



CPQ on Salesforce Summer 2019 SOAP API Guide

30 July 2019

Table of Contents

About this Guide	6
What's New	8
About Apttus Configure Price Quote	9
Apttus CPQ Model	9
Getting Started	10
API Supported Packages	10
Document Setup	10
API Standards and Development Platforms	11
Standards	11
Development Platforms	11
Field Types	11
Recommendations	12
API Reference	14
CPQ Web Service	14
Creating a Cart from a Quote	16
Retrieving Categories for a Price List	18
Retrieving Products and List Prices for a Price List	21
Retrieving Products and List Prices for a Price List and Category	25
Retrieving Products and List Prices For a Price List and Search Text	29
Retrieving Products and List Prices For a Price List Category and Search Text	32
Retrieving Option Groups, Options, and List Prices for a Price List Product	36

Modifying Assets (Deprecated)	42
Fetching Asset Line Items	57
Changing Assets	58
Renewing Assets	60
Swapping Assets	61
Getting a list of Products to be swapped with Assets	63
Getting a count of Asset Line Items	65
Terminating Assets	66
Retrieving Asset Line Items	68
Comparing Products	70
Adding a Bundle to a Cart	77
Adding Options to a Bundle	82
Adding Products to a Cart	85
Adding Price Ramps to a Cart (CPQ Web Service)	88
Associating Constraint Rules to a Cart	92
Applying Constraint Rules to a Cart	94
Removing a Bundle from a Cart	95
Removing Multiple Bundles from a Cart	97
Applying Constraint Rules to Deleted Products	98
Retrieving Constraint Rules Results	99
Computing the Net Price for a Bundle	101
Finalizing a Cart's Contents	103
Synchronizing a Cart	104
Abandoning a Cart	106
Updating Quote Terms	107
Price Breakup for a Cart or Specific Line Item	112
Updating Price For A Cart	115
Computing Shipping for Cart Line Items	119
Computing Taxes for Cart Line Items	121
Retrieving Incentives on the Cart	126
Cloning Bundle Line Items on the Cart	132
Cloning Line Items on the Cart	134
Cloning Option Line Items on the Cart	136
Removing Line Items from the Cart	139
Updating Category Hierarchy for a single Category	141
Updating Category Hierarchy for multiple Categories	142
Creating Favorite Configuration	143

BatchUpdate Web Service	145
Quote/Proposal Config Web Service	147
Accepting a Quote	148
Collaboration Structure	149
Adding Products to a Collaboration Request	149
Asset Service	151
Incrementing Assets	151
Scenarios	154
Working with Products in the Shopping Cart	154
Searching Products	155
Configuring and Adding Products	156
Displaying Recommendations	157
Updating Price	157
Viewing Price Break-up Details	158
Adding Price Ramps to a Cart	158
Updating Taxes and Shipping for an Order	163
Creating and Updating Quote and Quote Line Items	174
Creating and Updating Order and Order Line Items	174
Asset Based Ordering	175
Finalizing the Cart	176
Updating Quote Terms to Modify Dates	177
Apttus Copyright Disclaimer	178


```

        }
        ApexPages.addMessage(new ApexPages.Message(ApexPages.
severity.info, 'Option Count: ' + lstProductOptionWrapper.size()));
    }
}
}

```

Modifying Assets (Deprecated)

This method enables you to complete standard asset based ordering actions of renew, terminate, amend, upgrade and increment.

This method creates the cart line item and associates it to the corresponding asset id. Actions can be performed on a single line item or a selected set of asset line items. Actions can also be performed on standalone product or bundle product.

When doing a renewal, you can use any of the standard adjustment types, including *% Uplift* which is used to pass an uplift percentage that is applied to the *base price* of the renewed line item, resulting in an updated *net renewal* price.

Note

CartId, *AssetLineItemId_c*, and *AssetActions* are required parameters, the use of other parameters depends on your business case.

modifyAsset

Parameters		
Name	Type	Description
Apttus_CPQApi.CPQ.ModifyAssetRequestDO	request	This is the request call made by the method.


```

//Fetch asset line item number
    Apttus_Config2__LineItem__c liSO = new
Apttus_Config2__LineItem__c();
    liSO.Apttus_Config2__Comments__c = 'Comments Added my
Modify Asset API Code';
    liSO.Apttus_Config2__PricingStatus__c = 'Complete';
    assetActDO.CustomData = liSO;

    assetActDOList.add(assetActDO);
}
}
//Invoke the modify assets API and specify the cart for which the
API is invoked.
    Apttus_CPQApi.CPQ.ModifyAssetRequestDO modifyRequest = new
Apttus_CPQApi.CPQ.ModifyAssetRequestDO();
    modifyRequest.CartId = cartID;
    modifyRequest.AssetActions = assetActDOList;

    Apttus_CPQApi.CPQ.ModifyAssetResponseDO modifyReponse =
Apttus_CPQApi.CPQWebservice.modifyAsset(modifyRequest);

    List<Decimal> lineNumberList = modifyReponse.LineNumbers;

    List<Apttus_CPQApi.CPQ.AssetActionDO> assetList = modifyReponse.
AssetActions;

    Boolean bIsPending = false;

    Integer iCounter = 0;

    while (iCounter < 1000)
    {
        bIsPending = false;
        for(Apttus_CPQApi.CPQ.AssetActionDO objAsset: assetList)
        {
            if(objAsset.Pending == true)
            {
                bIsPending = true;
                break;
            }
        }
    }
//If asset Action is pending execute the following loop
    if(bIsPending == true)
    {
        modifyReponse = Apttus_CPQApi.CPQWebservice.modifyAsset
(modifyRequest);
        lineNumberList = modifyReponse.LineNumbers;

```

```

        assetList = modifyReponse.AssetActions;
    }
    else
    {
        break;
    }
    iCounter++;
}

bindAssets();
}

//This function enables you to bind updated asset.
public void bindAssets()
{
    List<Apttus_Config2__AssetLineItem__c> aliSOList = [select id,
Apttus_Config2__Description__c,
    Apttus_Config2__Quantity__c, Apttus_Config2__StartDate__c,
Apttus_Config2__EndDate__c from Apttus_Config2__AssetLineItem__c
    where Apttus_Config2__AccountID__c = :accountID];

    lstAssetWrapper = new List<AssetWrapperClass>();

    for(Apttus_Config2__AssetLineItem__c aliSO : aliSOList)
    {
        AssetWrapperClass objAssetWrapperClass = new
AssetWrapperClass ();
        objAssetWrapperClass.AssetId = aliSO.ID;
        objAssetWrapperClass.ProductId = aliSO.
Apttus_Config2__Description__c;
        objAssetWrapperClass.Quantity = aliSO.
Apttus_Config2__Quantity__c;
        objAssetWrapperClass.StartDate = aliSO.
Apttus_Config2__StartDate__c;
        objAssetWrapperClass.EndDate = aliSO.
Apttus_Config2__EndDate__c;

        lstAssetWrapper.Add(objAssetWrapperClass);
    }
}

```

You can increment the quantity or the dates of the asset items using the increment asset action. For example, the user selects an asset item and clicks Increment. You can use the sample below to update the quantity and the end date by five months and bind the updated assets to the account.


```

        ApexPages.addMessage(new ApexPages.Message(ApexPages.
severity.info, 'Please add a bundle to the cart.));
    }
    else
    {
//Selected products are added to the cart. The request object
comprises the id of the cart and the line number in the cart
        for(Decimal line: lineNumber)
        {
            Apttus_CPQApi.CPQ.ComputeNetPriceRequestDO request = new
Apttus_CPQApi.CPQ.ComputeNetPriceRequestDO();
            request.CartId = cartId;
            request.LineNumber = line;

            Apttus_CPQApi.CPQ.ComputeNetPriceResponseDO
priceResponse = Apttus_CPQApi.CPQWebService.computeNetPriceForBundle
(request);

            ApexPages.addMessage(new ApexPages.Message(ApexPages.
severity.info, 'Success: ' + priceResponse.IsSuccess));
        }
    }
}

```

Finalizing a Cart's Contents

This finalizes the cart, synchronizing the cart line items with the quote/proposal.

finalizeCart

Parameters		
Name	Type	Description
request	CPQ.FinalizeCartRequestDO	The request data object.

Request Data Object - CPQ.FinalizeCartRequestDO		
Field	Type	Description
CartId	ID	The Id of the cart.

Response Data Object - CPQ.FinalizeCartResponseDO		
Field	Type	Description
IsSuccess	Boolean	Indicates whether finalizing the cart items was successful.

Code Sample

The sample enables you to finalize a cart with a specific cartID and synchronizes the products (line items) added to the cart with the quote or proposal used to generate the cart. You can create a button finalize cart which the user will click once the products in the cart are adjusted and final. Invoke this API on click of the button.

```

public void finalizeCart()
{
    // create the finalize cart request
    Apttus_CpqApi.CPQ.FinalizeCartRequestDO request = new
Apttus_CpqApi.CPQ.FinalizeCartRequestDO();
    // add request parameters
    request.CartId = CartId;
    // finalize the cart
    Apttus_CpqApi.CPQ.FinalizeCartResponseDO response =
Apttus_CpqApi.CPQWebService.finalizeCart(request);
}

```

Synchronizing a Cart

This is used to sync the shopping cart with the product configuration on the quote record.

SynchronizeCart

The sample below enables you to send an API request for abandon cart along with the cart ID. The cart whose id has been passed will be abandoned when the request is successful. The response message in the sample indicates whether the request was successful. When the user clicks Abandon Cart the abandonCart API will be invoked and the cartID associated with the request is abandoned.

```
public void abandonCart()
{
    Apttus_CPQApi.CPQ.AbandonCartRequestDO request = new
Apttus_CPQApi.CPQ.AbandonCartRequestDO();
    request.CartId = cartId;

    Apttus_CPQApi.CPQ.AbandonCartResponseDO response = Apttus_CPQApi.
CPQWebService.abandonCart(request);
    ApexPages.addMessage(new ApexPages.Message(ApexPages.severity.
info, 'Success: ' + response.IsSuccess));
}
```

Updating Quote Terms

This method enables you to update the expected start date and expected end date of the quote, along with its selling term. This method does the following:

- Updates *Expected Start Date* and *Expected End Date* for the Apttus Quote/Proposal object.
- Retrieves the finalized cart associated with the quote.
- Creates a new cart by checking out the finalized cart.
- Updates Expected Start Date & Expected End Date in the cart object. Note this is for the new *Apttus Product Configuration* cart object.
- Updates Start Date, End Date, and Selling Term on line item terms that have a valid start date, i.e. when the start date is not null.
- Selling Term may be set to null to clear out the existing selling term which triggers a recalculation of the selling term using the start and end dates. Alternatively, if start date and selling term are provided, the end date is automatically calculated.
- Updates the net price and total price for the cart.
- The ID of the new cart is returned to the caller.
- It is the caller's responsibility to finalize the cart.

CPQ on Salesforce Summer 2019 SOAP API Guide

- All services that expire in 30 days.
- All assets that are in the End of Life or End of Sale state by the vendor.
- All services that are no longer active or are cancelled should be displayed.
- All software licenses setup for a site.
- All products under the CPQ product category.

The assets should be filtered and displayed based on asset status, price type, and subscription expiration.

The displayed results of the asset list the product name, ID, asset name, asset ID, SKU ID, quantity, price type, price UOM, purchase date, start and end date, base price, adjustment, net price, asset status, and renewal term.

To view assets

Retrieve asset results that fulfill the criteria using the [Retrieving Asset Line Items](#) API. Using this API, you can fetch the assets.

To modify assets

1. Retrieve asset results that fulfill the criteria using the [Retrieving Asset Line Items](#) API. Using this API, you can fetch the assets.
2. You can renew, amend, increment, or terminate the assets using the code snippets for [Modifying Assets \(Deprecated\)](#).

To create assets from orders

Create assets from an order using the Create Asset Line Items for an Order API. Using this API, you can create assets.

Finalizing the Cart

The customer can finalize the cart in the following scenarios:

- All the required products have been added.
- All requisite prices have been adjusted accordingly.

Invoke the [Finalizing a Cart's Contents](#) to enable the user to finalize the cart.

Updating Quote Terms to Modify Dates

For a finalized quote, a customer can choose to apply add-ons. The term, validity period, and pricing have to be recalculated after the quote is finalized, approved, and a subsequent proposal is generated. Use the [Updating Quote Terms](#) to achieve the scenarios highlighted.

The updateQuoteTerms API behavior is as follows:

- Update the carts in Finalized or Ready for Finalization status. The latest cart from the quote in any of the two statuses are used as basis to updating the quote.
- Create a new version of the cart in case of Finalized cart and supersede the current version of the cart. The status of the new version of the cart will be same as the status of the current version of the cart. If the cart was in Finalized status, the new cart will also be in finalized status and system will update the "Finalized date" on the quote header.
- Auto-Synch cart line items with the Proposal Line Items.

To synchronize cart and update quote

1. Finalize the cart using the [Finalize Cart](#) API.
2. Synchronize the finalized cart with the quote using the [Synchronize](#) API.
3. Update the selling terms of the products associated to the quote using the [Update Quote Terms](#) API.

Apttus Copyright Disclaimer

Copyright © 2019 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>.

DOC ID: CPQSFSUM19APIG20190730