

Name: MEGAN KIRUPA ROXANNE.R
Email: kirupa19063@cse.ssn.edu.in
College: SSN College of Engineering

Trailhead URL: <https://trailblazer.me/id/megankiruparoxanner>

Apex Specialist Superbadge

Steps to setting up development org

1. Creating a new Trailhead Playground.
2. Installing this package (package ID: 04t6g000008av9iAAA) containing metadata.
3. Add picklist values Repair and Routine Maintenance to the Type field on the Case object.
4. Updating the Case page layout assignment to use the **Case (HowWeRoll) Layout** for your profile.
5. Renaming the tab/label for the Case tab to **Maintenance Request**.
6. Updating the Product page layout assignment to use the **Product (HowWeRoll) Layout**.
7. Renaming the tab/label for the Product object to **Equipment**.
8. Using App Launcher to navigate to the **Create Default Data** tab of the **How We Roll Maintenance** app and clicking **Create Data** to generate sample data for the application.
9. The newly created records are generated to the data model.

Challenge 1 - Automated Record Creation

Steps:

1. Go to the App Launcher and 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 it.
2. Feed -> Close Case = save it.
3. Go to the Object Manager -> Maintenance Request ->Field & Relationships ->New ->Lookup Relationship -> next -> select Equipment ->next -> Field Label = Equipment ->next->next->next -> save it .
4. Go to the developer console use below code.

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));
        } else {
            nc.Date_Due__c = Date.today().addDays((Integer)
cc.Equipment__r.maintenance_Cycle__c);
        }

        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

```

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

```

Compilation Steps:

1. 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 , save it.
2. Feed -> Close Case = save it.

3. Run all

Challenge 2 - Synchronize Salesforce data with an external system

Steps:

1. Setup -> Search in quick find box -> click Remote Site Settings -> Name = Warehouse URL , Remote Site URL = <https://th-superbadge-apex.herokuapp.com> , make sure active is selected.
2. Go to the developer console use below code.

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);
    }

}
}
}
}

1 * public with sharing class WarehouseCalloutService {
2
3     private static final String WAREHOUSE_URL = 'https://th-superbadge-apex.herokuapp.com/equipment';
4
5     //@future(callout=true)
6     public static void runWarehouseEquipmentSync(){
7
8         Http http = new Http();
9         HttpRequest request = new HttpRequest();
10
11         request.setEndpoint(WAREHOUSE_URL);
12         request.setMethod('GET');
13         HttpResponse response = http.send(request);
14
15
16         List<Product2> warehouseEq = new List<Product2>();
17
18         if (response.getStatusCode() == 200){
19             List<Object> jsonResponse = (List<Object>)JSON.deserializeUntyped(response.getBody());
20             System.debug(response.getBody());
21
22             for (Object eq : jsonResponse){
23                 Map<String, Object> mapJson = (Map<String, Object>)eq;
24                 Product2 myEq = new Product2();
25                 myEq.Replacement_Part__c = (Boolean) mapJson.get('replacement');
26                 myEq.Name = (String) mapJson.get('name');
27                 myEq.Maintenance_Cycle__c = (Integer) mapJson.get('maintenanceperiod');
28                 myEq.Lifespan_Months__c = (Integer) mapJson.get('lifespan');
29                 myEq.Cost__c = (Decimal) mapJson.get('lifespan');
30                 myEq.Warehouse_SKU__c = (String) mapJson.get('sku');
31                 myEq.Current_Inventory__c = (Double) mapJson.get('quantity');
32                 warehouseEq.add(myEq);
33             }
34
35             if (warehouseEq.size() > 0){
36                 upsert warehouseEq;
37                 System.debug('Your equipment was synced with the warehouse one');
38                 System.debug(warehouseEq);
39             }
40
41         }
42     }
43 }

```

The screenshot shows the Salesforce IDE interface. The main area displays the Apex code for the `WarehouseCalloutService`. The code implements a static method `runWarehouseEquipmentSync` that sends a GET request to a warehouse URL to retrieve equipment data, then processes each item and adds it to a list of `Product2` objects. If the list is not empty, it performs an upsert operation. The code uses the `Http` and `HttpRequest` classes to handle the API call.

Below the code editor, the **Test** tab of the Test Results panel is active. It shows a single test run named "707lw0000000C2AM" that was enqueued on Wednesday, June 29, 2022, at 23:37:27 GM... The test passed with 0 failures and 2 total tests. The overall code coverage is 100%, with 45/45 lines covered. The coverage details table includes rows for `CreateDefaultData`, `MaintenanceRequest`, `MaintenanceRequestHelper`, `WarehouseCalloutService`, and `WarehouseSyncSchedule`.

Class	Percent	Lines
Overall	100%	
CreateDefaultData	100%	45/45
MaintenanceRequest	100%	2/2
MaintenanceRequestHelper	100%	35/35
WarehouseCalloutService	100%	22/22
WarehouseSyncSchedule	100%	2/2

Compilation Steps:

After saving the code open execute anonymous window (CTRI+E) and run this method, System.enqueueJob(new WarehouseCalloutService());

Challenge 3 - Schedule synchronization using Apex code

Steps:

Go to the developer console use below code.

WarehouseSyncSchedule.apxc :-

```
global class WarehouseSyncSchedule implements Schedulable {  
    global void execute(SchedulableContext ctx) {  
  
        WarehouseCalloutService.runWarehouseEquipmentSync();  
    }  
}
```



Compilation Steps:

1. Save the code.
 2. Go to setup -> Search in Quick find box -> Apex Classes -> click Schedule Apex and Jb Name = WarehouseSyncScheduleJob , Apex Class = WarehouseSyncSchedule as below.
 3. Run all.
-

Challenge 4 - Test automation logic

Steps:

Go to the developer console use below code.

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);
}
}

```

```

1  @isTest
2  public with sharing class MaintenanceRequestHelperTest {
3
4      private static final string STATUS_NEW = 'New';
5      private static final string WORKING = 'Working';
6      private static final string CLOSED = 'Closed';
7      private static final string REPAIR = 'Repair';
8      private static final string REQUEST_ORIGIN = 'Web';
9      private static final string REQUEST_TYPE = 'Routine Maintenance';
10     private static final string REQUEST SUBJECT = 'Testing subject';
11
12     PRIVATE STATIC Vehicle__c createVehicle(){
13         Vehicle__c Vehicle = new Vehicle__C(name = 'SuperTruck');
14         return Vehicle;
15     }
16
17     PRIVATE STATIC Product2 createEq(){
18         product2 equipment = new product2(name = 'SuperEquipment',
19         lifespan_months__C = 10,

```

```

20                         maintenance_cycle__C = 10,
21                         replacement_part__c = true);
22
23     return equipment;
24 }
25
26 PRIVATE STATIC Case createMaintenanceRequest(id vehicleId, id equipmentId){
27     case cs = new case(Type=REPAIR,
28                         Status=STATUS_NEW,
29                         Origin=REQUEST_ORIGIN,
30                         Subject=REQUEST SUBJECT,
31                         Equipment__c=equipmentId,
32                         Vehicle__c=vehicleId);
33
34     return cs;
35 }
36
37 PRIVATE STATIC Equipment_Maintenance_Item__c createWorkPart(id equipmentId,id requestId){
38     Equipment_Maintenance_Item__c wp = new Equipment_Maintenance_Item__c(Equipment__c = equipmentId,
39                                         Maintenance_Request__c = requestId);
40
41     return wp;
42 }
43
44 @istest
45 private static void testMaintenanceRequestPositive(){
46     Vehicle__c vehicle = createVehicle();
47     insert vehicle;
48     id vehicleId = vehicle.Id;
49
50     Product2 equipment = createEq();
51     insert equipment;
52     id equipmentId = equipment.Id;
53
54     case somethingToUpdate = createMaintenanceRequest(vehicleId,equipmentId);
55     insert somethingToUpdate;
56
57     Equipment_Maintenance_Item__c workP = createWorkPart(equipmentId,somethingToUpdate.id);
58     insert workP;
59
60     somethingToUpdate.status = CLOSED;
61     update somethingToUpdate;
62     test.stopTest();
63
64     Case newReq = [Select id, subject, type, Equipment__c, Date_Reported__c, Vehicle__c, Date_Due__c
65                   from case
66                   where status =:STATUS_NEW];
67
68     Equipment_Maintenance_Item__c workPart = [select id
69                                         from Equipment_Maintenance_Item__c
70                                         where Maintenance_Request__c =:newReq.Id];
71
72     system.assert(workPart != null);
73     system.assert(newReq.Subject != null);
74     system.assertEquals(newReq.Type, REQUEST_TYPE);
75     SYSTEM.assertEquals(newReq.Equipment__c, equipmentId);
76     SYSTEM.assertEquals(newReq.Vehicle__c, vehicleId);
77     SYSTEM.assertEquals(newReq.Date_Reported__c, system.today());
78 }
79
80 @istest
81 private static void testMaintenanceRequestNegative(){
82     Vehicle__c vehicle = createVehicle();
83     insert vehicle;
84     id vehicleId = vehicle.Id;
85
86     product2 equipment = createEq();
87     insert equipment;
88     id equipmentId = equipment.Id;
89
90     case emptyReq = createMaintenanceRequest(vehicleId,equipmentId);
91     insert emptyReq;
92
93     Equipment_Maintenance_Item__c workP = createWorkPart(equipmentId, emptyReq.Id);
94     insert workP;
95
96     test.startTest();
97     emptyReq.Status = WORKING;
98     update emptyReq;
99     test.stopTest();

```

```

99
100    list<case> allRequest = [select id
101                      from case];
102
103    Equipment_Maintenance_Item__c workPart = [select id
104                      from Equipment_Maintenance_Item__c
105                      where Maintenance_Request__c = :emptyReq.Id];
106
107    system.assert(workPart != null);
108    system.assert(allRequest.size() == 1);
109 }
110
111 @istest
112 private static void testMaintenanceRequestBulk(){
113     list<Vehicle__C> vehicleList = new list<Vehicle__C>();
114     list<Product2> equipmentList = new list<Product2>();
115     list<Equipment_Maintenance_Item__c> workPartList = new list<Equipment_Maintenance_Item__c>();
116     list<case> requestList = new list<case>();
117     list<id> oldRequestIds = new list<id>();
118
119     for(integer i = 0; i < 300; i++){
120         vehicleList.add(createVehicle());
121         equipmentList.add(createEq());
122     }
123     insert vehicleList;
124     insert equipmentList;
125
126     for(integer i = 0; i < 300; i++){
127         requestList.add(createMaintenanceRequest(vehicleList.get(i).id, equipmentList.get(i).id));
128     }
129     insert requestList;
130
131     for(integer i = 0; i < 300; i++){
132         workPartList.add(createWorkPart(equipmentList.get(i).id, requestList.get(i).id));
133     }
134     insert workPartList;
135
136     test.startTest();
137     for(case req : requestList){
138         req.Status = CLOSED;
139         oldRequestIds.add(req.Id);
140     }
141     update requestList;
142     test.stopTest();
143
144     list<case> allRequests = [select id
145                             from case
146                             where status =: STATUS_NEW];
147
148     list<Equipment_Maintenance_Item__c> workParts = [select id
149                             from Equipment_Maintenance_Item__c
150                             where Maintenance_Request__c in: oldRequestIds];
151
152     system.assert(allRequests.size() == 300);
153 }
154 }
```

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>();
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;
}
}
```

MaintenanceRequestHelper.apxc MaintenanceRequest.apxt WarehouseSyncSchedule.apxc MaintenanceRequestHelperTest.apxc WarehouseCalloutServiceMock.apxc WarehouseCalloutServiceTest.apxc WarehouseCalloutService.apxc

Code Coverage: None API Version: 45 Go To

```
1 public with sharing class MaintenanceRequestHelper {
2     public static void updateWorkOrders(List<Case> updWorkOrders, Map<Id, Case> nonUpdCaseMap) {
3         Set<Id> validIds = new Set<Id>();
4
5
6         for (Case c : updWorkOrders){
7             if (nonUpdCaseMap.get(c.Id).Status != 'Closed' && c.Status == 'Closed'){
8                 if (c.Type == 'Repair' || c.Type == 'Routine Maintenance'){
9                     validIds.add(c.Id);
10
11                 }
12             }
13         }
14     }
15
16     if (!validIds.isEmpty()){
17         List<Case> newCases = new List<Case>();
18         Map<Id, Case> closedCasesM = new Map<Id, Case>([SELECT Id, Vehicle__c, Equipment__c, Equipment__r.Maintenance_Cycle__c, (SELECT Id, Equipment__r.Maintenance_Cycle__c FROM Case WHERE Id IN :validIds)]);
19
20         Map<Id, Decimal> maintenanceCycles = new Map<ID, Decimal>();
21         AggregateResult[] results = [SELECT Maintenance_Request__c, MIN(Equipment__r.Maintenance_Cycle__c)cycle FROM Equipment_Maintenance_Iteration];
22
23         for (AggregateResult ar : results){
24             maintenanceCycles.put((Id) ar.get('Maintenance_Request__c'), (Decimal) ar.get('cycle'));
25         }
26
27         for(Case cc : closedCasesM.values()){
28             Case nc = new Case (
29                 ParentId = cc.Id,
30                 Status = 'New',
31                 Subject = 'Routine Maintenance',
32                 Type = 'Routine Maintenance',
33                 Vehicle__c = cc.Vehicle__c,
34                 Equipment__c = cc.Equipment__c,
35                 Origin = 'Web',
36                 Date_Reported__c = Date.Today()
37             );
38         };
39     }
40 }
```

```

39             If (maintenanceCycles.containsKey(cc.Id)){
40                 nc.Date_Due__c = Date.today().addDays((Integer) maintenanceCycles.get(cc.Id));
41             }
42         }
43         newCases.add(nc);
44     }
45
46     insert newCases;
47
48     List<Equipment_Maintenance_Item__c> clonedWPs = new List<Equipment_Maintenance_Item__c>();
49     for (Case nc : newCases){
50         for (Equipment_Maintenance_Item__c wp : closedCasesM.get(nc.ParentId).Equipment_Maintenance_Items__r){
51             Equipment_Maintenance_Item__c wpClone = wp.clone();
52             wpClone.Maintenance_Request__c = nc.Id;
53             clonedWPs.add(wpClone);
54         }
55     }
56 }
57
58     insert ClonedWPs;
59 }
60 }
61 }
```

MaintenanceRequest.apxt :-

```

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

Compilation Steps:

Run all

Challenge 5 - Test callout logic

Steps:

Go to the developer console use below code.

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);
}

}
}
}

```

```

1 * public with sharing class WarehouseCalloutService {
2
3     private static final String WAREHOUSE_URL = 'https://th-superbadge-apex.herokuapp.com/equipment';
4
5     //@future(callout=true)
6     public static void runWarehouseEquipmentSync(){
7
8         Http http = new Http();
9         HttpRequest request = new HttpRequest();
10
11         request.setEndpoint(WAREHOUSE_URL);
12         request.setMethod('GET');
13         HttpResponse response = http.send(request);
14
15
16         List<Product2> warehouseEq = new List<Product2>();
17
18         if (response.getStatusCode() == 200){
19             List<Object> jsonResponse = (List<Object>)JSON.deserializeUntyped(response.getBody());
20             System.debug(response.getBody());
21
22             for (Object eq : jsonResponse){
23                 Map<String, Object> mapJson = (Map<String, Object>)eq;
24                 Product2 myEq = new Product2();
25                 myEq.Replacement_Part__c = (Boolean) mapJson.get('replacement');
26                 myEq.Name = (String) mapJson.get('name');
27                 myEq.Maintenance_Cycle__c = (Integer) mapJson.get('maintenanceperiod');
28                 myEq.Lifespan_Months__c = (Integer) mapJson.get('lifespan');
29                 myEq.Cost__c = (Decimal) mapJson.get('lifespan');
30                 myEq.Warehouse_SKU__c = (String) mapJson.get('sku');
31                 myEq.Current_Inventory__c = (Double) mapJson.get('quantity');
32                 warehouseEq.add(myEq);
33             }
34
35             if (warehouseEq.size() > 0){
36                 upsert warehouseEq;
37                 System.debug('Your equipment was synced with the warehouse one');
38                 System.debug(warehouseEq);
39             }
40
41         }
42     }
43 }

```

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]);
    }
}

```

```

1  @isTest
2
3  private class WarehouseCalloutServiceTest {
4      @isTest
5      static void testWareHouseCallout(){
6          Test.startTest();
7          // implement mock callout test here
8          Test.setMock(HTTPCalloutMock.class, new WarehouseCalloutServiceMock());
9          WarehouseCalloutService.runWarehouseEquipmentSync();
10         Test.stopTest();
11         System.assertEquals(1, [SELECT count() FROM Product2]);
12     }
13 }

```

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,"name":"Generator 1000 kW","maintenanceperiod":365,"lifespan":120,"cost":5000,"sku":"100003"}]');
        response.setStatusCode(200);
        return response;
    }
}

```

```

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

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

7          // Create a fake response
8          HttpResponse response = new HttpResponse();
9          response.setHeader('Content-Type', 'application/json');
10         response.setBody('[{"_id":"55d66226726b611100aaf741","replacement":false,"quantity":5,"name":"Generator 1000 kW","maintenanceperiod":365,"lifespan":120,"cost":5000,"sku":"100003"}]');
11         response.setStatusCode(200);
12         return response;
13     }
14 }
15
16 }

```

Compilation Steps:

Run all.

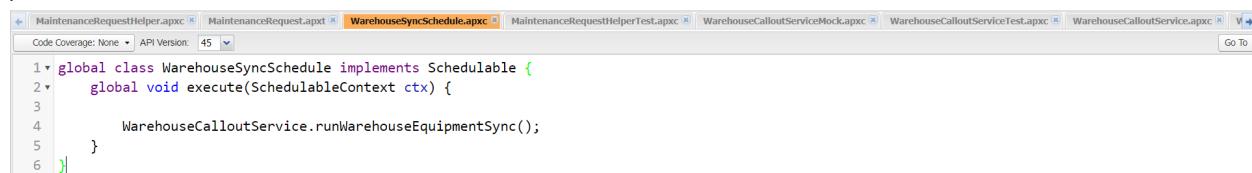
Challenge 6 - Test scheduling logic

Steps:

Go to the developer console use below code.

WarehouseSyncSchedule.apxc :-

```
global class WarehouseSyncSchedule implements Schedulable {  
    global void execute(SchedulableContext ctx) {  
  
        WarehouseCalloutService.runWarehouseEquipmentSync();  
    }  
}
```



```
1 ▾ global class WarehouseSyncSchedule implements Schedulable {  
2 ▾     global void execute(SchedulableContext ctx) {  
3  
4         WarehouseCalloutService.runWarehouseEquipmentSync();  
5     }  
6 }
```

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 ');  
  
    }  
}
```

```
+ MaintenanceRequest.apxc | WarehouseSyncSchedule.apxc | MaintenanceRequestHelperTest.apxc | WarehouseCalloutServiceMock.apxc | WarehouseCalloutServiceTest.apxc | WarehouseCalloutService.apxc | WarehouseSyncScheduleTest.apxc +  
Code Coverage: None | API Version: 45 | Run Test | Go To  
1  @isTest  
2  public class WarehouseSyncScheduleTest {  
3  
4  @isTest static void WarehousescheduleTest(){  
5      String scheduleTime = '00 00 01 * * ?';  
6      Test.startTest();  
7      Test.setMock(HttpCalloutMock.class, new WarehouseCalloutServiceMock());  
8      String jobID=System.schedule('Warehouse Time To Schedule to Test', scheduleTime, new WarehouseSyncSchedule());  
9      Test.stopTest();  
10     //Contains schedule information for a scheduled job. CronTrigger is similar to a cron job on UNIX systems.  
11     // This object is available in API version 17.0 and later.  
12     CronTrigger a=[SELECT Id FROM CronTrigger WHERE NextFireTime > today];  
13     System.assertEquals(jobID, a.Id,'Schedule ');  
14  
15 }  
16  
17 }
```

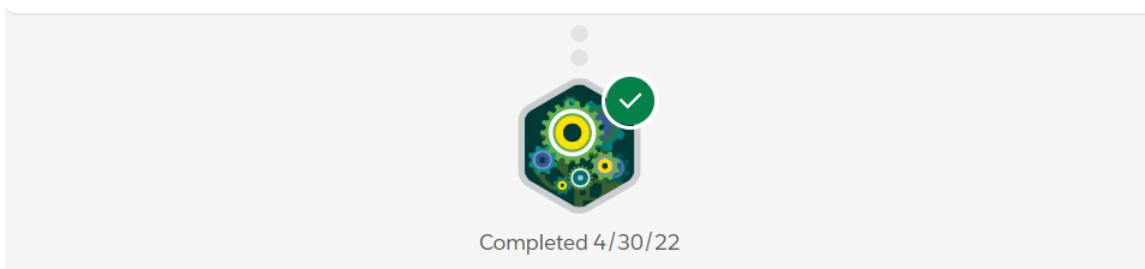
Compilation Steps:

Run all.

SUPERBADGE COMPLETE!

+13000 Points

[Discover more trailmixes](#)



Process Automation Specialist Superbadge

Steps:

1. Create a new Trailhead Playground for this superbadge.
2. Use Lightning Experience.
3. Install the Process Automation superbadge unmanaged package(package ID 04t46000001Zch4).

Challenge 1 - Automate Leads

Validation Rule on Lead - Search for Validation Rule and create a new under Lead in Object Manager.

Rule name: Country_Name

Error Condition Formula:

OR(AND(LEN(State) > 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:N
E:NV:NH:NJ:NM:NY:NC:ND:OH:OK:OR:PA:RI:SC:SD:TN:TX:UT:VT:VA:WA:WV:WI:WY", State)), NOT(OR(Country
="US",Country ="USA",Country ="United States", ISBLANK(Country))))

Lead Validation Rule Help for this Page 

[Back to Lead Validation Rules](#)

Validation Rule Detail

Rule Name	Country_Name	Active			
Error Condition Formula	OR(AND(LEN(State) > 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:N E:NV:NH:NJ:NM:NY:NC:ND:OH:OK:OR:PA:RI:SC:SD:TN:TX:UT:VT:VA:WA:WV:WI:WY", State)), NOT(OR(Country ="US",Country ="USA",Country ="United States", ISBLANK(Country))))	✓			
Error Message	Given value is not applicable	Error Location			
Description	Megan Kirupa Roxanne_R	Created By			
Created By	Megan Kirupa Roxanne_R	4/30/2022, 9:40 AM	Modified By	Megan Kirupa Roxanne_R	4/30/2022, 9:40 AM
Edit Clone					

Create Two Queues - Create the below queues for Lead as Supported Objects

Queues Help for this Page 

Queues allow groups of users to manage a shared workload more effectively. A queue is a location where records can be routed to await processing by a group member. The records remain in the queue until a user accepts them for processing or they are transferred to another queue. You can specify the set of objects that are supported by each queue, as well as the set of users that are allowed to retrieve records from the queue.

View: [All](#) [Edit](#) [Create New View](#)

Action	Label	Queue Name	Queue Email	Supported Objects	Modified By	Last Modified Date
Edit Del	Assembly_System_Sales	Assembly_System_Sales	Lead	R_Megan Kirupa Roxanne	Megan Kirupa Roxanne	4/30/2022, 9:41 AM
Edit Del	Rainbow_Sales	Rainbow_Sales	Lead	R_Megan Kirupa Roxanne	Megan Kirupa Roxanne	4/30/2022, 9:42 AM

Search under Quick Find for Lead Assignment Rules and create two rules entries as below:

Rule Name: LRule1

SETUP Lead Assignment Rules

Lead Assignment Rules

Automatically assign leads to users or queues based on criteria you define. You can create multiple rules with different conditions, but only one rule can be active at a time.

Click a rule name to add or edit rule entries.

Action	Rule Name	Active	Created By	Created On
Rename Del	LRule1	<input checked="" type="checkbox"/>	Megan Kirupa Roxanne R	4/30/2022
Rename Del	Standard	<input type="checkbox"/>	Megan Kirupa Roxanne R	4/30/2022

SETUP Lead Assignment Rules

Lead Assignment Rule LRule1

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		Edit			
Rule Name	LRule1	Active	<input checked="" type="checkbox"/>		
Created By	Megan Kirupa Roxanne R	4/30/2022, 9:43 AM	Modified By	Megan Kirupa Roxanne R	4/30/2022, 9:45 AM
Edit					

Rule Entries		New	Reorder	
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 NOTEQUAL TO Web	Assembly_System_Sales	<input type="checkbox"/>

Challenge 2 - Automate Accounts

Under Account in Object Manager,
Create four Roll-Up Summary Fields as below:

Account Custom Field
Number of deals

[Help for this Page](#)

[Back to Account Fields](#)

[Custom Field Definition Detail](#)

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

Field Information

Field Label	Number of deals	Object Name	Account		
Field Name	Number_of_deals				
API Name	Number_of_deals_c				
Description					
Help Text					
Data Owner					
Field Usage					
Data Sensitivity Level					
Compliance Categorization					
Created By	Megan Kirupa Roxanne R	4/30/2022, 9:47 AM	Modified By	Megan Kirupa Roxanne R	4/30/2022, 9:47 AM

Roll-Up Summary Options

Data Type	Roll-Up Summary	Summary Type	COUNT
Summarized Object	Opportunity		
Filter Criteria			

Account Custom Field
Number of won deals
[Back to Account Fields](#)

[Help for this Page](#)

Custom Field Definition Detail

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

Field Information

Field Label	Number of won deals	Object Name	Account
Field Name	<code>Number_of_won_deals</code>		
API Name	<code>Number_of_won_deals__c</code>		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Megan Kirupa Roxanne R , 4/30/2022, 9:49 AM	Modified By	Megan Kirupa Roxanne R , 4/30/2022, 9:49 AM

Roll-Up Summary Options

Data Type	Roll-Up Summary	Summary Type	COUNT
Summarized Object	Opportunity		
Filter Criteria	Stage EQUALS Closed Won		

Account Custom Field
Last won deal date
[Back to Account Fields](#)

[Help for this Page](#)

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	<code>Last_won_deal_date</code>		
API Name	<code>Last_won_deal_date__c</code>		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Megan Kirupa Roxanne R , 4/30/2022, 9:50 AM	Modified By	Megan Kirupa Roxanne R , 4/30/2022, 9:50 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		

Account Custom Field
Amount of won deals
[Back to Account Fields](#)

[Help for this Page](#)

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	<code>Amount_of_won_deals</code>		
API Name	<code>Amount_of_won_deals__c</code>		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Megan Kirupa Roxanne R , 4/30/2022, 9:51 AM	Modified By	Megan Kirupa Roxanne R , 4/30/2022, 9:51 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		

Create two Formula Fields as below:

Account Custom Field
Deal win percent
[Back to Account Fields](#)

[Help for this Page](#)

Custom Field Definition Detail

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

Field Information

Field Label	Deal win percent	Object Name	Account
Field Name	Deal_win_percent		
API Name	Deal_win_percent_c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Megan Kirupa Roxanne R. 4/30/2022, 9:54 AM	Modified By	Megan Kirupa Roxanne R. 4/30/2022, 10:17 AM

Formula Options

Data Type	Formula
Decimal Places	2
Number_of_won_deals_c / Number_of_deals_c	

Account Custom Field
Call for Service
[Back to Account Fields](#)

[Help for this Page](#)

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	Megan Kirupa Roxanne R. 4/30/2022, 9:55 AM	Modified By	Megan Kirupa Roxanne R. 4/30/2022, 10:04 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")	

Create two Validation Rules as below:

Validation Rules

2 Items, Sorted by Rule Name

[New](#)

Rule Name	Error Location	Error Message	Active	Modified By
Name_Change	Top of Page	You cannot change Account Name for "customer direct" or "customer channel"	✓	Megan Kirupa Roxanne R. 4/30/2022, 9:58 AM
US_Address	Top of Page	Billing address should be valid US state abbreviations	✓	Megan Kirupa Roxanne R. 4/30/2022, 9:57 AM

Account Validation Rule

[Back to Account Validation Rules](#)

[Help for this Page](#)

Validation Rule Detail

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

Rule Name	US_Address	Active	✓
Error Condition 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 = "United States", ISBLANK(BillingCountry))), NOT(OR(ShippingCountry = "US", ShippingCountry = "United States", ISBLANK(ShippingCountry))))		
Error Message	Billing address should be valid US state abbreviations	Error Location	Top of Page
Description			
Created By	Megan Kirupa Roxanne R. 4/30/2022, 9:57 AM	Modified By	Megan Kirupa Roxanne R. 4/30/2022, 9:57 AM

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

Account Validation Rule

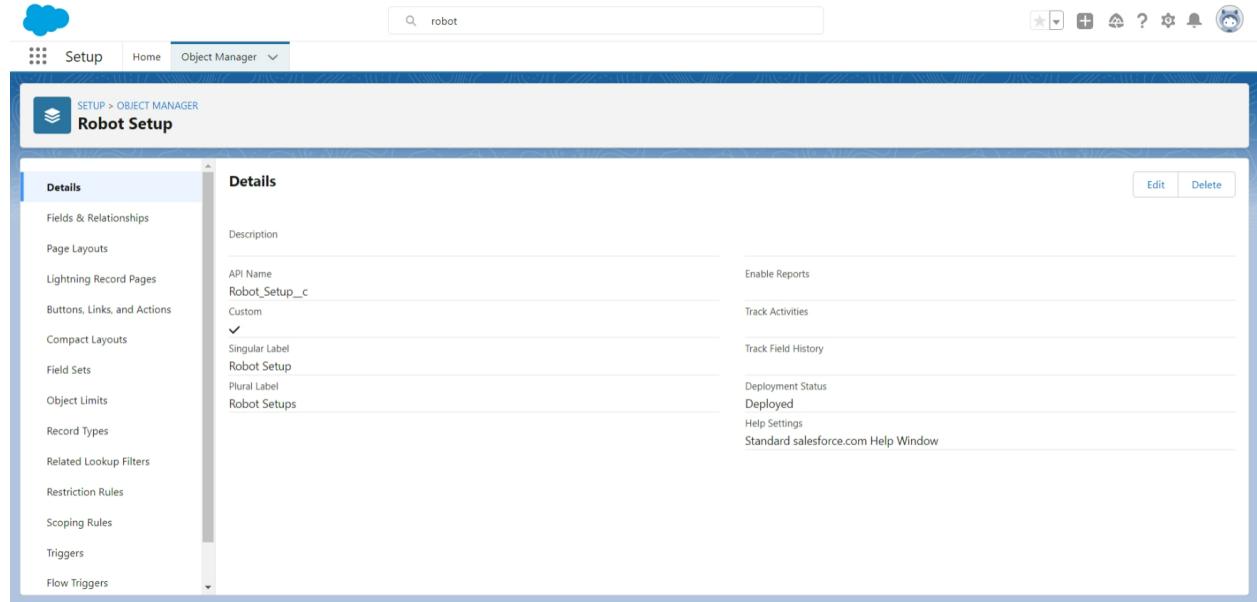
Help for this Page 

[Back to Account Validation Rules](#)

Validation Rule Detail		Edit	Clone
Rule Name	Name_Change	Active	<input checked="" type="checkbox"/>
Error Condition Formula	ISCHANGED(Name) && (OR(ISPICKVAL(Type , 'Customer - Direct') ,ISPICKVAL(Type , 'Customer - Channel')))	Error Location	Top of Page
Error Message	You cannot change Account Name for "customer direct" or "customer channel"	Created By	Megan Kirupa Roxanne_R, 4/30/2022, 9:58 AM
Description		Modified By	Megan Kirupa Roxanne_R, 4/30/2022, 9:58 AM
Created By	Megan Kirupa Roxanne_R, 4/30/2022, 9:58 AM	Modified By	Megan Kirupa Roxanne_R, 4/30/2022, 9:58 AM

Challenge 3 - Create Robot Setup Object

Create a custom object **Robot Setup**.



The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes a cloud icon, 'Setup', 'Home', and 'Object Manager'. A search bar contains the text 'robot'. The main area displays the 'Robot Setup' object details. On the left, a sidebar lists various configuration options: 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, and Flow Triggers. The main 'Details' section shows the following fields:

Field	Value
Description	
API Name	Robot_Setup_c
Custom	<input checked="" type="checkbox"/>
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

At the top right of the main window are 'Edit' and 'Delete' buttons.

Assign to it a Master-Detail Relationship to Opportunity. Use the following Field Names- Date, Notes, Day of the Week.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Date	Date__c	Date		
Day of the Week	Day_of_the_Week__c	Formula (Text)		
Last Modified By	LastModifiedById	Lookup(User)		
Notes	Notes__c	Text(255)		
Opportunity	Opportunity__c	Master-Detail(Opportunity)		
Robot Setup Name	Name	Auto Number		

Include Autonumber with the display format ROBOT SETUP-{0000}

Field Label	Robot Setup Name
Data Type	Auto Number
Description	
Data Owner	
Field Usage	
Data Sensitivity Level	
Compliance Categorization	
Display Format	ROBOT SETUP-{0000}

Challenge 4 - Create Sales Process and Validate Opportunities

Steps:

1. Create Sales Process in Opportunity; the name should be RB Robotics Sales Process.
2. Create a record type; the name should be RB Robotics Process RT.

The screenshot shows the Salesforce Setup interface under the Object Manager for the Opportunity object. The current tab is 'Record Types'. A specific record type, 'RB Robotics Process RT', is selected. The details pane shows the following configuration:

- Record Type Label:** RB Robotics Process RT
- Sales Process:** RB Robotics Sales Process
- Record Type Name:** RB_Robotics_Process_RT
- Namespace Prefix:** (empty)
- Description:** (empty)
- Created By:** Megan Kirupa Roxanne B. 4/30/2022, 10:29 AM
- Modified By:** Megan Kirupa Roxanne B. 4/30/2022, 10:29 AM
- Active:** checked

Below the main details, there is a section titled 'Picklists Available for Editing' which lists three fields:

Action	Field	Modified Date
Edit	Delivery/Installation Status	4/30/2022, 10:29 AM
Edit	Lead Source	4/30/2022, 10:29 AM
Edit	Type	4/30/2022, 10:29 AM

3. Add Awaiting Approval value in opportunity Stage picklist values.

The screenshot shows the Salesforce Setup interface under the Object Manager for the Opportunity object. The current tab is 'Fields & Relationships'. A specific field, 'Stage', is selected. The details pane shows the following configuration:

- Field Label:** Stage
- Data Type:** Picklist
- Help Text:** (empty)
- Description:** (empty)
- Data Owner:** (empty)
- Field Usage:** (empty)
- Data Sensitivity Level:** (empty)
- Compliance Categorization:** (empty)

Below the field information, there is a section titled 'Opportunity Stages Picklist Values' which lists the following stages:

Action	Stage Name	API Name	Type	Probability	Forecast Category	Chart Colors	Modified By
Edit	Prospecting	Prospecting	Open	10%	Pipeline	Assigned dynamically	Megan Kirupa Roxanne B. 4/30/2022, 4:51 AM
Edit	Qualification	Qualification	Open	10%	Pipeline	Assigned dynamically	Megan Kirupa Roxanne B. 4/30/2022, 4:51 AM
Edit	Needs Analysis	Needs Analysis	Open	20%	Pipeline	Assigned dynamically	Megan Kirupa Roxanne B. 4/30/2022, 4:51 AM
Edit	Value Proposition	Value Proposition	Open	20%	Pipeline	Assigned dynamically	Megan Kirupa Roxanne B. 4/30/2022, 4:51 AM
Edit	Decision Making	Decision Making	Open	20%	Pipeline	Assigned dynamically	Megan Kirupa Roxanne B. 4/30/2022, 4:51 AM
Edit	Presentation Analysis	Presentation Analysis	Open	20%	Pipeline	Assigned dynamically	Megan Kirupa Roxanne B. 4/30/2022, 4:51 AM
Edit	Proposal/Price Quote	Proposal/Price Quote	Open	75%	Pipeline	Assigned dynamically	Megan Kirupa Roxanne B. 4/30/2022, 4:51 AM
Edit	Negotiation/Review	Negotiation/Review	Open	50%	Pipeline	Assigned dynamically	Megan Kirupa Roxanne B. 4/30/2022, 4:51 AM
Edit	Closed Won	Closed Won	Closed/Won	100%	Closed	Assigned dynamically	Megan Kirupa Roxanne B. 4/30/2022, 4:51 AM
Edit	Closed Lost	Closed Lost	Closed/Lost	0%	Omitted	Assigned dynamically	Megan Kirupa Roxanne B. 4/30/2022, 4:51 AM
Edit	Awaiting Approval	Awaiting Approval	Open	20%	Pipeline	Assigned dynamically	Megan Kirupa Roxanne B. 4/30/2022, 10:29 AM

At the bottom of the stage configuration, there is a note: "No inactive values defined".

4. Process RT record type.

The screenshot shows the Salesforce Setup interface under the Sales Processes section. On the left, there's a sidebar with a search bar and navigation links for Feature Settings, Sales, and Sales Processes. The main area is titled "Sales Processes" and shows the "RB Robotics Sales Process". It lists "Opportunity Stages" with two columns: "Available Values" and "Selected Values". The "Available Values" column contains stages like "Needs Analysis (Open, 20%, Pipeline)" and "Value Proposition (Open, 50%, Pipeline)". The "Selected Values" column contains stages like "Prospecting (Open, 10%, Pipeline)" and "Qualification (Open, 10%, Pipeline)". There are "Add" and "Remove" buttons between the two columns. At the bottom are "Save" and "Cancel" buttons.

5. Create a Checkbox field and Name it Approved.
6. Write a validation rule as below:

The screenshot shows the "Opportunity Validation Rule" creation page. The title is "Validation Rule Detail". It has fields for Rule Name (RB_High_Value_Opp), Error Condition Formula (IF((Amount > 100000 && Approved__c <> True && ISPICKVAL(StageName,'Closed Won')),True,False)), Error Message (appropriate value should be given), Description (Megan Kirupa Roxanne_R 4/30/2022, 10:27 AM), Active (checked), Error Location (Top of Page), and Modified By (Megan Kirupa Roxanne_R 4/30/2022, 10:27 AM). There are "Edit" and "Clone" buttons at the bottom.

Challenge 5 - Automate Opportunities

Create three email templates:

Finance: Account Creation,

SALES: Opportunity Needs Approval,

Sales: Opportunity Approval Status

Create related Email Alert from the Search box for the above templates.

Approval Process Definition Detail

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

Process Definition Detail

Process Name	Prospect Approval	Active	<input checked="" type="checkbox"/>
Unique Name	Prospect_Approval	Next Automated Approver Determined By	
Description			
Entry Criteria	(Opportunity: Stage equals Negotiation/Review) AND (Opportunity: Amount GREATER THAN 100000)		
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests	<input type="checkbox"/>
Approval Assignment Email Template	Sales_Opportunity_Approval_Status_Email		
Initial Submitters	Opportunity Owner		
Created By	Megan Kirupa Roxanne R, 4/30/2022, 10:35 AM		
Modified By	Megan Kirupa Roxanne R, 4/30/2022, 10:47 AM		

Initial Submission Actions [Add Existing](#) [Add New](#)

Action	Type	Description
Record Lock		Lock the record from being edited
Edit Remove	Field Update	Approval

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

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
	1	Approval for Prospect			User Nushi Davoud	Final Rejection

Final Approval Actions [Add Existing](#) [Add New](#)

Action	Type	Description
Edit	Record Lock	Lock the record from being edited
Edit Remove	Field Update	Stage-Closed Won
Edit Remove	Email Alert	Sales_Opportunity_Approval_Request_Mail
Edit Remove	Field Update	Approved Check

Final Rejection Actions [Add Existing](#) [Add New](#)

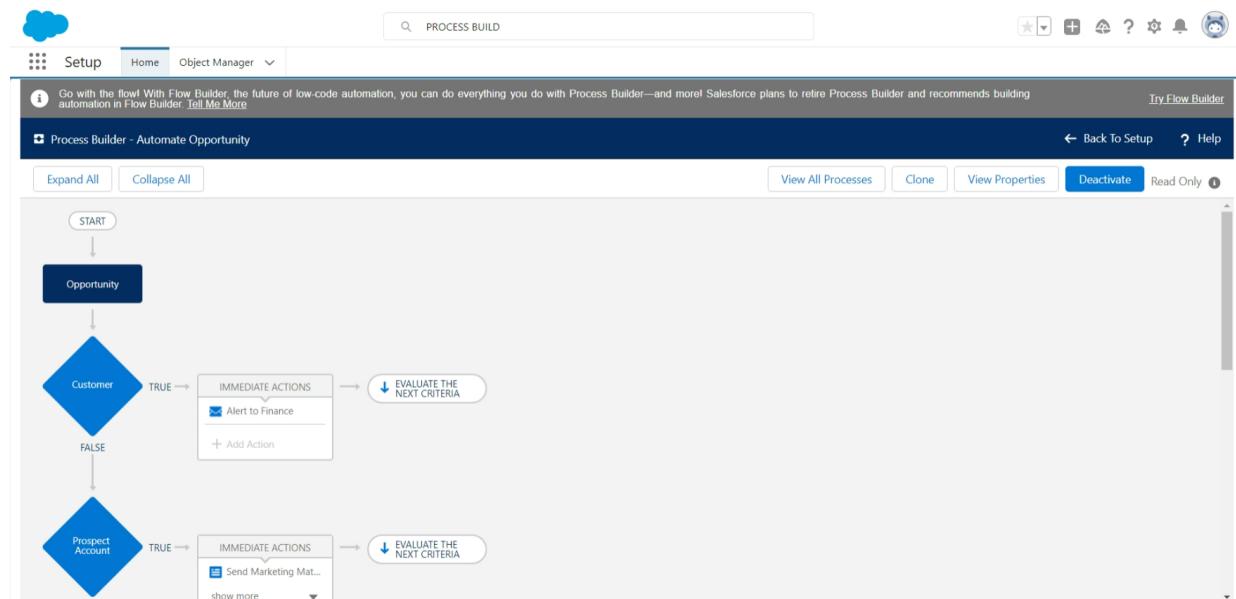
Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit Remove	Field Update	State_Nego
Edit Remove	Email Alert	Sales_Approval_Email

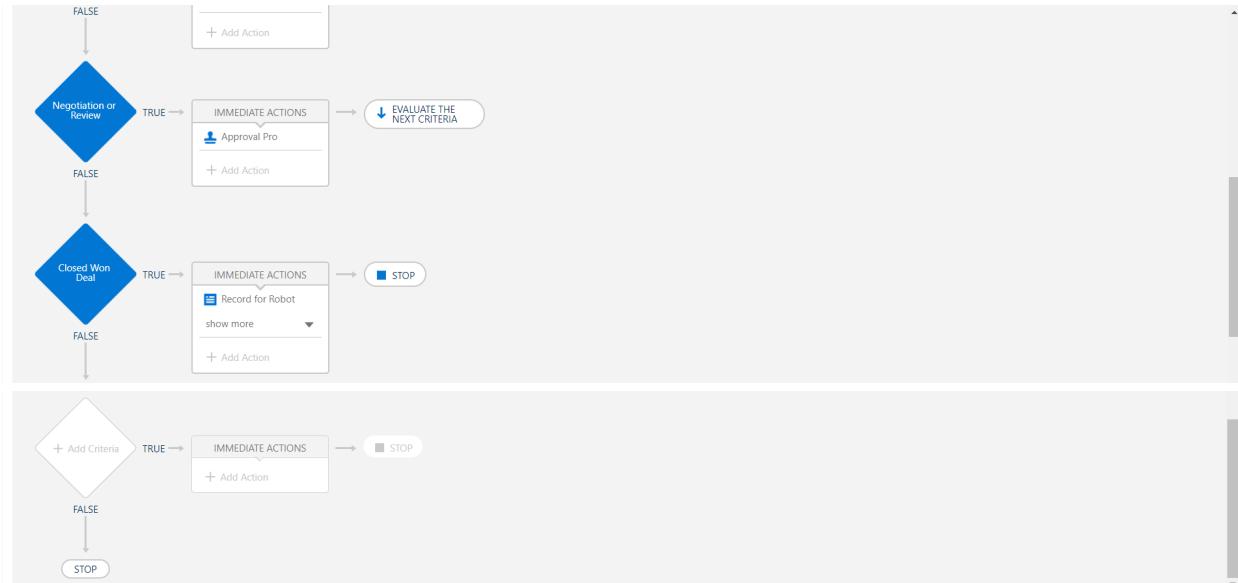
Recall Actions [Add Existing](#) [Add New](#)

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

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

Create Process with Process Builder - Automate Opportunity





Challenge 6 - Create Flow for Opportunities

Create a flow named Product Quick Search

The screenshot shows the Salesforce Flow Builder interface for a flow named 'Product Quick Search - V1'. The top navigation bar includes back, forward, search, and settings icons, along with tabs for 'Flow Builder' and 'Product Quick Search - V1'. On the right, there are buttons for 'Free-Form', 'Version 1: Active—Last modified 2 months ago', 'Run', 'Debug', 'Deactivate', 'Save As', and 'Save'.

The left sidebar contains a 'Toolbox' with sections for 'Elements' (Interaction, Logic, Data) and 'Manager' (Screens, Actions, Subflows). The 'Interaction' section is expanded, showing icons for Screen, Action, and Subflow. The 'Logic' section is also expanded, showing Assignment, Decision, Loop, Collection Sort, and Collection Filter. The 'Data' section is expanded, showing Create Records, Update Records, Get Records, Delete Records, and Roll Back Records.

The main workspace displays the flow logic. It starts with a 'Start Screen Flow' step, which branches into two parallel paths. The first path leads to a 'Screen' component labeled 'Product Quick Search'. From this screen, two arrows branch out: one to a 'Screen' component labeled 'Final Screen' and another to a 'Get Records' component labeled 'Product'. The 'Get Records' component has a feedback loop arrow pointing back to the 'Product Quick Search' screen.

Element 1 - Screen Component - Product Quick Search

Edit Screen

The screenshot shows the 'Edit Screen' interface. On the left, there's a sidebar titled 'Components' with a search bar and a list of components under 'Input (24)'. The list includes Address, Call Script, Checkbox, Checkbox Group, Currency, Date, Date & Time, Dependent Picklists, and Display Image. Below this is a link to 'Get more on the AppExchange'. In the center, there's a preview of a 'Product Quick Search' component. This preview shows a list of 'Product Type' options: RainbowBot, CloudyBot, and Assembly System. At the bottom of the preview are buttons for 'Pause', 'Previous', and 'Finish'. On the right, there's a 'Screen Properties' panel for the 'Product Quick Search' component, which has the ID '(Product_Quick_Search)'. The properties section includes 'Configure Header' and 'Configure Footer' options. At the bottom right of the main area are 'Cancel' and 'Done' buttons.

Element 2 - Get Record

Edit Get Records

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

Product (Product)

Get Records of This Object

* Object
Equipment

Filter Equipment Records

Condition Requirements
All Conditions Are Met (AND)

Field	Operator	Value
Name	Equals	Aa Product_Type X

+ Add Condition

Sort Equipment Records

Sort Order

Not Sorted



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

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 **Equipment** records in the flow, store their fields in variables.

Where to Store Field Values

- Together in a record variable
- In separate variables

Select Variable to Store Equipment

*Record

(x) FilterResult

Select Equipment Fields to Store in Variable

Field

ID

Field

Name



Add Field

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

Cancel

Done

Challenge 7 - Automate Setups

Change the datatype for “Day of the week” field from TEXT to Formula (TEXT) and use the following the formula to get Day of the week:

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

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		Object Name	Robot Setup
Field Label	Day of the Week	Object Name	Robot Setup
Field Name	Day_of_the_Week	Object Name	Robot Setup
API Name	Day_of_the_Week__c	Object Name	Robot Setup
Description		Object Name	Robot Setup
Help Text		Object Name	Robot Setup
Data Owner		Object Name	Robot Setup
Field Usage		Object Name	Robot Setup
Data Sensitivity Level		Object Name	Robot Setup
Compliance Categorization		Object Name	Robot Setup
Created By	Megan.Kirupa.Roxanne_R. 4/30/2022, 10:21 AM	Modified By	Megan.Kirupa.Roxanne_R. 4/30/2022, 10:21 AM

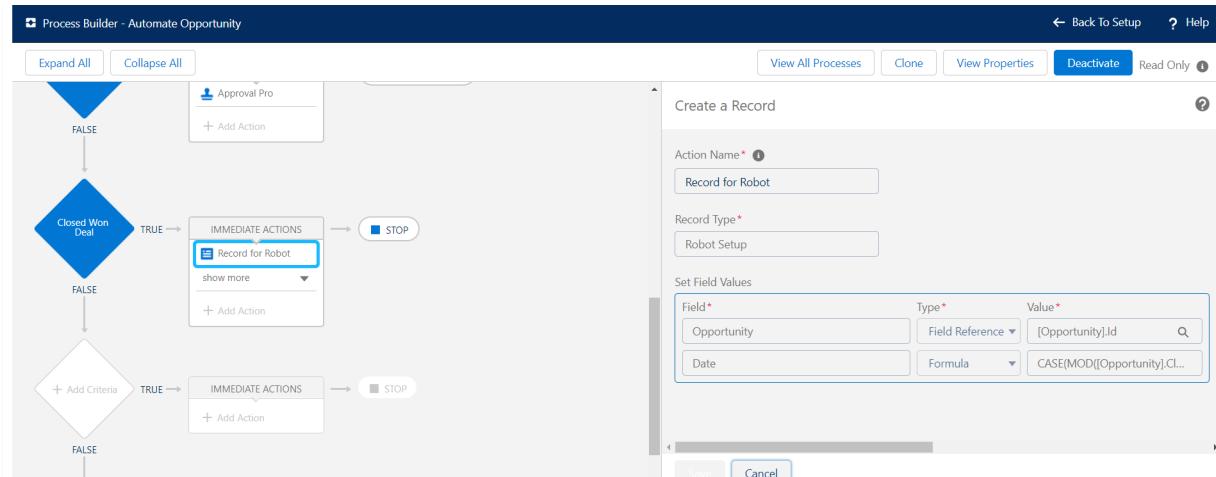
Formula Options

Data Type: Formula

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

Create clone of Process in Challenge 5. Go to last node and change formula of date field as below:

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



Activate the process.

SUPERBADGE COMPLETE!

+10000 Points

[Discover more trailmixes](#)



Completed 4/30/22
