TekSyscode

Remote site Setting:

'https://postman-echo.com/get?foo1=bar1&foo2=bar2'

|  |
| --- |
| public class tekSysHttpCallout {    public static String getArgs(Integer id)  {  String sArgs = '';  Http http = new Http();  HttpRequest request = new HttpRequest();  request.setEndpoint('https://postman-echo.com/get?foo1=bar1&foo2=bar2');  request.setMethod('GET');  HttpResponse response = http.send(request);    if (response.getStatusCode() == 200)  {    Map<String, Object> results = (Map<String, Object>) JSON.deserializeUntyped(response.getBody());  // Deserializes JSON string  Map<string,object> argsJson = (map<string,object>) results.get('args');  System.debug('Args:' + argsJson );  sArgs = string.valueof(argsJson.get('foo1'));  System.debug('sArgs = ' + sArgs );  sArgs = string.valueof(argsJson.get('foo2'));  System.debug('sArgs = ' + sArgs );  }  return sArgs ;  }        } |

Contact Count Trigger:

|  |
| --- |
| trigger tekSystemCodeContactCountonAccounts on Contact (after insert,after delete) {  Set<Id> actToUpdateId = new Set<Id>(); // var to store accountID    // check contact delete or insert contact  for (Contact cntRec : Trigger.isDelete ? Trigger.old : Trigger.new) {  //yes, you can have a contact without an account  if (cntRec.AccountId != null) {  actToUpdateId.add(cntRec.AccountId);  }  }    List<Account> actToUpdate = new List<Account>();  //two ways to do it  /\*  //using Aggregate  for (AggregateResult cC : [SELECT AccountId AcctId, Count(id) cCount FROM Contact WHERE AccountId in: actToUpdateId  GROUP BY AccountId]){  Account a = new Account();  a.Id = (Id) cC.get('AcctId');  a.Number\_of\_Contacts\_\_c = (Integer) cC.get('cCount');  actToUpdate.add(a);  }  \*/    List<Account> accountList = new List<Account>([Select id ,Name, Number\_of\_Contacts\_\_c, (Select id, Name From Contacts) from Account Where id in:actToUpdate]);  FOR(Account acc : accountList){  List<Contact> contactList = acc.Contacts;  acc.Number\_of\_Contacts\_\_c = contactList.size();  actToUpdate.add(acc);  }    update actToUpdate;  } |

Lightning App /Component:

**tekSysAccountListEditApp**

|  |
| --- |
| <aura:application extends="force:slds">  <c:tekSysAccountListEditComponent />  </aura:application> |

**tekSysAccountListEditComponent**

|  |
| --- |
| <aura:component controller=" tekSysAccountListEditController" implements="force:appHostable">  <aura:attribute access="PUBLIC" name="manageView" type="Object[]"/>  <aura:handler name="init" value="{!this}" action="{!c.getAccountRecord}" />    <div class="slds">  <table data-reactroot="" class="slds-table slds-table\_bordered slds-table\_cell-buffer">  <thead>  <tr class="slds-text-title\_caps">  <th scope="col">  <div class="slds-truncate" title="Account Name">Account Name</div>  </th>  <th scope="col">  <div class="slds-truncate" title="Industry">Industry</div>  </th>  <th scope="col">  <div class="slds-truncate" title="Phone">Phone</div>  </th>  <th scope="col">  <div class="slds-truncate" title="Action"></div>  </th>  </tr>  </thead>  <tbody>  <aura:iteration items="{!v.manageView}" var="row">  <c: tekSysAccountListEditComponentList view="{!row}"/>  </aura:iteration>  </tbody>  </table>  </div>  </aura:component> |

|  |
| --- |
| ({  getAccountRecord : function(component, event, helper) {  var action = component.get("c.getAllRecords");  action.setCallback(this, function(a)  {  if (a.getState() === "SUCCESS")  {  component.set('v.manageView',a.getReturnValue());  }  else if (a.getState() === "ERROR")  {  console.log(a.getError());  }  });  $A.enqueueAction(action);  }  }) |

|  |
| --- |
| public class tekSysAccountListEditController {  @AuraEnabled  public static List<Account> getAllRecords()  {  return [select id, Name,Industry, Phone from Account ORDER BY CreatedDate DESC LIMIT 10];  }  @AuraEnabled  public static Account saveAccount(Account account){  upsert account;  return account;  }  } |

**tekSysAccountListEditComponentList**

|  |
| --- |
| <aura:component controller=" tekSysAccountListEditController ">  <aura:attribute access="PUBLIC" name="view" type="Object"/>  <tr>  <th scope="row" data-label="Account Name">  <div class="slds-truncate" >  <ui:inputText value="{!v.view.Name}" />  </div>  </th>  <td data-label="Industry">  <div class="slds-truncate" >{!v.view.Industry}</div>  </td>  <td data-label="Close Date">  <div class="slds-truncate" >{!v.view.Phone}</div>  </td>  <td data-label="">  <div class="slds-truncate" >  <button onclick="{!c.callUpdateRecord}" class="slds-button slds-button\_icon" title="Edit Record">  <lightning:icon iconName="action:edit" size="xx-small" alternativeText="Edit"/>  <span class="slds-assistive-text">Edit</span>  </button>    </div>  </td>  </tr>  </aura:component> |

|  |
| --- |
| ({  callUpdateRecord : function(component, event, helper) {  var currentRecordId = component.get("v.view.Id");  var currentRecord = component.get("v.view");  var currentRecordName = component.get("v.view.Name");  //alert(currentRecordName);  var action = component.get("c.saveAccount");  action.setParams({"account": currentRecord});  action.setCallback(this, function() { console.log('SAVED.'); } );  $A.enqueueAction(action);  }  }) |