Apex specialist super-badge

**Maintenance Request Helper**

**public with sharing class MaintenanceRequestHelper {**  
**public static void updateworkOrders(List<Case> updWorkOrders, Map<Id,Case> nonUpdCaseMap) {**  
**Set<Id> validIds = n (Case c : updWorkOrde      if (nonUpdCaseMap.get(c.Id).Status != 'Closed' && c.Status == 'Closed'){**  
**if (c.Type == 'Repair' || c.Type == 'Routine Maintenance'){**  
**validIds.add(c.I }**  
**}**  
**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 are : results){**   
**maintenanceCycles.put((Id) ar.get('Maintenance\_Request\_\_c'), (Decimal) ar.get('cycle'));**  
**}**  
**for(Case cc : closedCasesM.values()){**

**Case no = new Case (**  
**ParentId =** [**cc.Id**](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbDVmMWt1S3d2UGlBdnFrNnY0aDBtNFl3TjQtd3xBQ3Jtc0tsVjdJeFlPeTdabEJWR1o1RTFJY0dGbzdsWk1RaDEwUTVDX3RFdl90MVRNRkNuRGNRb3FDWTYwWVNPMXUwTkV0b3ExYjk0aFpOQ2NVU2Vmck9mMVNqbl80VmhGSk1xbnZwbHlhX3E1NGFwUGN5UG5YTQ&q=http%3A%2F%2Fcc.id%2F&stzid=UgxUSVPdvMN0MySSaDN4AaABAg)**,**  
**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\_Mainten**ance\_Item\_\_c>();

**or (Case nc : newCases){**  
**for (Equipment\_Maintenance\_Item\_\_c we : closedCasesM.get(nc.ParentId).Equipment\_Maintenance\_Items\_\_r){   Equipment\_Maintenance\_Item\_\_c wpClone = wp.clone();**  
**wpClone.Maintenance\_Request\_\_c =** [**nc.Id**](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqa3R2WFphc0VGWFBMcmxVUlF5dUZxaHlVdmZyd3xBQ3Jtc0trOEYyMTR6SVRNOHJESTYxMWpvLUZ4S1lTQWtEcmVUT2JLbmtnMEtnZnJBMV9scnZEazdlb2hTVG1aT2owOVJWMFVTZUdqMTd5VGw4d0Z6QjRneWtzRW1zbTNUR3lYZVFfTlZ4R2x6OHlGdEx3N19zTQ&q=http%3A%2F%2Fnc.id%2F&stzid=UgxUSVPdvMN0MySSaDN4AaABAg)**;**  
**ClonedWPs.add(wpClone);**  
**}**  
**}**  
**insert ClonedWPs;**  
**}**

**}**  
**}**

**Maintenance Request**

**trigger MaintenanceRequest on Case (before update, after update) {**  
**if(Trigger.isUpdate && Trigger.isAfter){**  
**MaintenanceRequestHelper.updateWorkOrders([Trigger.New](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqa3NyTFg2WlBzWHlpSU9CWjVpeFVScUNXZldLQXxBQ3Jtc0tsb19FZ0Nad05jVGNtU2d0LWhrUkYwY21aSFM3S0ZoSFU2UHA5Nl9mSkJCdWsyN2E1aThSR2RRdnFFRVBxVVA4ZHd2cGpzcnpBZ3BOdURya2hIQUo3Y0NCZl9RSGFjSko2YVF4OW1zMjNETEtGUjgzRQ&q=http%3A%2F%2Ftrigger.new%2F&stzid=UgxUSVPdvMN0MySSaDN4AaABAg), Trigger.OldMap);**  
**}**  
**}**

**warehousecalloutservice:**

**public with sharing class WarehouseCalloutService implements Queueable {**

**private static final String WAREHOUSE\_URL = '**[**https://th-superbadge-apex.herokuapp.com/equipment';**](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbEVrbUdzQWRCUG90QUkzYmJyZDctaWRVQ0c0Z3xBQ3Jtc0ttRWhzWUFkdlljeWdJbGZBTDFtVFdzSVUzYXlXaHB0QjRweTdBNU80VllmZFVzalF4TC1lQ191Q1V3OUxTUW1Od01nOWhNSHlXYm1IY09hZThieUlQUHlaaTBRZWtmSGlQSC0wWHgyelI0blFjOXVtSQ&q=https%3A%2F%2Fth-superbadge-apex.herokuapp.com%2Fequipment%27%3B&stzid=UgzL1Vp4L-tKOhc4pUJ4AaABAg)

**//class that makes a REST callout to an external warehouse system to get a list of equipment that needs to be updated.**  
**//The callout’s JSON response returns the equipment records that you upsert in Salesforce.**   
**@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());**  
**//class maps the following fields: replacement part (always true), cost, current inventory, lifespan, maintenance cycle, and warehouse SKU**

**//warehouse SKU will be external ID for identifying which equipment records to update within Salesforce**  
**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](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqblRyOVNHano4YTgxdmgtdUVNZjRHNHJublYyZ3xBQ3Jtc0tsZGtkWUVwU2psOGpQNUpIM0VQSThNQmt5Q3hIT2NIem0zUUR3TWJyRUlMVDVDei13cFVMUHZqWEVrYU5YMVBBYmpObzl6TWJZZ3lYZWRGMEJBdEhtYVBQbUtWVlBMc0gwZ2kwa0JUamFWWm5UZzViaw&q=http%3A%2F%2Fmyeq.name%2F&stzid=UgzL1Vp4L-tKOhc4pUJ4AaABAg) = (String) mapJson.get('name');**

**myEq.Maintenance\_Cycle\_\_c = (Integer) mapJson.get('maintenanceperiod');**  
**myEq.Lifespan\_Months\_\_c = (Integer) mapJson.get('lifespan');**

**myEq.Cost\_\_c = (Integer) mapJson.get('cost');**  
**myEq.Warehouse\_SKU\_\_c = (String) mapJson.get('sku');**

**myEq.Current\_Inventory\_\_c = (Double) mapJson.get('quantity');**  
**myEq.ProductCode = (String) mapJson.get('\_id');**  
  
  **warehouseEq.add(myEq);**  
**}**  
**if (warehouseEq.size() > 0){**  
**upsert warehouseEq;**  
**System.debug('Your equipment was synced with the warehouse one');**  
**}**  
**}**  
**}**  
**public static void execute (QueueableContext context){**  
**runWarehouseEquipmentSync();**  
**}**  
**}**

warehousesyncschedule:

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

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

**} }**

**Maintenance request helper test:**

**@istpublic with sharing class MaintenanceRequestHelperTest {**

**private static final string STATUS\_NE**

**private static final string WORKING = 'Working'**

**private static final string CLOSED = 'Closed';**  
**private static final string REPAIR = 'Repair private static final string REQUEST private static final string REQUEST\_TYP**  
**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**](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbkFFVHozMEp1T1p4NUFTdmFqbjA3azFVWkpCQXxBQ3Jtc0trMGJfSkpLLTVUU3FocWZyUzBKNUtNa3NGSzVNNG1NelZiWEJMNzZqZkQxT2JjOTI4cl9RRWE0VWZCcWYwSUw3UTAyR3VEbnQ4bW5VRGNlb1NkWUJOVU84b0x3dnU5WFo5b3B2dlZPYkFLSUltVWRodw&q=http%3A%2F%2Fvehicle.id%2F&stzid=UgzLbrAdydQfYqkhoUR4AaABAg)**;  
  
          
  
        Product2 equipment = createEq();  
  
        insert equipment;  
  
        id equipmentId =** [**equipment.Id**](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbUtDVHdYU0o0bG5KSmRrbnotVEZvX3lxZU4xZ3xBQ3Jtc0ttTDRLVWZkRnF1RFlVY2VZakZUWjdxUWIzZG1iSUZ3SHBLMldlcUxKZWF6bUwtWW5JZGE0TVdzNXFCLVUxSG1TYndkUVZoOS1rLXZRWDU2LUhoSzNmdlVvcjE2a1Z0VW9mb0JlVGtlc001YjhOSXUzMA&q=http%3A%2F%2Fequipment.id%2F&stzid=UgzLbrAdydQfYqkhoUR4AaABAg)**;  
  
          
  
        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**](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbk1QQ2JOUzRKU3pyNFc4Ulh3ajV0dDdkaXB1d3xBQ3Jtc0ttTndlYnlCbWVZbGVXVlIxOFY2X2YzNnBsV0RneFFScFNnOGdEMTQtN1B3N1ZQOFdUMnNSR1prNE5fZEFkN0JJQjJYTzJKY1dJWjg0RVFhbEtHeTB5cmFMUkpWRHlJY1dTQ0hNNFhKeEtPMnZqZ015OA&q=http%3A%2F%2Fvehicle.id%2F&stzid=UgzLbrAdydQfYqkhoUR4AaABAg)**;  
  
          
  
        product2 equipment = createEq();  
  
        insert equipment;  
  
        id equipmentId =** [**equipment.Id**](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbUFfWENfdGFld1pwbGJ4ZHROcmp5OVp6TEtvUXxBQ3Jtc0trWFRjanVVWUQtWGRmbGpPcDRpSkhkbW1HSFk3cGZ5RlpITE9aVkY3R041ZmNGRzhZVVVsRUtVRTVNN252aWpYSW4zUzBCSlFfa0ZITjBlamR5XzNwQml3WTNtSzJ2azNDRDJvdElOYktkU2lOLS1WSQ&q=http%3A%2F%2Fequipment.id%2F&stzid=UgzLbrAdydQfYqkhoUR4AaABAg)**;  
  
          
  
        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];**  
  
          
  
        E**quipment\_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, requestLis**t**.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);**  
  
    }  
  
}

**MaintenanceRequestHelper:**

**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**](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqa1hoUGN5VngwV2xRaXA5dVNBMkpUZ2s1cTZvZ3xBQ3Jtc0tuN2Y0S2ZvblNTSXQ0WVFUak43aUV3RzR1XzhFUS16RXdnTEdWRlV6XzZSX0RRMG9lczhFM2htQW5oci1pT0t2cW5GWExhNnhDX1FNdUtoYXctNGxUbjE1bU9Oa1dFX3dlRF9yZDBLQzNBRXFXZW90SQ&q=http%3A%2F%2Fcc.id%2F&stzid=UgzLbrAdydQfYqkhoUR4AaABAg)**,**  
  
**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**](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbVFjU2d4Z2lLR014NnpsTE1tb0M5ZjFka0xVd3xBQ3Jtc0trN1ZGb1libWFlanpZQ2pyU25wQ0hmaFpZRVZJNC14SVd6c2FqcEZ2cGdnWVVBMjQtVnZzbVRTZG1pdnVfTS1aN3N2dDMtVFZOQVoyNVJBeXY1ZmR1bFlsZjNGTVBYRnhPMHRUSlhkTHpNc2pyLVlvUQ&q=http%3A%2F%2Fnc.id%2F&stzid=UgzLbrAdydQfYqkhoUR4AaABAg)**;**  
**ClonedWPs.add(wpClone);}}**  
  
                      
  
                 
            **insert ClonedWPs;}}}**

**MaintenanceRequest**

**trigger MaintenanceRequest on Case (before update, after update) {  
  
    if(Trigger.isUpdate && Trigger.isAfter){  
  
        MaintenanceRequestHelper.updateWorkOrders([Trigger.New](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbHlja0FIRnp5TlJDQ3ZWQy1nMnZ6aXQ0Q3BHUXxBQ3Jtc0tuWFY1WjRuclYwSEN1eURwYlNDSzFVemlUdU1uRGVlUUdpTTBrM3dhZHAxQV9OcHR5MEpObGpWajZSYktNZmFxVi1adnBURnUwYi1OcHZRdmFNaHZlbnZpZkRmZUY2bWpQSVBsUkVVZS1jaS1DNjBTYw&q=http%3A%2F%2Ftrigger.new%2F&stzid=UgzLbrAdydQfYqkhoUR4AaABAg), Trigger.OldMap);}}**

**WarehouseCalloutService  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
  
public with sharing class WarehouseCalloutService {  
  
  
  
    private static final String WAREHOUSE\_URL = '**[**https://th-superbadge-apex.herokuapp.com/equipment';**](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbUk0Mi1pOUhRVDBsQmhfQUZtdHN6bE9vdVBPZ3xBQ3Jtc0ttNnR1eklTUl9XcENJYzdVVUVaZk5vS2xqdVp0NWdYWUFlZzhmN3R0Um5tTHJaT3MyWUk0QzFpSTdEa2o3a1FaNmNQaW1vSnNJNDFoVlFuVzVQQ2VzRkRINjNjX2wzelRST2hDc1NMeHFtVnI0SG5idw&q=https%3A%2F%2Fth-superbadge-apex.herokuapp.com%2Fequipment%27%3B&stzid=UgxQDc13tkm_fkC78MV4AaABAg) **//@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](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqa2k2U0RlU09BTm5TVExtQ3dlcmxxelVDMWxCd3xBQ3Jtc0tuNG9UMU5BenNqTldEbFVtOFBqdzJXZ2VmZzd0bUpWS28tXzdEbHhWSlJIejJrMXM4WDQtUFJyZndRNTZGenI2Wmh2MFN6Mks1dHd1OVNwNjBSRWpLZG5Hd2hKbmFUMjZaNzdCQkpiSGVQMG5nVmEzRQ&q=http%3A%2F%2Fmyeq.name%2F&stzid=UgxQDc13tkm_fkC78MV4AaABAg) = (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  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**@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  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
@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'**](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbWN5YlY3dUl4RW1RT3RORndoSXhvUll0RkQ5QXxBQ3Jtc0tuWl9nUkgxNHpGSlZCUy0tZ2Nhc0xtbWxhUmJBMUx5RGM5UVlQUGY0dmowZHlYeFBsMDV6TnFHdGNGU1BHUHdOVmhmRktUX0M0NFVRTUFERGloNnlvTHczS3lHRk1JZHZ3RUFCdFlVVEd1SHEzRE5kZw&q=https%3A%2F%2Fth-superbadge-apex.herokuapp.com%2Fequipment%27&stzid=UgxQDc13tkm_fkC78MV4AaABAg)**, 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;  
  
    }  
  
}**