Description
billToAccountIds	Set <Id>	Yes	Set of bill to Account IDs
targetDateTime	Datetime	Yes	Process through Date
invoiceDate	Date	Yes	Invoice Creation Date

Code Sample

```

Set<ID> billToAccountIDs = new Set<ID>();
Date invoiceDate = Date.newInstance(2017, 1, 1);
Date myDate = Date.newInstance(2017, 1, 1);
Time myTime = Time.newInstance(3, 3, 3, 0);
DateTime targetDateTime = DateTime.newInstance(myDate, myTime);
Apttus_Billing.BillingService.createInvoices(billToAccountIDs, invoiceDate, targetDateTime);

```

createInvoices(billToAccountIDs, options)

This API is used to create invoices for given Account IDs. It accepts a set of Account IDs and a class containing invoice creation options. the class contains options that influence the creation of an Invoice such as Invoice Date, Invoice Through Date and Auto Approve.

Request			
Field	Type	Required?	Description
billToAccountIDs	Set <Id>	Yes	Set of bill to Account IDs
options	Apttus_Billing.InvoiceCreationOptions	Yes	Class holding the Invoice Creation Options

DataObject - Billing.InvoiceCreationOptions		
Field	Type	Description
abort	Boolean	If the process is aborted.
autoApprove	Boolean	Mandatory Constructor. If autoApprove for Invoice is set to true.
autoApproveAmount	Decimal	Mandatory Constructor. The amount provided for auto approval.
autoApproveCreditMemo	Boolean	Mandatory Constructor. If the autoApproveCreditMemo is set to true
autoApproveOperator	String	Mandatory Constructor. The selected autoApproveOperator such as Greater than, Less than etc.

DataObject - Billing.InvoiceCreationOptions		
Field	Type	Description
invoiceDate	Date	Mandatory Constructor. The Invoice Date.
invoiceDateType	String	The Invoice Date Type such as month or year.
invoiceRun	Apttus_Billing __InvoiceRun__c	The Invoice Run Object API Name.
invoiceRunResult	Apttus_Billing __InvoiceRunResult__c	The Invoice Run Result Object API Name.
numberAccountsProcessed	Integer	The number of accounts processed as part of the Invoice Runs.
numberCreditMemosGenerated	Integer	The number of Credit Memos generated.
numberInvoicesGenerated	Integer	The number of Invoices generated.
numberOfAccountsThatGeneratedCreditMemos	Integer	Number of accounts for which Credit Memos are generated.
numberOfAccountsThatGeneratedInvoices	Integer	Number of accounts for which Invoices are generated.
numberOfAutoApprovedInvoices	Integer	Number of Invoices that are auto-approved.
numberOfCreditMemosAutoApproved	Integer	Number of Credit Memos that are auto-approved.
numberOfSuppressedInvoices	Integer	Number of Invoices that are suppressed.
processThruDate	Date	Mandatory Constructor. The Process through date for invoice
splitInvoicesByOrder	Boolean	Mandatory Constructor. If Split Invoices By Order is set to true.

DataObject - Billing.InvoiceCreationOptions

Field	Type	Description
suppressInvoicesAmount	Decimal	Mandatory Constructor.The amount till which you want to suppress invoices.
suppressInvoicesOperator	String	Mandatory Constructor. .The selected Suppress Invoices operator such as Greater than, Less than etc.
taxCallbackStatus	String	The status of Tax Callback.

Code Sample

```

Set<ID> billToAccountIDs = new Set<ID>();
Date invoiceDate = Date.newInstance(2017, 1, 1);
Date processThruDate = Date.newInstance(2017, 1, 1);
Boolean autoApprove = true;
Boolean splitInvoicesByOrder = true;
String autoApproveOperator = 'Greater than';
Decimal autoApproveAmount = 5.00 ;
String suppressInvoicesOperator = 'Less than';
Decimal suppressInvoicesAmount = 2.00;
Boolean autoApproveCreditMemo = true;
Apttus_Billing.InvoiceCreationOptions options = new Apttus_Billing.InvoiceCreationOptions(
invoiceDate,
processThruDate,
autoApprove,
splitInvoicesByOrder,
autoApproveOperator,
autoApproveAmount,
suppressInvoicesOperator,
suppressInvoicesAmount,
autoApproveCreditMemo);

Apttus_Billing.BillingService.createInvoices(billToAccountIDs, options);

```

Creating Invoices for Orders

createInvoicesforOrder API is used to create invoices for given orders. It accepts a List of order IDs as input and produces invoices for each order ID.

createInvoicesForOrder(orderId, targetDateTime, invoiceDate)

This API is used to create invoices for given order ID. It accepts the order ID, targetDateTime and invoiceDate as input parameters. It creates invoices for the billing schedules that have status as pending billing and end date less than targetDateTime. All the new invoices are created with Invoice Date as the value mentioned in invoiceDate.

Request			
Field	Type	Required?	Description
orderId	ID	Yes	Order Id
targetDateTime	Datetime	Yes	Invoice Process Through Date
invoiceDate	Date	Yes	Invoice Creation Date

Response		
Field	Type	Description
Apttus_Config2__Invoice_c	List of Invoice objects	Invoices created for each order ID

Code Sample

```

ID orderId = new ID();
Date invoiceDate = Date.newInstance(2017, 1, 1);
Date myDate = Date.newInstance(2017, 1, 1);
Time myTime = Time.newInstance(3, 3, 3, 0);
DateTime targetDateTime= DateTime.newInstance(myDate, myTime);
List <Invoice_c> invoices = Apttus_Billing.BillingService.createInvoicesForOrder(orderID,
targetDateTime, invoiceDate);

```

createInvoicesForOrder(orderID, options)

This API is used to create invoices for the given order ID. It accepts the order IDs and a class containing invoice creation options. the class contains options that influence the creation of an Invoice such as Invoice Date, Invoice Through Date and Auto Approve. It returns a list of invoices.

Request			
Field	Type	Required?	Description
orderId	ID	Yes	Order Id
options	Apttus_Billing.InvoiceCreationOptions	Yes	Class holding the Invoice Creation Options

Response		
Field	Type	Description
Apttus_Config2__Invoice_c	List of Invoice objects	Invoices created for each order ID

DataObject - Billing.InvoiceCreationOptions			
Field	Type	Option Type	Description
abort	Boolean	Output	If the process is aborted.
autoApprove	Boolean	Input	Mandatory Constructor. If autoApprove for Invoice is set to true.
autoApproveAmount	Decimal	Input	Mandatory Constructor. The amount provided for auto approval.
autoApproveCreditMemo	Boolean	Input	Mandatory Constructor. If autoApproveCreditMemo is set to true

DataObject - Billing.InvoiceCreationOptions			
Field	Type	Option Type	Description
autoApproveOperator	String	Input	Mandatory Constructor. The selected autoApproveOperator such as Greater than, Less than etc.
invoiceDate	Date	Input	Mandatory Constructor. The Invoice Date.
invoiceDateType	String	Input	The Invoice Date Type such as month or year.
invoiceRun	Apttus_Billing__InvoiceRun__c	Input	The Invoice Run Object API Name.
invoiceRunResult	Apttus_Billing__InvoiceRunResult__c	Input/Output	The Invoice Run Result Object API Name.
numberAccountsProcessed	Integer	Output	The number of accounts processed as part of the Invoice Runs.
numberCreditMemosGenerated	Integer	Output	The number of Credit Memos generated.
numberInvoicesGenerated	Integer	Output	The number of Invoices generated.
numberOfAccountsThatGeneratedCreditMemos	Integer	Output	Number of accounts for which Credit Memos are generated.
numberOfAccountsThatGeneratedInvoices	Integer	Output	Number of accounts for which Invoices are generated.
numberOfAutoApprovedInvoices	Integer	Output	Number of Invoices that are auto-approved.

DataObject - Billing.InvoiceCreationOptions			
Field	Type	Option Type	Description
numberOfCreditMemosAutoApproved	Integer	Output	Number of Credit Memos that are auto-approved.
numberOfSuppressedInvoices	Integer	Output	Number of Invoices that are suppressed.
processThruDate	Date	Input	Mandatory Constructor. The Process through date for invoice
splitInvoicesByOrder	Boolean	Input	Mandatory Constructor. If Split Invoices By Order is set to true.
suppressInvoicesAmount	Decimal	Input	Mandatory Constructor. The amount till which you want to suppress invoices.
suppressInvoicesOperator	String	Input	Mandatory Constructor. The selected Suppress Invoices operator such as Greater than, Less than etc.
taxCallbackStatus	String	Output	The status of Tax Callback.

Code Sample

```

ID orderID = new ID();
Date invoiceDate = Date.newInstance(2017, 1, 1);
Date processThruDate = Date.newInstance(2017, 1, 1);
Boolean autoApprove = true;
Boolean splitInvoicesByOrder = true;
String autoApproveOperator = 'Greater than';
Decimal autoApproveAmount = 5.00 ;
String suppressInvoicesOperator = 'Less than';
Decimal suppressInvoicesAmount = 2.00;
Boolean autoApproveCreditMemo = true;

Apttus_Billing.InvoiceCreationOptions options = new Apttus_Billing.InvoiceCreationOptions(
    invoiceDate,
    processThruDate,
    autoApprove,
    splitInvoicesByOrder,
    autoApproveOperator,
    autoApproveAmount,
    suppressInvoicesOperator,
    suppressInvoicesAmount,
    autoApproveCreditMemo);

Apttus_Billing.BillingService.createInvoicesForOrder(orderID, options);

```

Previewing Pending Usage Inputs

previewPendingUsageInputs API is used to preview the rating amount of usage inputs without altering the Billing Schedules, Usage Schedules, and Usage Inputs. You can rate a maximum of 2000 Usage Inputs at a time.

previewPendingUsageInputs(Set<ID> usageInputIds)

The API accepts a set of Usage Input IDs as input. It validates the number of Usage Inputs to process. If the number of Usage Input Ids entered is less than 2000, it returns a map containing the rating amount for each Usage Input ID. If the number of Usage Input IDs given is more than 2000, Billing Management system throws an error.

Request			
Field	Type	Required?	Description
usageInputIds	Set <Id>	Yes	Set of Usage Input IDs

Response		
Field	Type	Description
mapRatedUsagInputIds	Map	Map containing Usage Inputs for each UsageInputId

Code Sample

```
Set<ID> usageInputIDs = new Set<ID>();
Map<ID, Apttus_Billing__UsageInput__c> mapRatedUsageInputs =
Apttus_Billing.BillingService.previewPendingUsageInputs(usageInputIDs);
for(Apttus_Billing__UsageInput__c usageInput : mapRatedUsageInputs.values()) {
    //Use usageInput.Apttus_Billing__RatedAmount__c
}
```

Processing Pending Usage Inputs

processPendingUsageInput API is used to process Usage Inputs. On completion of the batch job, the user receives a batch job status email.

You can process the usage inputs in two ways:

- Process all the loaded usage inputs
- Process an explicit set of usage inputs

processPendingUsageInput()

This API processes all the usage inputs with status as 'Loaded'.

Code Sample

```
Apttus_Billing.BillingService.processPendingUsageInput();
```

processPendingUsageInput(Set<ID> usageInputIDs)

This API processes usage inputs for all the usage inputs IDs with status as 'Loaded'. It accepts a Set of Usage Input IDs as input and checks the Usage Input Status for each Usage Input ID. It processes all the usage inputs with status as 'Loaded'.

Request			
Field	Type	Required?	Description
usageInputIDs	Set<Id>	No	Set of usage input IDs

Code Sample

```
Set<Id> usageInputIds = new Set<ID>{usageInput1.Id, usageInput2.Id};
Apttus_Billing.BillingService.processPendingUsageInput(usageInputIds);
```

processPendingUsageInput(Set<ID> usageInputIDs, Boolean processSynchronously)

This API processes usage inputs for all the usage inputs IDs with status as 'Loaded'. It accepts a Set of Usage Input IDs as input and checks the Usage Input Status for each Usage Input ID. It processes all the usage inputs with status as 'Loaded'. If processSynchronously is set to true, the usage inputs are processed synchronously without any batch job. You can process only a maximum of 1000 usage inputs. If processSynchronously is set to false, the API will schedule a batch job to process all the given usage inputs.

Request			
Field	Type	Required?	Description
usageInputIDs	Set<Id>	No	Set of usage input IDs
processSynchronously	Boolean	No	Indicates whether to process the Usage Inputs synchronously or asynchronously. The default value is false.

Code Sample

```
// To process Usage Inputs Synchronously
Set<ID> usageInputIDs = new Set<ID>(); //Max it can be 1000 IDs
Apttus_Billing.BillingService. processPendingUsageInput(usageInputIDs, true);
// To process Usage Inputs Asynchronously
Set<ID> usageInputIDs = new Set<ID>(); //Max it can be any number of IDs
Apttus_Billing.BillingService. processPendingUsageInput(usageInputIDs, false);
```

Processing Rated Usage Inputs

This API is used to unrate the processed usage input. It also reverts the Amount and Quantity from Billing Schedules, Usage Schedules, and Revenue Fee Schedules.

processRatedUsageInput(Set<ID> usageInputIDs)

It accepts a Set of Usage Input IDs as input. For all entered Usage Input IDs, it unrates all the usage Inputs and reverts the amount and quantity from the related billing schedules, usage scheules, and revenue fee schedules. On completion of the batch job, the user receives a batch job status email.

Request			
Field	Type	Required?	Description
usageInputIDs	Set <Id>	Yes	Set of usage input IDs

Code Sample

```
Set<Id> usageInputIds = new Set<ID>{usageInput1.Id, usageInput2.Id};
Apttus_Billing.BillingService.processRatedUsageInput(usageInputIds);
```

Updating Tax Calculations And Breakups on Credit Memos

updateCreditMemoTaxCalculationsAndBreakups API calculates Tax Amount and populates it on the Credit Memo.

updateCreditMemoTaxCalculationsAndBreakups(Set<ID> CreditMemoIDs, Boolean autoTransitionFromPendingApprovedToApproved)

This API calculates Tax Amount and Tax Breakups for credit memos with status as 'Draft' or 'Pending Approved'. After tax computation, it updates the Tax Amount on each credit memo line item and creates or

updates Credit Memo Line Item Tax Breakups. It then calculates the Total Tax Amount for the credit memo and creates or updates the Credit Memo Tax Breakups.

If `autoTransitionFromPendingApprovedToApproved` is set to true, it updates the status of Credit Memo from Pending Approved to Approved.

⚠ The billing administrator must register a Tax Callback class for `updateCreditMemoTaxCalculationsAndBreakups` API to work. For details on how to register a Tax Callback class, refer to Custom Settings for Tax.

Request			
Field	Type	Required?	Description
<code>creditMemoIDs</code>	Set <Id>	Yes	Set of Credit Memo Ids
<code>autoTransitionFromPendingApprovedToApproved</code>	Boolean	Yes	If set to true, the API updates the status of Credit Memo from Pending Approved to Approved. If set to false, the credit memo status is not updated.

Code Sample

The code sample below helps you calculate and fetch Tax Amount for credit memos.

```
Set<ID> creditMemoIDs = new Set<ID>();
//[{creditMemos[0].Id, creditMemos[1].Id};]
Apttus_Billing.BillingService.updateCreditMemoTaxCalculationsAndBreakups(creditMemoIDs, true);
```

Updating Tax Calculations and Breakups on Invoice

`updateInvoiceMemoTaxCalculationsAndBreakups` API calculates Tax Amount and populates it on the Invoice.

`updateInvoiceTaxCalculationsAndBreakups(Set<ID> InvoiceIds, Boolean autoTransitionFromPendingApprovedToApproved)`

This API calculates Tax Amount and Tax Breakups for invoices with status as 'Draft' or 'Pending Approved'. After tax computation, it updates the Tax Amount on each invoice line item and creates or updates Invoice Line Item Tax Breakups. It then calculates the Total Tax Amount for the invoice and creates or updates the Invoice Tax Breakups.

If `autoTransitionFromPendingApprovedToApproved` is set to true, it updates the status of the invoice from Pending Approved to Approved.

⚠ The billing administrator must register a Tax Callback class for `updateInvoiceTaxCalculationsAndBreakups` API to work. For details on how to register a Tax Callback class, refer to [Custom Settings for Tax](#).

Request			
Field	Type	Required?	Description
<code>invoiceIDs</code>	List <Id>	Yes	List of Invoice Ids
<code>autoTransitionFromPendingApprovedToApproved</code>	Boolean	Yes	If set to true, the API updates the status of the Invoice from Pending Approved to Approved. If set to false, the invoice status is not updated.

Code Sample

The code sample below helps you calculate and fetch Tax Amount for invoices.

```
Set<ID> invoiceIDs = new Set<ID>();
invoiceIDs.add('invoiceID')
Apttus_Billing.BillingService.updateInvoiceTaxCalculationsAndBreakups(invoiceIDs, true);
```

REST Services

Apttus Billing Management offers the following REST Services:

- [Creating Invoices for Orders - REST Service](#)
- [Creating Invoices - REST Service](#)
- [Processing Pending Usage Inputs - REST Service](#)

Creating Invoices for Orders - REST Service

`createInvoicesforOrder` API is used to create invoices for given orders. It accepts a List of order IDs as input and produces invoices for each order ID.

createInvoicesForOrder(orderId, targetDateTime, invoiceDate)

This REST Service is used to create invoices for given order IDs. It accepts a list of order ID, targetDateTime and invoiceDate as input parameters.

It creates invoices for billing schedules that have status as pending billing, and ReadyForInvoiceDate less than the targetDate. All the new invoices are created with Invoice Date as the value mentioned in invoiceDate.

Code Sample


```

MainClass.cs
=====
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            SalesforceClient client = new SalesforceClient();
            client.Username = "[yourUsername]";
            client.Password = "[yourPassword]";
            client.Token = "[Token]";
            client.ClientId = "[ConnectedApp_ClientID]";
            client.ClientSecret = "[ConnectedApp_SecretKey]";
            client.Login();

            client.createInvoiceForOrder();
        }
    }
}

SalesforceClient.cs
=====
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Text;

namespace ConsoleApplication1
{
    public class SalesforceClient
    {
        private const string LOGIN_ENDPOINT = "https://test.salesforce.com/services/oauth2/token";
        private const string API_ENDPOINT = "/services/apexrest/";

        public string Username { get; set; }
        public string Password { get; set; }
        public string Token { get; set; }
        public string ClientId { get; set; }
        public string ClientSecret { get; set; }
        public string AuthToken { get; set; }
        public string InstanceUrl { get; set; }

        static SalesforceClient()

```

```

    {
        // SF requires TLS 1.1 or 1.2
        ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12 |
SecurityProtocolType.Tls11;
    }

    public void Login()
    {
        String jsonResponse;
        using (var client = new HttpClient())
        {
            var request = new FormUrlEncodedContent(new Dictionary<string, string>
                {
                    {"grant_type", "password"},
                    {"client_id", ClientId},
                    {"client_secret", ClientSecret},
                    {"username", Username},
                    {"password", Password + Token}
                }
            );
            request.Headers.Add("X-PrettyPrint", "1");
            var response = client.PostAsync(LOGIN_ENDPOINT, request).Result;
            jsonResponse = response.Content.ReadAsStringAsync().Result;
        }
        Console.WriteLine(jsonResponse);

        var values = JsonConvert.DeserializeObject<Dictionary<string, string>>(jsonResponse);
        AuthToken = values["access_token"];
        InstanceUrl = values["instance_url"];
    }

    public void createInvoiceForOrder()
    {
        using (var client = new HttpClient())
        {
            string restRequest = InstanceUrl + API_ENDPOINT + "Apttus_Billing/
CreateInvoicesForOrder/v1/";

            string requestMessage = "{\\"orderId\\":\\"a2nW00000010D2G\\", \\"targetDate\\":
\\"2018-03-03\\", \\"invoiceDate\\":\\"2018-03-03\\"}";
            HttpContent content = new StringContent(requestMessage, Encoding.UTF8, "application/
json");

            HttpRequestMessage request = new HttpRequestMessage(HttpMethod.Post, restRequest);
            request.Headers.Add("X-PrettyPrint", "1");
            request.Headers.Add("Authorization", "Bearer " + AuthToken);
        }
    }

```

```

        request.Headers.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
        request.Content = content;

        var response = client.SendAsync(request);
        string result = response.Result.Content.ReadAsStringAsync().Result;

        Console.WriteLine(result);
        Console.ReadLine();
    }
}
}
}
}

```

Input Format

Internally the input format is converted to JSON as follows:

```

{
    "orderId" : "a2nW00000010D2A",
    "invoiceDate" : "2018-03-03",
    "targetDate" : "2018-03-03"
}

```

Output Format

Invoice is created for the given Order ID using the Invoice Date and targetDate parameters. Invoice is created only for billing schedules that have status as pending billing, and ReadyForInvoiceDate less than the targetDate.

```

[ {
    "attributes" : {
        "type" : "Apttus_Billing__Invoice__c",
        "url" : "/services/data/v31.0/objects/Apttus_Billing__Invoice__c/a5zW00000005EytIAE"
    },
    "Id" : "a5zW00000005EytIAE",
    "Name" : "INV-00000012",
    "Apttus_Billing__Type__c" : "Standard",
    "Apttus_Billing__TotalInvoiceAmount__c" : 2400.00000,
    "Apttus_Billing__ShipToAccountId__c" : "001W000000LSAJHIA5",
    "Apttus_Billing__InvoiceDate__c" : "2018-03-03T12:39:23.000+0000",
    "Apttus_Billing__DueDate__c" : "2018-03-03T12:39:23.000+0000",
    "Apttus_Billing__BillToAccountId__c" : "001W000000LSAJHIA5"
} ]

```

Creating Invoices - REST Service

createInvoices API is used to create Invoices automatically. It accepts a list of Account IDs and creates Invoices for each AccountID.

createInvoices(billToAccountIDs, invoiceDate, targetDateTime)

This API is used to create invoices for given Account IDs. It accepts a list of Account IDs, invoiceDate, and targetDateTime as input parameters. It creates invoices for billing scheules that have status as pending billing, and ReadyForInvoiceDate less than the targetDate. All the new invoices are created with Invoice Date as the value mentioned in invoiceDate.

Code Sample

```

MainClass.cs
=====
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            SalesforceClient client = new SalesforceClient();
            client.Username = "[yourUsername]";
            client.Password = "[yourPassword]";
            client.Token = "[Token]";
            client.ClientId = "[ConnectedApp_ClientID]";
            client.ClientSecret = "[ConnectedApp_SecretKey]";
            client.Login();

            client.createInvoice();
        }
    }
}

SalesforceClient.cs
=====
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Text;

namespace ConsoleApplication1
{
    public class SalesforceClient
    {
        private const string LOGIN_ENDPOINT = "https://test.salesforce.com/services/oauth2/token";
        private const string API_ENDPOINT = "/services/apexrest/";

        public string Username { get; set; }
        public string Password { get; set; }
        public string Token { get; set; }
        public string ClientId { get; set; }
        public string ClientSecret { get; set; }
        public string AuthToken { get; set; }
        public string InstanceUrl { get; set; }

        static SalesforceClient()

```

```

    {
        // SF requires TLS 1.1 or 1.2
        ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12 |
SecurityProtocolType.Tls11;
    }

    public void Login()
    {
        String jsonResponse;
        using (var client = new HttpClient())
        {
            var request = new FormUrlEncodedContent(new Dictionary<string, string>
                {
                    {"grant_type", "password"},
                    {"client_id", ClientId},
                    {"client_secret", ClientSecret},
                    {"username", Username},
                    {"password", Password + Token}
                }
            );
            request.Headers.Add("X-PrettyPrint", "1");
            var response = client.PostAsync(LOGIN_ENDPOINT, request).Result;
            jsonResponse = response.Content.ReadAsStringAsync().Result;
        }
        Console.WriteLine(jsonResponse);

        var values = JsonConvert.DeserializeObject<Dictionary<string, string>>(jsonResponse);
        AuthToken = values["access_token"];
        InstanceUrl = values["instance_url"];
    }

    public void createInvoice()
    {
        using (var client = new HttpClient())
        {
            string restRequest = InstanceUrl + API_ENDPOINT + "Apttus_Billing/CreateInvoices/
v1/";

            string requestMessage = "{ \"billToAccountIds\": [ \"001W000000eclP8\",
\\\"001W000000dkQ2c\\\", \\\"001W000000LSAJH\\\" ], \"targetDate\": \"2018-03-03\", \"invoiceDate\":
\\\"2018-03-03\\\" }";
            HttpContent content = new StringContent(requestMessage, Encoding.UTF8, "application/
json");

            HttpRequestMessage request = new HttpRequestMessage( HttpMethod.Post, restRequest);

```

```

request.Headers.Add("X-PrettyPrint", "1");
request.Headers.Add("Authorization", "Bearer " + AuthToken);
request.Headers.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
request.Content = content;

var response = client.SendAsync(request);
string result = response.Result.Content.ReadAsStringAsync().Result;

Console.Write(result);
Console.ReadLine();

    }
}
}
}

```

Input Format

Internally the input format is converted to JSON as follows:

```

{
    "billToAccountIds" : ["001W000000ec1P8", "001W000000dkQ2c", "001W000000LSAJH"],
    "targetDate" : "2018-03-03",
    "invoiceDate" : "2018-03-03"
}

```

Output

Invoice is created for the given list of Account IDs using the Invoice Date and targetDate parameters. Invoice is created only for billing schedules that have status as pending billing, and ReadyForInvoiceDate less than the targetDate. This API does not return any output.

Processing Pending Usage Inputs - REST Service

processPendingUsageInput API is used to process Usage Inputs.

processPendingUsageInput()

This API processes all the usage inputs with status as 'Loaded'. On completion of the batch job, the user receives a batch job status email.

Code Sample


```

MainClass.cs
=====
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            SalesforceClient client = new SalesforceClient();
            client.Username = "[yourUsername]";
            client.Password = "[yourPassword]";
            client.Token = "[Token]";
            client.ClientId = "[ConnectedApp_ClientID]";
            client.ClientSecret = "[ConnectedApp_SecretKey]";
            client.Login();

            client.processPendingUsageInputs();
        }
    }
}

SalesforceClient.cs
=====
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Text;

namespace ConsoleApplication1
{
    public class SalesforceClient
    {
        private const string LOGIN_ENDPOINT = "https://test.salesforce.com/services/oauth2/token";
        private const string API_ENDPOINT = "/services/apexrest/";

        public string Username { get; set; }
        public string Password { get; set; }
        public string Token { get; set; }
        public string ClientId { get; set; }
        public string ClientSecret { get; set; }
        public string AuthToken { get; set; }
        public string InstanceUrl { get; set; }

        static SalesforceClient()
    }
}

```

```

    {
        // SF requires TLS 1.1 or 1.2
        ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12 |
SecurityProtocolType.Tls11;
    }

    public void Login()
    {
        String jsonResponse;
        using (var client = new HttpClient())
        {
            var request = new FormUrlEncodedContent(new Dictionary<string, string>
                {
                    {"grant_type", "password"},
                    {"client_id", ClientId},
                    {"client_secret", ClientSecret},
                    {"username", Username},
                    {"password", Password + Token}
                }
            );
            request.Headers.Add("X-PrettyPrint", "1");
            var response = client.PostAsync(LOGIN_ENDPOINT, request).Result;
            jsonResponse = response.Content.ReadAsStringAsync().Result;
        }
        Console.WriteLine(jsonResponse);

        var values = JsonConvert.DeserializeObject<Dictionary<string, string>>(jsonResponse);
        AuthToken = values["access_token"];
        InstanceUrl = values["instance_url"];
    }

    public void processPendingUsageInputs()
    {
        using (var client = new HttpClient())
        {
            string restRequest = InstanceUrl + API_ENDPOINT + "Apttus_Billing/ProcessUsageInput/
v1/";

            var request = new HttpRequestMessage(HttpMethod.Post, restRequest);
            request.Headers.Add("Authorization", "Bearer " + AuthToken);
            request.Headers.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
            request.Headers.Add("X-PrettyPrint", "1");
            var response = client.SendAsync(request).Result;
            Console.WriteLine(response);
            Console.ReadLine();
        }
    }
}

```

```
}
}
```

Input Format

This API requires no input. All the Usagelnputs with status as Loaded are processed.

Output Format

This API does not return any output. On completion of the batch job, the user receives a batch job status email.

WSDL Services

Apttus Billing Management offers the following WSDL Services:

- [Applying Credit Memos to Invoices](#)
- [Applying Payments to Invoices](#)
- [Adding a A/R Transaction - Deprecated](#)
- [Adding Multiple A/R Transactions - Deprecated](#)

Prerequisites for invoking WSDL Services:

1. You need to Generate Entripse WSDL and Apex WSDL for ARTransactionService. For information on generating enterprise WSDL, refer to [Generating WSDL](#).
2. Add both the Generated WSDLs as a reference in your .Net Project.

Applying Credit Memos to Invoices

This API indirectly applies a Credit Memo to an Invoice by creating Related A/R Transaction records to reduce the tracked balance of the Invoice and the Credit Memo (double entry accounting).

You can use this API for a single invoice or multiple invoices. For each Related A/R Transaction Input passed in, a Related A/R Transaction Result will be returned. However, the order of the results may not be the same as the order of the inputs.

For more details on A/R Transactions, refer to Related A/R transactions in Billing Management User Guide.

applyCreditMemosToInvoices(List<RelatedARTxnInput2> inputs)

Request			
Field	Type	Required?	Description
inputs	List <RelatedARTxnInput2>	Yes	List of RelatedARTaxInput2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput2		
Field	Type	Description
Description	String	Description for the transaction.
DestinationObjId	ID	Id of the Destination Object.
ExternalSystemStatus	String	Status of the External System.
IntegrationDate	Datetime	Date of the system Integration.
ReasonCode	String	Reason for applying the A/R transaction.
SourceObjId	ID	Id of the Source Object.
transactionAmount	Decimal	The transaction amount.
transactionDate	Datetime	The transaction date.
transactionISOCurrency	String	The transaction Currency.
transactionNumber	String	The transaction number.
transactionSubType	String	Sub-type of transaction (picklist value).
transactionType	String	Type of transaction.

Response		
Field	Type	Description
results	List <RelatedARTxnResult2>	List of RelatedARTaxResult2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnResult2		
Field	Type	Description
destinationARTransactionId	ID	A/R transaction Id for destination object.

Apttus_Billing.RelatedARTransactionService.RelatedARTxnResult2		
destinationObjId	ID	The id of Destination object.
errorString	String	The Error message. Null value implies success and non-empty string value implies failure.
sourceObjId	ID	The id of Source object.
Status	String	The Status of a transaction will be success or failure.
transactionNumber	String	The transaction number.

Code Sample

```

using System;
using sforce = ConsoleApplication1.Sforce1;
using ARService = ConsoleApplication1.ARTransactionService;
using System.Net;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            String sessionId;
            using (sforce.SoapClient client = new sforce.SoapClient())
            {
                ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12 |
SecurityProtocolType.Tls11;

                //call login service
                sforce.LoginResult result = client.login(null, "[ORG_UserName]", "[ORG_Password]");
                //extract authentication token
                sessionId = result.sessionId;
                //serviceUrl = result.serverUrl;
                Console.WriteLine("Token generated: " + sessionId);
            }

            //call Related AR Transaction SOAP API
            ARService.SessionHeader header = new ARService.SessionHeader();
            header.sessionId = sessionId;

            using (ARService.RelatedARTransactionServicePortTypeClient soapClient = new
ARService.RelatedARTransactionServicePortTypeClient())
            {
                ARService.RelatedARTxnInput2[] inputs = new ARService.RelatedARTxnInput2[1];
                ARService.RelatedARTxnInput2 input2 = new ARService.RelatedARTxnInput2();
                input2.destinationObjId = "[InvoiceID]";
                input2.sourceObjId = "[CreditMemoID]";
                input2.transactionAmount = 100.00M;
                input2.description = "test description";
                input2.integrationDate = DateTime.Today;
                input2.transactionDate = DateTime.Today;
                input2.transactionNumber = "AR12342";
                input2.transactionType = "Credit Memo";
            }
        }
    }
}

```

```

        inputs[0] = input2;
        ARService.RelatedARTxnResult2[] arResults = new ARService.RelatedARTxnResult2[1];
        soapClient.applyCreditMemosToInvoices(header, null, null, null, inputs, out
arResults);

        Console.WriteLine("ARResult is: " + arResults[0].destinationARTransactionId);
        Console.ReadLine();
    }
}
}
}
}

```

Applying Payments to Invoices

This API indirectly applies a Payment to an Invoice by creating Related A/R Transaction records to reduce the tracked balance of the Invoice and the Payment (double entry accounting).

If no Payment exists for the specified Transaction Number then a Payment will be created with the Payment Amount set to the Transaction Amount.

The API can be for a single Invoice or multiple Invoices. For each Related A/R Transaction Input passed in, a Related A/R Transaction Result will be returned. However, the order of the results may not be the same as the order of the inputs.

For more details on A/R Transactions, refer to Related A/R transactions in Billing Management User Guide.

applyPaymentsToInvoices(List<RelatedARTxnInput2> inputs)

Request		
Field	Type	Description
inputs	List <RelatedARTxnInput2>	List of RelatedARTaxInput2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput2		
Field	Type	Description
Description	String	Description for the transaction.
destinationObjId	ID	Id of the Destination Object.

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput2		
externalSystemStatus	String	Status of the External System.
integrationDate	Datetime	Date of the system Integration.
reasonCode	String	Reason for applying the A/R transaction.
sourceObjId	ID	Id of the Source Object.
transactionAmount	Decimal	The transaction amount.
transactionDate	Datetime	The transaction date.
transactionISOCurrency	String	The transaction Currency.
transactionNumber	String	The transaction number.
transactionSubType	String	Sub-type of transaction (picklist value).
transactionType	String	Type of transaction.

Response		
Field	Type	Description
results	List <RelatedARTxnResult2>	List of RelatedARTaxResult2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnResult2		
Field	Type	Description
destinationARTransactionId	ID	A/R transaction Id for destination object.
destinationObjId	ID	The id of Destination object.
errorString	String	The Error message.Null value implies success and non-empty string value implies failure.

Apttus_Billing.RelatedARTransactionService.RelatedARTxnResult2		
sourceObjId	ID	The id of Source object.
Status	String	The Status of a transaction will be success or failure.
transactionNumber	String	The transaction number.

Code Sample

```

using System;
using sforce = ConsoleApplication1.Sforce1;
using ARService = ConsoleApplication1.ARTransactionService;
using System.Net;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            String sessionId;
            using (sforce.SoapClient client = new sforce.SoapClient())
            {
                ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12 |
SecurityProtocolType.Tls11;

                //call login service
                sforce.LoginResult result = client.login(null, "[ORG_UserName]", "[ORG_Password]");
                //extract authentication token
                sessionId = result.sessionId;
                //serviceUrl = result.serverUrl;
                Console.WriteLine("Token generated: " + sessionId);
            }

            //call Related AR Transaction SOAP API
            ARService.SessionHeader header = new ARService.SessionHeader();
            header.sessionId = sessionId;

            using (ARService.RelatedARTransactionServicePortTypeClient soapClient = new
ARService.RelatedARTransactionServicePortTypeClient())
            {
                ARService.RelatedARTxnInput2[] inputs = new ARService.RelatedARTxnInput2[1];
                ARService.RelatedARTxnInput2 input2 = new ARService.RelatedARTxnInput2();
                input2.destinationObjId = "[InvoiceID]";
                input2.transactionAmount = 100.00M;
                input2.description = "test description";
                input2.integrationDate = DateTime.Today;
                input2.transactionDate = DateTime.Today;
                input2.transactionNumber = "AR12342";
                input2.transactionType = "Payment";
            }
        }
    }
}

```

```

        inputs[0] = input2;
        ARService.RelatedARTxnResult2[] arResults = new ARService.RelatedARTxnResult2[1];
        soapClient.applyPaymentsToInvoices(header, null, null, null, inputs, out arResults);
        Console.WriteLine("ARResult is: " + arResults[0].destinationARTransactionId);
        Console.ReadLine();
    }
}
}
}

```

Adding a A/R Transaction - Deprecated

addRelatedARTransaction(RelatedARTxnInput input)

This API applies Credit Memo to Invoices as part of the A/R transaction.


Request		
Field	Type	Description
input	RelatedARTxnInput	List of RelatedARTaxInput2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput		
Field	Type	Description
collectionStatus	String	The Collection Status.
description	String	Description for A/R Transaction.
externalSystemStatus	String	The Status of external system.
impactARAMount	Decimal	The A/R impact amount.
integrationDate	Datetime	Date of Integration.
invoiceId	ID	Invoice Id.
reasonCode	String	Reason code for creating A/R transaction.

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput		
relatedTransactionId	String	Related transaction id.
transactionAmount	Decimal	Transaction amount.
transactionDate	Datetime	Transaction date.
transactionEffectiveDate	Datetime	The effective date of transaction.
transactionNumber	String	Transaction number.
transactionType	String	The type of transaction.

Response		
Field	Type	Description
result	RelatedARTxnResult	List of RelatedARTaxResult2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnResult		
Field	Type	Description
errorString	String	Error string showing the error message.
invoiceId	ID	Invoice Id.
relatedARTransactionId	ID	The related A/R transaction Id.
Status	String	Status of the transaction.
transactionNumber	String	Transaction number.

 This API is deprecated. Calling this API will result in an error.

Adding Multiple A/R Transactions - Deprecated

This API invokes RelatedAR Transactions.

`addRelatedARTransactions(List<RelatedARTxnInput> inputs)`

This API is used to accept A/R Transactions in bulk.

Request			
Field	Type	Required?	Description
inputs	RelatedARTxnInput	Yes	List of RelatedARTxnInput2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput		
Field	Type	Description
collectionStatus	String	The Collection Status.
description	String	Description for A/R Transaction.
externalSystemStatus	String	The Status of external system.
impactARAMount	Decimal	The A/R impact amount.
integrationDate	Datetime	Date of Integration.
invoiceId	ID	Invoice Id.
reasonCode	String	Reason code for creating A/R transaction.
relatedTransactionId	String	Related transaction id.

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput		
transactionAmount	Decimal	Transaction amount.
transactionDate	Datetime	Transaction date.
transactionEffectiveDate	Datetime	The effective date of transaction.
transactionNumber	String	Transaction number.
transactionType	String	The type of transaction.

Response		
Field	Type	Description
results	List <RelatedARTxnResult>	List of RelatedARTaxResult2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnResult		
Field	Type	Description
errorString	String	Error string showing the error message.
invoiceId	ID	Invoice Id.
relatedARTransactionId	ID	The related A/R transaction Id.
Status	String	Status of the transaction.
transactionNumber	String	Transaction number.

 This API is deprecated. Calling this API will result in an error.

Apttus Copyright Disclaimer

Copyright © 2018 Apttus Corporation (“Apttus”) and/or its affiliates. All rights reserved.

No part of this document, or any information linked to or referenced herein, may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written consent of Apttus. All information contained herein is subject to change without notice and is not warranted to be error free.

This document may describe certain features and functionality of software that Apttus makes available for use under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not, in any form, or by any means, use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part of the software. Reverse engineering, disassembly, decompilation of, or the creation of derivative work(s) from, the software is strictly prohibited. Additionally, this document may contain descriptions of software modules that are optional and for which you may not have purchased a license. As a result, your specific software solution and/or implementation may differ from those described in this document.

U.S. GOVERNMENT END USERS: Apttus software, including any operating system(s), integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Neither the software nor the documentation were developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Apttus and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Apttus and X-Author are registered trademarks of Apttus and/or its affiliates.

The documentation and/or software may provide links to Web sites and access to content, products, and services from third parties. Apttus is not responsible for the availability of, or any content provided by third parties. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Apttus is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Apttus is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

For additional resources and support, please visit <https://community.apttus.com>.