

Apex Triggers

Get Started with Apex Triggers :

AccountAddressTrigger.apxt

```
trigger AccountAddressTrigger on Account (before insert) {  
  
    for(Account a:Trigger.New){  
        if(a.Match_Billing_Address__c ==true){  
            a.ShippingPostalCode = a.BillingPostalCode;  
  
        }  
  
    }  
  
}
```

Bulk Apex Triggers:

ClosedOpportunityTrigger.apxt

```
trigger ClosedOpportunityTrigger on Opportunity (after insert, after update) {  
    List<Task> taskList = new List<Task>();  
    //first way  
    for(Opportunity opp : [SELECT Id, StageName FROM Opportunity WHERE  
    StageName='Closed Won' AND Id IN : Trigger.New]){  
        taskList.add(new Task(Subject='Follow Up Test Task', WhatId = opp.Id));  
    }  
  
    insert tasklist;  
}  
  
}
```

Apex Testing

Get Started with Apex Unit Tests:

VerifyDate.apxc

```
public class VerifyDate {  
  
    //method to handle potential checks against two dates  
  
    public static Date CheckDates(Date date1, Date date2) {  
  
        //if date2 is within the next 30 days of date1, use date2. Otherwise use  
        the end of the month  
  
        if(DateWithin30Days(date1,date2)) {  
  
            return date2;  
  
        } else {  
  
            return SetEndOfMonthDate(date1);  
  
        }  
  
    }  
  
    //method to check if date2 is within the next 30 days of date1  
  
    private static Boolean DateWithin30Days(Date date1, Date date2) {
```

```
//check for date2 being in the past

if( date2 < date1) { return false; }

//check that date2 is within (>=) 30 days of date1

Date date30Days = date1.addDays(30); //create a date 30 days away from date1

if( date2 >= date30Days ) { return false; }

else { return true; }

}

//method to return the end of the month of a given date

private static Date SetEndOfMonthDate(Date date1) {

    Integer totalDays = Date.daysInMonth(date1.year(), date1.month());

    Date lastDay = Date.newInstance(date1.year(), date1.month(), totalDays);

    return lastDay;

}
```

TestVerifyDate.apxc

```
@isTest
public class TestVerifyDate {
    @isTest static void test1(){
        Date d =
VerifyDate.CheckDates(Date.parse('01/01/2022'),Date.parse('01/03/2022'));
        System.assertEquals(Date.parse('01/03/2022'),d);
    }
    @isTest static void test2(){
        Date d =
VerifyDate.CheckDates(Date.parse('01/01/2022'),Date.parse('03/03/2022'));
        System.assertEquals(Date.parse('01/31/2022'),d);
    }
}
```

Test Apex Triggers:

RestrictContactByName.apxt

```
trigger RestrictContactByName on Contact (before insert, before update) {

    //check contacts prior to insert or update for invalid data
    For (Contact c : Trigger.New) {
        if(c.LastName == 'INVALIDNAME') {      //invalidname is invalid
            c.AddError('The Last Name "'+c.LastName+'" is not allowed for
DML');
        }
    }
}
```

TestRestrictContactByName.apxc

```
@isTest
public class TestRestrictContactByName {
```

```

@isTest

public static void testConsole(){

    Contact ct =new Contact();

    ct.LastName='INVALIDNAME';

    Database.SaveResult res = Database.insert(ct,false);

    System.assertEquals('The last Name "INVALIDNAME" is not allowed for
DML',res.getErrors()[0].getMessage());

}

}

```

Create Test Data for Apex Tests:

RandomContactFactory.apxc

```

public class RandomContactFactory {
    public static List<Contact> generateRandomContacts(Integer num,String lastName){
        List<Contact> contactList =new List<Contact>();
        for(Integer i=1;i<=num;i++){
            Contact ct = new Contact(FirstName ='Test'+i,LastName = lastName);
            contactList.add(ct);
        }
        return contactList;
    }
}

```

Asynchronous Apex

Use Future Methods:

AccountProcessor.apxc

```
public class AccountProcessor
{
    @future
    public static void countContacts(Set<id> setId)
    {
        List<Account> lstAccount = [select id,Number_of_Contacts__c , (select id from
        contacts ) from account where id in :setId ];
        for( Account acc : lstAccount )
        {
            List<Contact> lstCont = acc.contacts ;
            acc.Number_of_Contacts__c = lstCont.size();
        }
        update lstAccount;
    }
}
```

AccountProcessorTest.apxc

```
@IsTest
public class AccountProcessorTest {
    public static testmethod void TestAccountProcessorTest()
    {
        Account a = new Account();
        a.Name = 'Test Account';
        Insert a;

        Contact cont = New Contact();
        cont.FirstName ='Bob';
        cont.LastName ='Masters';
        cont.AccountId = a.Id;
        Insert cont;

        set<Id> setAccId = new Set<ID>();
```

```

        setAcctId.add(a.id);

        Test.startTest();
        AccountProcessor.countContacts(setAcctId);
        Test.stopTest();

        Account ACC = [select Number_of_Contacts__c from Account where id = :a.id LIMIT
1];
        System.assertEquals ( Integer.valueOf(ACC.Number_of_Contacts__c) ,1);
    }

}

```

Use Batch Apex:

LeadProcessor.apxc

```

global class LeadProcessor implements
Database.Batchable<sObject>, Database.Stateful {

    // instance member to retain state across transactions
    global Integer recordsProcessed = 0;

    global Database.QueryLocator start(Database.BatchableContext bc) {
        return Database.getQueryLocator('SELECT Id, LeadSource FROM Lead');
    }

    global void execute(Database.BatchableContext bc, List<Lead> scope){
        // process each batch of records
        List<Lead> leads = new List<Lead>();
        for (Lead lead : scope) {

            lead.LeadSource = 'Dreamforce';
            // increment the instance member counter
            recordsProcessed = recordsProcessed + 1;

        }
        update leads;
    }

    global void finish(Database.BatchableContext bc){

```

```
        System.debug(recordsProcessed + ' records processed. Shazam!');

    }

}
```

LeadProcessorTest.apxc

```
@isTest
public class LeadProcessorTest {
    @testSetup
    static void setup() {
        List<Lead> leads = new List<Lead>();
        // insert 200 leads
        for (Integer i=0;i<200;i++) {
            leads.add(new Lead(LastName='Lead '+i,
                Company='Lead', Status='Open - Not Contacted'));
        }
        insert leads;
    }

    static testmethod void test() {
        Test.startTest();
        LeadProcessor lp = new LeadProcessor();
        Id batchId = Database.executeBatch(lp, 200);
        Test.stopTest();

        // after the testing stops, assert records were updated properly
        System.assertEquals(200, [select count() from lead where LeadSource =
'Dreamforce']);
    }
}
```

Control Processes with Queueable Apex:

AddPrimaryContact.apxc

```
public class AddPrimaryContact implements Queueable{
    Contact con;
    String state;

    public AddPrimaryContact(Contact con, String state){
        this.con = con;
        this.state = state;
    }
    public void execute(QueueableContext qc){
        List<Account> lstOfAccs = [SELECT Id FROM Account WHERE BillingState = :state
LIMIT 200];

        List<Contact> lstOfConts = new List<Contact>();
        for(Account acc : lstOfAccs){
            Contact conInst = con.clone(false,false,false,false);
            conInst.AccountId = acc.Id;

            lstOfConts.add(conInst);
        }

        INSERT lstOfConts;
    }
}
```

AddPrimaryContactTest.apxc

```
@isTest
public class AddPrimaryContactTest{
    @testSetup
    static void setup(){
        List<Account> lstOfAcc = new List<Account>();
        for(Integer i = 1; i <= 100; i++){
            if(i <= 50)
                lstOfAcc.add(new Account(name='AC'+i, BillingState = 'NY'));
            else
                lstOfAcc.add(new Account(name='AC'+i, BillingState = 'CA'));
        }
    }
}
```

```

    INSERT lstOfAcc;
}

static testmethod void testAddPrimaryContact(){
    Contact con = new Contact(LastName = 'TestCont');
    AddPrimaryContact addPCIns = new AddPrimaryContact(CON 'CA');

    Test.startTest();
    System.enqueueJob(addPCIns);
    Test.stopTest();

    System.assertEquals(50, [select count() from Contact]);
}
}

```

AddPrimaryContactTest.apxc

```

@isTest
public class AddPrimaryContactTest{
    @testSetup
    static void setup(){
        List<Account> lstOfAcc = new List<Account>();
        for(Integer i = 1; i <= 100; i++){
            if(i <= 50)
                lstOfAcc.add(new Account(name='AC'+i, BillingState = 'NY'));
            else
                lstOfAcc.add(new Account(name='AC'+i, BillingState = 'CA'));
        }

        INSERT lstOfAcc;
    }

    static testmethod void testAddPrimaryContact(){
        Contact con = new Contact(LastName = 'TestCont');
        AddPrimaryContact addPCIns = new AddPrimaryContact(CON 'CA');

        Test.startTest();
        System.enqueueJob(addPCIns);
        Test.stopTest();
    }
}

```

```
        System.assertEquals(50, [select count() from Contact]);
    }
}
```

Schedule Jobs Using the Apex Scheduler:

DailyLeadProcessorTest.apxc

```
@isTest
public class DailyLeadProcessorTest {

    @isTest
    public static void testing() {

        List<lead> l_lst = new List<lead>();
        for(Integer i=0;i<200;i++) {
            lead l = new lead();
            l.lastname = 'lastname'+i;
            l.Company = 'company'+i;
            l_lst.add(l);
        }
        insert l_lst;

        Test.startTest();
        DailyLeadProcessor dlp = new DailyLeadProcessor ();
        String jobId = System.Schedule('dailyleadprocessing','0 0 0 1 12 ? 2022',dlp);
        Test.stopTest();

        List<lead> l_lst_chk = new List<lead>([select id,leadsource from lead where
leadsource != 'Dreamforce']);
        System.assertequals(0,l_lst_chk.size());
    }
}
```

Apex Integration Services

Apex Integration Overview:

The image contains two side-by-side screenshots of the Salesforce Setup interface, both titled "Remote Site Settings".

Screenshot 1 (Top): The search bar at the top contains "Quick Find". The sidebar on the left shows "Setup Home", "Service Setup Assistant", "Multi-Factor Authentication Assistant", "Release Updates", "Lightning Experience Transition Assistant", "New Salesforce Mobile App QuickStart", "Lightning Usage", "Optimizer", "ADMINISTRATION" (with "Users", "Data", "Email"), and "PLATFORM TOOLS" (with "Apps", "Feature Settings", "Slack", "Workflow Services", "Einstein", "Objects and Fields"). The main content area displays the "Remote Site Details" for a site named "animals_http" with URL "https://rh-apex-http-callout.herokuapp.com". It includes fields for "Disable Protocol Security" (unchecked), "Description" ("Trailhead animal service: HTTP"), "Active" (checked), and "Created By" ("Sravya Reddy, 6/9/2022, 9:24 AM").

Screenshot 2 (Bottom): The search bar at the top contains "remo". The sidebar on the left shows "Custom Code", "Remote Access", "Security", and "Remote Site Settings" (which is selected). The main content area displays the "Remote Site Details" for a site named "animals_soap" with URL "https://rh-apex-soap-service.herokuapp.com". It includes fields for "Disable Protocol Security" (unchecked), "Description" ("Trailhead animal service: SOAP"), "Active" (checked), and "Created By" ("Sravya Reddy, 6/9/2022, 9:25 AM").

Apex REST Callouts:

AnimalLocator.apxc

```
public class AnimalLocator
{
    public static String getAnimalNameById(Integer id)
    {
```

```

Http http = new Http();
HttpRequest request = new HttpRequest();
request.setEndpoint('https://th-apex-http-callout.herokuapp.com/animals/'+id);
request.setMethod('GET');
HttpResponse response = http.send(request);
String strResp = "";
system.debug('*****response '+response.getStatusCode());
system.debug('*****response '+response.getBody());
// If the request is successful, parse the JSON response.
if (response.getStatusCode() == 200)
{
    // Deserializes the JSON string into collections of primitive data types.
    Map<String, Object> results = (Map<String, Object>)
JSON.deserializeUntyped(response.getBody());
    // Cast the values in the 'animals' key as a list
    Map<string,object> animals = (map<string,object>) results.get('animal');
    System.debug('Received the following animals:' + animals );
    strResp = string.valueof(animals.get('name'));
    System.debug('strResp >>>>' + strResp );
}
return strResp ;
}

}

```

AnimalLocatorTest.apxc

```

@isTest
private class AnimalLocatorTest{
    @isTest static void AnimalLocatorMock1() {
        Test.SetMock(HttpCallOutMock.class, new AnimalLocatorMock());
        string result=AnimalLocator.getAnimalNameById(3);
        string expectedResult='chicken';
        System.assertEquals(result, expectedResult);
    }
}

```

AnimalLocatorMock.apxc

```
@isTest
global class AnimalLocatorMock implements HttpCalloutMock {
    global HttpResponseMessage respond(HTTPRequest request) {
        HttpResponse response = new HttpResponse();
        response.setHeader('Content-Type', 'application/json');
        response.setBody('{"animal":{"id":1,"name":"chicken","eats":"chicken
food","says":"cluck cluck"}}');
        response.setStatusCode(200);
        return response;
    }
}
```

Apex SOAP Callouts:

ParkLocator.apxc

```
public class ParkLocator {
    public static String[] country(String country){
        ParkService.ParksImplPort parks = new ParkService.ParksImplPort();
        String[] parksname = parks.byCountry(country);
        return parksname;
    }
}
```

ParkLocatorTest.apxc

```
@isTest
private class ParkLocatorTest{
    @isTest
    static void testParkLocator() {
        Test.setMock(WebServiceMock.class, new ParkServiceMock());
        String[] arrayOfParks = ParkLocator.country('India');

        System.assertEquals('Park1', arrayOfParks[0]);
    }
}
```

ParkServiceMock.apxc

```

@isTest
global class ParkServiceMock implements WebServiceMock {
    global void doInvoke(
        Object stub,
        Object request,
        Map<String, Object> response,
        String endpoint,
        String soapAction,
        String requestName,
        String responseNS,
        String responseName,
        String responseType) {
        ParkService.byCountryResponse response_x = new
        ParkService.byCountryResponse();
        List<String> lstOfDummyParks = new List<String> {'Park1','Park2','Park3'};
        response_x.return_x = lstOfDummyParks;

        response.put('response_x', response_x);
    }
}

```

Apex Web Services:

AccountManager.apxc

```

@RestResource(urlMapping='/Accounts/*/contacts')
global with sharing class AccountManager{
    @HttpGet
    global static Account getAccount(){
        RestRequest req = RestContext.request;
        String accId = req.requestURI.substringBetween('Accounts/', '/contacts');
        Account acc = [SELECT Id, Name, (SELECT Id, Name FROM Contacts)
                      FROM Account WHERE Id = :accId];

        return acc;
    }
}

AccountManagerTest.apxc
@IsTest

```

```

private class AccountManagerTest{
    @isTest static void testAccountManager(){
        Id recordId = getTestAccountId();
        // Set up a test request
        RestRequest request = new RestRequest();
        request.requestUri =
            'https://ap5.salesforce.com/services/apexrest/Accounts/'+ recordId +'/' + 'contacts';
        request.httpMethod = 'GET';
        RestContext.request = request;

        // Call the method to test
        Account acc = AccountManager.getAccount();

        // Verify results
        System.assert(acc != null);
    }

    private static Id getTestAccountId(){
        Account acc = new Account(Name = 'TestAcc2');
        Insert acc;

        Contact con = new Contact(LastName = 'TestCont2', AccountId = acc.Id);
        Insert con;

        return acc.Id;
    }
}

```

Apex Specialist

Automate record creation:

The screenshot shows the 'Maintenance Request' object's field configuration. The 'Type' section is selected, showing details like Data Type (Picklist), Help Text, Description, Data Owner, Field Usage, Data Sensitivity Level, and Compliance Categorization. It also displays 'Field Information' and 'Case Type Picklist Values' sections.

This screenshot provides a more detailed view of the 'Case Type Picklist Values' section, showing a table of active and inactive values. Active values include Mechanical, Electrical, Electronic, Structural, Other, Repair, and Routine Maintenance. Inactive values include Inspection.

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	Mechanical	Mechanical	<input type="checkbox"/>	Assigned dynamically	Sriya Reddy, 4/12/2022, 1:47 AM
Edit Del Deactivate	Electrical	Electrical	<input type="checkbox"/>	Assigned dynamically	Sriya Reddy, 4/12/2022, 1:47 AM
Edit Del Deactivate	Electronic	Electronic	<input type="checkbox"/>	Assigned dynamically	Sriya Reddy, 4/12/2022, 1:47 AM
Edit Del Deactivate	Structural	Structural	<input type="checkbox"/>	Assigned dynamically	Sriya Reddy, 4/12/2022, 1:47 AM
Edit Del Deactivate	Other	Other	<input type="checkbox"/>	Assigned dynamically	Sriya Reddy, 4/12/2022, 1:47 AM
Edit Del Deactivate	Repair	Repair	<input type="checkbox"/>	Assigned dynamically	putnala.sriya, 6/22/2022, 9:23 AM
Edit Del Deactivate	Routine Maintenance	Routine Maintenance	<input type="checkbox"/>	Assigned dynamically	putnala.sriya, 6/22/2022, 9:23 AM

Action	Values	API Name	Modified By
Del Activate	Inspection'	Inspection'	putnala.sriya, 6/22/2022, 9:48 AM

MaintenanceRequest.apxt

```
trigger MaintenanceRequest on Case (before update, after update) {
    if(Trigger.isUpdate && Trigger.isAfter){
        MaintenanceRequestHelper.updateWorkOrders(Trigger.New, Trigger.OldMap);
    }
}
```

MaintenanceRequestHelper.apxc

```
public with sharing class MaintenanceRequestHelper {
    public static void updateWorkOrders(List<Case> updWorkOrders, Map<Id,Case> nonUpdCaseMap) {
        Set<Id> validIds = new Set<Id>();
```

```

For (Case c : updWorkOrders){
    if (nonUpdCaseMap.get(c.Id).Status != 'Closed' && c.Status == 'Closed'){
        if (c.Type == 'Repair' || c.Type == 'Routine Maintenance'){
            validIds.add(c.Id);

        }
    }
}

if (!validIds.isEmpty()){
    List<Case> newCases = new List<Case>();
    Map<Id,Case> closedCasesM = new Map<Id,Case>([SELECT Id, Vehicle__c,
Equipment__c, Equipment__r.Maintenance_Cycle__c,(SELECT
Id,Equipment__c,Quantity__c FROM Equipment_Maintenance_Items__r)
FROM Case WHERE Id IN :validIds]);
    Map<Id,Decimal> maintenanceCycles = new Map<ID,Decimal>();
    AggregateResult[] results = [SELECT Maintenance_Request__c,
MIN(Equipment__r.Maintenance_Cycle__c)cycle FROM
Equipment_Maintenance_Item__c WHERE Maintenance_Request__c IN :ValidIds GROUP
BY Maintenance_Request__c];

    for (AggregateResult ar : results){
        maintenanceCycles.put((Id) ar.get('Maintenance_Request__c'), (Decimal)
ar.get('cycle'));
    }

    for(Case cc : closedCasesM.values()){
        Case nc = new Case (
            ParentId = cc.Id,
            Status = 'New',
            Subject = 'Routine Maintenance',
            Type = 'Routine Maintenance',
            Vehicle__c = cc.Vehicle__c,
            Equipment__c =cc.Equipment__c,
            Origin = 'Web',

```

```

        Date_Reported__c = Date.Today()

    );

    If (maintenanceCycles.containsKey(cc.Id)){
        nc.Date_Due__c = Date.today().addDays((Integer)
maintenanceCycles.get(cc.Id));
    }

    newCases.add(nc);
}

insert newCases;

List<Equipment_Maintenance_Item__c> clonedWPs = new
List<Equipment_Maintenance_Item__c>();
for (Case nc : newCases){
    for (Equipment_Maintenance_Item__c wp :
closedCasesM.get(nc.ParentId).Equipment_Maintenance_Items__r){
        Equipment_Maintenance_Item__c wpClone = wp.clone();
        wpClone.Maintenance_Request__c = nc.Id;
        ClonedWPs.add(wpClone);

    }
}
insert ClonedWPs;
}
}
}

```

MaintenanceRequestHelperTest.apxc

```

@istest
public with sharing class MaintenanceRequestHelperTest {

    private static final string STATUS_NEW = 'New';
    private static final string WORKING = 'Working';
    private static final string CLOSED = 'Closed';
    private static final string REPAIR = 'Repair';

```

```

private static final string REQUEST_ORIGIN = 'Web';
private static final string REQUEST_TYPE = 'Routine Maintenance';
private static final string REQUEST SUBJECT = 'Testing subject';

PRIVATE STATIC Vehicle__c createVehicle(){
    Vehicle__c Vehicle = new Vehicle__C(name = 'SuperTruck');
    return Vehicle;
}

PRIVATE STATIC Product2 createEq(){
    product2 equipment = new product2(name = 'SuperEquipment',
        lifespan_months__C = 10,
        maintenance_cycle__C = 10,
        replacement_part__c = true);
    return equipment;
}

PRIVATE STATIC Case createMaintenanceRequest(id vehicleId, id equipmentId){
    case cs = new case(Type=REPAIR,
        Status=STATUS_NEW,
        Origin=REQUEST_ORIGIN,
        Subject=REQUEST_SUBJECT,
        Equipment__c=equipmentId,
        Vehicle__c=vehicleId);
    return cs;
}

PRIVATE STATIC Equipment_Maintenance_Item__c createWorkPart(id equipmentId,id
requestId){
    Equipment_Maintenance_Item__c wp = new
Equipment_Maintenance_Item__c(Equipment__c = equipmentId,
                                Maintenance_Request__c = requestId);
    return wp;
}
@istest
private static void testMaintenanceRequestPositive(){
    Vehicle__c vehicle = createVehicle();

```

```

insert vehicle;
id vehicleId = vehicle.Id;

Product2 equipment = createEq();
insert equipment;
id equipmentId = equipment.Id;

case somethingToUpdate = createMaintenanceRequest(vehicleId,equipmentId);
insert somethingToUpdate;

Equipment_Maintenance_Item__c workP =
createWorkPart(equipmentId,somethingToUpdate.id);
insert workP;

test.startTest();
somethingToUpdate.status = CLOSED;
update somethingToUpdate;
test.stopTest();

Case newReq = [Select id, subject, type, Equipment__c, Date_Reported__c,
Vehicle__c, Date_Due__c
    from case
    where status =:STATUS_NEW];

Equipment_Maintenance_Item__c workPart = [select id
    from Equipment_Maintenance_Item__c
    where Maintenance_Request__c =:newReq.Id];

system.assert(workPart != null);
system.assert(newReq.Subject != null);
system.assertEquals(newReq.Type, REQUEST_TYPE);
SYSTEM.assertEquals(newReq.Equipment__c, equipmentId);
SYSTEM.assertEquals(newReq.Vehicle__c, vehicleId);
SYSTEM.assertEquals(newReq.Date_Reported__c, system.today());
}

@istest
private static void testMaintenanceRequestNegative(){

```

```

Vehicle__C vehicle = createVehicle();
insert vehicle;
id vehicleId = vehicle.Id;

product2 equipment = createEq();
insert equipment;
id equipmentId = equipment.Id;

case emptyReq = createMaintenanceRequest(vehicleId,equipmentId);
insert emptyReq;

Equipment_Maintenance_Item__c workP = createWorkPart(equipmentId,
emptyReq.Id);
insert workP;

test.startTest();
emptyReq.Status = WORKING;
update emptyReq;
test.stopTest();

list<case> allRequest = [select id
                           from case];

Equipment_Maintenance_Item__c workPart = [select id
                                             from Equipment_Maintenance_Item__c
                                             where Maintenance_Request__c = :emptyReq.Id];

system.assert(workPart != null);
system.assert(allRequest.size() == 1);
}

@istest
private static void testMaintenanceRequestBulk(){
    list<Vehicle__C> vehicleList = new list<Vehicle__C>();
    list<Product2> equipmentList = new list<Product2>();
    list<Equipment_Maintenance_Item__c> workPartList = new
list<Equipment_Maintenance_Item__c>();
    list<case> requestList = new list<case>();
}

```

```

list<id> oldRequestIds = new list<id>();

for(integer i = 0; i < 300; i++){
    vehicleList.add(createVehicle());
    equipmentList.add(createEq());
}
insert vehicleList;
insert equipmentList;

for(integer i = 0; i < 300; i++){
    requestList.add(createMaintenanceRequest(vehicleList.get(i).id,
equipmentList.get(i).id));
}
insert requestList;

for(integer i = 0; i < 300; i++){
    workPartList.add(createWorkPart(equipmentList.get(i).id, requestList.get(i).id));
}
insert workPartList;

test.startTest();
for(case req : requestList){
    req.Status = CLOSED;
    oldRequestIds.add(req.Id);
}
update requestList;
test.stopTest();

list<case> allRequests = [select id
                           from case
                           where status =: STATUS_NEW];

list<Equipment_Maintenance_Item__c> workParts = [select id
                                                   from Equipment_Maintenance_Item__c
                                                   where Maintenance_Request__c in: oldRequestIds];

system.assert(allRequests.size() == 300);
}

```

```
}
```

Synchronize salesforce data with an external system:

WarehouseCalloutService.apxc

```
public with sharing class WarehouseCalloutService {  
  
    private static final String WAREHOUSE_URL = 'https://th-superbadge-apex.herokuapp.com/equipment';  
  
    //@future(callout=true)  
    public static void runWarehouseEquipmentSync(){  
  
        Http http = new Http();  
        HttpRequest request = new HttpRequest();  
  
        request.setEndpoint(WAREHOUSE_URL);  
        request.setMethod('GET');  
        HttpResponse response = http.send(request);  
  
        List<Product2> warehouseEq = new List<Product2>();  
  
        if (response.getStatusCode() == 200){  
            List<Object> jsonResponse =  
(List<Object>)JSON.deserializeUntyped(response.getBody());  
            System.debug(response.getBody());  
  
            for (Object eq : jsonResponse){  
                Map<String, Object> mapJson = (Map<String, Object>)eq;  
                Product2 myEq = new Product2();  
                myEq.Replacement_Part__c = (Boolean) mapJson.get('replacement');  
                myEq.Name = (String) mapJson.get('name');  
                myEq.Maintenance_Cycle__c = (Integer) mapJson.get('maintenanceperiod');  
                myEq.Lifespan_Months__c = (Integer) mapJson.get('lifespan');  
                myEq.Cost__c = (Decimal) mapJson.get('lifespan');  
                myEq.Warehouse_SKU__c = (String) mapJson.get('sku');  
                myEq.Current_Inventory__c = (Double) mapJson.get('quantity');  
                warehouseEq.add(myEq);  
            }  
        }  
    }  
}
```

```

    }

    if (warehouseEq.size() > 0){
        upsert warehouseEq;
        System.debug('Your equipment was synced with the warehouse one');
        System.debug(warehouseEq);
    }

}

}
}
}

```

Schedule synchronization:

WarehouseSyncSchedule.apxc

```

global class WarehouseSyncSchedule implements Schedulable {
    global void execute(SchedulableContext ctx) {

        WarehouseCalloutService.runWarehouseEquipmentSync();
    }
}

```

Test automation logic:

MaintenanceRequestHelperTest.apxc

```

@istest
public with sharing class MaintenanceRequestHelperTest {

```

```

    private static final string STATUS_NEW = 'New';
    private static final string WORKING = 'Working';
    private static final string CLOSED = 'Closed';
    private static final string REPAIR = 'Repair';
    private static final string REQUEST_ORIGIN = 'Web';
    private static final string REQUEST_TYPE = 'Routine Maintenance';
    private static final string REQUEST SUBJECT = 'Testing subject';

```

```

PRIVATE STATIC Vehicle__c createVehicle(){
    Vehicle__c Vehicle = new Vehicle__C(name = 'SuperTruck');

```

```

        return Vehicle;
    }

PRIVATE STATIC Product2 createEq(){
    product2 equipment = new product2(name = 'SuperEquipment',
        lifespan_months__C = 10,
        maintenance_cycle__C = 10,
        replacement_part__c = true);
    return equipment;
}

PRIVATE STATIC Case createMaintenanceRequest(id vehicleId, id equipmentId){
    case cs = new case(Type=REPAIR,
        Status=STATUS_NEW,
        Origin=REQUEST_ORIGIN,
        Subject=REQUEST SUBJECT,
        Equipment__c=equipmentId,
        Vehicle__c=vehicleId);
    return cs;
}

PRIVATE STATIC Equipment_Maintenance_Item__c createWorkPart(id equipmentId,id
requestId){
    Equipment_Maintenance_Item__c wp = new
    Equipment_Maintenance_Item__c(Equipment__c = equipmentId,
                                    Maintenance_Request__c = requestId);
    return wp;
}

@istest
private static void testMaintenanceRequestPositive(){
    Vehicle__c vehicle = createVehicle();
    insert vehicle;
    id vehicleId = vehicle.Id;

    Product2 equipment = createEq();
    insert equipment;
    id equipmentId = equipment.Id;
}

```

```

case somethingToUpdate = createMaintenanceRequest(vehicleId,equipmentId);
insert somethingToUpdate;

Equipment_Maintenance_Item__c workP =
createWorkPart(equipmentId,somethingToUpdate.id);
insert workP;

test.startTest();
emptyReq.Status = WORKING;
update emptyReq;
test.stopTest();

list<case> allRequest = [select id
                           from case];

Equipment_Maintenance_Item__c workPart = [select id
                                              from Equipment_Maintenance_Item__c
                                              where Maintenance_Request__c = :emptyReq.Id];

system.assert(workPart != null);
system.assert(allRequest.size() == 1);
}

@istest
private static void testMaintenanceRequestBulk(){
    list<Vehicle__C> vehicleList = new list<Vehicle__C>();
    list<Product2> equipmentList = new list<Product2>();
    list<Equipment_Maintenance_Item__c> workPartList = new
list<Equipment_Maintenance_Item__c>();
    list<case> requestList = new list<case>();
    list<id> oldRequestIds = new list<id>();

    for(integer i = 0; i < 300; i++){
        vehicleList.add(createVehicle());
        equipmentList.add(createEq());
    }
    insert vehicleList;
    insert equipmentList;
}

```

```

for(integer i = 0; i < 300; i++){
    requestList.add(createMaintenanceRequest(vehicleList.get(i).id,
equipmentList.get(i).id));
}
insert requestList;

for(integer i = 0; i < 300; i++){
    workPartList.add(createWorkPart(equipmentList.get(i).id, requestList.get(i).id));
}
insert workPartList;

test.startTest();
for(case req : requestList){
    req.Status = CLOSED;
    oldRequestIds.add(req.Id);
}
update requestList;
test.stopTest();

list<case> allRequests = [select id
                           from case
                           where status =: STATUS_NEW];

list<Equipment_Maintenance_Item__c> workParts = [select id
                                                   from Equipment_Maintenance_Item__c
                                                   where Maintenance_Request__c in: oldRequestIds];

system.assert(allRequests.size() == 300);
}
}

```

Test callout logic:

[WarehouseCalloutServiceMock.apxc](#)

```

@isTest
global class WarehouseCalloutServiceMock implements HttpCalloutMock {
    // implement http mock callout
    global static HttpResponse respond(HttpRequest request){

```

```

        System.assertEquals('https://th-superbadge-apex.herokuapp.com/equipment',
request.getEndpoint());
        System.assertEquals('GET', request.getMethod());

        // Create a fake response
        HttpResponse response = new HttpResponse();
        response.setHeader('Content-Type', 'application/json');

response.setBody('[{"_id":"55d66226726b611100aaaf741","replacement":false,"quantity":5
,"name":"Generator 1000
kW","maintenanceperiod":365,"lifespan":120,"cost":5000,"sku":"100003"}]');
        response.setStatusCode(200);
        return response;
    }
}

@isTest
WarehouseCalloutServiceTest.apxc
private class WarehouseCalloutServiceTest {
    @isTest
    static void testWareHouseCallout(){
        Test.startTest();
        // implement mock callout test here
        Test.setMock(HTTPCalloutMock.class, new WarehouseCalloutServiceMock());
        WarehouseCalloutService.runWarehouseEquipmentSync();
        Test.stopTest();
        System.assertEquals(1, [SELECT count() FROM Product2]);
    }
}

```

Test Scheduling logic:

WarehouseSyncScheduleTest.apxc

```

@isTest

public class WarehouseSyncScheduleTest {

    @isTest static void WarehousescheduleTest(){

String scheduleTime = '00 00 01 * * ?';

```

```
Test.startTest();
Test.setMock(HttpCalloutMock.class, new WarehouseCalloutServiceMock());
String jobID=System.schedule('Warehouse Time To Schedule to Test',
scheduleTime, new WarehouseSyncSchedule());
Test.stopTest();
//Contains schedule information for a scheduled job. CronTrigger is similar to a
cron job on UNIX systems.
// This object is available in API version 17.0 and later.
CronTrigger a=[SELECT Id FROM CronTrigger where NextFireTime > today];
System.assertEquals(jobID, a.Id,'Schedule ');
}
}
```

Process Automation Specialist superbadge

Formulas and Validations

Use Formula Fields:

The screenshot shows the Salesforce Setup interface for creating a custom field. The object selected is 'Contract'. The 'Fields & Relationships' tab is active. A new custom field is being created with the following details:

- Field Label:** Days Remaining
- Field Name:** Days_Remaining
- API Name:** Days_Remaining__c
- Data Type:** Formula
- Formula:** EndDate-TODAY()
- Decimal Places:** 2

The 'Custom Field Definition Detail' section includes fields like Field Usage, Data Sensitivity Level, and Compliance Categorization. The 'Object Name' is set to 'Contract'. The 'Created By' and 'Modified By' fields show 'Stravya Reddy' with the timestamp '6/6/2022, 9:17 AM'.

Implement Roll-Up Summary Fields:

The screenshot shows the Salesforce Setup interface for creating a custom field. The object selected is 'Account'. The 'Fields & Relationships' tab is active. A new custom field is being created with the following details:

- Field Label:** Potential Value
- Field Name:** Potential_Value
- API Name:** Potential_Value__c
- Data Type:** Roll-Up Summary
- Summarized Object:** Opportunity
- Field to Aggregate:** Opportunity_Amount
- Filter Criteria:** None

The 'Custom Field Definition Detail' section includes fields like Field Usage, Data Sensitivity Level, and Compliance Categorization. The 'Object Name' is set to 'Account'. The 'Created By' and 'Modified By' fields show 'Stravya Reddy' with the timestamp '6/6/2022, 9:32 AM'. The 'Summary Type' is set to 'SUM'.

Create Validation Rules:

The screenshot shows the Salesforce Setup interface with the following details:

- Page Header:** SETUP > OBJECT MANAGER
- Object:** Contact
- Section:** Validation Rule Detail
- Role Name:** Contact_must_be_in_Account_ZIP_Code
- Error Condition Formula:** AND(NOT(ISBLANK(AccountId)), MailingPostalCode <> Account.ShippingPostalCode)
- Error Message:** Mailing postal code should be equal to Accounts shipping postal code
- Description:** (empty)
- Created By:** Sravya.Reddy, 6/6/2022, 10:04 AM
- Active:** checked
- Error Location:** Top of Page
- Modified By:** Sravya.Reddy, 6/6/2022, 10:04 AM

The left sidebar lists various setup categories such as Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Hierarchy Columns, Scoping Rules, and Triggers.

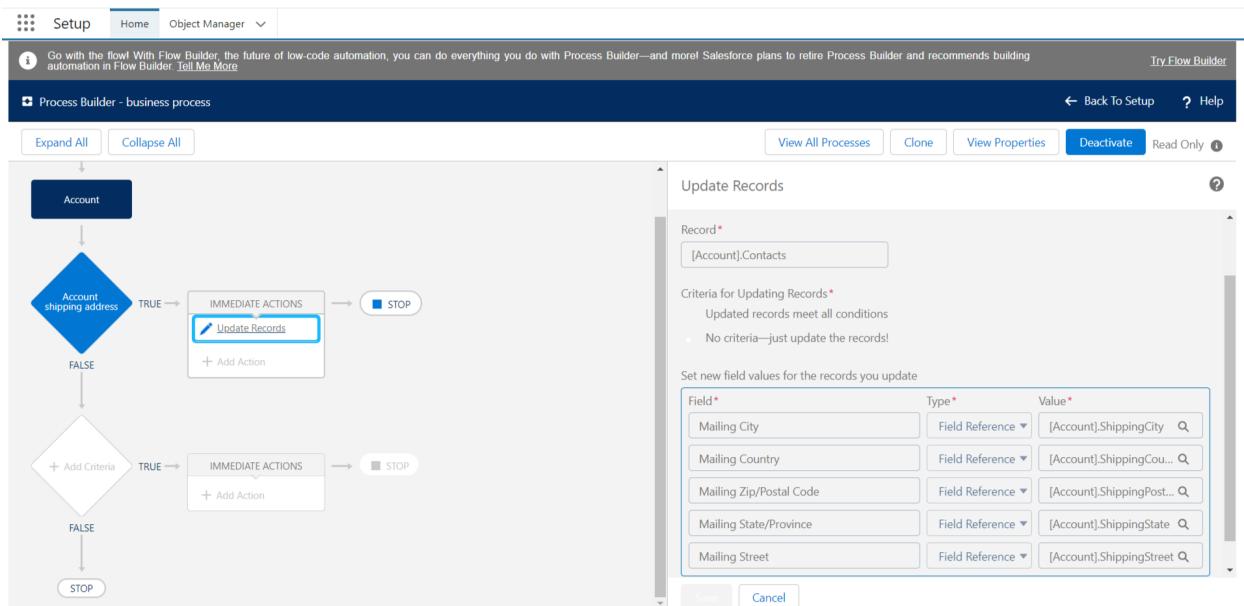
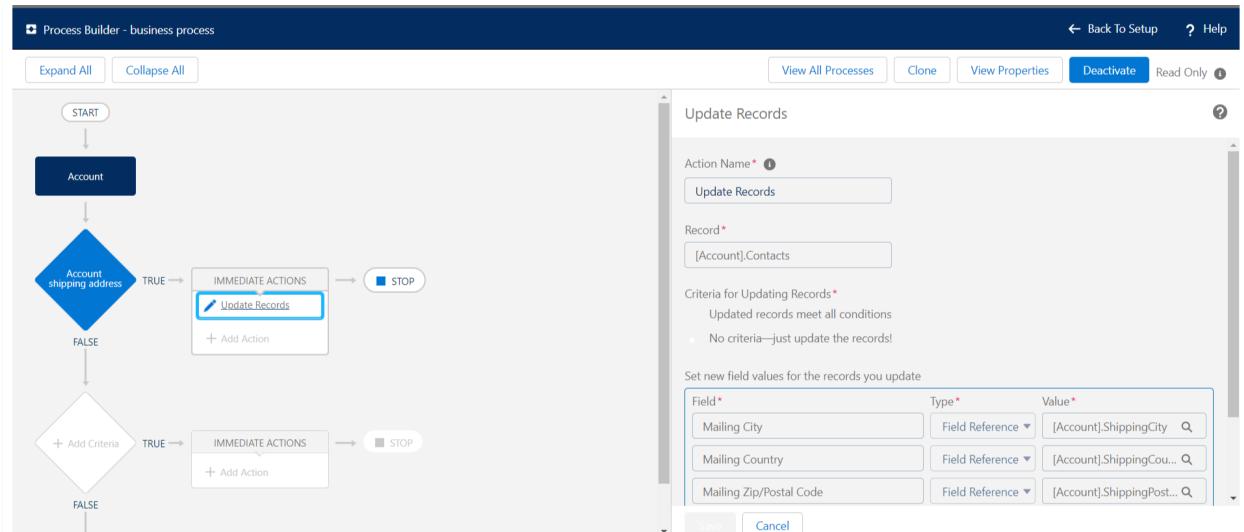
Salesforce Flow

Automate Simple Business Processes with Process Builder:

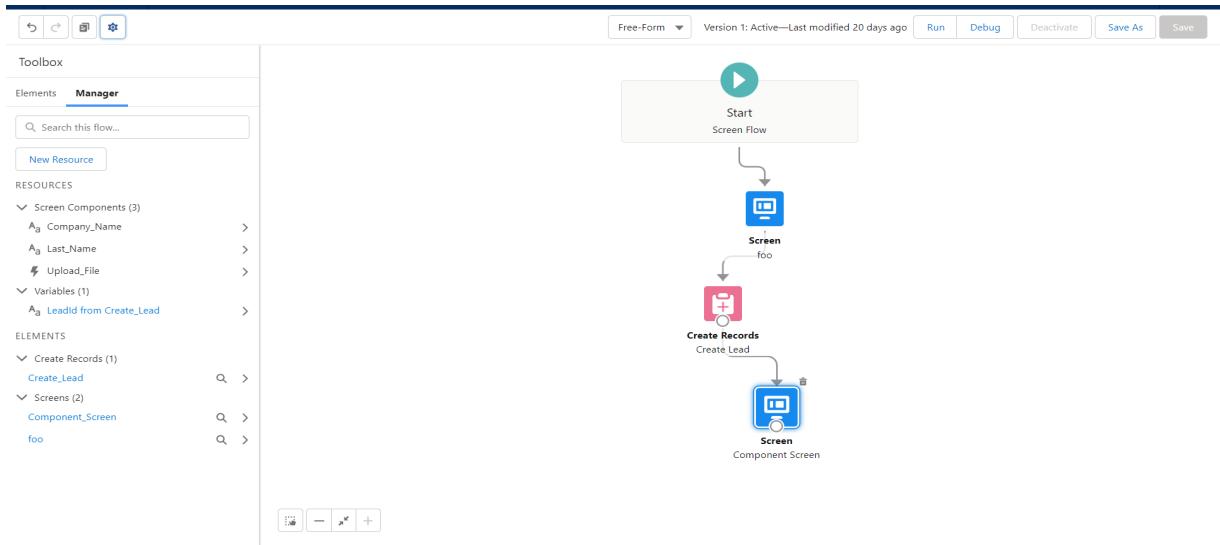
The screenshot shows the Salesforce Process Builder interface with the following details:

- Header:** Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder. [Tell Me More](#) Try Flow Builder
- Section:** Process Builder - business process
- Flow Diagram:**
 - Start with an **Account** object.
 - An **Account shipping address** decision diamond:
 - TRUE**: Immediate actions: **Update Records**.
 - FALSE**: An **+ Add Criteria** decision diamond:
 - TRUE**: Immediate actions: **+ Add Action**.
 - FALSE**: STOP.
- Right Panel: Define Criteria for this Action Group**
 - Criteria for Executing Actions:**
 - Conditions are met
 - Formula evaluates to true
 - No criteria—just execute the actions!
 - Set Conditions:**

Field*	Operator*	Type*	Value*
[Account].ShippingAddress	Is changed	Boolean	True
[Account].ShippingAddress	Is changed	Boolean	True
[Account].ShippingAddress	Is changed	Boolean	True
[Account].ShippingAddress	Is changed	Boolean	True
[Account].ShippingAddress	Is changed	Boolean	True



Guide Users Through Your Business Processes with Flow Builder:



Edit Create Records

Create Salesforce records using values from the flow.

Create Lead (Create_Lead)

How Many Records to Create

- One
- Multiple

How to Set the Record Fields

- Use all values from a record
- Use separate resources, and literal values

Create a Record of This Object

Object: Lead

Set Field Values for the Lead

Field	Value
Company	← A_a Company_Name X
LastName	← A_a Last_Name X

Cancel Done

Customize How Records Get Approved with Approvals:

The screenshot shows the Salesforce Setup interface for 'Approval Processes'. A process named 'Approve New Account' is displayed. The process details include:

- Process Name:** Approve New Account
- Unique Name:** Approve_New_Account
- Description:** Next Automated Approver Determined By
- Entry Criteria:** (Account: Type EQUALS Prospect) AND (Account: Employees GREATER THAN 500)
- Record Editability:** Administrator ONLY
- Approval Assignment Email Template:** Marketing_Product_Inquiry_Response
- Initial Submitters:** Account Owner
- Created By:** Sravya Reddy, 6/8/2022, 5:15 AM
- Modified By:** Sravya Reddy, 6/8/2022, 5:20 AM

Initial Submission Actions:

Action	Type	Description
Record Lock		Lock the record from being edited
Field Update		Account_Type To Pending

Approval Steps:

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit	1	Step 1			User Sravya Reddy	Final Rejection

Final Approval Actions:

Action	Type	Description
--------	------	-------------

The screenshot shows the same Approval Processes page with a modified process definition. The process now includes:

Initial Submission Actions:

Action	Type	Description
Record Lock		Lock the record from being edited
Field Update		Account_Type To Pending

Approval Steps:

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit	1	Step 1			User Sravya Reddy	Final Rejection

Final Approval Actions:

Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit Remove	Field Update	Account_Type To Customer

Final Rejection Actions:

Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit Remove	Field Update	Account_Type To Prospect

Recall Actions:

Action	Type	Description
Record Lock		Unlock the record for editing

Leads & Opportunities for Lightning Experience

Create and Convert Leads as Potential Customers:

The image displays two screenshots of the Salesforce Lightning Experience interface, specifically for a Lead record.

Screenshot 1: Activity View

This screenshot shows the Activity tab of a Lead record for "Mr. Chantal Smith". The lead's information includes Title (Mr.), Company (Get Cloudy South), and Phone (2). The status bar at the top indicates "Working - Contacted". The activity section shows a task titled "Call Ms. Smith" with a due date set to "No due date". A message box says "You have an upcoming task". Below this, a note states "No past activity. Past meetings and tasks marked as done show up here."

Screenshot 2: Details View

This screenshot shows the Details tab of the same Lead record. The lead's information is identical. The status bar at the top indicates "Working - Contacted". The details section lists various contact information fields: Lead Owner (Sravya Reddy), Name (Mr. Chantal Smith), Company (Get Cloudy South), Title, Lead Source, Industry, Annual Revenue, and Region. To the right, the Related section displays a message: "We found no potential duplicates of this Lead." and a "Campaign History (0)" section.

Process Automation Specialist

Automate Leads

The screenshot shows the 'Lead Validation Rule' setup page in the Salesforce Object Manager. The left sidebar lists various configuration options for the 'Lead' object. The main content area displays the details of a specific validation rule:

Validation Rule Detail	
Rule Name	CheckifUSOrNot
Error Condition Formula	OR(NOT(CONTAINS("AL AK AZ AR CA CO CT DE DC FL GA HI ID IL IN IA KS KY LA ME MD MA MI MN MS MO MT NE NV NH NJ NM NY NC ND OH OK OR PA RI SC SD TN TX UT VT VA WA WV WI WY", State)), LEN(State) <> 2, NOT(OR(Country = "US", Country = "United States", ISBLANK(Country))))
Error Message	We're a US based company and for now we can do business in the United States. I can't do anything Error Location Top of Page
Description	
Created By	Sravya Reddy, 6/23/2022, 6:06 AM
Modified By	Sravya Reddy, 6/23/2022, 6:06 AM

The screenshot shows the 'Assembly System Sales' queue setup page in the Salesforce Object Manager. The left sidebar lists various queue-related settings. The main content area displays the details of the queue:

Queue	
Label	Assembly System Sales
Queue Email	
Supported Objects	Lead
Created By	Sravya Reddy, 6/23/2022, 6:11 AM
Modified By	Sravya Reddy, 6/23/2022, 6:11 AM

Below the table, a message states: "No members." with a "View All Users" button.

Setup Home Object Manager

Queues

Queue Rainbow Sales

Label Rainbow Sales Queue Name Rainbow_Sales

Queue Email Send Email to Members

Supported Objects Lead

Created By Sravya Reddy, 6/23/2022, 6:10 AM Modified By Sravya Reddy, 6/23/2022, 6:10 AM

No members

View All Users

Help for this Page

Setup Home Object Manager

Lead Updates

Feature Settings

Marketing

Lead Assignment Rules

Lead Auto-Response Rules

Lead Processes

Lead Settings

LinkedIn Lead Gen

Lead Gen Fields

LinkedIn Accounts

Web-to-Lead

Sales

Social Accounts and Contacts Settings

User Engagement

Learning Paths

Rule Detail

Rule Name Rainbow Sales Active ✓

Created By Sravya Reddy, 6/23/2022, 6:13 AM Modified By Sravya Reddy, 6/23/2022, 6:17 AM

Rule Entries

Action	Order	Criteria	Assign To	Email
Edit Del	1	Lead: Lead Source EQUALS Web	Rainbow Sales	<input type="checkbox"/>
Edit Del	2	Lead: Lead Source NOTEQUALTO Web	Assembly_System_Sales	<input type="checkbox"/>

Help for this Page

The screenshot shows the Salesforce Setup interface under the Object Manager section for the Account object. A custom field named "Number of deals" has been created. The field information includes:

Field Label	Number of deals
Field Name	Number_of_deals
API Name	Number_of_deals_c
Description	(empty)
Help Text	(empty)
Data Owner	(empty)
Field Usage	(empty)
Data Sensitivity Level	(empty)
Compliance Categorization	(empty)
Created By	Srayya Reddy, 6/23/2022, 6:21 AM
Modified By	Srayya Reddy, 6/23/2022, 6:21 AM

Under Roll-Up Summary Options, the settings are:

Data Type	Roll-Up Summary
Summarized Object	Opportunity
Filter Criteria	(empty)

The summary type is set to COUNT.

Automate Accounts

Setup | Home | Object Manager

SETUP > OBJECT MANAGER
Account

Details

Fields & Relationships

- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts
- List View Button Layout
- Hierarchy Columns
- Scoping Rules
- Triggers

Last won deal date

Back to Account Fields

Custom Field Definition Detail

Edit | Set Field-Level Security | View Field Accessibility | Where is this used?

Field Information

Field Label	Last won deal date	Object Name	Account
Field Name	Last_won_deal_date		
API Name	Last_won_deal_date_c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Sravya Reddy, 6/23/2022, 6:46 AM	Modified By	Sravya Reddy, 6/23/2022, 6:46 AM

Roll-Up Summary Options

Data Type	Roll-Up Summary	Summary Type	MAX
Summarized Object	Opportunity		
Field to Aggregate	Opportunity.Close Date		
Filter Criteria	Stage EQUALS Closed Won		

Help for this Page

Setup | Home | Object Manager

SETUP > OBJECT MANAGER
Account

Details

Fields & Relationships

- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts
- List View Button Layout
- Hierarchy Columns
- Scoping Rules
- Triggers

Amount of won deals

Back to Account Fields

Custom Field Definition Detail

Edit | Set Field-Level Security | View Field Accessibility | Where is this used?

Field Information

Field Label	Amount of won deals	Object Name	Account
Field Name	Amount_of_won_deals		
API Name	Amount_of_won_deals_c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Sravya Reddy, 6/23/2022, 6:52 AM	Modified By	Sravya Reddy, 6/23/2022, 6:52 AM

Roll-Up Summary Options

Data Type	Roll-Up Summary	Summary Type	SUM
Summarized Object	Opportunity		
Field to Aggregate	Opportunity.Amount		
Filter Criteria	Stage EQUALS Closed Won		

Sales | Approve | Recen | Get Cr | Lightning | Approve | Lightning | Approve | Lightning | Ac x | Lightning | Lead | YouTube | Apex | forums | Savy | Process +

creative-narwhal-wjwc29-dev-ed.lightning.force.com/lightning/setup/ObjectManager/Account/FieldsAndRelationships/00N5i00000AvWL/view

The screenshot shows the 'Custom Field Definition Detail' page for the 'Deal win percent' field on the Account object. The field is defined as a formula type with a decimal places value of 2. The formula is: Number_of_won_deals__c/Number_of_deals__c. The field has a label of 'Deal win percent' and an API name of 'Deal_win_percent__c'. It is active and was created by Sravya Reddy on June 23, 2022, at 6:50 AM.

The screenshot shows the 'Validation Rule Detail' page for the 'ValidationForBilling' rule on the Account object. The rule is active and was created by Sravya Reddy on June 23, 2022, at 7:04 AM. The validation formula is: OR(NOT(CONTAINS("AL,AK,AZ,AR,CA,CO,CT,DE,DC,FL,GA,HI,ID,IL,IN,IA,KY,LA,ME,MD,MA,MI,MN,MS,MO,MT,NE,NV,NH,NJ,NM,NY,NC,ND,OH,OK,OR,PA,RI,SC,SD,TN,TX,UT,VT,VA,WA,WV,WI,WY", BillingState)), LEN(BillingState) < 2, NOT(OR(BillingCountry = "US", BillingCountry = "United States", ISBLANK(BillingCountry))), NOT(CONTAINS("AL,AK,AZ,AR,CA,CO,CT,DE,DC,FL,GA,HI,ID,IL,IN,IA,KY,LA,ME,MD,MA,MI,MN,MS,MO,MT,NE,NV,NH,NJ,NM,NY,NC,ND,OH,OK,OR,PA,RI,SC,SD,TN,TX,UT,VT,VA,WA,WV,WI,WY", ShippingState)), LEN(ShippingState) < 2, NOT(OR(ShippingCountry = "US", ShippingCountry = "United States", ISBLANK(ShippingCountry))))). The error message is: you have to make sure nobody makes a new account unless the shipping and billing fields are valid US state abbreviations ,and the country field is either blank or US,USA or United States.

Setup | Home | Object Manager | Account

Account Validation Rule

[Back to Account Validation Rules](#)

Validation Rule Detail

Rule Name	ValidationForType	Active	<input type="checkbox"/>
Error Condition Formula	ISCHANGED(Name) && (OR(ISPICKVAL(Type, 'Customer - Direct'), ISPICKVAL(Type, 'Customer - Channel'))))	Error Message	the account type is either 'Customer-Direct' or 'Customer-channel' we don't want anybody to change
Description	Error Location Account Name		
Created By	Sravya Reddy, 6/23/2022, 7:14 AM		
Modified By	Sravya Reddy, 6/23/2022, 10:16 AM		

[Edit](#) [Clone](#)

Help for this Page 

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Hierarchy Columns
Scoping Rules
Triggers

Setup | Home | Object Manager | Account

Account Custom Field Call for Service

[Back to Account Fields](#)

Custom Field Definition Detail

Edit	Set Field-Level Security	View Field Accessibility	Where is this used?
Field Information			
Field Label	Call for Service	Object Name	Account
Field Name	Call_for_Service		
API Name	Call_for_Service__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Sravya Reddy, 6/23/2022, 6:58 AM		
Modified By	Sravya Reddy, 6/23/2022, 6:58 AM		
Formula Options			
Data Type	Formula		
IF(DATE(YEAR(Last_won_deal_date__c)+2,MONTH(Last_won_deal_date__c),DAY(Last_won_deal_date__c))<= TODAY(),"YES","NO")			

Help for this Page 

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Hierarchy Columns
Scoping Rules
Triggers

Create Robot Setup Object

Setup | Home | Object Manager

Robot Setup

- Details**
- Fields & Relationships
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Restriction Rules
- Scoping Rules
- Triggers
- Flow Triggers
- Validation Rules

Details

Description	
API Name	Robot_Setup_c
Custom	✓
Singular Label	Robot Setup
Plural Label	Robot Setups
Enable Reports	
Track Activities	
Track Field History	
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

[Edit](#) [Delete](#)

<https://creative-narwhal-wyw29-dev-ed.lightning.force.com/one/one.app#/setup/ObjectManager/015i000000V7C1/FieldsAndRelationships/view>

Setup | Home | Object Manager

Robot Setup

- Details**
- Fields & Relationships**
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Restriction Rules
- Scoping Rules
- Triggers
- Flow Triggers
- Validation Rules

Robot Setup Custom Field Date

[Back to Robot Setup](#) [Help for this Page](#)

Custom Field Definition Detail
[Edit](#) [Set Field-Level Security](#) [View Field Accessibility](#) [Where is this used?](#)

Field Information	
Field Label	Date
Field Name	Date
API Name	Date_c
Description	
Help Text	
Data Owner	
Field Usage	
Data Sensitivity Level	
Compliance Categorization	
Created By	Scavya Reddy, 6/23/2022, 8:32 AM
Modified By	Scavya Reddy, 6/23/2022, 8:32 AM

General Options	
Required	<input type="checkbox"/>
Default Value	

Validation Rules	Validation Rules Help
No validation rules defined.	New

[^ Back To Top](#) [Always show me more records per related list](#)

Setup Home Object Manager

SETUP > OBJECT MANAGER Robot Setup

Robot Setup Custom Field Notes Back to Robot Setup Validation Rules Help for this Page

Custom Field Definition Detail Edit Set Field-Level Security View Field Accessibility Where is this used?

Field Information Field Label Notes Field Name Notes API Name Notes_c Description Help Text Data Owner Field Usage Data Sensitivity Level Compliance Categorization Created By Sravya Reddy, 6/23/2022, 8:34 AM Modified By Sravya Reddy, 6/23/2022, 8:34 AM Object Name Robot Setup Data Type Text

General Options Required Unique Case Sensitive External ID Default Value

Text Options Length 100

Fields & Relationships Details Page Layouts Lightning Record Pages Buttons, Links, and Actions Compact Layouts Field Sets Object Limits Record Types Related Lookup Filters Restriction Rules Scoping Rules Triggers Flow Triggers Validation Rules

Setup Home Object Manager

SETUP > OBJECT MANAGER Robot Setup

Robot Setup Custom Field Day of the Week Back to Robot Setup Help for this Page

Custom Field Definition Detail Edit Set Field-Level Security View Field Accessibility Where is this used?

Field Information Field Label Day of the Week Field Name Day_of_the_Week API Name Day_of_the_Week_c Description Help Text Data Owner Field Usage Data Sensitivity Level Compliance Categorization Created By Sravya Reddy, 6/23/2022, 8:44 AM Modified By Sravya Reddy, 6/23/2022, 8:44 AM Object Name Robot Setup Data Type Formula

Formula Options

```
CASE(weekday(Date__c),  
1."Sunday",  
2."Monday",  
3."Tuesday",  
4."Wednesday",  
5."Thursday",  
6."Friday",  
7."Saturday",  
Text(weekday(Date__c)))
```

Fields & Relationships Details Page Layouts Lightning Record Pages Buttons, Links, and Actions Compact Layouts Field Sets Object Limits Record Types Related Lookup Filters Restriction Rules Scoping Rules Triggers Flow Triggers Validation Rules

The screenshot shows the Salesforce Setup interface under the Object Manager. A custom field named "Robot_Setup" has been created on the Opportunity object. The field is of type Master-Detail, related to the "Robot_Setups" object. It has a field label of "Opportunity" and a field name of "Opportunity". The API name is "Opportunity__c". The field is described as "Help Text" and owned by "Data Owner". The field usage is "Field Usage". The data sensitivity level is "Data Sensitivity Level". The compliance categorization is "Compliance Categorization". The field was created by "Sravya Reddy" on 6/23/2022 at 8:38 AM and modified by "Sravya Reddy" on the same date at 8:38 AM.

Create Sales Process and Validate Opportunities

The screenshot shows the Salesforce Setup interface under the Sales Processes section. An opportunity stage named "RB Robotics Sales Process" is being configured. The stage is part of the "Sales Process" named "RB Robotics Sales Process". The stage has a namespace prefix and a description. The available values for the stage include "Prospecting (Open, 10%, Pipeline)", "Qualification (Open, 10%, Pipeline)", "Proposal/Price Quote (Open, 75%, Pipeline)", "Negotiation/Review (Open, 90%, Pipeline)", "Closed Won (Closed/Won, 100%, Closed)", "Closed Lost (Closed/Lost, 0%, Omitted)", and "Awaiting Approval (Open, 100%, Pipeline)". The selected values for the stage include "Needs Analysis (Open, 20%, Pipeline)", "Value Proposition (Open, 50%, Pipeline)", "Id: Decision Makers (Open, 60%, Pipeline)", and "Perception Analysis (Open, 70%, Pipeline)".

Setup | Home | Object Manager

Opportunity Validation Rule

[Back to Opportunity Validation Rules](#)

Validation Rule Detail

Rule Name	ValidationForHighValue	Active	<input type="checkbox"/>
Error Condition Formula	<code> (Amount>1000 && Approved__c = false && ISPICKVAL(StageName, "Closed Won"))</code> , true, false)		
Error Message	cannot have a validation rule	Error Location	Top of Page
Description		Modified By	Savya Reddy, 6/24/2022, 2:35 AM
Created By	Savya Reddy, 6/23/2022, 9:09 AM		

[Edit](#) [Clone](#)

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Scoping Rules
Triggers
Flow Triggers

Automate Opportunities

Setup | Home | Object Manager

Q approv

v Data
Mass Transfer Approval Requests
v Process Automation
Approval Processes

Approval Processes
Opportunity: Automate Opportunity Process
[Back to Approval Process List](#)

Process Definition Detail

Process Name	Automate_Opportunity_Process	Active	<input checked="" type="checkbox"/>
Unique Name	Automate_Opportunity_Process	Next Automated Approver Determined By	Manager of Record Submitter
Description			
Entry Criteria	Opportunity: Amount GREATER OR EQUAL 100000		
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests	<input type="checkbox"/>
Approval Assignment Email Template	SALES_Opportunity_Needs_Approval		
Initial Submitters	User:Nushi Davoud		
Created By	Savya Reddy, 6/24/2022, 2:29 AM	Modified By	Savya Reddy, 6/24/2022, 2:31 AM

[Edit](#) [Clone](#) [Deactivate](#)

Initial Submission Actions

Action Type	Description
Record Lock	Lock the record from being edited

[Add Existing](#) [Add New](#)

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions	1	Step 1			User:Nushi Davoud	Final Rejection

[Show Actions](#) [Edit](#)

Final Approval Actions

Action	Type	Description

[Add Existing](#) [Add New](#)

Setup Home Object Manager

Search: approv

v Data Mass Transfer Approval Requests

v Process Automation Approval Processes

Didn't find what you're looking for? Try using Global Search.

SETUP Approval Processes

Initial Submitters: User_Nushi Davoud Created By: Savya Reddy, 6/24/2022, 2:29 AM Modified By: Savya Reddy, 6/24/2022, 2:31 AM

Initial Submission Actions		Add Existing Add New
Action	Type	Description
	Record Lock	Lock the record from being edited

Approval Steps					
Action	Step Number	Name	Description	Criteria	Assigned Approver
Show Actions Edit	1	Step 1			User_Nushi Davoud
Reject Behavior: Final Rejection					

Final Approval Actions		Add Existing Add New
Action	Type	Description
Edit	Record Lock	Lock the record from being edited

Final Rejection Actions		Add Existing Add New
Action	Type	Description
Edit	Record Lock	Unlock the record for editing

Recall Actions		Add Existing Add New
Action	Type	Description
	Record Lock	Unlock the record for editing

[Back To Top](#) Always show me **more** records per related list

Setup Home Object Manager

Search: approv

v Data Mass Transfer Approval Requests

v Process Automation Approval Processes

Didn't find what you're looking for? Try using Global Search.

SETUP Approval Processes

Opportunity: prospect

Help for this Page

Process Definition Detail

Process Name:	prospect	Active: <input checked="" type="checkbox"/>
Unique Name:	prospect	Next Automated Approver Determined By: Manager of Record Submitter
Description:		
Entry Criteria:	(Opportunity: Amount GREATER THAN 100000) AND (Opportunity: Stage EQUALS Negotiation/Review)	
Record Editability:	Administrator ONLY	Allow Submitters to Recall Approved Requests: <input type="checkbox"/>
Approval Assignment Email Template:	SALES_Opportunity_NeedsApproval	
Initial Submitters:	User_Nushi Davoud, Opportunity Owner, User_Savya Reddy	
Created By:	Savya Reddy, 6/24/2022, 1:23 AM	Modified By: Savya Reddy, 6/24/2022, 1:47 AM

Initial Submission Actions		Add Existing Add New
Action	Type	Description
	Record Lock	Lock the record from being edited
	Field Update	update

Approval Steps					
Action	Step Number	Name	Description	Criteria	Assigned Approver
Show Actions Edit	1	step1			User_Nushi Davoud
Show Actions Edit	2	Step 1			Manually Chosen
Reject Behavior: Final Rejection					

Search: approv

- Mass Transfer Approval Requests
- Process Automation
- Approval Processes**

Didn't find what you're looking for? Try using Global Search.

SETUP Approval Processes

Action	Type	Description
Record Lock	Lock the record from being edited	
Field Update	update	

Approval Steps [1]

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit	1	step1			User:Nushi Dayoud	Final Rejection
Show Actions Edit	2	Step 1			Manually Chosen	Final Rejection

Final Approval Actions [1]

Action	Type	Description
Edit	Record Lock	Lock the record from being edited
Edit Remove	Field Update	w/in deal

Final Rejection Actions [1]

Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit Remove	Field Update	negotiate

Recall Actions [1]

Action	Type	Description
Record Lock		Unlock the record for editing

Always show me [more](#) records for related list.

Setup Home Object Manager

Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder. [Tell Me More](#) Try Flow Builder

Process Builder - OpportunityProcess

Expand All Collapse All

View All Processes Clone View Properties Deactivate Read Only

```

graph TD
    START([START]) --> Opportunity[Opportunity]
    Opportunity --> CustomAccount{CustomAccount}
    CustomAccount -- TRUE --> IMMEDIATE_ACTIONS1["IMMEDIATE ACTIONS<br>emailAlerts"]
    IMMEDIATE_ACTIONS1 --> STOP1([STOP])
    CustomAccount -- FALSE --> ProspectAccount{ProspectAccount}
    ProspectAccount -- TRUE --> IMMEDIATE_ACTIONS2["IMMEDIATE ACTIONS<br>emailAlerts<br>show more"]
    IMMEDIATE_ACTIONS2 --> STOP2([STOP])
    ProspectAccount -- FALSE --> javascript:void(0);
  
```

Define Criteria for this Action Group

Criteria Name *

Criteria for Executing Actions *

- Conditions are met
- Formula evaluates to true
- No criteria—just execute the actions!

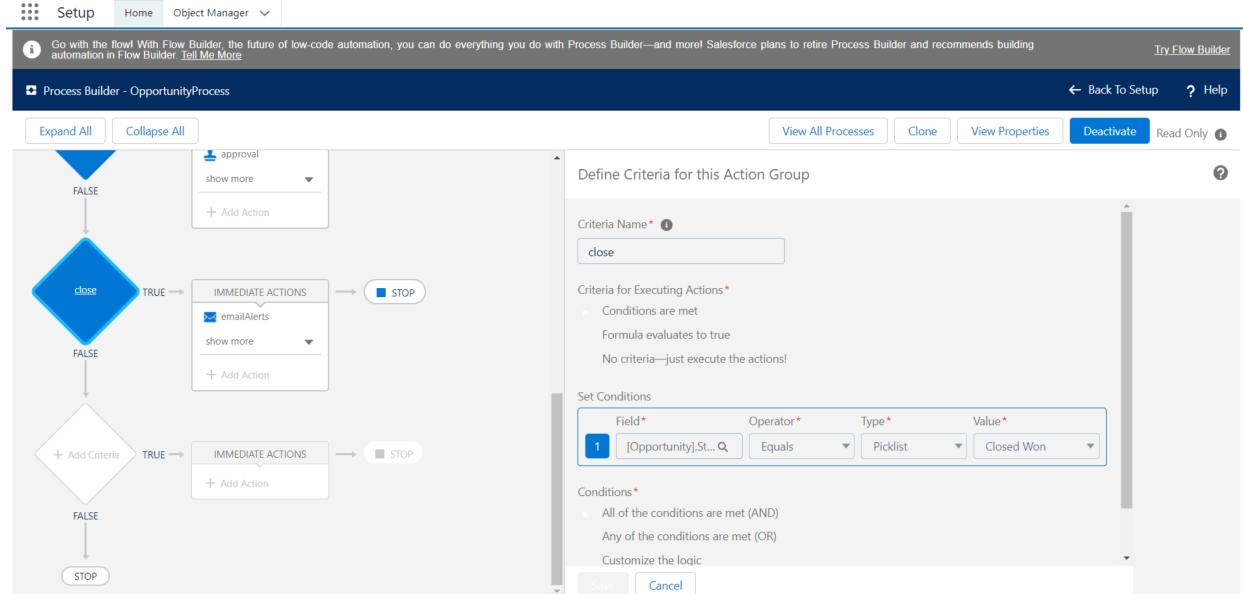
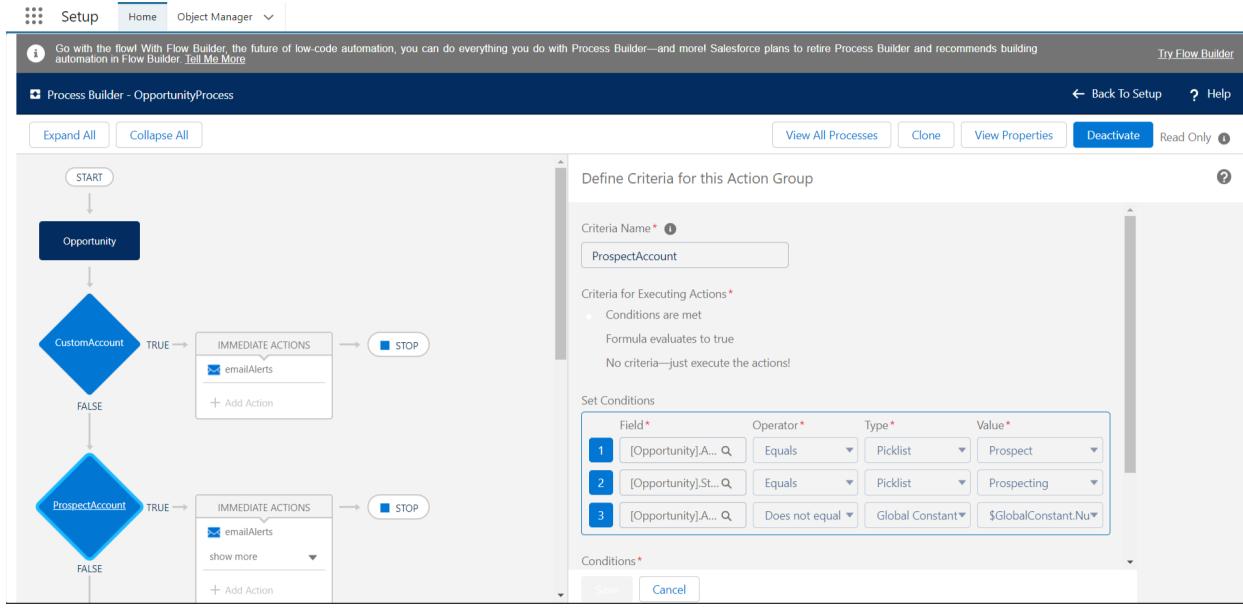
Set Conditions

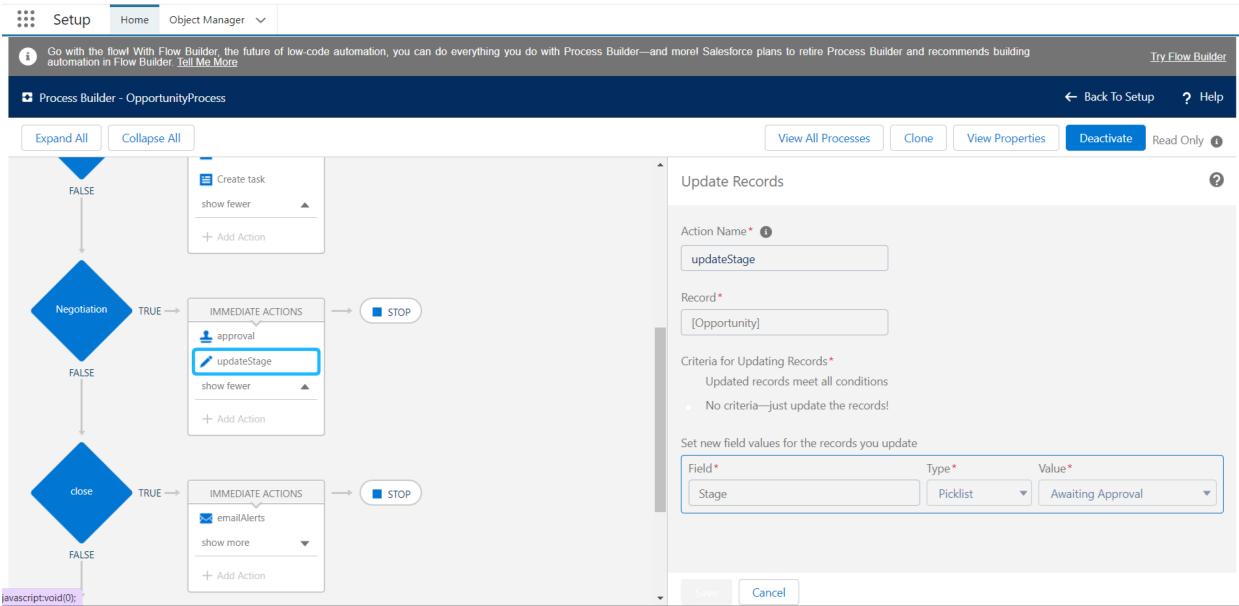
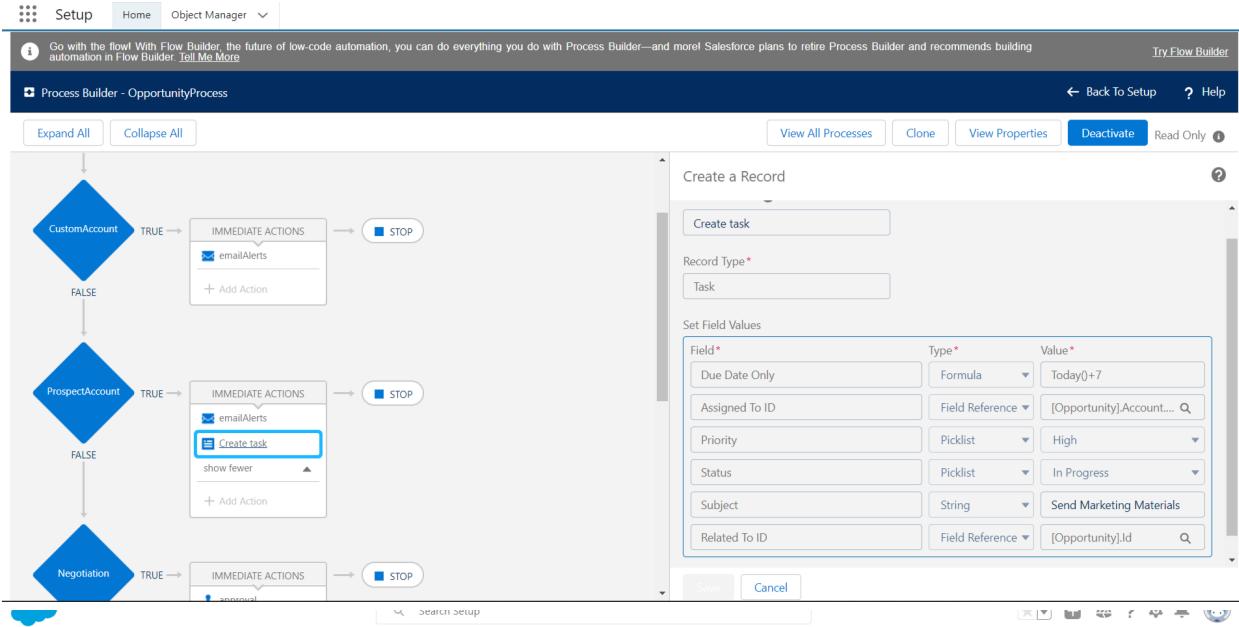
Field *	Operator *	Type *	Value *
[Opportunity].A... Q	Equals	Picklist	Customer - Direct
[Opportunity].A... Q	Does not equal	Global Constant	\$GlobalConstant.Nu

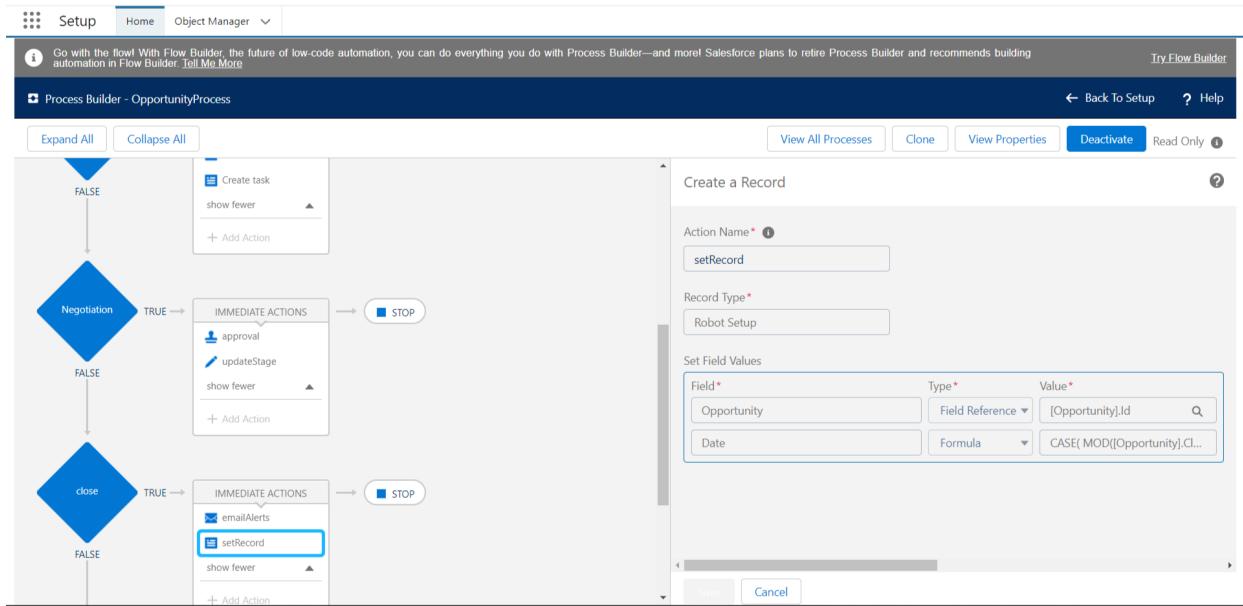
Conditions *

- All of the conditions are met (AND)

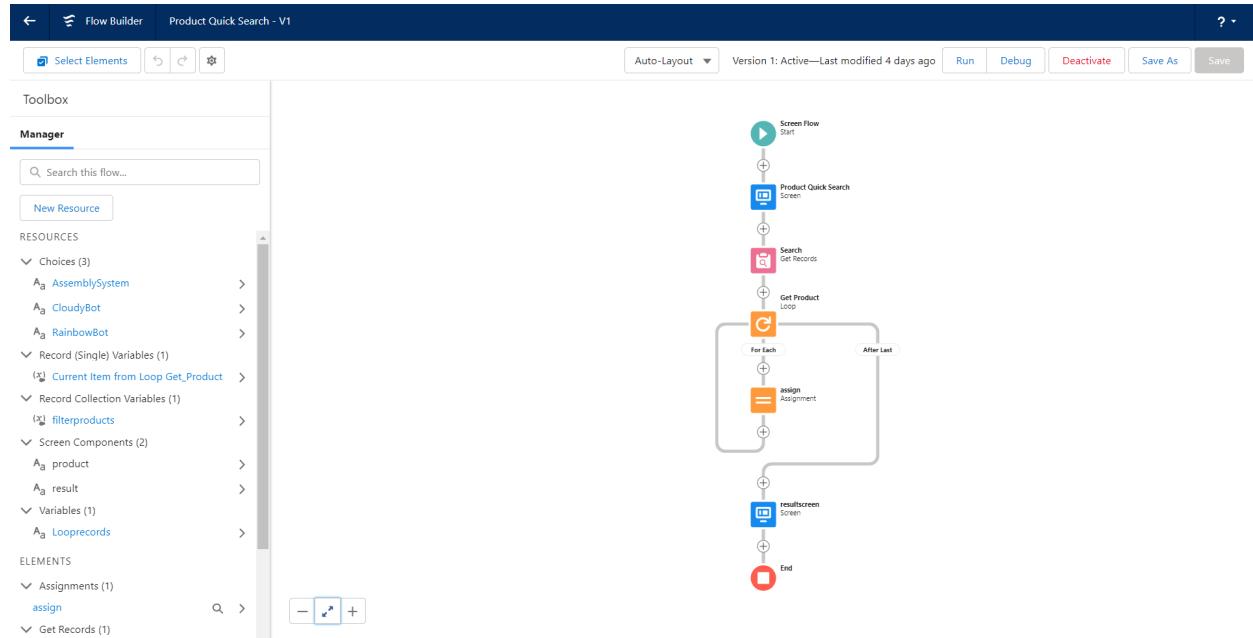
Save Cancel







Create flow for opportunities and Automate Setups



Edit Screen

Product Quick Search

product

- RainbowBot
- <em style="color: #51, 51, 51; background-color: #255, 255, 255; font-size: 16px; font-family: "Salesforce Sans"; -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, sans-serif;>CloudyBot
- <em style="color: #51, 51, 51; background-color: #255, 255, 255; font-size: 16px; font-family: "Salesforce Sans"; -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, sans-serif;>Assembly System

Pause Previous Finish

Screen Properties

Product Quick Search (Product_Quick_Search)

Configure Header Configure Footer

Get more on the AppExchange

Flow Builder Product Quick Search - V1

Edit Get Records

Find Salesforce records and store their field values in flow variables.

Search (Search)

Get Records of This Object

Object: Product

Filter Product Records

Condition Requirements: All Conditions Are Met (AND)

Field	Operator	Value
Name	Contains	A_a product

+ Add Condition

Sort Product Records

Sort Order: Not Sorted

⚠️ If you store only the first record, filter by a unique field, such as ID.

Cancel Done

Edit Get Records

How Many Records to Store
 Only the first record
 All records

How to Store Record Data
 Automatically store all fields
 Choose fields and let Salesforce do the rest
 Choose fields and assign variables (advanced)

To use the returned **Product** records in the flow, store their fields in variables.

Select Variable to Store Product Records
Record Collection
 filterproducts X

Select Product Fields to Store in Variable

Field

Field

When no records are returned, set specified variables to null.

Screen Flow

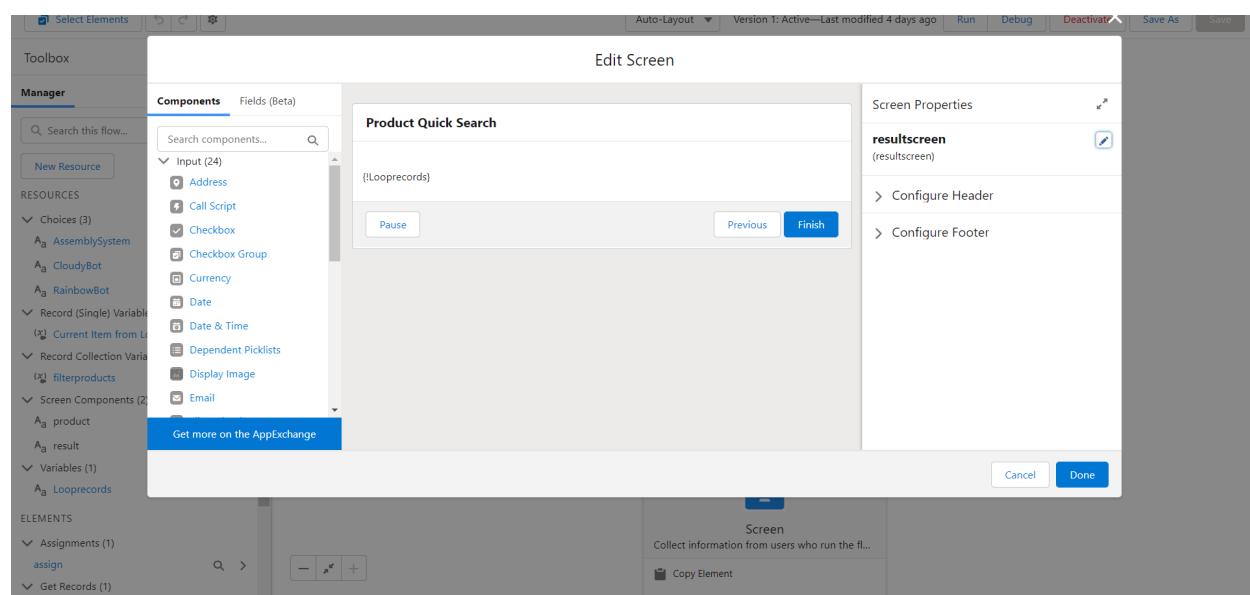
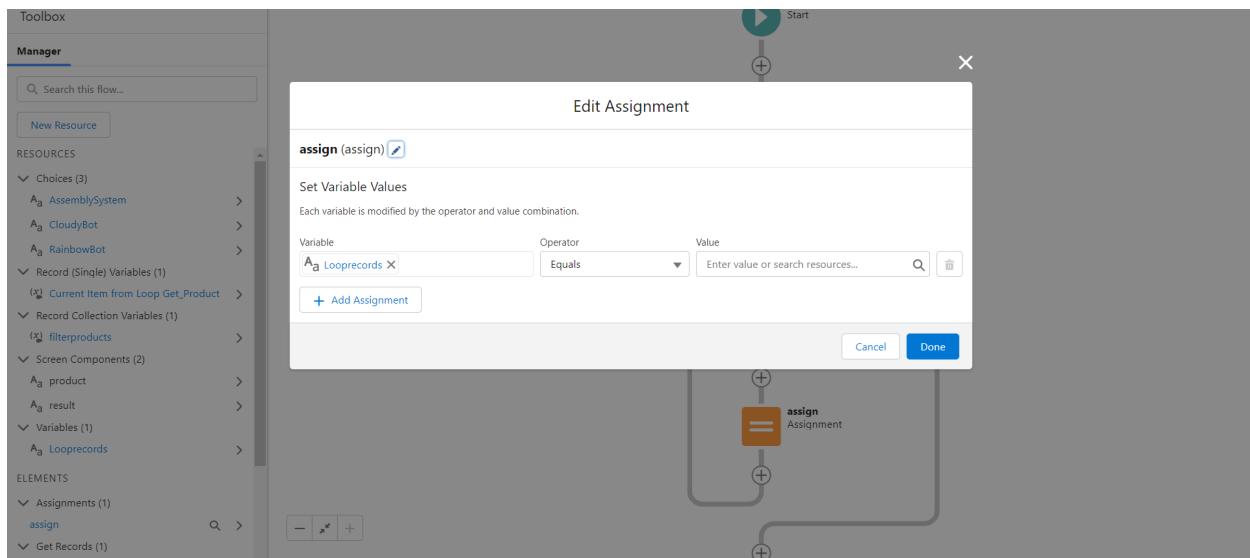
Start a loop path for iterating over items in a collection variable. For each iteration, the flow temporarily stores the item in the loop variable.

Get Product (Get_Product)

Select Collection Variable
Collection Variable
 filterproducts

Specify Direction for Iterating Over Collection
Direction
 First item to last item
 Last item to first item

To use the current item in other elements in the loop, use the API name of the Loop element. Example: if your flow iterates over accounts with a Loop element named "My_Account_Loop" you can reference the current item from that loop element. Just start typing "My_Account_Loop" and select "Current Item from Loop My_Account_Loop".



SETUP > OBJECT MANAGER

Opportunity

Lightning Page
Product_Quick_Search

Lightning Page Detail [Edit](#) [Clone](#) [Delete](#)

Help for this Page [?](#)

Information

Name	Description	Label
Product_Quick_Search		Product Quick Search

[Edit](#) [Clone](#) [Delete](#)

Assignments By App
No Assignments to display

Assignments By App, Record Type, and Profile
No Assignments to display

[Back To Top](#) Always show me [more](#) records per related list

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Scoping Rules
Triggers
Flow Triggers