<aura:component controller="BeneficialOwner" implements="flexipage:availableForAllPageTypes,force:lightningQuickAction,force:hasRecordId,force:hasSObjectName,force:appHostable,lightning:actionOverride" access="global">

<aura:handler name="sampleComponentEvent" action="{!c.handleSampleEvent}" event="c:IBO\_Close" />

<aura:attribute name="acc" type="Account" default="{'sobjectType':'Account'}"/>

<aura:attribute name="isActive" type="Boolean" default="true"/>

<aura:attribute name="reqSection" type="Boolean" default="false"></aura:attribute>

<aura:attribute name="bolst" type="Beneficial\_Owner\_\_c[]"/>

<aura:attribute name="ownershipMap" type="Map"/>

<aura:attribute name="oppId" type="Id" default="006g000000IyoHaAAJ" />

<aura:handler name="init" value="{!this}" action="{!c.doInit}"/>

<aura:attribute name="recordId" type="String"/>

<lightning:overlayLibrary aura:id="overlayLib"/>

<!-- <aura:attribute name="options" type="List" default="[

{'label': 'Non - Profit', 'value': 'Non - Profit'},

{'label': 'Public Sector', 'value': 'Public Sector'},

{'label': 'Publicly Traded', 'value': 'Publicly Traded'},

{'label': 'Other', 'value': 'Other'},

]"/> -->

<aura:attribute name="options" type="List"/>

<aura:attribute name="value" type="String" default=""/>

<div class="slds-box slds-theme--default c-container">

<!--<div class="slds-box">-->

<div class="slds-notify slds-notify\_alert slds-theme\_alert-texture slds-theme\_error" role="alert">

<span class="slds-assistive-text">error</span>

<h2>As part of the Know Your Customer Initiative, ADP Governance now requires you to identify any and all individuals that own 25% or more of the company ADP will process payroll for. It is Imperative that we take the time to identify individuals and enter their information below

</h2>

</div>

<br/>

<br/>

<div>

<lightning:icon iconName="action:new\_account" size="xx-small"/>

<span style="font-weight: bold;">&nbsp; Account Ownership Status</span>

</div>

<div>

<lightning:radioGroup name="radioGroupRequired"

label="The current Account Ownership status is Multi-Partner that mandates to have Beneficial Ownership information. If this information is inappropriate would you like to change it?"

options="{!v.options}"

value="{!v.value}"

type="radio"

onchange ="{!c.onChange}"

/>

</div>

<br/>

<div class="slds slds-hide" aura:id="conditionallyVisible">

<lightning:icon iconName="action:add\_contact" size="xx-small"/>

<span style="font-weight: bold;">&nbsp; Individual Beneficial Owners</span>

<lightning:button name="modal" label="New" onclick="{!c.handleNewIBO}" class="pull-right"/>

<lightning:layout multipleRows="true">

<aura:iteration items="{!v.bolst}" var="accs1" >

<lightning:layoutItem padding="around-small" size="6">

<div id="Tilu" class="slds-box">

<lightning:tile >

<aura:set attribute="media">

<lightning:icon iconName="action:add\_contact" size="xx-small"/>

</aura:set>

<dl class="slds-dl\_horizontal" style="align:right">

<lightning:icon iconName="utility:edit" size="xx-small" />

<lightning:button label="Delete" title="Neutral action" value="{v.isActive}" onclick="{!c.handleDelete}"/>

<lightning:layoutItem size="4" smallDeviceSize="12" mediumDeviceSize="12" largeDeviceSize="12" >

<span class="slds-form-element\_\_label" for="form-element-01">First Name</span>

<ui:outputText value="{!accs1.First\_Name\_\_c}"/><br/>

</lightning:layoutItem>

<lightning:layoutItem size="4" smallDeviceSize="12" mediumDeviceSize="12" largeDeviceSize="12" >

<span class="slds-form-element\_\_label" for="form-element-01">Last Name</span>

<ui:outputText value="{!accs1.Last\_Name\_\_c}"/><br/>

</lightning:layoutItem>

<lightning:layoutItem size="4" smallDeviceSize="12" mediumDeviceSize="12" largeDeviceSize="12" >

<span class="slds-form-element\_\_label" for="form-element-01">Country</span>

<ui:outputText value="{!accs1.Country\_\_c}"/><br/>

</lightning:layoutItem>

</dl>

</lightning:tile>

</div>

</lightning:layoutItem>

</aura:iteration>

</lightning:layout>

</div>

<div class="slds slds-hide" aura:id="conditionallyVisible2">

<lightning:icon iconName="action:add\_contact" size="xx-small"/>

<span style="font-weight: bold;">&nbsp; Business as Owners</span>

<lightning:button name="modal" label="New" onclick="{!c.handleNewBO}" class="pull-right"/>

<lightning:layout multipleRows="true">

<aura:iteration items="{!v.bolst}" var="accs1" >

<lightning:layoutItem padding="around-small" size="6">

<div id="Tilu" class="slds-box">

<lightning:tile >

<aura:set attribute="media">

<lightning:icon iconName="action:add\_contact" size="xx-small"/>

</aura:set>

<dl class="slds-dl\_horizontal">

<aura:set attribute="media">

<lightning:icon iconName="action:add\_contact" size="xx-small"/>

</aura:set>

<lightning:layoutItem size="4" smallDeviceSize="12" mediumDeviceSize="12" largeDeviceSize="12" >

<span class="slds-form-element\_\_label" for="form-element-01">Company Name</span>

<ui:outputText value="{!accs1.Last\_Name\_\_c}"/><br/>

</lightning:layoutItem>

</dl>

</lightning:tile>

</div>

</lightning:layoutItem>

</aura:iteration>

</lightning:layout>

</div>

</div>

</aura:component>

**Controller**:

({

doInit : function(component, event, helper) {

helper.myAction(component, event, helper);

var action =component.get("c.getAllBO");

console.log('The action value is: '+action);

action.setCallback(this, function(a){

component.set("v.bolst", a.getReturnValue());

console.log('result');

console.log(component.get("v.bolst"));

});

$A.enqueueAction(action);

},

handleNewIBO: function(component, evt, helper) {

var modalBody;

//alert(component.get("v.oppId"));

$A.createComponent("c:IndividualBusinessOwner", {oppId :component.get("v.oppId")},

function(content, status) {

if (status === "SUCCESS") {

modalBody = content;

component.find('overlayLib').showCustomModal({

header: "New Individual Beneficial Owner",

body: modalBody,

showCloseButton: true,

cssClass: "mymodal",

closeCallback: function() {

}

})

}

});

},

handleNewBO: function(component, evt, helper) {

var modalBody;

$A.createComponent("c:BusinessAsOwner", {oppId :component.get("v.oppId")},

function(content, status) {

if (status === "SUCCESS") {

modalBody = content;

component.find('overlayLib').showCustomModal({

header: "New Business as User",

body: modalBody,

showCloseButton: true,

cssClass: "mymodal",

closeCallback: function() {

}

})

}

});

},

onChange:function(cmp,event,helper){

var optionalDiv= cmp.find("conditionallyVisible");

var optionalDiv2= cmp.find("conditionallyVisible2");

var chkVal= cmp.get("v.options");

if(chkVal){

$A.util.removeClass(optionalDiv, 'slds-hide');

$A.util.removeClass(optionalDiv2, 'slds-hide');

cmp.set("v.reqSection", true);

}

else{

$A.util.addClass(optionalDiv, 'slds-hide');

$A.util.addClass(optionalDiv2, 'slds-hide');

cmp.set("v.reqSection", false);

}

},

handleSampleEvent:function(cmp,event,helper){

var message = event.getParam("message");

alert(message);

alert('hi');

},

handleDelete : function(component, event, helper) {

var a = component.get("v.isActive",true);

a.isActive\_\_c = true;

component.set("v.isActive",a);

var btnClicked = event.getSource();

btnClicked.set("v.disabled",false);

}

})

**Helper**

({

myAction : function(component, event, helper) {

var action = component.get("c.getPicklistOptions");

var self=this;

action.setCallback(this, function(result){

if (component.isValid() && result.getState() === 'SUCCESS'){

var options = [];

var resultSet = result.getReturnValue();

/\* options.push({

'value': '--None--',

'label': '--None--'

});\*/

console.log(resultSet);

for (var i = 0; i<resultSet.length; i++){

options.push({value: resultSet[i].value, label: resultSet[i].label});

if(resultSet[i].isDefault == true){

//component.set("v.value", resultSet[i].value);

}

}

component.set("v.options", options);

this.getPickVal(component, event, helper);

}

});

$A.enqueueAction(action);

},

//get Industry Picklist Value

getPickVal : function(component, event, helper) {

var action = component.get("c.getRecordPickval");

var recId = component.get("v.recordId");

action.setParams({"recordId":recId});

action.setCallback(this, function(response) {

var state = response.getState();

if (state === "SUCCESS") {

var result = response.getReturnValue();

if(result!=null || result!='')

component.set("v.value", result);

/\* var industryMap = [];

for(var key in result){

industryMap.push({label: result[key], value: key});

}

component.set("v.industryMap", industryMap);\*/

}

});

$A.enqueueAction(action);

},

})

**Apex Class:**

public class BeneficialOwner {

@AuraEnabled

public static List<SelectOptionObj> getPicklistOptions() {

List<SelectOptionObj> picklistOptions = new List<SelectOptionObj>();

try {

Schema.DescribeFieldResult fieldResult = Account.Ownership.getDescribe();

List<Schema.PicklistEntry> ple = fieldResult.getPicklistValues();

SelectOptionObj option = null;

Integer count = 0;

for (Schema.PicklistEntry pickListVal : ple) {

option = new SelectOptionObj(pickListVal.getLabel(), pickListVal.getValue(), pickListVal.isDefaultValue());

picklistOptions.add(option);

count++;

if(count == 4){

option = new SelectOptionObj('Other','Other', false);

picklistOptions.add(option);

break;

}

}

} catch (Exception ex) {

throw new AuraHandledException('Exception in getting PicklistOptions');

}

return picklistOptions;

}

@AuraEnabled

public static String getRecordPickval(String recordId) {

String val =null;

try {

Opportunity[] acc = [Select Id, Account.Ownership from Opportunity where Id = '006g000000IyoHaAAJ'];

List<SelectOptionObj> records = getPicklistOptions();

if(acc.size()!=0)

{

for(SelectOptionObj rec:records)

{

if(acc[0].Account.Ownership.containsIgnoreCase(rec.value)){

val = acc[0].Account.Ownership;

return val;

}

else {

val = 'Other';

}

}

}

return val;

}catch (Exception ex) {

throw new AuraHandledException('Exception in getting PicklistOptions val');

}

}

public class SelectOptionObj {

@AuraEnabled

public String label { get; set; }

@AuraEnabled

public String value { get; set; }

@AuraEnabled

public Boolean isDefault {get; set;}

public SelectOptionObj(String label, String val,Boolean isDefault) {

this.label = label;

this.value = val;

this.isDefault = isDefault;

}

}

/\* @auraenabled

public static string getAccountOwnerShip(String oppId){

string str = [Select Id, Account.ownership from Opportunity where id= '006g000000IyoHaAAJ'].Account.ownership;

return str;

}\*/

/\*@auraenabled

public static string getAccountOwnerShip(String oppId){

string str = [Select Id, AccountId,Account.Ownership from Opportunity where Id='006g000000IyoHaAAJ'].Account.Ownership;

return str;

}\*/

@auraEnabled

public static list<Beneficial\_Owner\_\_c > getAllBO()

{

list<Beneficial\_Owner\_\_c> bo = new list<Beneficial\_Owner\_\_c>();

bo =[select id,First\_Name\_\_c ,Last\_Name\_\_c, Country\_\_c from Beneficial\_Owner\_\_c where Last\_Name\_\_c != NULL LIMIT 2];

return bo;

}

@auraEnabled

public static list<Beneficial\_Owner\_\_c > getAllBO\_LNNULL()

{

list<Beneficial\_Owner\_\_c> bolnull = new list<Beneficial\_Owner\_\_c>();

bolnull= [select id,First\_Name\_\_c ,Last\_Name\_\_c, Country\_\_c from Beneficial\_Owner\_\_c where Last\_Name\_\_c = NULL LIMIT 2 ];

return bolnull;

}

@auraenabled

public static Boolean processRecords(Beneficial\_Owner\_\_c objbo, String oppId, String reqType){

Id accId = [Select Id, AccountId,Account.Ownership from Opportunity where Id=:oppId].AccountId;

system.debug('objbo'+objbo);

for(Beneficial\_Owner\_Relationship\_\_c bor: [Select Id,Account\_\_c,Beneficial\_Owner\_\_r.First\_Name\_\_c ,Beneficial\_Owner\_\_r.Last\_Name\_\_c ,Beneficial\_Owner\_\_r.Street\_\_c ,Beneficial\_Owner\_\_r.City\_\_c from Beneficial\_Owner\_Relationship\_\_c where Account\_\_c=:accId]){

if(reqType== 'Individual'&& objbo.Last\_Name\_\_c == bor.Beneficial\_Owner\_\_r.Last\_Name\_\_c && objbo.City\_\_c == bor.Beneficial\_Owner\_\_r.City\_\_c){

return false;

}

if(reqType == 'Business' && objbo.First\_Name\_\_c == bor.Beneficial\_Owner\_\_r.First\_Name\_\_c && objbo.City\_\_c == bor.Beneficial\_Owner\_\_r.City\_\_c){

return false;

}

}

insert objbo;

Beneficial\_Owner\_Relationship\_\_c bor1 = new Beneficial\_Owner\_Relationship\_\_c();

bor1.Account\_\_c = accId;

bor1.Beneficial\_Owner\_\_c = objbo.Id;

insert bor1;

return true;

}

}