**Apex class for creating HTTP req for the merchandise**

|  |  |
| --- | --- |
|  | @RestResource(urlMapping='/Merchandise/\*') global with sharing class MerchandiseManager {  @HttpGet global static Merchandise\_\_c getMerchandiseById() { RestRequest req = RestContext.request; String marchId = req.requestURI.substring( req.requestURI.lastIndexOf('/')+1); Merchandise\_\_c result =[SELECT Name,Description\_\_c,Price\_\_c,Total\_Inventory \_c FROM Merchandise\_\_c WHERE Id = :merchId];  return result; }  @HttpPost global static String createMerchandise(String name, String description, Decimal price, Double inventory) { Merchandise\_\_c m = new Merchandise\_\_c( Name=name, Description\_\_c=description, Price\_\_c=price, Total\_Inventory\_\_c=inventory);  insert m; return m.Id; } } |

**Apex test class for merchandise invoice:**

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 | @isTest public class testApexWorkbook {     static testmethod void testLoadData() {         // creating the test data         Test.startTest();         apexWorkbook.loadData();         List<Merchandise\_\_c> merchandise = [SELECT Id FROM Merchandise\_\_c];         List<Invoice\_Statement\_\_c> invoices = [SELECT Id FROM Invoice\_Statement\_\_c];         List<Line\_Item\_\_c> line\_items = [SELECT Id FROM Line\_Item\_\_c];         Test.stopTest();         // doing assertions          System.assertEquals(merchandise.size(), 2);         System.assertEquals(invoices.size(), 1);         System.assertEquals(line\_items.size(), 1);     } } |

**Apex Trigger to prevent deletion of invoice:**

trigger RestrictInvoiceDeletion on Invoice\_Statement\_\_c (before delete) {

for (Invoice\_Statement\_\_c invoice : [SELECT Id FROM Invoice\_Statement\_\_c WHERE Id IN (SELECT Invoice\_Statement\_\_c FROM Line\_Item\_\_c) AND Id IN :Trigger.old])

{  
Trigger.oldMap.get(invoice.Id).addError( 'Cannot delete invoice statement with line items');  
}  
}