Lista 01

<aura:component implements="force:appHostable,flexipage:availableForAllPageTypes,flexipage:availableForRecordHome,force:hasRecordId" access="global" >

<div class="slds-box slds-theme\_default" aura:id="leads">

<lightning:recordEditForm aura:id="recordEditForm"

objectApiName="Lead"

onsuccess="{!c.showToast}">

<lightning:messages />

<div class="slds-grid">

<div class="slds-col slds-p-horizontal\_medium">

<lightning:inputField fieldName="LastName" />

</div>

<div class="slds-col slds-p-horizontal\_medium">

<lightning:inputField fieldName="Company"/>

<lightning:inputField fieldName="Status"/>

</div>

</div>

<lightning:button class="slds-m-top\_small" type="submit" label="Novo Lead" />

</lightning:recordEditForm>

</div>

</aura:component>

({

showToast: function( component, event, helper){

var toastEvent = $A.get("e.force:showToast");

toastEvent.setParams({

"title":"Sucesso",

"message": "Seu lead foi criado!",

"type": "success"

});

toastEvent.fire();

$A.get('e.force:refreshView').fire();

}

})

Lista 02

public class OpportunityC {

@AuraEnabled

public static List<Opportunity> getOpportunitiesNR(){

return[

SELECT Name, Amount, StageName, closeDate

FROM opportunity

where stageName = 'Negotiation/Review'

ORDER BY createdDate desc

Limit 5

];

}

}

<aura:component controller="lista" implements="flexipage:availableForAllPageTypes" access="global" >

<aura:attribute name="account" type="Account[]" />

<!--aura:handler-->

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

<lightning:card iconName="standard:opportunity" title="Contas em Negociação">

<div class="slds-p-around\_medium">

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

<p>------------</p>

<p>{!opp.Name}</p>

<p>{!opp.Amount}</p>

<p>{!opp.StageName}</p>

<p>{!opp.CloseDate}</p>

</aura:iteration>

</div>

</lightning:card>

</aura:component>

({

handleRefresh : function(component, event, helper) {

helper.loadAccounts(component);

},

handleSort : function(component, event, helper) {

var sortBy = event.getParam('fieldName');

var sortDirection = event.getParam('sortDirection');

component.set("v.sortField", sortBy);

component.set("v.sortOrder", sortDirection);

helper.sortAccounts(component, sortBy, sortDirection);

}

})

({

loadAccounts : function(component) {

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

action.setCallback(this, function(response) {

var state = response.getState();

if (state === "SUCCESS") {

component.set("v.accounts", response.getReturnValue());

}

});

$A.enqueueAction(action);

},

sortAccounts : function(component, sortBy, sortDirection) {

var accounts = component.get("v.accounts");

var reverse = sortDirection !== 'asc';

accounts.sort(this.sortBy(sortBy, reverse));

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

},

sortBy : function(field, reverse, primer) {

var key = primer

? function(x) { return primer(x[field]); }

: function(x) { return x[field]; };

reverse = !reverse ? 1 : -1;

return function(a, b) {

return a = key(a), b = key(b), reverse \* ((a > b) - (b > a));

};

}

})

LISTA 03

public class classexer3 {

@AuraEnabled

public static list<account> getlistacc(){

return [select Name, Estagio\_\_c, Rating from account where Rating like 'Hot'];

}

}

<aura:component controller ="classexer3" implements="force:appHostable,flexipage:availableForAllPageTypes,flexipage:availableForRecordHome,force:hasRecordId" access="global" >

<aura:attribute name="accs" type="Account[]"/>

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

<lightning:card iconName="standard:account" title="Contas em Negociação">

<div class='slds-p-arround\_medium'>

<aura:iteration items="{!v.accs}" var="accounts">

<p>--------------------------------------------------</p>

<p>Nome: {!accounts.Name}</p>

<p>Estágio: {!accounts.Estagio\_\_c}</p>

<p>Rating: {!accounts.Rating}</p>

</aura:iteration>

</div>

</lightning:card>

</aura:component>

({

doInit : function(component, event, helper) {

helper.getAccounts(component);

}

})

({

getAccounts : function(component) {

let action = component.get("c.getlistacc");

action.setCallback(this, function(response){

let state = response.getState();

console.log(state);

if(state =="SUCCESS"){

component.set("v.accs", response.getReturnValue());

console.log(response.getReturnValue());

}

});

$A.enqueueAction(action);

}

})

LISTA 04

public class classex4 {

@AuraEnabled

public static list<Lista\_Produtos\_\_c> matCompra(){

return [select Name,Produto\_\_c, Valor\_\_c, Quantidade\_\_c From Lista\_Produtos\_\_c];

}

}

<aura:component controller="classex4" implements="force:appHostable,flexipage:availableForAllPageTypes,flexipage:availableForRecordHome,force:hasRecordId" access="global" >

<div class="slds-box slds-theme\_default">

<lightning:recordEditForm aura:id="recordEditForm"

objectApiName="Lista\_Produtos\_\_c"

onsuccess="{!c.showToast}">

<lightning:messages />

<lightning:inputField fieldName="Name" />

<lightning:inputField fieldName="Produto\_\_c" />

<lightning:inputField fieldName="Valor\_\_c" />

<lightning:inputField fieldName="Quantidade\_\_c" />

<lightning:button class="slds-m-top\_small" type="submit" label="Adicionar a Lista" />

</lightning:recordEditForm>

</div>

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

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

<aura:iteration items="{!v.getList}" var="con">

<lightning:card title="Lista materiais a comprar">

<table class="slds-table slds-table--bordered slds-table--striped">

<thead>

<tr>

<th>Produto</th>

<th>Valor</th>

<th>Qtde</th>

</tr>

</thead>

<tbody>

<aura:iteration items="{!v.getList}" var="con">

<tr>

<td>{!con.Produto\_\_c}</td>

<td>{!con.Valor\_\_c}</td>

<td>{!con.Quantidade\_\_c}</td>

</tr>

</aura:iteration>

</tbody>

</table>

</lightning:card>

</aura:iteration>

</aura:component>

({

myAction : function(component, event, helper) {

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

action.setCallback(this,function(response){

var state = response.getState();

if(state == 'SUCCESS'){

var returnData = response.getReturnValue();

component.set("v.getList", returnData )

}

});

$A.enqueueAction(action)

},

showToast: function( component, event, helper){

var toastEvent = $A.get("e.force:showToast");

toastEvent.setParams({

"title":"Sucesso",

"message": "Seu item foi adicionado com sucesso",

"type": "success"

});

toastEvent.fire();

$A.get('e.force:refreshView').fire();

}

})