conga

Intelligent Workflow Approvals on Salesforce Spring 2020 SOAP API Guide

Table of Contents

About This Guide

With the Intelligent Workflow Approvals on Salesforce SOAP API Guide, you can find out how Application Programming Interfaces (API) for you to extend the features offered by Apttus. These extensions add more functionality to the features available through configuration on Salesforce.

Topic	Description
What's Covered	This guide is designed to provide APIs provided to work with Intelligent Workflow and Approvals objects.
Primary Audience	Admin users responsible for setting up approvals and users for Apttus Intelligent Workflow Approvals.
IT Environment	Refer to the latest Apttus Intelligent Workflow Approvals on Salesforce Release Notes for information on System Requirements and Supported Platforms.
Updates	For a comprehensive list of updates to this guide for each release, see the What's New topic.
Other Resources	• Intelligent Workflow Approvals Administrator Guide: Refer to this guide for basic admin tasks and end user experience.

This guide describes the APIs provided to work with Intelligent Workflow and Approvals objects.

Before using Intelligent Workflow Approvals, you must be familiar with the following:

- Basic Salesforce administration
- Salesforce and Apttus terms and definitions

What's New

The following table lists changes in the documentation to support each release.

Release	Topic	Description
Spring 2020	Submitting For Approvals With Comments JSON	Added new API - Submit For Approval With Comments API for MAX using JSON parameter to pass comments.
	Submitting For Approvals With Comments	Modified topic. Updated code sample.
Winter 2019	Previewing Adhoc Approvals	Added API info.
	Submit For Approvals Using An Adhoc Approval Specification	Added API info.
	Submit For Approvals With Comments Using An Adhoc Approval Specification	Added API info.
	Delete An Adhoc Approval Step	Added API info.
	Create a Runtime Adhoc Approval Process	Added API info.
	AdhocApprovalProcessDTO	Added DTO info for adhoc approval process API.
	Update a Runtime Adhoc Approval Process	Added API info.

Release	Topic	Description
	Retrieve a Runtime Adhoc Approval Process	Added API info.
	Delete a Runtime Adhoc Approval Process	Added API info.
	Add Comment to a Runtime Adhoc Approval Process	Added API info.
	Add Attachments to a Runtime Adhoc Approval Process	Added API info.
	Delete Attachments to a Runtime Adhoc Approval Process	Added API info.
	Sample Code	Added sample code for adhoc approval process runtime APIs.
Summer 2019	N/A	No new API added.
Spring 2019	N/A	No new API added.
Winter 2018	Adding Comments to Approval Requests	Added API info.
	Approving Approval Requests	Added API info.
	Cancelling Approvals	Added API info.

Release	Topic	Description
	Checking If an Approval is Required	Added API info.
	Checking if User is Authorized to Approve/Reject	Added API info.
	Creating Ad-hoc Approvals	Added API info.
	Escalating Approval Requests	Added API info.
	Previewing Approvals	Added API info.
	Reassigning Approval Requests	Added API info.
	Rejecting Approval Requests	Added API info.
	Retrieving Add Comment Page URL	Added API info.
	Retrieving Approval History	Added API info.
	Retrieving Approval Page URL	Added API info.
	Retrieving Approve or Reject Page URL	Added API info.
	Retrieving Reassign Page URL	Added API info.

Release	Topic	Description
	Submitting an Approval Request	Added API info.
	Taking Ownership of an Approval Request	Added API info.

About Apttus Intelligent Workflow Approvals

Intelligent Workflow Approvals enables you to trigger an approval request for any object and send an email notification to the concerned stakeholders. Using Intelligent Workflow Approvals, you can set an approval request on an object header, line item, or both. Once an approval request is triggered, an approver can approve or reject a request via email or via Salesforce by logging into their org and viewing all the available requests in the My Approval or Home Page.Apttus Quote-to-Cash enhances the visibility and agility of all business processes from initiating a prospective customer, to preparing a quote to managing contracts and renewals with its intelligent workflow. By implementing with your Salesforce1 data, you can improve enterprise-wide collaboration, streamline approvals, and significantly cut down sales cycles.

Apttus provides over a dozen intelligent workflow capabilities that can help you get much more out of standard and custom Salesforce1 objects. You can gain a greater control over your sales process, discounts and exceptions so you can get higher productivity and more wins.Apttus Intelligent Workflow Approvals APIs work with Intelligent Workflow Approvals objects to perform the following:

- Add Comments to Approval Requests
- Approve Approval Requests
- Cancell Approvals
- Check If an Approval is Required
- Check if User is Authorised to Approve or RejectCreate Ad-hoc ApprovalsSubmit for Enable auto re-approvals on Proposal and Proposal Line ItemsApprovals Escalate Approval RequestsPreview ApprovalsReassign Approval RequestsTake Reject Approval RequestsRetrieve Add Comment Page URLRetrieve Approval History Retrieve Approval Page URLRetrieve Approve or Reject Page URL Retrieve Reassign Page URL Ownership of an Approval Request
- Previewing Adhoc Approvals
- Submit For Approvals Using An Adhoc Approval Specification
- Submit For Approvals With Comments Using An Adhoc Approval Specification
- Delete An Adhoc Approval Step
- Create a Runtime Adhoc Approval Process
    - AdhocApprovalProcessDTO
- Update a Runtime Adhoc Approval Process
- Retrieve a Runtime Adhoc Approval Process
- Delete a Runtime Adhoc Approval Process

- Add Comment to a Runtime Adhoc Approval Process
- Add Attachments to a Runtime Adhoc Approval Process
- Delete Attachments to a Runtime Adhoc Approval Process
- Sample Code

Key Terminology

It is important to understand how terms are used when working with Apttus Intelligent Workflow Approvals.

Term	Description
Approval Matrix	Approval matrix determines the assignees based on given criteria. For example, approver authority may change based on the discount being offered on a sales contract. So different levels of approvers can be assigned based on approval authority on a given item.
Approval Process	The process definition that describes approval processes within your organizations. You can define approval processes for Opportunity, Agreement or Term Exceptions objects. You have to define entry criteria, initial submission actions, step groups (steps, step filters) and final actions within an approval process.
Approval Rule (Rule)	Approval rule uniquely identifies an approver based on combinations of various logical conditions per business process and policies. For example, an agreement needs to be sent for approval to CFO if contract value is above $1M OR (geography is emerging markets AND contract value is above $250K).
Assignee Type	The system supports the following: Approval Matrix, Auto Approval, Custom Queue, Custom Role, Custom Rule, Related User, Custom User, Queue, Role, User
Assignment Email Template	Email template used when an item is reassigned to another user. When an item is assigned to a user for approval, he/she always has an option to reassign the item to another user.

<table>
<tr><th>Term</th><th>Description</th></tr>
<tr><td>Auto Approval</td><td>This Assignee Type can be selected for Approval Rule Entry Criteria when you want the item in an approval process to be automatically progressed. For this to work, in an Approval Process step which references that Approval Rule using Auto Approval, the Assigned Approver > Assignee Type must be Subprocess.</td></tr>
<tr><td>Backup Approvers</td><td>When a user is out of office, all approvals can be automatically assigned to backup approvers. The user may choose to reassign in process approval requests to backup approvers as well.</td></tr>
<tr><td>Consolidated Approvals</td><td>If some approval process results in a situation where the same user is required to approve the same object record/line items, these multiple approvals can be consolidated in one approval. In other words, the approver has to approve or reject it only once to approve or reject all of their approval requests for a single approval process.Consider the following approval process with three steps:
<table>
<tr><th>Approval Step</th><th>Approver 1</th><th>Approver 2</th><th>Approver 3</th></tr>
<tr><td>Sales Level 1</td><td>John</td><td>Mary</td><td>Dave</td></tr>
<tr><td>Sales Level 2 (Dependent on 1)</td><td>James</td><td>John</td><td>Suze</td></tr>
<tr><td>Executive Sign-off</td><td>Art</td><td>-</td><td>-</td></tr>
</table>
When the object record is submitted for approval and the approvers are selected based on the criteria in the approval process, the approver John is included in two steps, including one dependent step.Once Mary and Dave have approved their requests for the Sales Level 1 step, John's approval request in the Sales Level 2 step will be Assigned to him, while his approval request in Sales Level 1 is on Hold. This enables him to approve both of his approval requests via a single action.</td></tr>
<tr><td>Custom Assignee</td><td>Custom assignee option allows you to configure custom code to be called to evaluate an assignee. If none of the other options are meeting the assignment requirements, custom code can be developed to evaluate the approver based on given criteria.</td></tr>
</table>

Term	Description
Custom Queue	This assignee type points to a field which contains a queue name in any custom object with a filter to narrow it down to a single row. i.There are two ways to specify a filter: 1. Example 1: Select the row where the 'Region__c' field has a value of 'Asia'. 2. Example 2: Select the row where the 'Region__c' field has a value which equals the value of 'MyRegion__c' field in the associated business object
Custom Role	This points to field which contains a role name in any custom object with filter to narrow down to a single row. For example: Custom_Object__c.Approver_RoleName__c
Custom User	This points to a user lookup reference field in any custom object with filter to narrow down to a single row. The custom object may have filter criteria enclosed in parenthesis like the Region filter in this example: Custom_Object__c(Region__c = 'Asia').Approver_UserId__c
Entry Criteria	You can define certain logical conditions as part of entry criteria for an approval process. If these conditions evaluate true, the approval process is evaluated. This criteria allows you to make sure that all agreements (or opportunities) do not process through given approval process.
Reassign	When an item is assigned to a user for approval, he/she always has an option to reassign the item to another user.
Step	A step is a business action of approval which will be performed by a user to approve or reject the record.
Step Assignee Type	The type of assignee to which the record will be assigned for approval.
User	This assignee type refers to a named user of the system. Use this assignee type when you want to dynamically assign an approval request to a user.

For more information about terms used with Apttus products, see Glossary.

SOAP API Guide Structure

Document Setup

The Apttus CPQ and CM API reference Guide is divided into two sections: *API Reference* and *Scenarios*.

API Reference	The *API reference* section details the APIs that you can use to manipulate Apttus objects through API calls and passing parameters. The API reference section also includes some code samples.
Scenarios	The *Scenarios* section details examples of the APIs you require to complete a specific task such as, adding products and constraint rules to the cart. The scenarios are classified by theme. You can refer to the generic examples of scenarios to identify the calls you can use to achieve your objective.

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 (WDSL) 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.

⚠ A later version of this content will contain sample code.

Field Types

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

The following table lists the APIs that Apttus provides. 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.
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	The **ID** field is an alphanumeric field that acts like 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. 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.
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.

Type	Description
String	The **String** field contains text and may have differing 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.

IWA Web Service

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

You can invoke Apttus IWA APIs belonging to the ApprovalsWebService Class from the following command:

```
Apttus_Approval.ApprovalWebService.<Name of the Function>
where the name of the function is API Name and it parameters.
```

Approvals Web Serice class contains the following APIs:

- Adding Comments to Approval Requests
- Approving Approval Requests
- Cancelling Approvals
- Checking If an Approval is Required
- Checking if User is Authorised to Approve or Reject
- Creating Ad-hoc Approvals
- Enabling auto re-approvals on Proposal and Proposal Line Items
- Escalating Approval Requests
- Previewing Approvals
- Reassigning Approval Requests
- Rejecting Approval Requests
- Retrieving Add Comment Page URL
- Retrieving Approval History
- Retrieving Approval Page URL
- Retrieving Approve or Reject Page URL
- Retrieving Reassign Page URL
- Submitting for Approvals
- Taking Ownership of an Approval Request
- Previewing Adhoc Approvals
- Submit For Approvals Using An Adhoc Approval Specification
- Submit For Approvals With Comments Using An Adhoc Approval Specification
- Delete An Adhoc Approval Step
- Adhoc Approval Process Runtime APIs

Adding Comments to Approval Requests

This API enables you to add comments to approval requests. This API accepts the ID of the approval request and comments as input parameters.

addCommentsToRequest

Request Parameters			
Name	**Type**	**Required?**	**Description**
requestId	ID	Yes	ID of the approval request.
comments	String	Yes	Comments to add to the approval request.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the comments are added to the approval request.

Approving Approval Requests

This API approves the approval request. This API accepts the ID of the approval request and comments as input parameters.

approveRequest

Request Parameters			
Name	**Type**	**Required?**	**Description**
requestId	ID	Yes	ID of the approval request.
comments	String	Yes	Comments to add to the approval request.

Response Parameter		
Name	Type	Description
result	Boolean	Returns true if the approval request is approved successfully.

Cancelling Approvals

This API cancels the entire approval process associated with the context object. Currently, assigned approvers will be notified via email. This API accepts the object type and the object ID of the approval request as input parameters.

cancelApprovals

Request Parameters			
Name	Type	Required?	Description
sObjectId	ID	Yes	ID of the approval context object.
sObjectType	String	Yes	Type of the approval context object.

Response Parameter		
Name	Type	Description
result	Boolean	Returns true if the approval is cancelled successfully.

Checking If an Approval is Required

This API determines if given context object, in its current state, requires approval. This API accepts ID of the context object and type of the context object as request parameters.

isApprovalRequired

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectId	ID	Yes	ID of the business context object.
sObjectType	String	Yes	Type of the context object.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if an approval is required.

checkIfApprovalRequired

This API determines if given list of context objects, in its current state, requires approval. It returns list of true/false values corresponding to the order in which ids were input.

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectIds	List<ID>	Yes	List of IDs of business context objects.

Response Parameter		
Name	**Type**	**Description**
result	List<Boolean>	Returns a list of boolean indicators. The indicator is true if an approval is required.

checkIfApprovalRequired2

This API determines if a context header object (For example: Agreement) or context child object (For example: AgreementLineItems), in its current state, requires approval. It returns a single consolidated list with updated approval status (a01U0000017FRf9IAG__Approval Required) against the corresponding Id.

headerIdStatus & childIdStatus are constructed as follows:
Example: (a01U0000017FRf9IAG__None) - ID and Approval_Status__c value of the object, separated by '__' (double-underscore)

Request Parameters			
Name	**Type**	**Required?**	**Description**
headerIdStatus	String	Yes	Header ID and approval status.
childIdStatusList	List<String>	Yes	List of approval status ID of child objects.
modifiedChildObjectIds	Lsit<ID>	Yes	List of IDs of modified child objects.

Response Parameter		
Name	**Type**	**Description**
result	List<String>	Returns a list of the header ID and child object id approval statuses separated by '__'.

Checking if User is Authorised to Approve or Reject

This API checks if the context user is authorized to approve or reject an approval request. This API accepts the request ID of the approval as an input parameter.

canApproveRejectRequest

Request Parameters			
Name	**Type**	**Required?**	**Description**
requestId	ID	Yes	ID of the approval request.

Response Parameter		
Name	Type	Description
result	Boolean	Returns true if the user is authorized to approve or reject a request.

Creating Ad-hoc Approvals

This API creates an ad-hoc approval step. This API accepts Context Request ID and creates an approval step for it.

createAdhocApproval

Request Parameters			
Name	Type	Required?	Description
contextRequestId	ID	Yes	ID of the context request in relation to which you want to create an approval step.
stepLable	String	Yes	Text label of the step.
assigneeType	String	Yes	Type of the assignee. Assignee can be: • User • Role • Queue • Related User
assigneeID	ID	Yes	ID of the assignee.
submissionComment	String	Yes	Comments for approval submission.
relativeLocationToContext	String	Yes	Relative Location can be: • Before • After • In-parallel
dependsOnContextRequestIdList	List<ID>	Yes	List of request identifiers that the ad-hoc approval depends on.

Request Parameters			
Name	**Type**	**Required?**	**Description**
dependsOnMeContextRequestIdList	List<ID>	Yes	List of request identifiers that depend on this ad-hoc approval.
sendEmail	Boolean	No	Set it as true to send an email request. The default value is true.
notifyOnly	Boolean	No	The default value is false

Response Parameter		
Name	**Type**	**Description**
result	Boolean	The API returns a true value if the ad-hoc approval is created successfully

Enabling auto re-approvals on Proposal and Proposal Line Items

This API takes several parameters and updates the reapproval data in the process instance so that proposal line items can be automatically reapproved when resubmitted after finalizing new items in the cart. Note that this API is available in the base Approvals package and can be used with the optional unmanaged custom package "Apttus Proposal Auto-Reapprovals" available on request with Apttus Release Management. This package contains the complete sample code that can used along with the API in the base package to make auto-reapprovals work with Proposals and Proposal Line Items.

updateApprovalData

Request Parameters			
Name	**Type**	**Required?**	**Description**
instanceId	ID	Yes	The process instance id that is being currently executed in the system.
sObjectType	String	Yes	ThesObjecttypethat is used to identify the object type.
contextObjId	ID	Yes	The ID of the context object.

Request Parameters			
Name	**Type**	**Required?**	**Description**
oldContextObjIds	List<ID>oldContextObjIds	Yes	The list of old approval data context object ids that are to be replaced with the new ones.
newContextObjIds	List<ID> newContextObjIds	Yes	The list of new approval data context object id store place with.

Response Parameter			
Name	**Type**	**Required**	**Description**
Ok	Boolean	Yes	Defines if the operation was successful. Returns *true* if yes, otherwise *false*.

Code Sample

The sample below enables you to execute auto re-approvals over Proposal Line Items by considering the input parameters, such as process instance, sObject type, context object, and lists of old and new context object ids. These IDs represent the Proposal Line Item IDs before and after the cart is finalized. You can use this API in a scenario where a Quote/Proposal is approved but the Approvals Manager decides to change the discount from 10 to 15% on a Proposal Line Item, you can resubmit the approval request for this for re-approval. The system honors the auto-reapproval criteria, using which a request once approved by certain assignees are auto-approved.

See the complete code included in the package to understand how this works, especially CustomProposalLineItemSupport.cls. Note that the code in this package is based on a trigger on the Proposal Line Item object which saves the old ids of the Proposal Line Items before they are deleted as well as the new ids of the Proposal Line Items that get inserted after the cart is finalized. You can extend this concept for other custom child objects to the proposal object by creating a trigger on them and using the same pattern in the sample code to save ids before and after new items are created.

```
/**
*Apttus Approvals Management
*CustomProposalLineItemTrigger
*@2017 Apttus Inc. All rights reserved.
*/


trigger CustomProposalLineItemTrigger on Apttus_Proposal__Proposal_Line_Item__c
(before delete, after insert)
{
    if (Trigger.isBefore && Trigger.isDelete)
    {
       // save map of old line items to attributes before they are deleted
        CustomProposalLineItemSupport.doBeforeDelete(Trigger.old, Trigger.oldMap);
    }
    if (Trigger.isAfter && Trigger.isInsert)
    {
        // update reapproval data with new line items after they are inserted
        CustomProposalLineItemSupport.doAfterInsert(Trigger.new, Trigger.newMap);
    }
}
/**
*Apttus Approvals Management
*CustomProposalLineItemSupport
* @2017 Apttus Inc. All rights reserved.
*/

public with sharing class CustomProposalLineItemSupport extends
 CustomApprovalsConstants
{
    // line item types
    private static final String LINETYPE_PRODUCT = 'Product/Service';
    private static final String LINETYPE_OPTION = 'Option';

    // proposal
    private static final String PROPOSAL_SOBJECT_TYPE =
'Apttus_Proposal__Proposal__c';
    private static ID quoteId = null;

    // associated process instance
    private static Apttus_Approval__ApprovalProcessInstance__c quoteProcessInstance =
null;
```

```
    // map of old line item ids to attribute key
    private static Map<ID,String> oldLineId2KeyMap = new Map<ID,String>();

    // map of new line item attribute key to id
    private static Map<String,ID> newLineKey2IdMap = new Map<String,ID>();

    // map of old line item ids to new line item ids
    private static Map<ID,ID> lineOldId2NewIdMap = new Map<ID,ID>();

    /**
     * Process old ProposalLineItems before they are deleted when a cart is finalized
     *  @param oldLineItems - a list of the old versions of the sObject records
     *  @param oldLineItemsMap - a map of IDs to the old versions of the sObject
records
     */

    public static void doBeforeDelete( List<Apttus_Proposal__Proposal_Line_Item__c>
oldLineItems, Map<ID, Apttus_Proposal__Proposal_Line_Item__c> oldLineItemsMap)
{
    // iterate over line items about to deleted
    for (Integer i=0; i<oldLineItems.size(); i++)
    {
        Apttus_Proposal__Proposal_Line_Item__c oldLineItem = oldLineItems[i];
    // get quote from line item
        quoteId = oldLineItem.Apttus_Proposal__Proposal__c;
    // get line item attributes
        String lineNumber =
String.valueOf(oldLineItem.Apttus_QPConfig__PrimaryLineNumber__c);
        String lineType = oldLineItem.Apttus_QPConfig__LineType__c;
        String productId = null;

        if (lineType == LINETYPE_PRODUCT)
        {
            productId = oldLineItem.Apttus_Proposal__Product__c;
        }
        else if (lineType == LINETYPE_OPTION)
        {
            productId = oldLineItem.Apttus_QPConfig__OptionId__c;

        //productId = oldLineItem.Apttus_QPConfig__ProductOptionId__c;
        }

        String chargeType = oldLineItem.Apttus_QPConfig__ChargeType__c;
```

```
        // create attribute key
            String oldLineKey = lineNumber + ':' + productId + ':' + chargeType;
        // save in map
            oldLineId2KeyMap.put(oldLineItem.Id, oldLineKey);
    }
}
/**
* Process new ProposalLineItems after they are inserted when a cart is finalized
* @param newLineItems - a list of the new versions of the sObject records
* @param newLineItemsMap - a map of IDs to the new versions of the sObject records
*/
    public static void doAfterInsert(List<Apttus_Proposal__Proposal_Line_Item__c>
newLineItems, Map<ID, Apttus_Proposal__Proposal_Line_Item__c> newLineItemsMap)
    {
        // iterate over line items about to deleted
        for (Integer i=0; i<newLineItems.size(); i++)
        {
            Apttus_Proposal__Proposal_Line_Item__c newLineItem = newLineItems[i];

        // get quote from line item
            quoteId = newLineItem.Apttus_Proposal__Proposal__c;

        // get line item attributes
            String lineNumber =
String.valueOf(newLineItem.Apttus_QPConfig__PrimaryLineNumber__c);
            String lineType = newLineItem.Apttus_QPConfig__LineType__c;
            String productId = null;

            if (lineType == LINETYPE_PRODUCT)
            {
                productId = newLineItem.Apttus_Proposal__Product__c;
            }
            else if(lineType == LINETYPE_OPTION)
            {
                productId = newLineItem.Apttus_QPConfig__OptionId__c;
                //productId = newLineItem.Apttus_QPConfig__ProductOptionId__c;
            }
            String chargeType = newLineItem.Apttus_QPConfig__ChargeType__c;

            // create attribute key
                String newLineKey = lineNumber + ':' + productId + ':' + chargeType;
            // save in map
                newLineKey2IdMap.put(newLineKey, newLineItem.Id);
```

```
        }
        // create map of old line item ids to new line item ids
            Set<ID> oldLineItemIds = oldLineId2KeyMap.keySet();
            List<ID> newLineItemIds = new List<ID>();

            for (String oldLineItemId : oldLineItemIds)
            {
                // get attribute key
                String attrKey = oldLineId2KeyMap.get(oldLineItemId);
                // lookup key in new line items map
                String newLineItemId = null;
                if(newLineKey2IdMap.containsKey(attrKey))
                {
                    newLineItemId = newLineKey2IdMap.get(attrKey);
                    newLineItemIds.add(newLineItemId);
                    // associate old key with new one
                    lineOldId2NewIdMap.put(oldLineItemId, newLineItemId);
                }

            }

      // update reapprovals data by calling API in approvals package

        List<ID> oldContextObjIds = new List<ID>(oldLineItemIds);
        List<ID> newContextObjIds = newLineItemIds;

        system.debug(LoggingLevel.INFO,'sObjectType='+PROPOSAL_SOBJECT_TYPE);

        system.debug(LoggingLevel.INFO,'contextObjId='+quoteId);

        system.debug(LoggingLevel.INFO,'oldContextObjIds='+oldContextObjIds);

        system.debug(LoggingLevel.INFO,'newContextObjIds='+newContextObjIds);

        if (!oldContextObjIds.isEmpty() && !newContextObjIds.isEmpty() &&
oldContextObjIds.size() == newContextObjIds.size())
        {
            // get process instance associated with the old quote
            Apttus_Approval__ApprovalProcessInstance__c instanceSO =
getProcessInstance(quoteId);
            system.debug(LoggingLevel.INFO,'instanceSO='+instanceSO);

            // call API to update reapproval data
```

```
            Boolean ok =
Apttus_Approval.ApprovalsWebService.updateApprovalData(instanceSO.Id,
PROPOSAL_SOBJECT_TYPE, quoteId, oldContextObjIds, newContextObjIds);


system.debug(LoggingLevel.INFO,
'Apttus_Approval.ApprovalsWebService.updateApprovalData='+ok);

        }
    }

/**
* Get process instance for the given proposal id
* @param proposalId
* @return process instance object
*/

    private static Apttus_Approval__ApprovalProcessInstance__c getProcessInstance(ID
proposalId)
    {
        List<Apttus_Approval__ApprovalProcessInstance__c> instanceList = [select Id,
Name, LastModifiedDate, LastModifiedById, LastActivityDate, CreatedDate, CreatedById,
Apttus_Approval__Status__c, Apttus_Approval__StartTime__c,
Apttus_Approval__ReassignmentEmailTemplate__c,
Apttus_Approval__PrevProcessInstanceId__c,
Apttus_Approval__NotifyOnlyEmailTemplate__c, Apttus_Approval__InstanceNumber__c,
Apttus_Approval__EscalationEmailTemplate__c, Apttus_Approval__EndTime__c,
Apttus_Approval__Data__c, Apttus_Approval__ConsolidationVersionNumber__c,
Apttus_Approval__CancellationEmailTemplate__c,
Apttus_Approval__BusinessObjectType__c, Apttus_Approval__BusinessObjectLink__c,
Apttus_Approval__BusinessObjectId__c, Apttus_Approval__AssignmentEmailTemplate__c,
Apttus_Approval__ApprovalProcessId__c From
Apttus_Approval__ApprovalProcessInstance__c where
Apttus_Approval__BusinessObjectId__c = :proposalId order by CreatedDate DESC limit 1]
;
        if ( ! nullOrEmpty(instanceList))
        {
            return instanceList[0];
        }

        return null;
    }
/**
```

```
* Checks if the given string value is null or empty.
* @param strValue the string to check
* @return <code>true</code> if the string value is null or empty, <code>false</code>
otherwise
*/
    public static Boolean nullOrEmpty(String strValue)
    {
        // check if null or zero length string
        return (strValue == null || strValue.trim().length() == 0);
    }

/**
* Checks if the given list of objects is null or empty.
* @param objList the list of objects to check
* @return <code>true</code> if the list is null or empty, <code>false</code>
otherwise
*/
    public static Boolean nullOrEmpty(List<Object> objList)
    {
        // check if null or empty
        return (objList == null || objList.isEmpty());
    }
}
```

Escalating Approval Requests

System follows the escalation scheme specified in the current approval process. Approval Request is reassigned accordingly. This API accepts the ID of the approval request as an input parameter.

escalateRequest

Request Parameters			
Name	**Type**	**Required?**	**Description**
requestId	ID	Yes	ID of the approval request.

Response Parameter		
Name	Type	Description
result	Boolean	Returns true if the approval request is escalated.

Previewing Approvals

This API generates required approval requests for the given context object. You must query the approval request table to fetch them. This API accepts the object type and the object ID of the approval request as input parameters.

previewApprovals

Request Parameters			
Name	Type	Required?	Description
sObjectId	ID	Yes	ID of the approval context object.
sObjectType	String	Yes	Type of the approval context object.

Response Parameter		
Name	Type	Description
result	Boolean	Returns true if the API is executed successfully.

Reassigning Approval Requests

This API reassigns the approval request. This API accepts the ID of the approval request, Id of the assignee and comments as input parameters.

reassignRequest

Request Parameters			
Name	**Type**	**Required?**	**Description**
requestId	ID	Yes	ID of the approval request.
toAssigneeId	ID	Yes	ID of the new assignee.
comments	String	Yes	Comments

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the approval request is reassigned.

Rejecting Approval Requests

This API rejects the approval request. This API accepts the ID of the approval request and comments as input parameters.

rejectRequest

Request Parameters			
Name	**Type**	**Required?**	**Description**
requestId	ID	Yes	ID of the approval request.
comments	String	Yes	Comments to add to the approval request.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the approval request is rejected successfully.

Retrieving Add Comment Page URL

This API gets the URL of the add comments page for the given approval request. This API accepts the approval request object as input parameter.

getAddCommentPageUrl

Request Parameters			
Name	**Type**	**Required?**	**Description**
Approval_Request__c requestSO	Object	Yes	The approval request object.

Response Parameter		
Name	**Type**	**Description**
addCommentPageURL	String	The URL of the add comment page.

Retrieving Approval History

This API gets the approval history associated with the approval request. This API accepts the approval request object as an input parameter and returns a list of the approval history objects.

getApprovalHistory

Request Parameters			
Name	**Type**	**Required?**	**Description**
Approval_Request__c requestSO	Object	Yes	The approval request object.

Response Parameter		
Name	**Type**	**Description**
approvalHistorySO	List<Approval_Request__c>	List of approval history objects.

Retrieving Approval Page URL

This API gets the approvals page URL for the given approval request. This API accepts the approval request object as input parameter.

getMyApprovalsPageUrl

Request Parameters			
Name	**Type**	**Required?**	**Description**
Approval_Request__c requestSO	Object	Yes	The approval request object.

Response Parameter		
Name	**Type**	**Description**
myApprovalPageURL	String	The URL of the approval page.

Retrieving Approve or Reject Page URL

This API gets the approve/reject page URL for the given approval request. This API accepts the approval request object as input parameter.

getApproveRejectPageUrl

Request Parameters			
Name	**Type**	**Required?**	**Description**
Approval_Request__c requestSO	Object	Yes	The approval request object.

Response Parameter		
Name	**Type**	**Description**
approveRejectPageURL	String	The URL of the approve/reject page.

Retrieving Reassign Page URL

This API gets the reassign page URL for the given approval request. This API accepts the approval request object as input parameter.

getReassignPageUrl

Request Parameters			
Name	**Type**	**Required?**	**Description**
Approval_Request__c requestSO	Object	Yes	The approval request object.

Response Parameter		
Name	**Type**	**Description**
reassignPageURL	String	The URL of the reassign page.

Submitting for Approvals

With this API, you can submit for approvals. This API accepts the object type and the object ID of the context object as input parameters.

submitForApprovals

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectId	ID	Yes	ID of the approval context object.
sObjectType	String	Yes	Type of the approval context object.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the API is executed successfully.

Submitting For Approvals With Comments

With this API, you can submit an approval context with comments.

submitForApprovalsWithComments

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectId	ID	Yes	ID of the approval context object.
sObjectType	String	Yes	Type of the approval context object.
comments	Object	Yes	The SubmissionComments object.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the API is executed successfully.

SubmissionComments Class

```
/**
 *  Apttus Approvals Management
 *  SubmissionComments
 *
 *  @2010-2019 Apttus Inc. All rights reserved.
 */
global with sharing class SubmissionComments {

    public static final String PROCESS_LEVEL_COMMENTS = 'Process';
    public static final String STEP_LEVEL_COMMENTS = 'Step';

    // comments level
    public String commentsLevel = null;
    // comments count
    public Integer commentsCount = 1;

    // process name
    public String processName = null;
    // process comments label
    public String processCommentLabel = null;
    // process comment mandatory
    public Boolean processCommentMandatory = false;
    // process comments
    public String processComment = null;

    // step comments list
    public List<StepComment> stepCommentList = new List<StepComment>();

    /**
     * Public constructor
     */
    public SubmissionComments() {

    }

    /**
     * Gets process name
     */
    global String getProcessName() {
        return processName;
    }
```

```
    /**
     * Gets process comment label
     */
    global String getProcessCommentLabel() {
        return processCommentLabel;
    }

    /**
     * Gets process comment
     */
    global String getProcessComment() {
        return processComment;
    }

    /**
     * Sets process comment
     * @param comment
     */
    global void setProcessComment(String comment) {
        this.processComment = comment;
    }

    /**
     * Is comment at process level?
     */
    global Boolean isProcessLevelComment() {
        return (PROCESS_LEVEL_COMMENTS == commentsLevel);
    }

    /**
     * Sets process comment to manadatory
     */
    global void setProcessCommentMandatory() {
        this.processCommentMandatory = true;
    }

    /**
     * Is comment at process level mandatory?
     */
    global Boolean isProcessLevelCommentMandatory() {
        return processCommentMandatory;
    }
```

```
/**
 * Is comment at step level?
 */
global Boolean isStepLevelComment() {
    return (STEP_LEVEL_COMMENTS == commentsLevel);
}

/**
 * Get comments count - max of 3 at step-level and 1 at process-level
 */
global Integer getCommentsCount() {
    return commentsCount;
}

/**
 * Gets step comment list
 */
global List<StepComment> getStepCommentList() {
    return stepCommentList;
}

/**
 * Add stepComment to list
 * @param comment
 */
global void addStepComment(StepComment comment) {
    stepCommentList.add(comment);
}

/**
 * Creates the json representation of the submission comments
 * @return the json string
 */
global String toJSON() {
    // get the json representation
    return System.JSON.serializePretty(this);

}

/**
 * Parses the submission comments object from the given JSON string
 * @param jsonString the json string to parse
 * @return the submission comments data object
```

```
     */
    global static SubmissionComments parse(String jsonString) {
        // load the class (salesforce class loading bug)
        System.Type wrapperType =
System.Type.forName(SystemUtil.getFQClassName('SubmissionComments'));
        // deserialize the object
        return (SubmissionComments) System.Json.deserialize(jsonString,
SubmissionComments.class);

    }

    /**
     * Inner class to hold step level comments
     */
    global class StepComment {

        // step name
        public String stepName = null;
        // step comment label
        public String stepCommentLabel = null;
        // step comments
        public String stepComment = null;

        /**
         * Public constructor
         */
        public StepComment() {

        }

        /**
         * Gets step name
         */
        global String getStepName() {
            return stepName;
        }

        /**
         * Gets step comment label
         */
        global String getStepCommentLabel() {
            return stepCommentLabel;
        }
```

```
        /**
         * Gets step comment
         */
        global String getStepComment() {
            return stepComment;
        }

        /**
         * Set step comment
         * @param comment
         */
        global void setStepComment(String comment) {
            this.stepComment = comment;
        }
    }

}
```

Submitting For Approvals With Comments JSON

With this API, you can submit an approval context with comments JSON.

submitForApprovalsWithComments

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectId	ID	Yes	ID of the approval context object.
sObjectType	String	Yes	Type of the approval context object.
commentsJSON	String	Yes	A JSON representation of the SubmissionComments object.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the API is executed successfully.

Code Sample

The sample below enables you to set process and step level comments.

```
// create process level submission comments JSON structure
SubmissionComments commentsInfo = new SubmissionComments();
commentsInfo.commentsLevel = SubmissionComments.PROCESS_LEVEL_COMMENTS;
commentsInfo.commentsCount = 1;
commentsInfo.processName = 'ProcessName';
commentsInfo.processCommentLabel = 'ProcessLabel';
commentsInfo.processCommentMandatory = false;
commentsInfo.processComment = 'Here is my comment';

String submitCommentsJSON = commentsInfo.toJSON();
/* submitCommentsJSON =
{
  "stepCommentList" : [],
    "processName" : "ProcessName",
    "processCommentMandatory" : false,
    "processCommentLabel" : "ProcessLabel",
    "processComment" : "Here is my comment",
    "commentsLevel" : "Process",
    "commentsCount" : 1
}
*/
//  get JSON representation for submission comments
String submitCommentsJSON = commentsInfo.toJSON();
// submit with comments
ApprovalsWebService.submitForApprovalsWithCommentsJSON(SObjectConstants.SOBJECT_TYPE_
AGREEMENT, 'a013l00000sR4iPAAS', submitCommentsJSON);



// create step level submission comments JSON structure
SubmissionComments commentsInfo = new SubmissionComments();
commentsInfo.commentsLevel = SubmissionComments.STEP_LEVEL_COMMENTS;
commentsInfo.commentsCount = 3;
// create step comment
SubmissionComments.StepComment stepComment1 = new SubmissionComments.StepComment();
stepComment1.stepName = 'User Assignee';
stepComment1.stepCommentLabel = 'GS_Business_Justification';
stepComment1.stepComment = 'This is comment 1';
commentsInfo.addStepComment(stepComment1);
// create step comment
SubmissionComments.StepComment stepComment2 = new SubmissionComments.StepComment();
stepComment2.stepName = 'Queue Assignee';
```

```
stepComment2.stepCommentLabel = 'GS_Deal_Strategy';
stepComment2.stepComment = 'This is comment 2';
commentsInfo.addStepComment(stepComment2);
// create step comment
SubmissionComments.StepComment stepComment3 = new SubmissionComments.StepComment();
stepComment3.stepName = 'Role Assignee';
stepComment3.stepCommentLabel = 'GS_Supporting_Information';
stepComment3.stepComment = 'This is comment 3';
commentsInfo.addStepComment(stepComment3);

/* submitCommentsJSON =
{
  "stepCommentList" : [ {
    "stepName" : "User Assignee",
    "stepCommentLabel" : "GS_Business_Justification",
    "stepComment" : "This is comment 1"
  }, {
    "stepName" : "Queue Assignee",
    "stepCommentLabel" : "GS_Deal_Strategy",
    "stepComment" : "This is comment 2"
  }, {
    "stepName" : "Role Assignee",
    "stepCommentLabel" : "GS_Supporting_Information",
    "stepComment" : "This is comment 3"
  } ],
  "processName" : null,
  "processCommentMandatory" : false,
  "processCommentLabel" : null,
  "processComment" : null,
  "commentsLevel" : "Step",
  "commentsCount" : 3
}
*/
//  get JSON representation for submission comments
String submitCommentsJSON = commentsInfo.toJSON();
// submit with comments
ApprovalsWebService.submitForApprovalsWithCommentsJSON(SObjectConstants.SOBJECT_TYPE_
AGREEMENT, 'a013l00000sR4iPAAS', submitCommentsJSON);
```

Taking Ownership of an Approval Request

This API takes ownership of the given approval request. This API accepts process definition ID, list of request IDs, user ID and ID of the context object as request parameters.

takeOwnership

Request Parameters			
Name	**Type**	**Required?**	**Description**
processDefnId	ID	Yes	ID of the process definition.
requestIdList	Set<ID>	Yes	Set of request identifiers.
userId	ID	Yes	ID of the requesting owner.
ctxObjectId	ID	Yes	ID of the business context object.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	The API returns a true value upon successful execution of the API.

Previewing Adhoc Approvals

This API enables you to preview approvals for a context object using the adhoc approval specification for that object. This API accepts the context object type and id as input parameters and returns true if the preview is successful.

previewAdhocApprovals

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectId	ID	Yes	ID of the approval context object.

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectType	String	Yes	Type of the approval context object.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the preview is successful.

Submit For Approvals Using An Adhoc Approval Specification

This API enables you to submit approvals for a context object using the adhoc approval specification for that object. This API accepts the context object type and id as input parameters and returns true if the submit is successful.

submitForAdhocApprovals

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectId	ID	Yes	ID of the business object.
sObjectType	String	Yes	Type of business object.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the approval request is submitted successfully.

Submit For Approvals With Comments Using An Adhoc Approval Specification

This API enables you to submit approvals for a context object with comments using the adhoc approval specification for that object. This API accepts the context object type and id, along with the submission comments structure as input parameters and returns true if the submit is successful.

> ⚠ The SubmissionComments structure is same as from previous API docs.

submitForAdhocApprovalsWithComments

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectId	ID	Yes	ID of the approval context object.
sObjectType	String	Yes	Type the approval context object.
comments	SubmissionComments	Yes	The comment to add.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the approval request is submitted successfully.

Delete An Adhoc Approval Step

This API enables you to delete an approval step which was added through the Add Adhoc Step process from the MyApprovals tab. This API accepts the id of the request to be removed as input parameter and returns true if the step is successfully deleted.

deleteAdhocApproval

Request Parameters			
Name	**Type**	**Required?**	**Description**
requestId	ID	Yes	ID of the request to be removed/deleted.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the approval step is deleted successfully.

Adhoc Approval Process Runtime APIs

Create a Runtime Adhoc Approval Process

This API enables you to create a runtime adhoc approval process that can be used to submit for adhoc approvals. This API accepts an approval process structure as input parameter and returns the same structure with record IDs populated. For details on the input structure, refer to AdhocApprovalProcessDTO.

createRuntimeAdhocApprovalProcess

Request Parameters			
Name	**Type**	**Required?**	**Description**
adhocApproverProcessDTO	adhocApproverProcessDTO	Yes	The adhoc approver process structure to create.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	The runtime adhoc approval process structure for the created process.

AdhocApprovalProcessDTO

```
/**
 *  Apttus Approvals Management
 *  AdhocApprovalProcessDTO
 *
 *  @2019 Apttus Inc. All rights reserved.
 */
global with sharing class AdhocApprovalProcessDTO {

    // adhoc approval process
    private AdhocApprovalProcess__c adhocApprovalProcessSO = null;

    // adhoc approval groups
    private List<AdhocApprovalGroupDTO> adhocApprovalGroups = new
 List<AdhocApprovalGroupDTO>();


    /**
     * Public constructor
     */
    global AdhocApprovalProcessDTO() {

    }

    /**
     * Class Constructor specifying initial values
     * @param adhocApprovalGroupSO the adhoc approval group
     */
    global AdhocApprovalProcessDTO(AdhocApprovalProcess__c
adhocApprovalProcessSO) {
        this();
        setAdhocApprovalProcessSO(adhocApprovalProcessSO);

    }

    /**
     * Gets adhoc approval process
     * @return adhoc approval process sobject
     */
    global AdhocApprovalProcess__c getAdhocApprovalProcessSO() {
        return adhocApprovalProcessSO;
```

```
39
40      }
41
42      /**
43       * Sets adhoc approval process
44       * @param adhocApprovalProcessSO the adhoc approval process
45       */
46      global void setAdhocApprovalProcessSO(AdhocApprovalProcess__c
    adhocApprovalProcessSO) {
47          this.adhocApprovalProcessSO = adhocApprovalProcessSO;
48
49      }
50
51      /**
52       * Determines if a process has one or more approval groups
53       * @return <code>true</code> approval groups exist, <code>false</code>
    otherwise
54       */
55      global Boolean hasAdhocApprovalGroups() {
56          return adhocApprovalGroups != null && adhocApprovalGroups.size() >
    0;
57
58      }
59
60      /**
61       * Gets list of adhoc approval groups
62       * @return list of adhoc approval groups
63       */
64      global List<AdhocApprovalGroupDTO> getAdhocApprovalGroups() {
65          return adhocApprovalGroups;
66
67      }
68
69      /**
70       * Add an adhoc approval group
71       * @param adhocApprovalGroup the adhoc approval group
72       */
73      global void addAdhocApprovalGroup(AdhocApprovalGroupDTO
    adhocApprovalGroup) {
74          this.adhocApprovalGroups.add(adhocApprovalGroup);
75
76      }
77
```

```
78      /**
79       * Add a list of adhoc approval group
80       * @param adhocApprovalGroups list of adhoc approval group
81       */
82      global void addAdhocApprovalGroups(List<AdhocApprovalGroupDTO>
    adhocApprovalGroups) {
83          this.adhocApprovalGroups.addAll(adhocApprovalGroups);
84
85      }
86
87      /**
88       * Creates the json representation of this object
89       * @return the json string
90       */
91      public String toJSON() {
92          // get the json representation
93          return System.JSON.serializePretty(this);
94
95      }
96
97
98      /**
99       * Inner class to hold adhoc approval groups
100      */
101     global class AdhocApprovalGroupDTO {
102
103         // adhoc approval group
104         public AdhocApprovalGroup__c adhocApprovalGroupSO = null;
105
106         // adhoc approvers
107         private List<AdhocApproverDTO> adhocApprovers = new
     List<AdhocApproverDTO>();
108
109         /**
110          * Class constructor
111          */
112         private AdhocApprovalGroupDTO() {
113
114         }
115
116         /**
117          * Class Constructor specifying initial values
118          * @param adhocApprovalGroupSO the adhoc approval group
119          */
```

```
120         global AdhocApprovalGroupDTO(AdhocApprovalGroup__c
    adhocApprovalGroupSO) {
121             this();
122             setAdhocApprovalGroupSO(adhocApprovalGroupSO);
123 
124         }
125 
126         /**
127          * Gets adhoc approval group
128          * @return adhoc approval group sobject
129          */
130         global AdhocApprovalGroup__c getAdhocApprovalGroupSO() {
131             return adhocApprovalGroupSO;
132 
133         }
134 
135         /**
136          * Set adhoc approval group
137          * @param adhocApprovalGroup the adhoc approval group
138          */
139         global void setAdhocApprovalGroupSO(AdhocApprovalGroup__c
    adhocApprovalGroupSO) {
140             this.adhocApprovalGroupSO = adhocApprovalGroupSO;
141 
142         }
143 
144         /**
145          * Determines if an approval group has one or more approvers
146          * @return <code>true</code> approval groups exist, <code>false</
    code> otherwise
147          */
148         global Boolean hasAdhocApprovers() {
149             return adhocApprovers != null && adhocApprovers.size() > 0;
150 
151         }
152 
153         /**
154          * Gets list of adhoc approvers
155          * @return list of adhoc approvers
156          */
157         global List<AdhocApproverDTO> getAdhocApprovers() {
158             return adhocApprovers;
159 
```

```
        }

        /**
         * Add an adhoc approver
         * @param adhocApprover the adhoc approver
         */
        global void addAdhocApprover(AdhocApproverDTO adhocApprover) {
            this.adhocApprovers.add(adhocApprover);

        }

        /**
         * Add a list of adhoc approvers
         * @param adhocApprovers list of adhoc approvers
         */
        global void addAdhocApprovers(List<AdhocApproverDTO>
adhocApprovers) {
            this.adhocApprovers.addAll(adhocApprovers);

        }

    }


    /**
     * Inner class to hold adhoc approver
     */
    global class AdhocApproverDTO {

        // adhoc approver
        public AdhocApprover__c adhocApproverSO = null;

        /**
         * Class constructor
         */
        private AdhocApproverDTO() {

        }

        /**
         * Class Constructor specifying initial values
         * @param adhocApproverSO the adhoc approver
         */
        global AdhocApproverDTO(AdhocApprover__c adhocApproverSO) {
            this();
```

```
204             setAdhocApproverSO(adhocApproverSO);
205 
206         }
207 
208         /**
209          * Get adhoc approver
210          * @return the adhoc approver sobject
211          */
212         global AdhocApprover__c getAdhocApproverSO() {
213             return adhocApproverSO;
214 
215         }
216 
217         /**
218          * Set adhoc approver
219          * @param adhocApprover the adhoc approver
220          */
221         global void setAdhocApproverSO(AdhocApprover__c adhocApprover) {
222             this.adhocApproverSO = adhocApprover;
223 
224         }
225 
226     }
227 
228 }
```

Update a Runtime Adhoc Approval Process

This API enables you to update a runtime adhoc approval process that can be used to submit for adhoc approvals. This API accepts an approval process structure as input parameter and returns the same structure with record IDs populated. For details on the input structure, refer to AdhocApprovalProcessDTO.

updateRuntimeAdhocApprovalProcess

Request Parameters			
Name	**Type**	**Required?**	**Description**
adhocApproverProcessDTO	adhocApproverProcessDTO	Yes	The adhoc approver process structure to update.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	The runtime adhoc approval process structure for the updated process.

Retrieve a Runtime Adhoc Approval Process

This API enables you to retrieve a runtime adhoc approval process. This API accepts the context object type and id as input parameters and returns the adhoc process definition.

GetRuntimeAdhocApprovalProcess

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectId	ID	Yes	Id of the approval context object.
sObjectType	Type	Yes	Type of the approval context object.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	The runtime adhoc approval process structure.

Code Sample

```
1  /**
2     * Retrieve a runtime adhoc approval process
3     * @param sObjectType the approval context sobject type
4     * @param sObjectId the approval context sobject identifier
5     * @return the runtime adhoc approval process structure
6     */
7    WebService static AdhocApprovalProcessDTO
GetRuntimeAdhocApprovalProcess(String sObjectType, ID sObjectId);
```

Delete a Runtime Adhoc Approval Process

This API enables you to delete a runtime adhoc approval process. This API accepts the context object type and id as input parameters and returns true if the adhoc process definition is successfully deleted.

DeleteRuntimeAdhocApprovalProcess

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectId	ID	Yes	Id of the approval context object.
sObjectType	Type	Yes	Type of the approval context object.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the approval request is deleted successfully.

Code Sample

```
 /**
     * Delete a runtime adhoc approval process
     * @param sObjectType the approval context sobject type
     * @param sObjectId the approval context sobject identifier
     * @return <code>true</code> if the process is deleted, <code>false</code> otherwise
     */
    WebService static Boolean DeleteRuntimeAdhocApprovalProcess(String sObjectType, ID sObjectId);
```

Add Comment to a Runtime Adhoc Approval Process

This API enables you to add a comment to a runtime adhoc approval process. This API accepts the context object type, id, and comment as input parameters and returns true if the comments are successfully added.

AddCommentToRuntimeAdhocApprovalProcess

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectId	ID	Yes	Id of the approval context object.
sObjectType	Type	Yes	Type of the approval context object.
sComment	String	Yes	The comment to add.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the comment is added successfully.

Code Sample

```
/**
     * Add comment to a runtime adhoc approval process
     * @param sObjectType the approval context sobject type
     * @param sObjectId the approval context sobject identifier
     * @param sComment the comment to add
     * @return <code>true</code> if the comment was added, <code>false</code> otherwise
     */
    WebService static Boolean AddCommentToRuntimeAdhocApprovalProcess(String sObjectType, ID sObjectId, String sComment);
```

Add Attachments to a Runtime Adhoc Approval Process

This API enables you to add one or more attachments to a runtime adhoc approval process. This API accepts the context object type, id, and list of attachment IDs to add as input parameters and returns true if the attachments are successfully added.

AddAttachmentsToRuntimeAdhocApprovalProcess

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectId	ID	Yes	Id of the approval context object.
sObjectType	Type	Yes	Type of the approval context object.
attachmentIds	List	Yes	List of attachment Id's to add.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the attachments are added successfully.

Code Sample

```
/**
     * Add attachments to a runtime adhoc approval process
     * @param sObjectType the approval context sobject type
     * @param sObjectId the approval context sobject identifier
     * @param attachmentIds the list of attachment ids to add
     * @return <code>true</code> if the attachments were added,
<code>false</code> otherwise
     */
    WebService static Boolean
AddAttachmentsToRuntimeAdhocApprovalProcess(String sObjectType, ID
sObjectId, List<ID> attachmentIds);
```

Delete Attachments to a Runtime Adhoc Approval Process

This API enables you to delete one or more attachments from a runtime adhoc approval process. This API accepts the context object type, id, and list of attachment IDs to delete as input parameters and returns true if the attachments were successfully deleted.

DeleteAttachmentsFromRuntimeAdhocApprovalProcess

Request Parameters			
Name	**Type**	**Required?**	**Description**
sObjectId	ID	Yes	Id of the approval context object.
sObjectType	Type	Yes	Type of the approval context object.
attachmentIds	List	Yes	List of attachment Id's to remove.

Response Parameter		
Name	**Type**	**Description**
result	Boolean	Returns true if the attachments are removed successfully.

Code Sample

```
/**
     * Delete attachments from a runtime adhoc approval process
     * @param sObjectType the approval context sobject type
     * @param sObjectId the approval context sobject identifier
     * @param attachmentIds the list of attachment ids to remove
     * @return <code>true</code> if the attachments were removed,
<code>false</code> otherwise
     */
    WebService static Boolean
DeleteAttachmentsFromRuntimeAdhocApprovalProcess(String sObjectType, ID
sObjectId, List<ID> attachmentIds);
```

Sample Code

This sample code shows how you can use the following 7 APIs to create, read, update, and retrieve adhoc approval runtime process info and be able to add comments and attachments to it.

```
/**
 *  Apttus Approvals Management
 *  ApprovalsWebServiceTestSample
 *
 *  @2019 Apttus Inc. All rights reserved.
 */
public with sharing class AdhocApprovalsWebServiceTestSample {

    /**
     * Class Constructor
     */
    public AdhocApprovalsWebServiceTestSample() {

    }

    // test webservice API calls by running as anonymous apex from developer console
    //AdhocApprovalsWebServiceTestSample.TestCreateRuntimeAdhocApprovalProcess('Apttus__APTS_Agreement__c','a07q000000EZfki');
    //AdhocApprovalsWebServiceTestSample.TestUpdateRuntimeAdhocApprovalProcess('Apttus__APTS_Agreement__c','a07q000000EZfki');
    //AdhocApprovalsWebServiceTestSample.TestGetRuntimeAdhocApprovalProcess('Apttus__APTS_Agreement__c','a07q000000EZfki');
    //AdhocApprovalsWebServiceTestSample.TestAddCommentToRuntimeAdhocApprovalProcess('Apttus__APTS_Agreement__c','a07q000000EZfki');
    //AdhocApprovalsWebServiceTestSample.TestAddAttachmentsToRuntimeAdhocApprovalProcess('Apttus__APTS_Agreement__c','a07q000000EZfki');
    //AdhocApprovalsWebServiceTestSample.TestDeleteAttachmentsFromRuntimeAdhocApprovalProcess('Apttus__APTS_Agreement__c','a07q000000EZfki');
    //AdhocApprovalsWebServiceTestSample.TestDeleteRuntimeAdhocApprovalProcess('Apttus__APTS_Agreement__c','a07q000000EZfki');

    /**
     * Create a runtime adhoc approval process
     * @param sObjectType the approval context sobject type
     * @param sObjectId the approval context sobject identifier
```

```
29      * @return the runtime adhoc approval process structure for the
    created process
30      */
31     public static void TestCreateRuntimeAdhocApprovalProcess(String
    sObjectType, ID sObjectId) {
32
33         // create insert structure
34         Apttus_Approval.AdhocApprovalProcessDTO adhocProcessDTO = new
     Apttus_Approval.AdhocApprovalProcessDTO();
35
36         // get users to add as approvers
37         List<User> users = [SELECT Id,Name
38                             FROM User
39                             WHERE UserType != 'AutomatedProcess'
40                             LIMIT 2];
41         User userSO1 = users[0];
42         User userSO2 = users[1];
43
44         // add process
45         Apttus_Approval__AdhocApprovalProcess__c adhocProcessSO = new
     Apttus_Approval__AdhocApprovalProcess__c(
46             Id = null,
47             Apttus_Approval__AttachmentIds__c = null,
48             Apttus_Approval__BusinessObjectId__c = sObjectId,
49             Apttus_Approval__BusinessObjectType__c = sObjectType,
50             Apttus_Approval__DisplayFields__c = null,
51             Apttus_Approval__DisplayHeaderFields__c = null,
52             Apttus_Approval__ProcessComments__c = null
53         );
54         adhocProcessDTO.setAdhocApprovalProcessSO(adhocProcessSO);
55         // create group 1
56         Apttus_Approval__AdhocApprovalGroup__c adhocGroupSO1 = new
     Apttus_Approval__AdhocApprovalGroup__c(
57             Apttus_Approval__AdhocApprovalProcessId__c = null,
58             Apttus_Approval__DependsOn__c = null,
59             Apttus_Approval__GroupName__c = 'Group 1',
60             Apttus_Approval__GroupSequence__c = 1
61         );
62         // add group 1 to process
```

```
63          Apttus_Approval.AdhocApprovalProcessDTO.AdhocApprovalGroupDTO
adhocGroupDTO1 = new

Apttus_Approval.AdhocApprovalProcessDTO.AdhocApprovalGroupDTO(adhocGroupSO
1);
64          adhocProcessDTO.addAdhocApprovalGroup(adhocGroupDTO1);
65          // create group 1 approver 1
66          Apttus_Approval__AdhocApprover__c adhocApproverSO1 = new
 Apttus_Approval__AdhocApprover__c(
67              Apttus_Approval__AdhocApprovalGroupId__c = null,
68              Apttus_Approval__ApproverSequence__c = 1,
69              Apttus_Approval__DependsOn__c = null,
70              Apttus_Approval__AssigneeId__c = userSO1.Id,
71              Apttus_Approval__AssigneeType__c = 'User',
72              Apttus_Approval__AssigneeValue__c = userSO1.Name,
73              Apttus_Approval__AutoReapprovalEnabled__c = true,
74              Apttus_Approval__IsReviewer__c = false
75          );
76          // add group 1 approver 1
77          Apttus_Approval.AdhocApprovalProcessDTO.AdhocApproverDTO
adhocApproverDTO1 = new

Apttus_Approval.AdhocApprovalProcessDTO.AdhocApproverDTO(adhocApproverSO1)
;
78          adhocGroupDTO1.addAdhocApprover(adhocApproverDTO1);
79          // create group 1 approver 2
80          Apttus_Approval__AdhocApprover__c adhocApproverSO2 = new
 Apttus_Approval__AdhocApprover__c(
81              Apttus_Approval__AdhocApprovalGroupId__c = null,
82              Apttus_Approval__ApproverSequence__c = 2,
83              Apttus_Approval__DependsOn__c = null,
84              Apttus_Approval__AssigneeId__c = userSO2.Id,
85              Apttus_Approval__AssigneeType__c = 'User',
86              Apttus_Approval__AssigneeValue__c = userSO2.Name,
87              Apttus_Approval__AutoReapprovalEnabled__c = true,
88              Apttus_Approval__IsReviewer__c = false
89          );
90          // add group 1 approver 2
91          Apttus_Approval.AdhocApprovalProcessDTO.AdhocApproverDTO
adhocApproverDTO2 = new

Apttus_Approval.AdhocApprovalProcessDTO.AdhocApproverDTO(adhocApproverSO2)
;
92          adhocGroupDTO1.addAdhocApprover(adhocApproverDTO2);
93          // create group 2
```

```
94          Apttus_Approval__AdhocApprovalGroup__c adhocGroupSO2 = new
 Apttus_Approval__AdhocApprovalGroup__c(
95              Apttus_Approval__AdhocApprovalProcessId__c = null,
96              Apttus_Approval__DependsOn__c = null,
97              Apttus_Approval__GroupName__c = 'Group 2',
98              Apttus_Approval__GroupSequence__c = 2
99          );
100         // add group 2 to process
101         Apttus_Approval.AdhocApprovalProcessDTO.AdhocApprovalGroupDTO
    adhocGroupDTO2 = new

    Apttus_Approval.AdhocApprovalProcessDTO.AdhocApprovalGroupDTO(adhocGroupSO
    2);
102         adhocProcessDTO.addAdhocApprovalGroup(adhocGroupDTO2);
103         // create group 2 approver 1
104         adhocApproverSO1 = new Apttus_Approval__AdhocApprover__c(
105             Apttus_Approval__AdhocApprovalGroupId__c = null,
106             Apttus_Approval__ApproverSequence__c = 1,
107             Apttus_Approval__DependsOn__c = null,
108             Apttus_Approval__AssigneeId__c = userSO1.Id,
109             Apttus_Approval__AssigneeType__c = 'User',
110             Apttus_Approval__AssigneeValue__c = userSO1.Name,
111             Apttus_Approval__AutoReapprovalEnabled__c = true,
112             Apttus_Approval__IsReviewer__c = false
113         );
114         // add group 2 approver 1
115         adhocApproverDTO1 = new

    Apttus_Approval.AdhocApprovalProcessDTO.AdhocApproverDTO(adhocApproverSO1)
    ;
116         adhocGroupDTO2.addAdhocApprover(adhocApproverDTO1);
117         // create group 2 approver 2
118         adhocApproverSO2 = new Apttus_Approval__AdhocApprover__c(
119             Apttus_Approval__AdhocApprovalGroupId__c = null,
120             Apttus_Approval__ApproverSequence__c = 2,
121             Apttus_Approval__DependsOn__c = null,
122             Apttus_Approval__AssigneeId__c = userSO2.Id,
123             Apttus_Approval__AssigneeType__c = 'User',
124             Apttus_Approval__AssigneeValue__c = userSO2.Name,
125             Apttus_Approval__AutoReapprovalEnabled__c = true,
126             Apttus_Approval__IsReviewer__c = false
127         );
128         // add group 2 approver 2
```

```
129          adhocApproverDTO2 = new
Apttus_Approval.AdhocApprovalProcessDTO.AdhocApproverDTO(adhocApproverSO2)
;
130         adhocGroupDTO2.addAdhocApprover(adhocApproverDTO2);
131
132         // call API to create process
133         Apttus_Approval.AdhocApprovalProcessDTO adhocProcessDTOResult =
Apttus_Approval.ApprovalsWebService.CreateRuntimeAdhocApprovalProcess(adho
cProcessDTO);
134         system.debug('adhocProcessDTOResult='+adhocProcessDTOResult);
135
136     }
137
138     /**
139      * Update a runtime adhoc approval process
140      * @param sObjectType the approval context sobject type
141      * @param sObjectId the approval context sobject identifier
142      * @return the runtime adhoc approval process structure for the edited
process
143      */
144     public static void TestUpdateRuntimeAdhocApprovalProcess(String
sObjectType, ID sObjectId) {
145
146         // create update structures
147         Apttus_Approval.AdhocApprovalProcessDTO adhocProcessDTO = new
 Apttus_Approval.AdhocApprovalProcessDTO();
148         Apttus_Approval.AdhocApprovalProcessDTO.AdhocApprovalGroupDTO
adhocGroupDTO = null;
149         Apttus_Approval.AdhocApprovalProcessDTO.AdhocApproverDTO
adhocApproverDTO = null;
150
151         // get users to add as approvers
152         List<User> users = [SELECT Id,Name
153                             FROM User
154                             WHERE UserType != 'AutomatedProcess'
155                             LIMIT 5];
156         User userSO1 = users[0];
157         User userSO2 = users[1];
158         User userSO3 = users[2];
159         User userSO4 = users[3];
160         User userSO5 = users[4];
161
162         // get process
```

```
163         Apttus_Approval__AdhocApprovalProcess__c processSO = [SELECT
    Id,Name,
164
    Apttus_Approval__BusinessObjectType__c,
165
    Apttus_Approval__BusinessObjectId__c
166                                             FROM
    Apttus_Approval__AdhocApprovalProcess__c
167                                             WHERE
    Apttus_Approval__BusinessObjectType__c = :sObjectType AND
168
    Apttus_Approval__BusinessObjectId__c = :sObjectId
169                                             LIMIT 1];
170         system.assert(processSO != null);
171
172         // update process
173         Apttus_Approval__AdhocApprovalProcess__c adhocProcessSO = new
     Apttus_Approval__AdhocApprovalProcess__c(
174             Id = processSO.Id,
175             Apttus_Approval__AttachmentIds__c = null,
176             Apttus_Approval__BusinessObjectId__c =
    processSO.Apttus_Approval__BusinessObjectId__c,
177             Apttus_Approval__BusinessObjectType__c =
    processSO.Apttus_Approval__BusinessObjectType__c,
178             Apttus_Approval__DisplayFields__c = null,
179             Apttus_Approval__DisplayHeaderFields__c =
    'Apttus__Total_Contract_Value__c',
180             Apttus_Approval__ProcessComments__c = 'new comment added'
181         );
182         adhocProcessDTO.setAdhocApprovalProcessSO(adhocProcessSO);
183
184         // get groups and approvers
185         List<Apttus_Approval__AdhocApprovalGroup__c> groups = [SELECT Id,
186                                                 Name,
187
    Apttus_Approval__AdhocApprovalProcessId__c,
188
    Apttus_Approval__AdhocApprovalProcessId__r.Apttus_Approval__BusinessObject
    Id__c,
189
    Apttus_Approval__GroupName__c,
190
    Apttus_Approval__GroupSequence__c,
```

```
191
    Apttus_Approval__DependsOn__c,
192                                                    (SELECT Id,
193                                                            Name,
194
    Apttus_Approval__AdhocApprovalGroupId__c,
195
    Apttus_Approval__ApproverSequence__c,
196
    Apttus_Approval__DependsOn__c,
197
    Apttus_Approval__AssigneeId__c,
198
    Apttus_Approval__AssigneeType__c,
199
    Apttus_Approval__AssigneeValue__c,
200
    Apttus_Approval__AutoReapprovalEnabled__c,
201
    Apttus_Approval__IsReviewer__c,
202
    Apttus_Approval__SendEmail__c
203                                                     FROM
    Apttus_Approval__AdhocApprovers__r
204                                                     ORDER BY
    Apttus_Approval__ApproverSequence__c)
205                                                FROM
    Apttus_Approval__AdhocApprovalGroup__c
206                                                WHERE
    Apttus_Approval__AdhocApprovalProcessId__c = :processSO.Id
207                                                ORDER BY
    Apttus_Approval__GroupSequence__c
208                                                LIMIT 100];
209
210         // iterate over groups to update
211         for (Apttus_Approval__AdhocApprovalGroup__c groupSO : groups) {
212             Apttus_Approval__AdhocApprovalGroup__c adhocGroupSO = null;
213             if (groupSO.Apttus_Approval__GroupSequence__c == 1) {
214                 // create group 1
215                 adhocGroupSO = new Apttus_Approval__AdhocApprovalGroup__c(
216                     Id = groupSO.Id,
217                     Apttus_Approval__AdhocApprovalProcessId__c =
    processSO.Id,
218                     Apttus_Approval__DependsOn__c = null,
219                     Apttus_Approval__GroupName__c = 'Group 1 - Edited',
```

```
220                     Apttus_Approval__GroupSequence__c = 1
221                 );
222                 // update group 1
223                 adhocGroupDTO = new
    Apttus_Approval.AdhocApprovalProcessDTO.AdhocApprovalGroupDTO(adhocGroupSO
    );
224                 adhocProcessDTO.addAdhocApprovalGroup(adhocGroupDTO);
225
226                 // iterate over approvers to update
227                 for (Apttus_Approval__AdhocApprover__c approverSO :
    groupSO.Apttus_Approval__AdhocApprovers__r) {
228                     Apttus_Approval__AdhocApprover__c adhocApproverSO =
    null;
229                     if (approverSO.Apttus_Approval__ApproverSequence__c ==
    1) {
230                         // create group 1 approver 1
231                         adhocApproverSO = new
     Apttus_Approval__AdhocApprover__c(
232                             Id = approverSO.Id,
233                             Apttus_Approval__AdhocApprovalGroupId__c =
    groupSO.Id,
234                             Apttus_Approval__ApproverSequence__c = 1,
235                             Apttus_Approval__DependsOn__c = null,
236                             Apttus_Approval__AssigneeId__c = userSO1.Id,
237                             Apttus_Approval__AssigneeType__c = 'User',
238                             Apttus_Approval__AssigneeValue__c =
    userSO1.Name,
239                             Apttus_Approval__AutoReapprovalEnabled__c =
    true,
240                             Apttus_Approval__IsReviewer__c = false
241                         );
242
243                     } else if
     (approverSO.Apttus_Approval__ApproverSequence__c == 2) {
244                         // create group 1 approver 2
245                         adhocApproverSO = new
     Apttus_Approval__AdhocApprover__c(
246                             Id = approverSO.Id,
247                             Apttus_Approval__AdhocApprovalGroupId__c =
    groupSO.Id,
248                             Apttus_Approval__ApproverSequence__c = 2,
249                             Apttus_Approval__DependsOn__c = '1',
```

```
250                             Apttus_Approval__AssigneeId__c = userSO2.Id,
251                             Apttus_Approval__AssigneeType__c = 'User',
252                             Apttus_Approval__AssigneeValue__c =
    userSO2.Name,
253                             Apttus_Approval__AutoReapprovalEnabled__c =
    true,
254                             Apttus_Approval__IsReviewer__c = false
255                         );
256
257                     }
258
259                     // update approver
260                     adhocApproverDTO = new
    Apttus_Approval.AdhocApprovalProcessDTO.AdhocApproverDTO(adhocApproverSO);
261                     adhocGroupDTO.addAdhocApprover(adhocApproverDTO);
262
263                 }
264
265             } else if (groupSO.Apttus_Approval__GroupSequence__c == 2) {
266                 // create group 2
267                 adhocGroupSO = new Apttus_Approval__AdhocApprovalGroup__c(
268                     Id = groupSO.Id,
269                     Apttus_Approval__AdhocApprovalProcessId__c =
    processSO.Id,
270                     Apttus_Approval__DependsOn__c = null,
271                     Apttus_Approval__GroupName__c = 'Group 2 - Edited',
272                     Apttus_Approval__GroupSequence__c = 2
273                 );
274                 // update group 2
275                 adhocGroupDTO = new
    Apttus_Approval.AdhocApprovalProcessDTO.AdhocApprovalGroupDTO(adhocGroupSO
    );
276                 adhocProcessDTO.addAdhocApprovalGroup(adhocGroupDTO);
277
278                 // iterate over approvers to update
279                 for (Apttus_Approval__AdhocApprover__c approverSO :
    groupSO.Apttus_Approval__AdhocApprovers__r) {
280                     Apttus_Approval__AdhocApprover__c adhocApproverSO =
    null;
281                     if (approverSO.Apttus_Approval__ApproverSequence__c ==
    1) {
282                         // create group 2 approver 1
```

```
283                         adhocApproverSO = new
 Apttus_Approval__AdhocApprover__c(
284                             Id = approverSO.Id,
285                             Apttus_Approval__AdhocApprovalGroupId__c =
groupSO.Id,
286                             Apttus_Approval__ApproverSequence__c = 1,
287                             Apttus_Approval__DependsOn__c = null,
288                             Apttus_Approval__AssigneeId__c = userSO3.Id,
289                             Apttus_Approval__AssigneeType__c = 'User',
290                             Apttus_Approval__AssigneeValue__c =
userSO3.Name,
291                             Apttus_Approval__AutoReapprovalEnabled__c =
true,
292                             Apttus_Approval__IsReviewer__c = false
293                         );
294
295                     } else if
 (approverSO.Apttus_Approval__ApproverSequence__c == 2) {
296                         // create group 2 approver 2
297                         adhocApproverSO = new
 Apttus_Approval__AdhocApprover__c(
298                             Id = approverSO.Id,
299                             Apttus_Approval__AdhocApprovalGroupId__c =
groupSO.Id,
300                             Apttus_Approval__ApproverSequence__c = 2,
301                             Apttus_Approval__DependsOn__c = '1',
302                             Apttus_Approval__AssigneeId__c = userSO4.Id,
303                             Apttus_Approval__AssigneeType__c = 'User',
304                             Apttus_Approval__AssigneeValue__c =
userSO4.Name,
305                             Apttus_Approval__AutoReapprovalEnabled__c =
true,
306                             Apttus_Approval__IsReviewer__c = false
307                         );
308
309                     }
310
311                     // update approver
312                     adhocApproverDTO = new

Apttus_Approval.AdhocApprovalProcessDTO.AdhocApproverDTO(adhocApproverSO);
313                     adhocGroupDTO.addAdhocApprover(adhocApproverDTO);
314
```

```
                }

                // create group 2 approver 3
                Apttus_Approval__AdhocApprover__c adhocApproverSO = new
 Apttus_Approval__AdhocApprover__c(
                    Id = null,
                    Apttus_Approval__AdhocApprovalGroupId__c = groupSO.Id,
                    Apttus_Approval__ApproverSequence__c = 3,
                    Apttus_Approval__DependsOn__c = '2',
                    Apttus_Approval__AssigneeId__c = userSO5.Id,
                    Apttus_Approval__AssigneeType__c = 'User',
                    Apttus_Approval__AssigneeValue__c = userSO5.Name,
                    Apttus_Approval__AutoReapprovalEnabled__c = true,
                    Apttus_Approval__IsReviewer__c = false
                );
                // update group 2 approver 3
                adhocApproverDTO = new

Apttus_Approval.AdhocApprovalProcessDTO.AdhocApproverDTO(adhocApproverSO);
                adhocGroupDTO.addAdhocApprover(adhocApproverDTO);

            }

        }

        // print input structure
        PrintAdhocApprovalProcess(adhocProcessDTO);

        // call API to update process
        Apttus_Approval.AdhocApprovalProcessDTO adhocProcessDTOResult =
Apttus_Approval.ApprovalsWebService.UpdateRuntimeAdhocApprovalProcess(adho
cProcessDTO);

        // print output structure
        PrintAdhocApprovalProcess(adhocProcessDTOResult);

    }

    /**
     * Retrieve a runtime adhoc approval process
     * @param sObjectType the approval context sobject type
     * @param sObjectId the approval context sobject identifier
     * @return the runtime adhoc approval process structure
     */
```

```
354     public static void TestGetRuntimeAdhocApprovalProcess(String
    sObjectType, ID sObjectId) {
355
356         // call API to get process
357         Apttus_Approval.AdhocApprovalProcessDTO adhocProcessDTOResult =
    Apttus_Approval.ApprovalsWebService.GetRuntimeAdhocApprovalProcess(sObject
    Type, sObjectId);
358
359         // print output structure
360         PrintAdhocApprovalProcess(adhocProcessDTOResult);
361
362     }
363
364     /**
365      * Delete a runtime adhoc approval process
366      * @param sObjectType the approval context sobject type
367      * @param sObjectId the approval context sobject identifier
368      * @return <code>true</code> if the process is deleted, <code>false</
    code> otherwise
369      */
370     public static void TestDeleteRuntimeAdhocApprovalProcess(String
    sObjectType, ID sObjectId) {
371
372         // call API to delete process
373         Boolean ok =
    Apttus_Approval.ApprovalsWebService.DeleteRuntimeAdhocApprovalProcess(sObj
    ectType, sObjectId);
374         system.assertEquals(ok, true);
375
376     }
377
378     /**
379      * Add comment to a runtime adhoc approval process
380      * @param sObjectType the approval context sobject type
381      * @param sObjectId the approval context sobject identifier
382      * @return <code>true</code> if the comment was added, <code>false</
    code> otherwise
383      */
384     public static void TestAddCommentToRuntimeAdhocApprovalProcess(String
    sObjectType, ID sObjectId) {
385
386         // call API to add comment
```

```
387         Boolean ok =
Apttus_Approval.ApprovalsWebService.AddCommentToRuntimeAdhocApprovalProces
s(sObjectType, sObjectId, 'MY NEW comment');
388         system.assertEquals(ok, true);
389
390         // call API to get process
391         Apttus_Approval.AdhocApprovalProcessDTO adhocProcessDTOResult =
Apttus_Approval.ApprovalsWebService.GetRuntimeAdhocApprovalProcess(sObject
Type, sObjectId);
392         // get comment from process
393         Apttus_Approval__AdhocApprovalProcess__c adhocProcessSO =
adhocProcessDTOResult.getAdhocApprovalProcessSO();
394         String comment =
adhocProcessSO.Apttus_Approval__ProcessComments__c;
395         system.assertEquals(comment, 'MY NEW comment');
396
397     }
398
399     /**
400      * Add attachments to a runtime adhoc approval process
401      * @param sObjectType the approval context sobject type
402      * @param sObjectId the approval context sobject identifier
403      * @return <code>true</code> if the attachments were added,
<code>false</code> otherwise
404      */
405     public static void
 TestAddAttachmentsToRuntimeAdhocApprovalProcess(String sObjectType, ID
sObjectId) {
406
407         // create list of attachments to add
408         List<ID> attachmentIds = new List<ID>{sObjectId, sObjectId};
409         // call API to add attachments
410         Boolean ok =
Apttus_Approval.ApprovalsWebService.AddAttachmentsToRuntimeAdhocApprovalPr
ocess(sObjectType, sObjectId, attachmentIds);
411         system.assertEquals(ok, true);
412
413         // call API to get process
414         Apttus_Approval.AdhocApprovalProcessDTO adhocProcessDTOResult =
Apttus_Approval.ApprovalsWebService.GetRuntimeAdhocApprovalProcess(sObject
Type, sObjectId);
415         // get attachments from process
416         Apttus_Approval__AdhocApprovalProcess__c adhocProcessSO =
adhocProcessDTOResult.getAdhocApprovalProcessSO();
```

```
417         String attachmentIdsStr =
adhocProcessSO.Apttus_Approval__AttachmentIds__c;
418         String attachmentIdsStrExpected = sObjectId + ',' + sObjectId;
419         system.assertEquals(attachmentIdsStr, attachmentIdsStrExpected);
420
421     }
422
423     /**
424      * Delete attachments from a runtime adhoc approval process
425      * @param sObjectType the approval context sobject type
426      * @param sObjectId the approval context sobject identifier
427      * @return <code>true</code> if the attachments were removed,
<code>false</code> otherwise
428      */
429     public static void
 TestDeleteAttachmentsFromRuntimeAdhocApprovalProcess(String sObjectType,
ID sObjectId) {
430
431         // create list of attachments to delete
432         List<ID> attachmentIds = new List<ID>{sObjectId};
433         // call API to delete attachments
434         Boolean ok =
Apttus_Approval.ApprovalsWebService.DeleteAttachmentsFromRuntimeAdhocAppro
valProcess(sObjectType, sObjectId, attachmentIds);
435         system.assertEquals(ok, true);
436
437         // call API to get process
438         Apttus_Approval.AdhocApprovalProcessDTO adhocProcessDTOResult =
Apttus_Approval.ApprovalsWebService.GetRuntimeAdhocApprovalProcess(sObject
Type, sObjectId);
439         // get attachments from process
440         Apttus_Approval__AdhocApprovalProcess__c adhocProcessSO =
adhocProcessDTOResult.getAdhocApprovalProcessSO();
441         String attachmentIdsStr =
adhocProcessSO.Apttus_Approval__AttachmentIds__c;
442         system.assertEquals(attachmentIdsStr, sObjectId);
443
444     }
445
446     /**
447      * Prints an adhoc process definition
448      * @param approvalProcessDTO the adhoc approval process definition
449      */
```

```
450     public static void
 PrintAdhocApprovalProcess(Apttus_Approval.AdhocApprovalProcessDTO
approvalProcessDTO) {
451
452         // print process info
453         Apttus_Approval__AdhocApprovalProcess__c processSO =
approvalProcessDTO.getAdhocApprovalProcessSO();
454         system.debug('processSO='+processSO);
455
456         // print approval groups and approvers
457         for (Apttus_Approval.AdhocApprovalProcessDTO.AdhocApprovalGroupDTO
groupDTO : approvalProcessDTO.getAdhocApprovalGroups()) {
458             Apttus_Approval__AdhocApprovalGroup__c groupSO =
groupDTO.getAdhocApprovalGroupSO();
459             system.debug('groupSO='+groupSO);
460
461             // get approvers
462             for (Apttus_Approval.AdhocApprovalProcessDTO.AdhocApproverDTO
approverDTO : groupDTO.getAdhocApprovers()) {
463                 Apttus_Approval__AdhocApprover__c approverSO =
approverDTO.getAdhocApproverSO();
464                 system.debug('approverSO='+approverSO);
465
466             }
467
468         }
469
470     }
471
472 }
```

Apttus Copyright Disclaimer

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.conga.com.

DOC ID: IWASFSPR20API20200408