Apex Specialist

Pre-work

Set Up Development Org

1. Create a new Trailhead Playground or Developer Edition Org for this superbadge. Using this org for any other reason might create problems when validating the challenge. If you choose to use a development org, make sure you deploy **My Domain** to all the users. The package you will install has some custom lightning components that only show when My Domain is deployed.
2. Install this unlocked package (package ID: 04t6g000008av9iAAA). This package contains metadata you'll use to complete this challenge. If you have trouble installing this package, follow the steps in the Install a Package or App to Complete a Trailhead Challenge help article.
3. Add picklist values Repair and Routine Maintenance to the **Type** field on the Case object.
4. Update the Case page layout assignment to use the **Case (HowWeRoll) Layout** for your profile.
5. Rename the tab/label for the Case tab to Maintenance Request.
6. Update the Product page layout assignment to use the **Product (HowWeRoll) Layout** for your profile.
7. Rename the tab/label for the Product object to Equipment.
8. Use App Launcher to navigate to the **Create Default Data** tab of the **How We Roll Maintenance** app. Click **Create Data** to generate sample data for the application.
9. Review the newly created records to get acquainted with the data model.

1. Automated Record Creation

Step 1: Go to the App Launcher -> Search How We Roll Maintenance -> click on Maintenance

Requests -> click on first case -> click Details -> change the type Repair to Routine Maintenance -> select Origin = Phone -> Vehicle = select Teardrop Camper-> save

Step 2: Feed -> Close Case -> save

Step 3: Go to the Object Manager -> Maintenance Request ->Field & Relationships ->New -

>Lookup

Relationship -> next -> select Equipment ->next -> Field Label = Equipment ->next->next

->next -> save

Step 4: Go to the developer console

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;
      }
   }
}
```

MaintenanceRequest.apxt

```
@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);
    }
}
```

After saving the code go back the How We Roll Maintenance , click on Maintenance Requests -> click on 2nd case -> click Details -> change the type Repair to Routine Maintenance -> select Origin = Phone -> Vehicle = select Teardrop Camper and then save.Finally,Feed -> Close Case -> save.

2.Synchronize Salesforce data with an external system

Setup -> Search in quick find box -> click Remote Site Settings -> Name = Warehouse URL , Remote Site URL = https://th-superbadge-apex.herokuapp.com.
Go to the developer console,

<u>WarehouseCalloutService.apxc</u>

```
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);
           }

        }
    }
}
```

After saving the code open execute anonymous window and run this method ,

```
System.enqueueJob(new WarehouseCalloutService());
```

3. Schedule synchronization using Apex code

Go to the developer console use below code ,

WarehouseSyncShedule.apxc

```
global with sharing class WarehouseSyncSchedule implements Schedulable{
   global void execute(SchedulableContext ctx){
      System.enqueueJob(new WarehouseCalloutService());
   }
}
```

4. Test automation logic

<u>MaintenanceRequestHelper.apxc</u>

```
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;
      }
   }
}
```

<u>MaintenanceRequestHelperTest.apxc</u>

```
@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);
    }
}
```

MaintenanceRequest.apxt :-

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

5. Test callout logic

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);
            }

        }
    }
}
```
WarehouseCalloutServiceTest.apxc
```
@isTest
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]);
    }
}
```

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":"55d66226726b611100aaf741","replacement":false,"quantity":5,"na
me":"Generator 1000
kW","maintenanceperiod":365,"lifespan":120,"cost":5000,"sku":"100003"}]');
        response.setStatusCode(200);
        return response;
    }
}
```

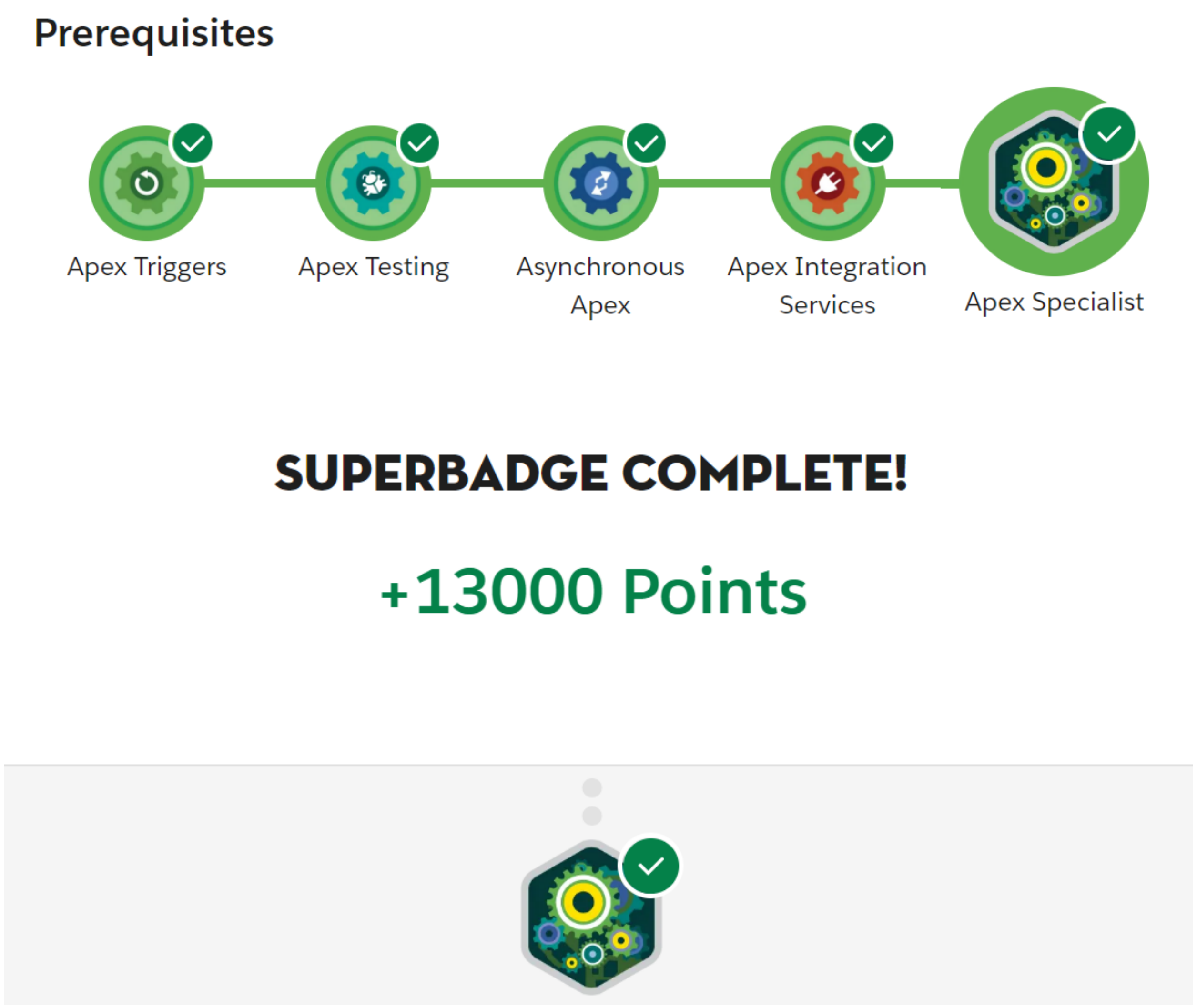
6. Test scheduling logic

<u>WarehouseSyncSchedule.apxc</u>

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

        WarehouseCalloutService.runWarehouseEquipmentSync();
    }
}
```

<u>WarehouseSyncScheduleTest.apxc</u>

```
@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 ');


    }
}
```

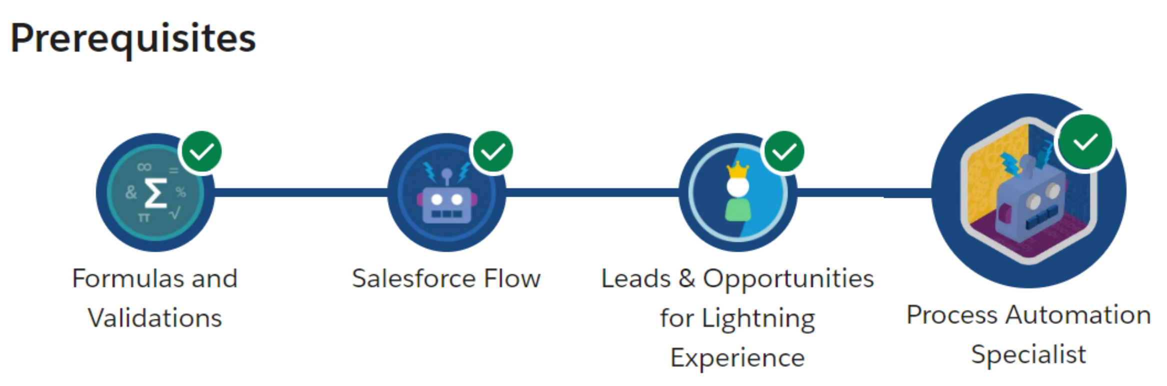


Prerequisites

SUPERBADGE COMPLETE!

+13000 Points

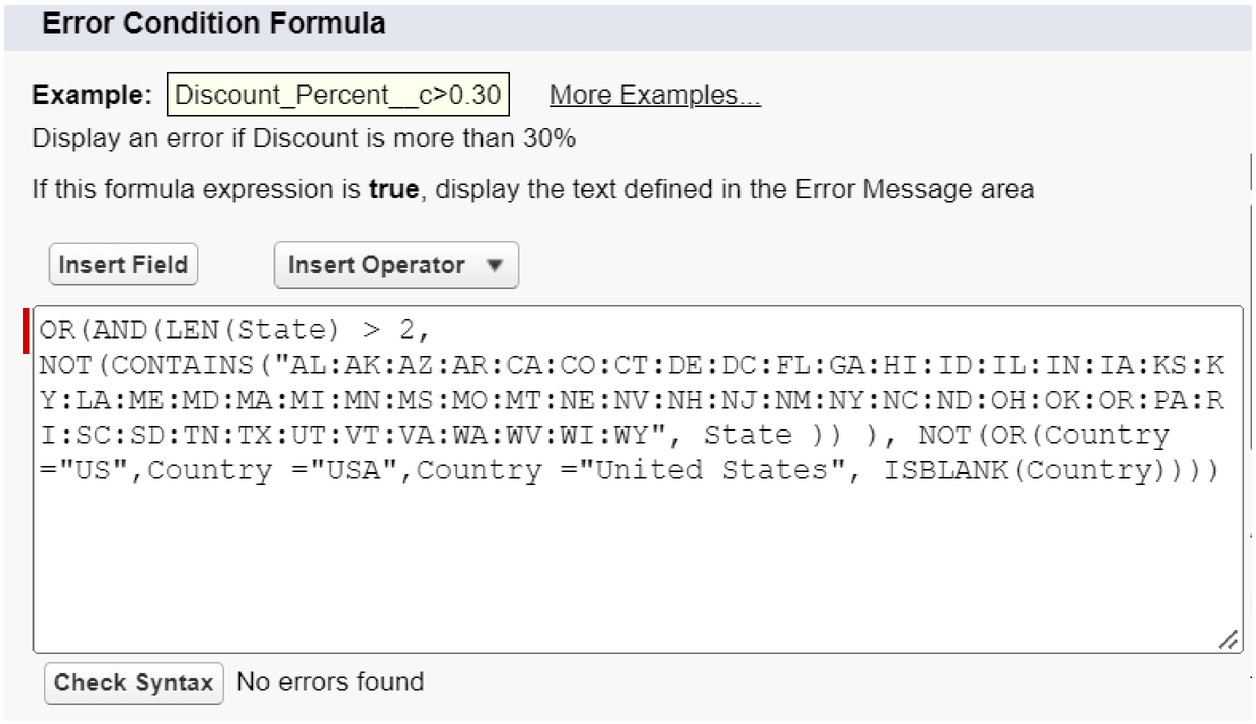
Process Automation Specialist


Prerequisites
Formulas and Validations
Salesforce Flow
Leads & Opportunities for Lightning Experience
Process Automation Specialist

1. Automate Leads

Create a new validation rule on Lead object.

Error condition formula



Error Condition Formula

Example: Discount_Percent__c>0.30 More Examples...

Display an error if Discount is more than 30%

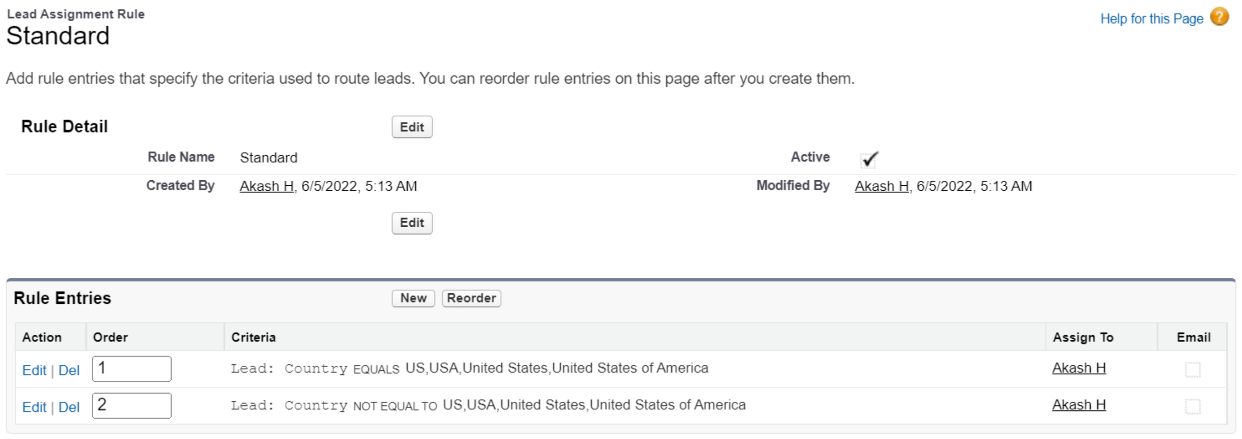
If this formula expression is **true**, display the text defined in the Error Message area

```
OR(AND(LEN(State) > 2,
NOT(CONTAINS("AL:AK:AZ:AR:CA:CO:CT:DE:DC:FL:GA:HI:ID:IL:IN:IA:KS:K
Y:LA:ME:MD:MA:MI:MN:MS:MO:MT:NE:NV:NH:NJ:NM:NY:NC:ND:OH:OK:OR:PA:R
I:SC:SD:TN:TX:UT:VT:VA:WA:WV:WI:WY", State )) ), NOT(OR(Country
="US",Country ="USA",Country ="United States", ISBLANK(Country))))
```

No errors found

Create Two Queues: 1. Rainbow Sales and 2. Assembly System Sales.

Create an lead assignment rule as shown below


Lead Assignment Rule
Standard

Add rule entries that specify the criteria used to route leads. You can reorder rule entries on this page after you create them.

Rule Detail

Rule Name	Standard	Active	✓
Created By	Akash H, 6/5/2022, 5:13 AM	Modified By	Akash H, 6/5/2022, 5:13 AM

Rule Entries

Action	Order	Criteria	Assign To	Email
Edit | Del	1	Lead: Country EQUALS US,USA,United States,United States of America	Akash H	☐
Edit | Del	2	Lead: Country NOT EQUAL TO US,USA,United States,United States of America	Akash H	☐

2. Automate Accounts

Create 4 roll up summary fields:

Number of deals-Count-Opportunity-None

Number of won deals-Count-Opportunity-Stage equals closed won

Last won deal date-Max-Opportunity:Close Date-Opportunity-Stage equals closed won

Amount of won deals-sum-Opportunity:Amont-Opportunity-Stage equals closed won

Create 2 formula fields

Deal win percent-Percent-2-(Number_of_won_deals__c/Number_of_deals___c)

Call for service-text-if(date(year(last_won_deal_date__c)+2,month(last_won_deal__c),day(last_won_deal_date__c))<=today(),"yes","no")

Create 2 validation rules

Validation rule: US_Address(something)

Error condtion formula: OR(AND(LEN(BillingState) > 2, 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", BillingState))

),AND(LEN(ShippingState) > 2, 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", ShippingState))

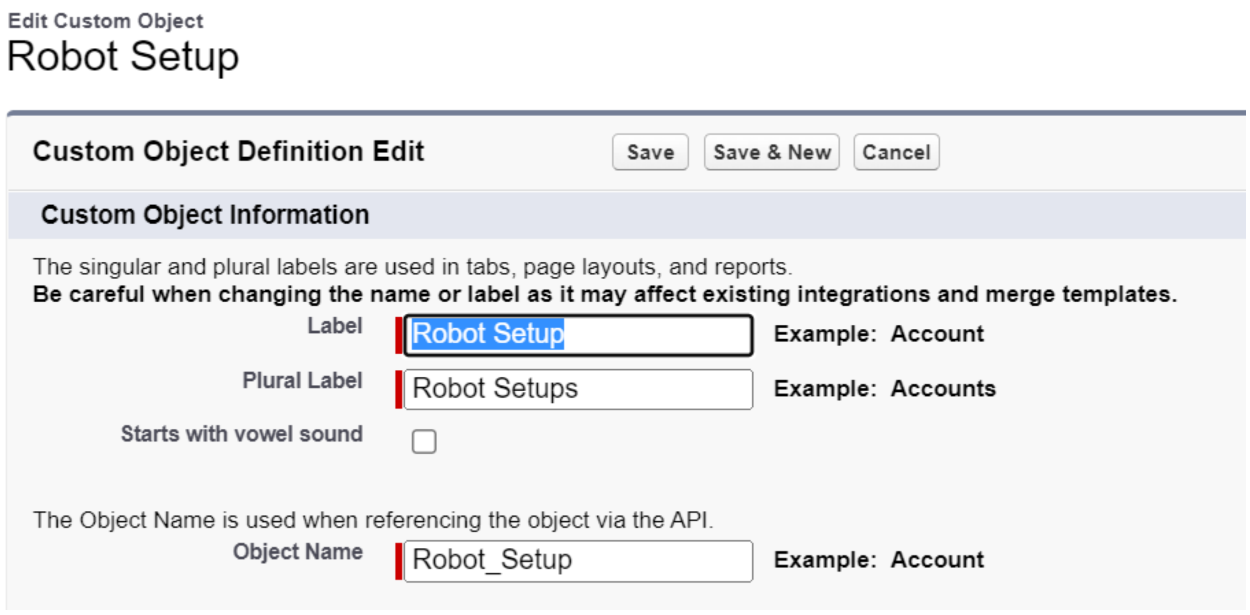
),NOT(OR(BillingCountry ="US",BillingCountry ="USA",BillingCountry ="United States", ISBLANK(BillingCountry))),

NOT(OR(ShippingCountry ="US",ShippingCountry ="USA",ShippingCountry ="United States", ISBLANK(ShippingCountry))))

Validation rule: Change Name(something)

ISCHANGED(Name) && (OR(ISPICKVAL(Type ,'Customer - Direct') ,ISPICKVAL(Type ,'Customer - Channel')))

3. Create Robot Setup



Edit Custom Object
Robot Setup

Custom Object Definition Edit

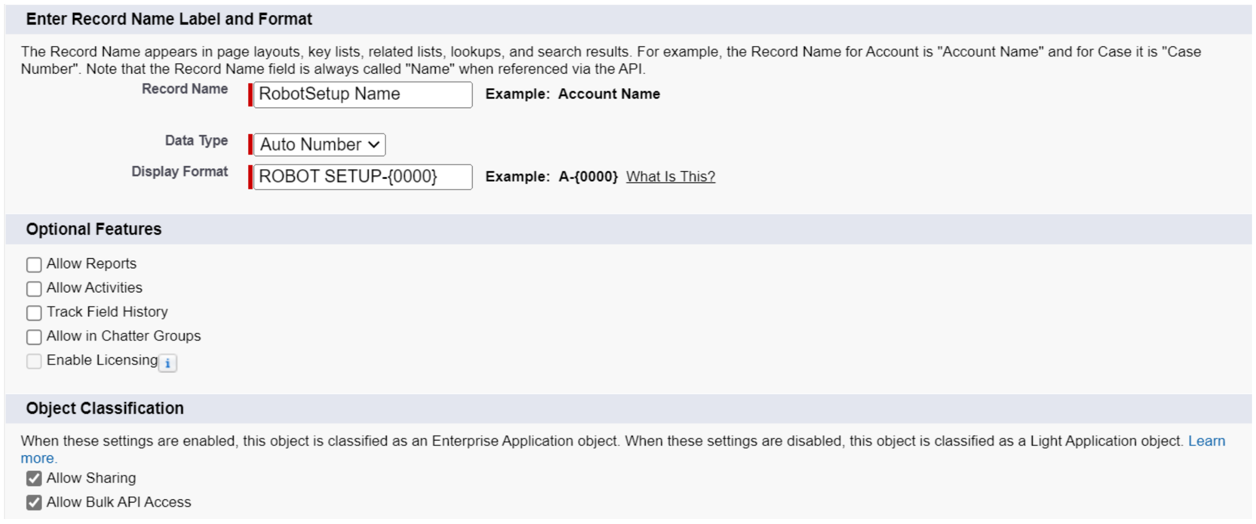
Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.
Be careful when changing the name or label as it may affect existing integrations and merge templates.

Label	Robot Setup	Example: Account
Plural Label	Robot Setups	Example: Accounts
Starts with vowel sound	☐	

The Object Name is used when referencing the object via the API.

Object Name	Robot_Setup	Example: Account



Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name	RobotSetup Name	Example: Account Name
Data Type	Auto Number	
Display Format	ROBOT SETUP-{0000}	Example: A-{0000} What Is This?

Optional Features

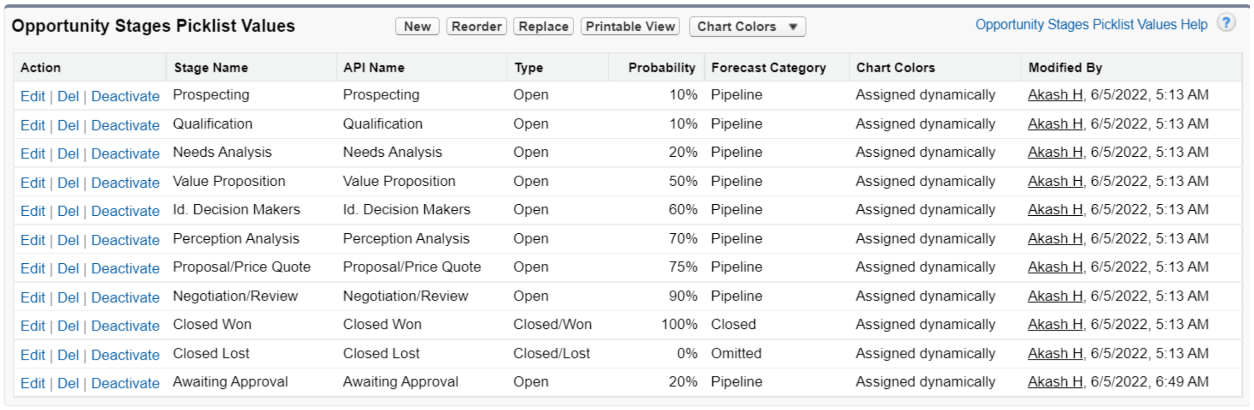
☐ Allow Reports
☐ Allow Activities
☐ Track Field History
☐ Allow in Chatter Groups
☐ Enable Licensing

Object Classification

When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. Learn more.

☑ Allow Sharing
☑ Allow Bulk API Access

4. Create sales process and validate opportunities



Opportunity Stages Picklist Values New | Reorder | Replace | Printable View | Chart Colors | Opportunity Stages Picklist Values Help

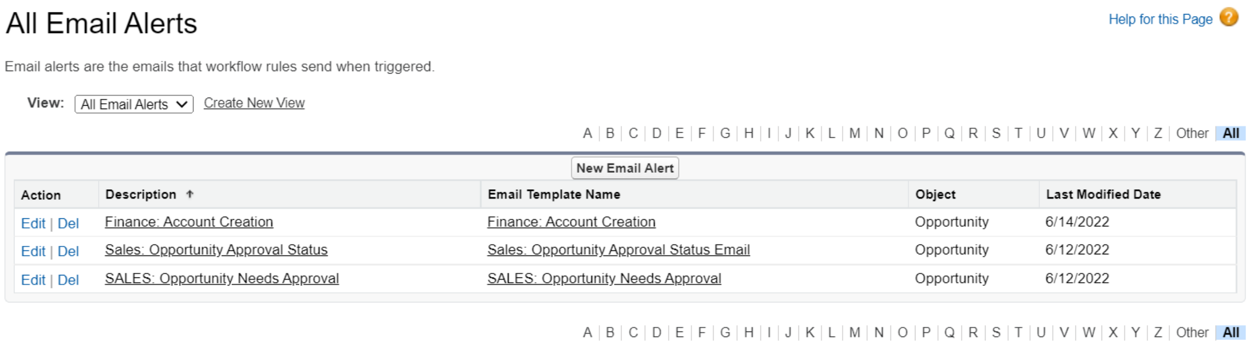
Action	Stage Name	API Name	Type	Probability	Forecast Category	Chart Colors	Modified By
Edit | Del | Deactivate	Prospecting	Prospecting	Open	10%	Pipeline	Assigned dynamically	Akash H, 6/5/2022, 5:13 AM
Edit | Del | Deactivate	Qualification	Qualification	Open	10%	Pipeline	Assigned dynamically	Akash H, 6/5/2022, 5:13 AM
Edit | Del | Deactivate	Needs Analysis	Needs Analysis	Open	20%	Pipeline	Assigned dynamically	Akash H, 6/5/2022, 5:13 AM
Edit | Del | Deactivate	Value Proposition	Value Proposition	Open	50%	Pipeline	Assigned dynamically	Akash H, 6/5/2022, 5:13 AM
Edit | Del | Deactivate	Id. Decision Makers	Id. Decision Makers	Open	60%	Pipeline	Assigned dynamically	Akash H, 6/5/2022, 5:13 AM
Edit | Del | Deactivate	Perception Analysis	Perception Analysis	Open	70%	Pipeline	Assigned dynamically	Akash H, 6/5/2022, 5:13 AM
Edit | Del | Deactivate	Proposal/Price Quote	Proposal/Price Quote	Open	75%	Pipeline	Assigned dynamically	Akash H, 6/5/2022, 5:13 AM
Edit | Del | Deactivate	Negotiation/Review	Negotiation/Review	Open	90%	Pipeline	Assigned dynamically	Akash H, 6/5/2022, 5:13 AM
Edit | Del | Deactivate	Closed Won	Closed Won	Closed/Won	100%	Closed	Assigned dynamically	Akash H, 6/5/2022, 5:13 AM
Edit | Del | Deactivate	Closed Lost	Closed Lost	Closed/Lost	0%	Omitted	Assigned dynamically	Akash H, 6/5/2022, 5:13 AM
Edit | Del | Deactivate	Awaiting Approval	Awaiting Approval	Open	20%	Pipeline	Assigned dynamically	Akash H, 6/5/2022, 6:49 AM

Add opportunity validation rule with the error formula :

IF((Amount > 100000 && Approved__c <> True && ISPICKVAL(StageName,'Closed Won')),True,False)

5. Automate opportunties

Create an approval process and select opportunity object



All Email Alerts

Help for this Page

Email alerts are the emails that workflow rules send when triggered.

View: All Email Alerts | Create New View

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

New Email Alert

Action	Description ↑	Email Template Name	Object	Last Modified Date
Edit | Del	Finance: Account Creation	Finance: Account Creation	Opportunity	6/14/2022
Edit | Del	Sales: Opportunity Approval Status	Sales: Opportunity Approval Status Email	Opportunity	6/12/2022
Edit | Del	SALES: Opportunity Needs Approval	SALES: Opportunity Needs Approval	Opportunity	6/12/2022

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

Create a process with the process builder

Start

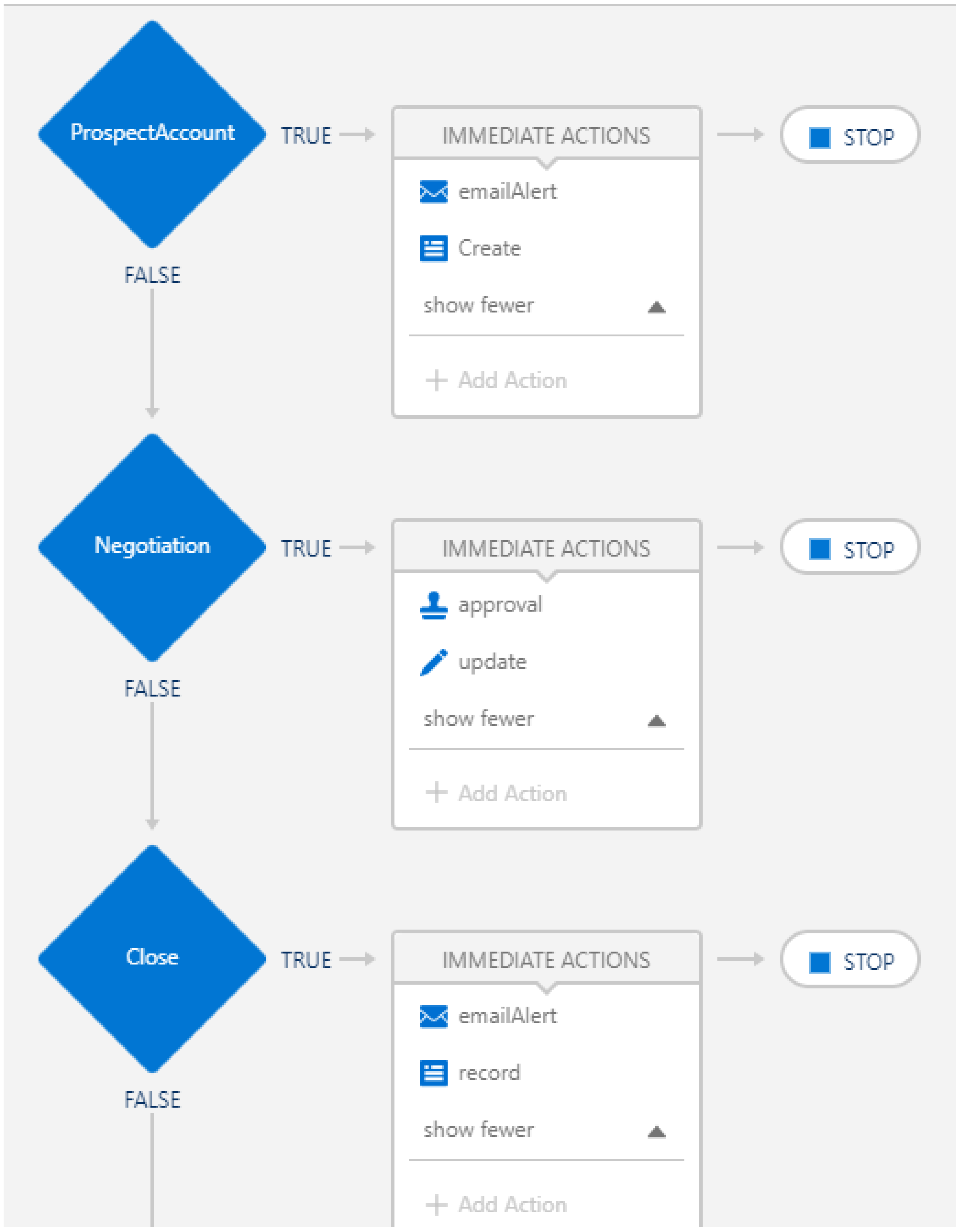
Opportunity

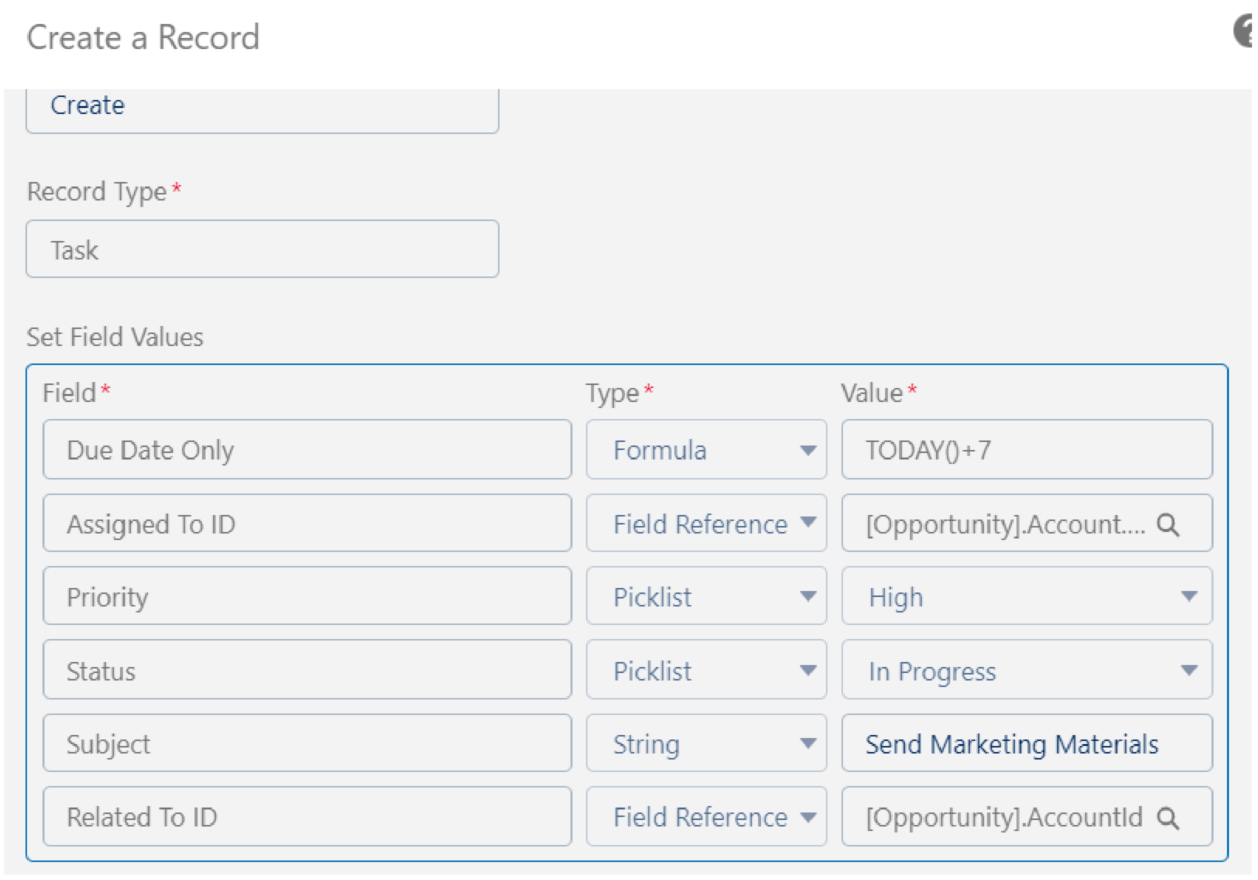
Customer Account->emailAlert

Prospect account->alertAlert->create

Negotiation->approval->update

Closed deal->alert->record


ProspectAccount
TRUE
IMMEDIATE ACTIONS
emailAlert
Create
show fewer
Add Action
STOP
FALSE
Negotiation
TRUE
IMMEDIATE ACTIONS
approval
update
show fewer
Add Action
STOP
FALSE
Close
TRUE
IMMEDIATE ACTIONS
emailAlert
record
show fewer
Add Action
STOP
FALSE


Create a Record

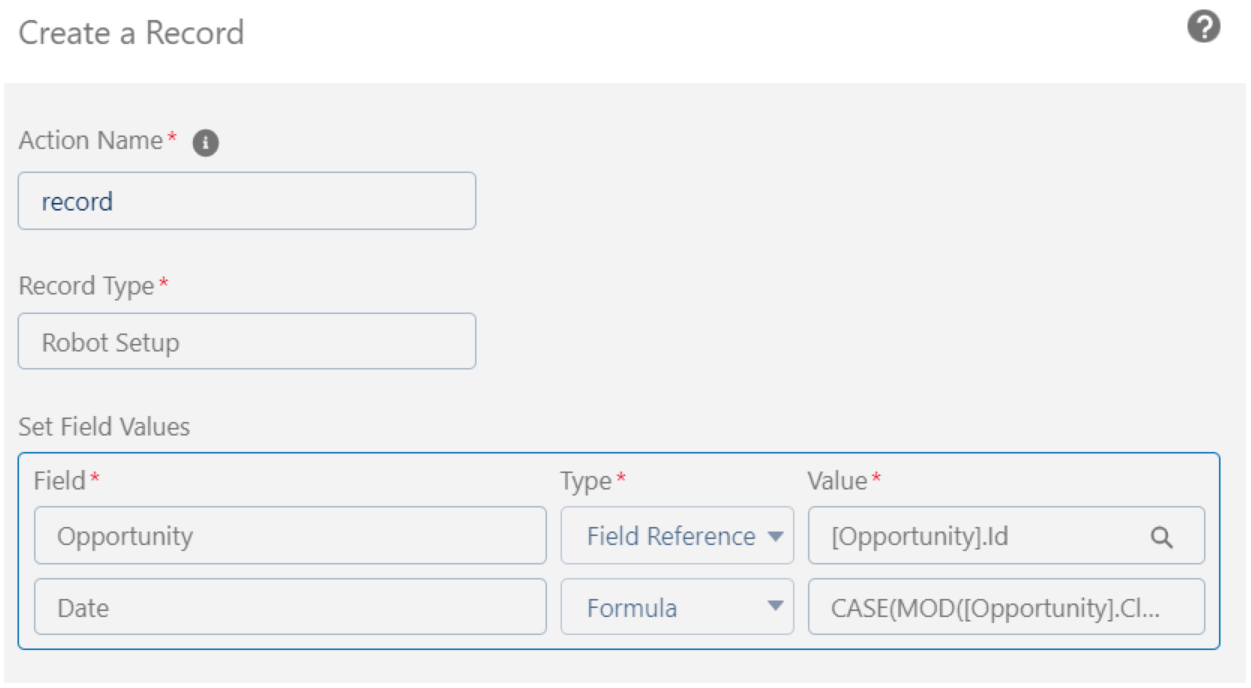
Create

Record Type *

Task

Set Field Values

Field *	Type *	Value *
Due Date Only	Formula	TODAY()+7
Assigned To ID	Field Reference	[Opportunity].Account....
Priority	Picklist	High
Status	Picklist	In Progress
Subject	String	Send Marketing Materials
Related To ID	Field Reference	[Opportunity].AccountId


Create a Record

Action Name *

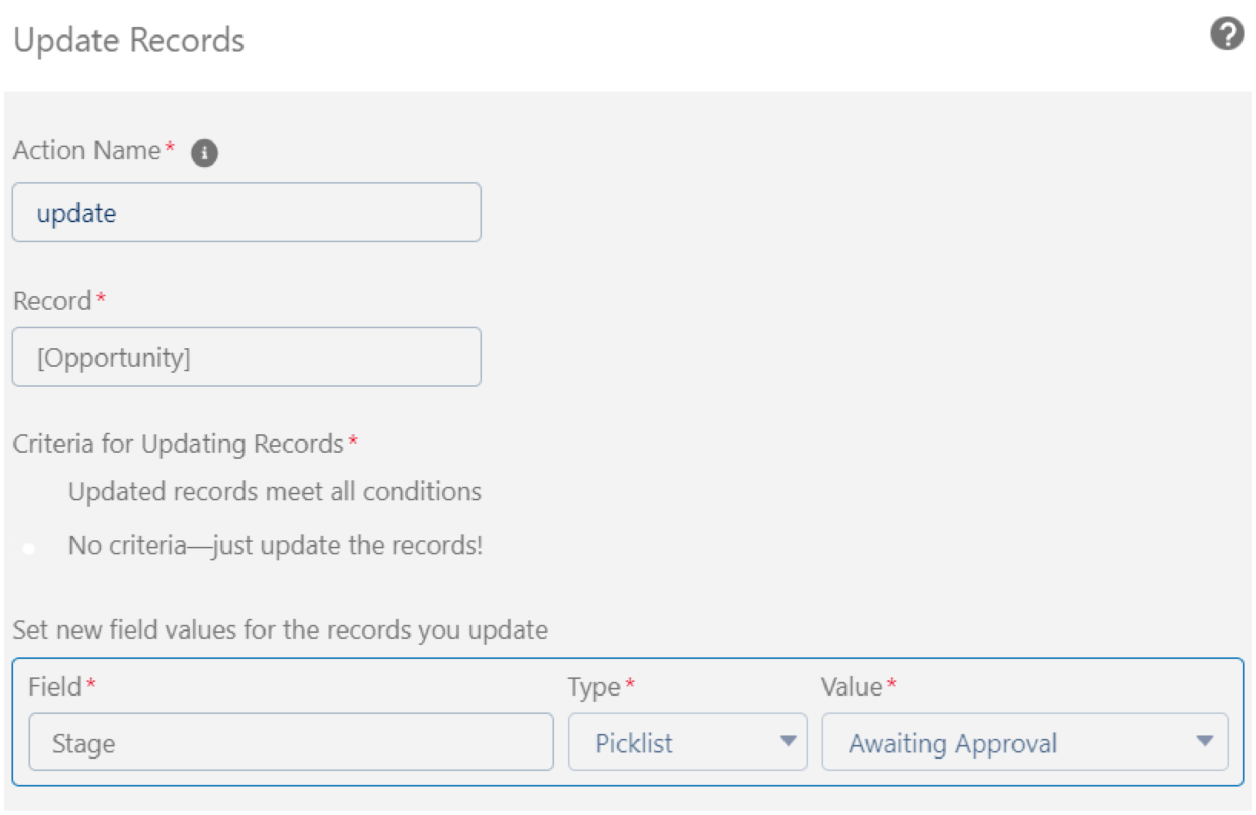
record

Record Type *

Robot Setup

Set Field Values

Field *	Type *	Value *
Opportunity	Field Reference	[Opportunity].Id
Date	Formula	CASE(MOD([Opportunity].Cl...


Update Records

Action Name *

update

Record *

[Opportunity]

Criteria for Updating Records *

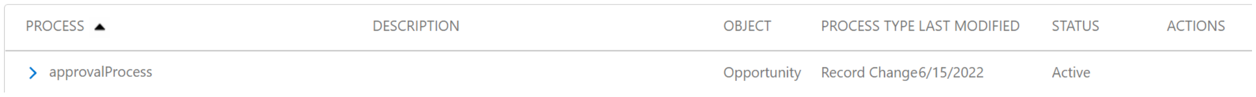
Updated records meet all conditions

No criteria—just update the records!

Set new field values for the records you update

Field *	Type *	Value *
Stage	Picklist	Awaiting Approval

Make sure the process is active



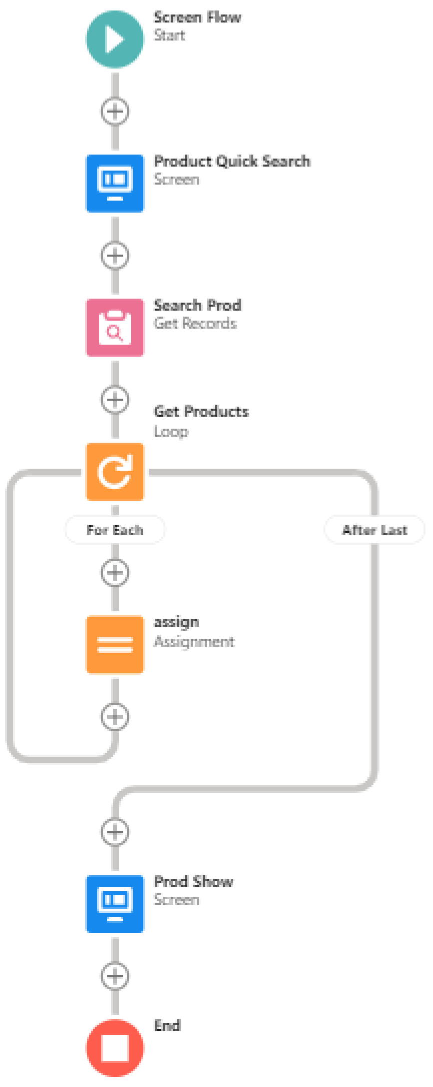
PROCESS	DESCRIPTION	OBJECT	PROCESS TYPE	LAST MODIFIED	STATUS	ACTIONS
approvalProcess		Opportunity	Record Change	6/15/2022	Active	

<u>6. Create flow for Opportunities</u>

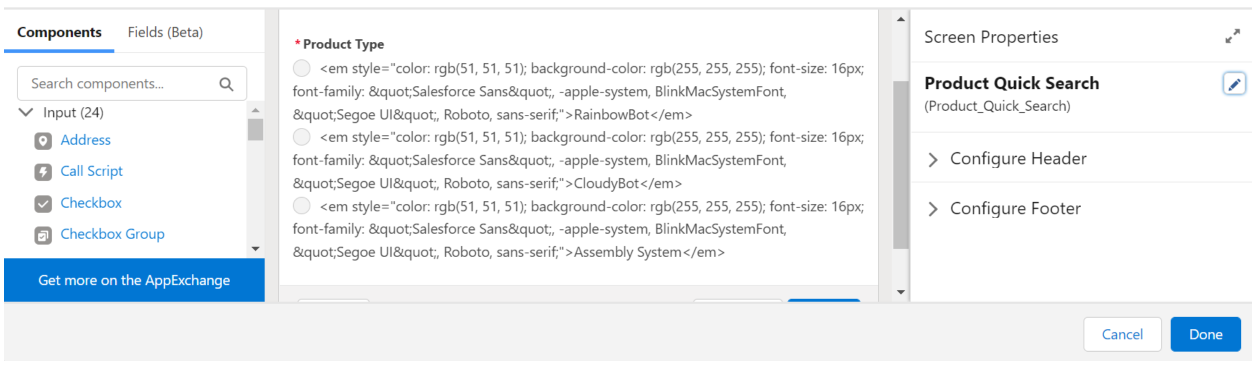
Create Flow named Product Quick Search as shown below

Create a radio button as Product Type

Create three choices as RainbowBot,CloudyBot and Assemble Systems



Edit Screen



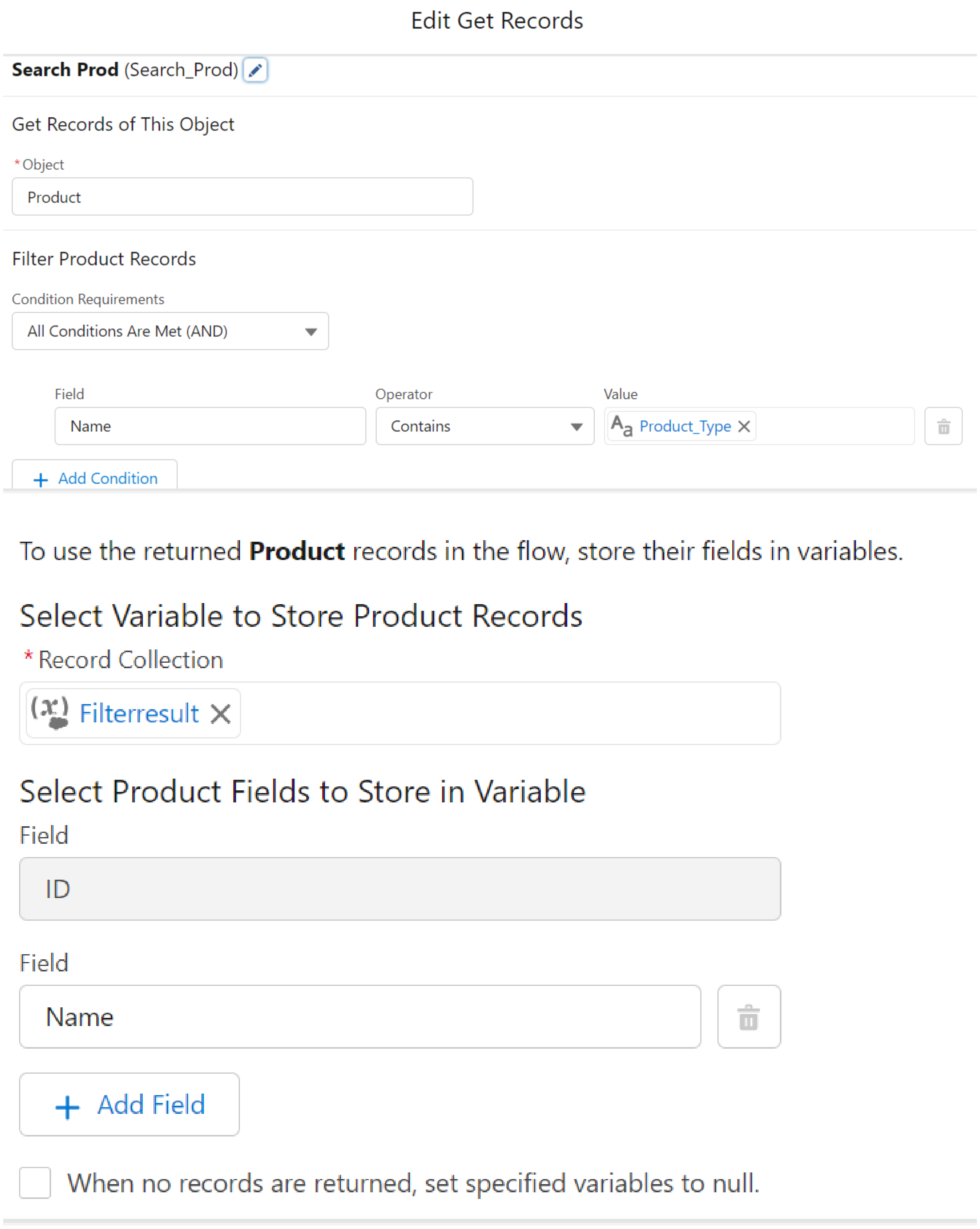
Product Quick Search
(Product_Quick_Search)

* Product Type

<em style="color: rgb(51, 51, 51); background-color: rgb(255, 255, 255); font-size: 16px; font-family: &quot;Salesforce Sans&quot;, -apple-system, BlinkMacSystemFont, &quot;Segoe UI&quot;, Roboto, sans-serif;">RainbowBot</em>

<em style="color: rgb(51, 51, 51); background-color: rgb(255, 255, 255); font-size: 16px; font-family: &quot;Salesforce Sans&quot;, -apple-system, BlinkMacSystemFont, &quot;Segoe UI&quot;, Roboto, sans-serif;">CloudyBot</em>

<em style="color: rgb(51, 51, 51); background-color: rgb(255, 255, 255); font-size: 16px; font-family: &quot;Salesforce Sans&quot;, -apple-system, BlinkMacSystemFont, &quot;Segoe UI&quot;, Roboto, sans-serif;">Assembly System</em>


Edit Get Records

Search Prod (Search_Prod)

Get Records of This Object

*Object

Product

Filter Product Records

Condition Requirements

All Conditions Are Met (AND)

Field	Operator	Value
Name	Contains	Product_Type

+ Add Condition

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

Select Variable to Store Product Records

* Record Collection

Filterresult

Select Product Fields to Store in Variable

Field

ID

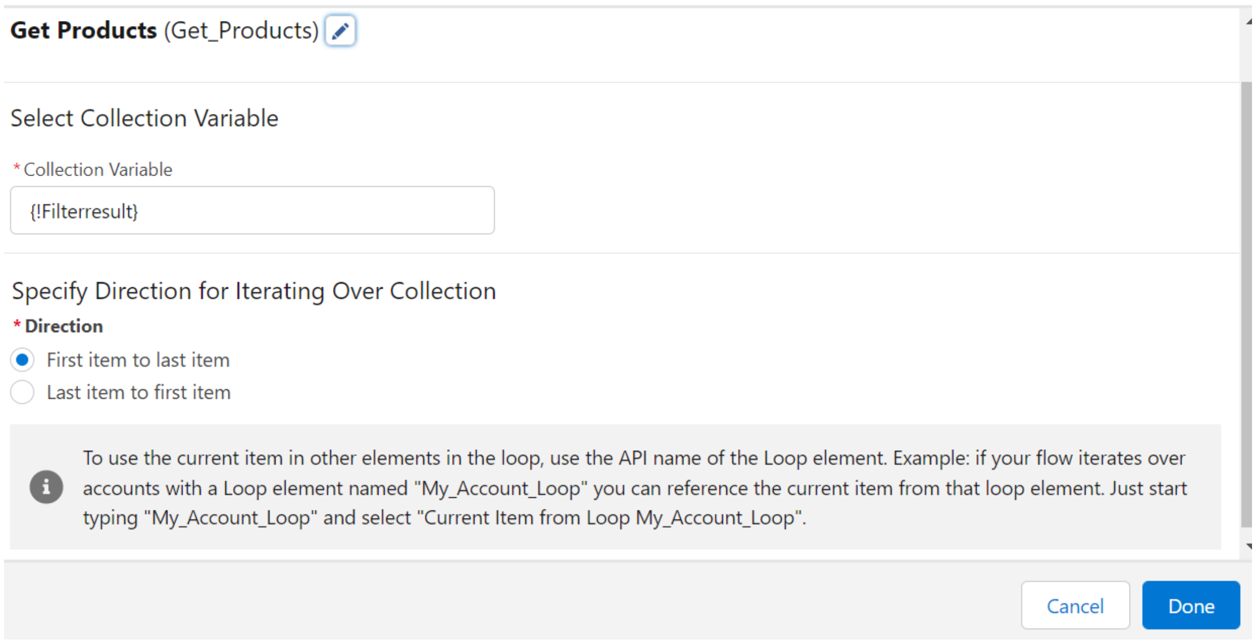
Field

Name

+ Add Field

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

Edit Loop



Get Products (Get_Products)

Select Collection Variable

* Collection Variable

{!Filterresult}

Specify Direction for Iterating Over Collection

* **Direction**

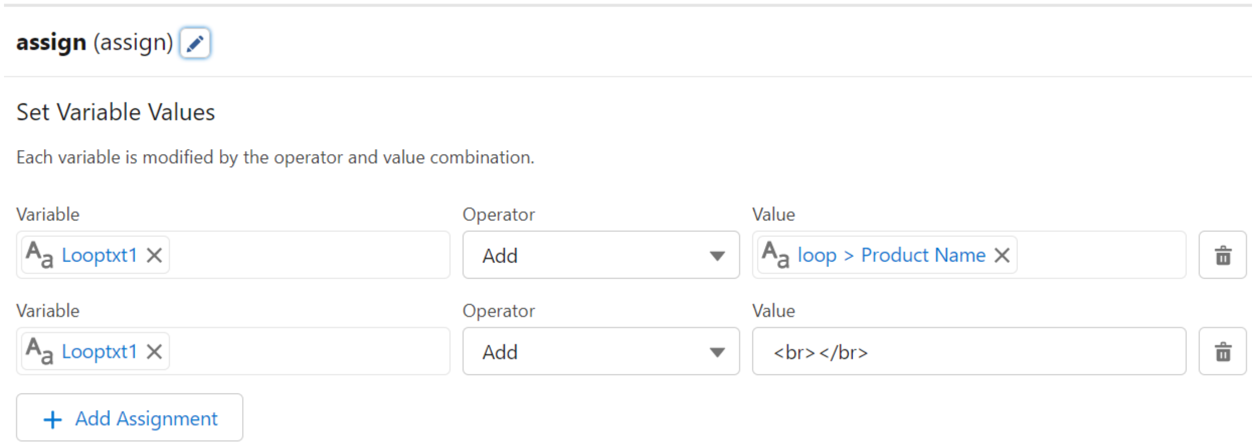
[✓ First item to last item]

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".

Cancel Done

Edit Assignment



assign (assign)

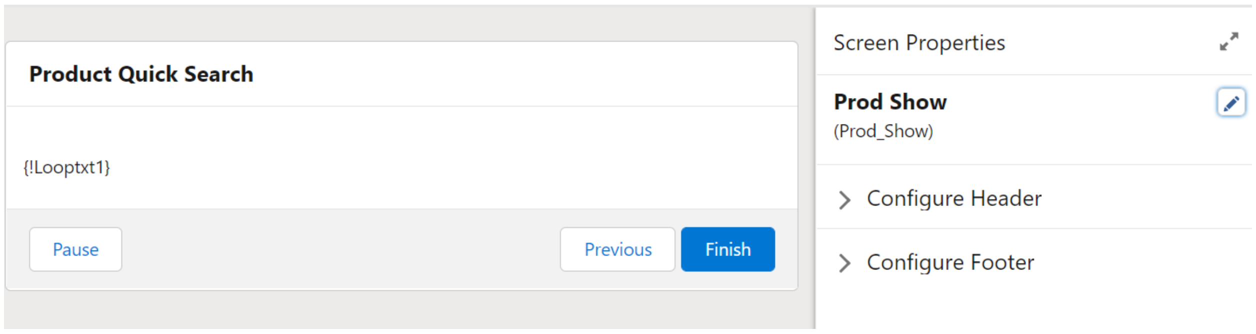
Set Variable Values

Each variable is modified by the operator and value combination.

Variable	Operator	Value
Looptxt1	Add	loop > Product Name
Looptxt1	Add	
</br>

+ Add Assignment

Edit Screen



Product Quick Search

{!Looptxt1}

Pause Previous Finish

Screen Properties

Prod Show
(Prod_Show)

Configure Header

Configure Footer

7. Automate Setups

In the Robot Setup object search for Day of the week and change the field type to formula with return type as text

Case (WEEKDAY(Date__c),

1,"Sunday",

2,"Monday",

3,"Tuesday",

4,"Wednesday",

5,"Thursday",

6,"Friday",

7,"Saturday",

Text(WEEKDay(Date__c)))

Change the formula of date field of the previously process "closed deal" criteria and update the formula as :

CASE(MOD([Opportunity].CloseDate + 180 - DATE(1900, 1, 7),7), 0, [Opportunity].CloseDate + 181, 6, [Opportunity].CloseDate + 182, [Opportunity].CloseDate + 180)