|  |
| --- |
| @isTest |
|  | public with sharing class MaintenanceRequestHelperTest { |
|  |  |
|  | private static Vehicle\_\_c createVehicle(){ |
|  | Vehicle\_\_c vehicle = new Vehicle\_\_C(name = 'Testing Vehicle'); |
|  | return vehicle; |
|  | } |
|  |  |
|  | private static Product2 createEquipment(){ |
|  | product2 equipment = new product2(name = 'Testing equipment', |
|  | lifespan\_months\_\_c = 10, |
|  | maintenance\_cycle\_\_c = 10, |
|  | replacement\_part\_\_c = true); |
|  | return equipment; |
|  | } |
|  |  |
|  | private static Case createMaintenanceRequest(id vehicleId, id equipmentId){ |
|  | case cse = new case(Type='Repair', |
|  | Status='New', |
|  | Origin='Web', |
|  | Subject='Testing subject', |
|  | Equipment\_\_c=equipmentId, |
|  | Vehicle\_\_c=vehicleId); |
|  | return cse; |
|  | } |
|  |  |
|  | private static Equipment\_Maintenance\_Item\_\_c createEquipmentMaintenanceItem(id equipmentId,id requestId){ |
|  | Equipment\_Maintenance\_Item\_\_c equipmentMaintenanceItem = new Equipment\_Maintenance\_Item\_\_c( |
|  | Equipment\_\_c = equipmentId, |
|  | Maintenance\_Request\_\_c = requestId); |
|  | return equipmentMaintenanceItem; |
|  | } |
|  |  |
|  | @isTest |
|  | private static void testPositive(){ |
|  | Vehicle\_\_c vehicle = createVehicle(); |
|  | insert vehicle; |
|  | id vehicleId = vehicle.Id; |
|  |  |
|  | Product2 equipment = createEquipment(); |
|  | insert equipment; |
|  | id equipmentId = equipment.Id; |
|  |  |
|  | case createdCase = createMaintenanceRequest(vehicleId,equipmentId); |
|  | insert createdCase; |
|  |  |
|  | Equipment\_Maintenance\_Item\_\_c equipmentMaintenanceItem = createEquipmentMaintenanceItem(equipmentId,createdCase.id); |
|  | insert equipmentMaintenanceItem; |
|  |  |
|  | test.startTest(); |
|  | createdCase.status = 'Closed'; |
|  | update createdCase; |
|  | test.stopTest(); |
|  |  |
|  | Case newCase = [Select id, |
|  | subject, |
|  | type, |
|  | Equipment\_\_c, |
|  | Date\_Reported\_\_c, |
|  | Vehicle\_\_c, |
|  | Date\_Due\_\_c |
|  | from case |
|  | where status ='New']; |
|  |  |
|  | Equipment\_Maintenance\_Item\_\_c workPart = [select id |
|  | from Equipment\_Maintenance\_Item\_\_c |
|  | where Maintenance\_Request\_\_c =:newCase.Id]; |
|  | list<case> allCase = [select id from case]; |
|  | system.assert(allCase.size() == 2); |
|  |  |
|  | system.assert(newCase != null); |
|  | system.assert(newCase.Subject != null); |
|  | system.assertEquals(newCase.Type, 'Routine Maintenance'); |
|  | SYSTEM.assertEquals(newCase.Equipment\_\_c, equipmentId); |
|  | SYSTEM.assertEquals(newCase.Vehicle\_\_c, vehicleId); |
|  | SYSTEM.assertEquals(newCase.Date\_Reported\_\_c, system.today()); |
|  | } |
|  |  |
|  | @isTest |
|  | private static void testNegative(){ |
|  | Vehicle\_\_C vehicle = createVehicle(); |
|  | insert vehicle; |
|  | id vehicleId = vehicle.Id; |
|  |  |
|  | product2 equipment = createEquipment(); |
|  | insert equipment; |
|  | id equipmentId = equipment.Id; |
|  |  |
|  | case createdCase = createMaintenanceRequest(vehicleId,equipmentId); |
|  | insert createdCase; |
|  |  |
|  | Equipment\_Maintenance\_Item\_\_c workP = createEquipmentMaintenanceItem(equipmentId, createdCase.Id); |
|  | insert workP; |
|  |  |
|  | test.startTest(); |
|  | createdCase.Status = 'Working'; |
|  | update createdCase; |
|  | test.stopTest(); |
|  |  |
|  | list<case> allCase = [select id from case]; |
|  |  |
|  | Equipment\_Maintenance\_Item\_\_c equipmentMaintenanceItem = [select id |
|  | from Equipment\_Maintenance\_Item\_\_c |
|  | where Maintenance\_Request\_\_c = :createdCase.Id]; |
|  |  |
|  | system.assert(equipmentMaintenanceItem != null); |
|  | system.assert(allCase.size() == 1); |
|  | } |
|  |  |
|  | @isTest |
|  | private static void testBulk(){ |
|  | list<Vehicle\_\_C> vehicleList = new list<Vehicle\_\_C>(); |
|  | list<Product2> equipmentList = new list<Product2>(); |
|  | list<Equipment\_Maintenance\_Item\_\_c> equipmentMaintenanceItemList = new list<Equipment\_Maintenance\_Item\_\_c>(); |
|  | list<case> caseList = new list<case>(); |
|  | list<id> oldCaseIds = new list<id>(); |
|  |  |
|  | for(integer i = 0; i < 300; i++){ |
|  | vehicleList.add(createVehicle()); |
|  | equipmentList.add(createEquipment()); |
|  | } |
|  | insert vehicleList; |
|  | insert equipmentList; |
|  |  |
|  | for(integer i = 0; i < 300; i++){ |
|  | caseList.add(createMaintenanceRequest(vehicleList.get(i).id, equipmentList.get(i).id)); |
|  | } |
|  | insert caseList; |
|  |  |
|  | for(integer i = 0; i < 300; i++){ |
|  | equipmentMaintenanceItemList.add(createEquipmentMaintenanceItem(equipmentList.get(i).id, caseList.get(i).id)); |
|  | } |
|  | insert equipmentMaintenanceItemList; |
|  |  |
|  | test.startTest(); |
|  | for(case cs : caseList){ |
|  | cs.Status = 'Closed'; |
|  | oldCaseIds.add(cs.Id); |
|  | } |
|  | update caseList; |
|  | test.stopTest(); |
|  |  |
|  | list<case> newCase = [select id |
|  | from case |
|  | where status ='New']; |
|  |  |
|  |  |
|  |  |
|  | list<Equipment\_Maintenance\_Item\_\_c> workParts = [select id |
|  | from Equipment\_Maintenance\_Item\_\_c |
|  | where Maintenance\_Request\_\_c in: oldCaseIds]; |
|  |  |
|  | system.assert(newCase.size() == 300); |
|  |  |
|  | list<case> allCase = [select id from case]; |
|  | system.assert(allCase.size() == 600); |
|  | } |
|  | } |