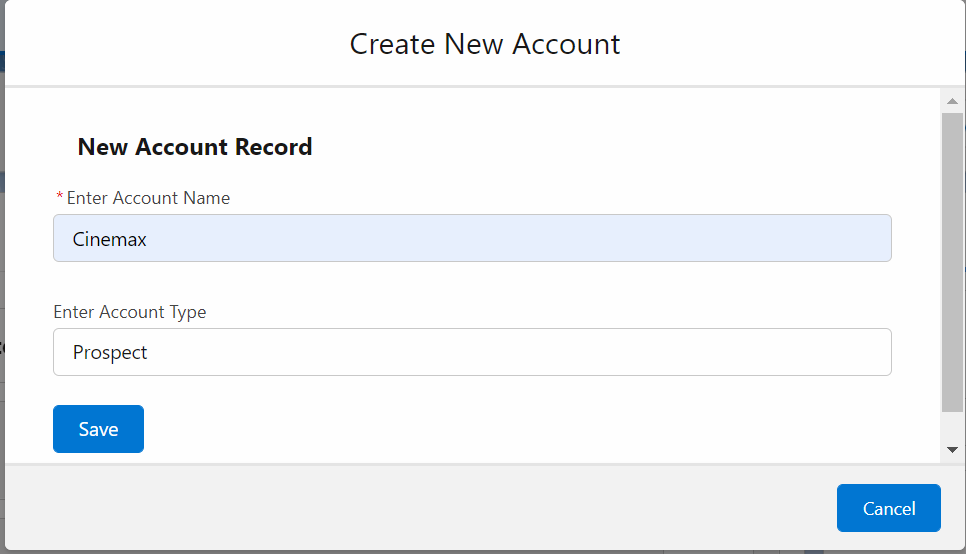
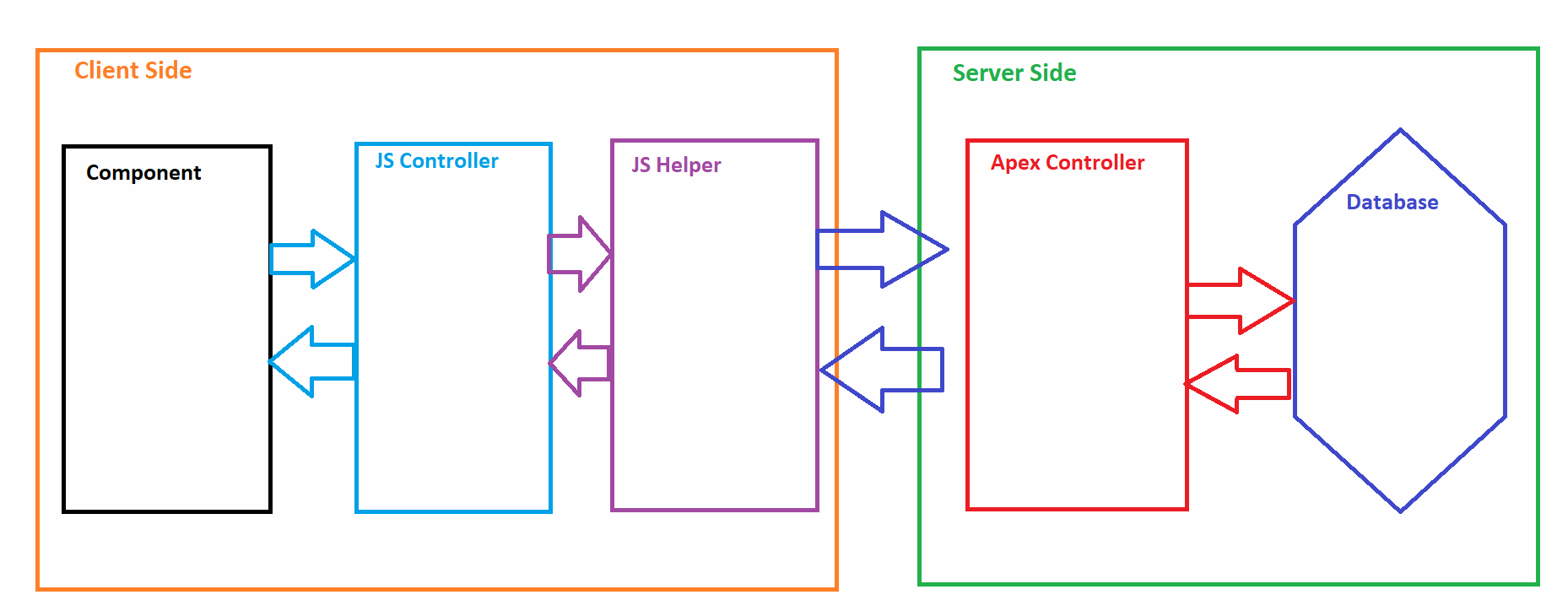
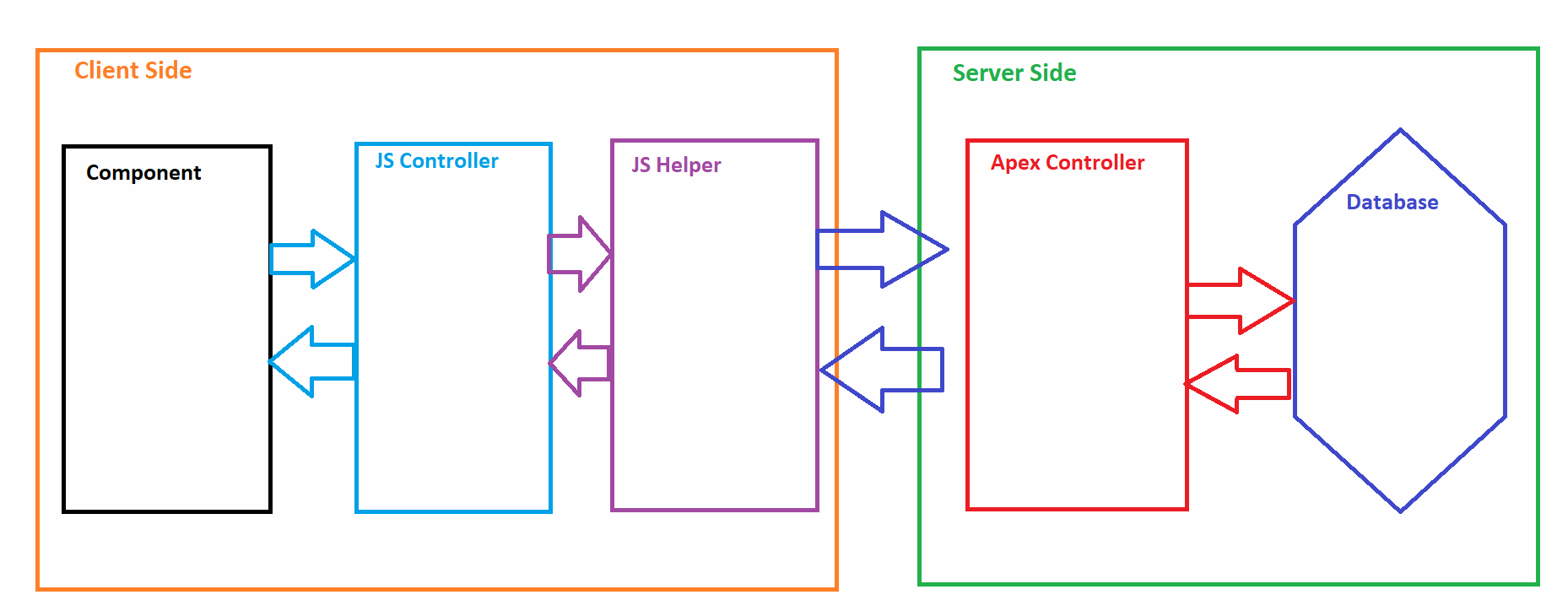
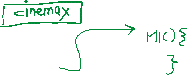
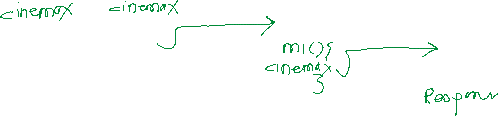
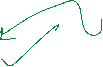
78. Custom Aura Lightning Component - 28 July 2022

User Story 1] Create a new quick action button “Aura Custom New Account” to create a new Account Record.









STEPS TO CALL APEX METHOD

1] Method must be static

2] Method should be annotated at @AuraEnabled

3] Provide Controller Name in the component

var action = component.get('c.createNewAccount');

action.setParams({'objAcc' : component.get('v.accountObject')});

action.setCallback(this, function(response){

var state = response.getState();

if(state === 'SUCCESS'){

console.log('success');

}

else{

console.log('error');

}

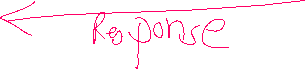
})

&A.enqueueAction(action);

$A.enqueueAction(action); //3











FULL CODE:

COMPONENT

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

<aura:attribute name="accountObject" type="Account" default="{'sObjectType' :'Account'}" />

<lightning:card>

<lightning:input type="text" name="accName" label="Enter Account Name" required="true" value="{!v.accountObject.Name}"/> <br/>

<lightning:input type="text" name="accType" label="Enter Type" value="{!v.accountObject.Type}"/> <br />

<lightning:input type="text" name="accSLA" label="Enter SLA" value="{!v.accountObject.SLA\_\_c}"/> <br/>

<lightning:button variant="brand" label="Save" title="Save" onclick="{!c.createAccountController}" />

</lightning:card>

</aura:component>

JS CONTROLLER:

({

createAccountController : function(component, event, helper) {

alert("Inside of JS Controller");

helper.createAccountHelper(component, event, helper);

}

})

JS HELPER:

({

createAccountHelper : function(component, event, helper) {

var action = component.get('c.createNewAccount'); //1

action.setParams({'objAcc' : component.get('v.accountObject')}); //2

$A.enqueueAction(action); //3

action.setCallback(this, function(response){

if(response.getState() === 'SUCCESS'){

console.log('success '+JSON.stringify(response.getReturnValue()));

}

else{

console.log('error');

}

});

}

})

Apex Controller:

public with sharing class AccountProvider {

@AuraEnabled

public static string createNewAccount(Account objAcc){

try {

if(objAcc != null){

Insert objAcc;

return 'New Account Record Created Successfully...!!!';

}

else{

return 'Locha, something went wrong...!!!';

}

} catch (Exception e) {

throw new AuraHandledException(e.getMessage());

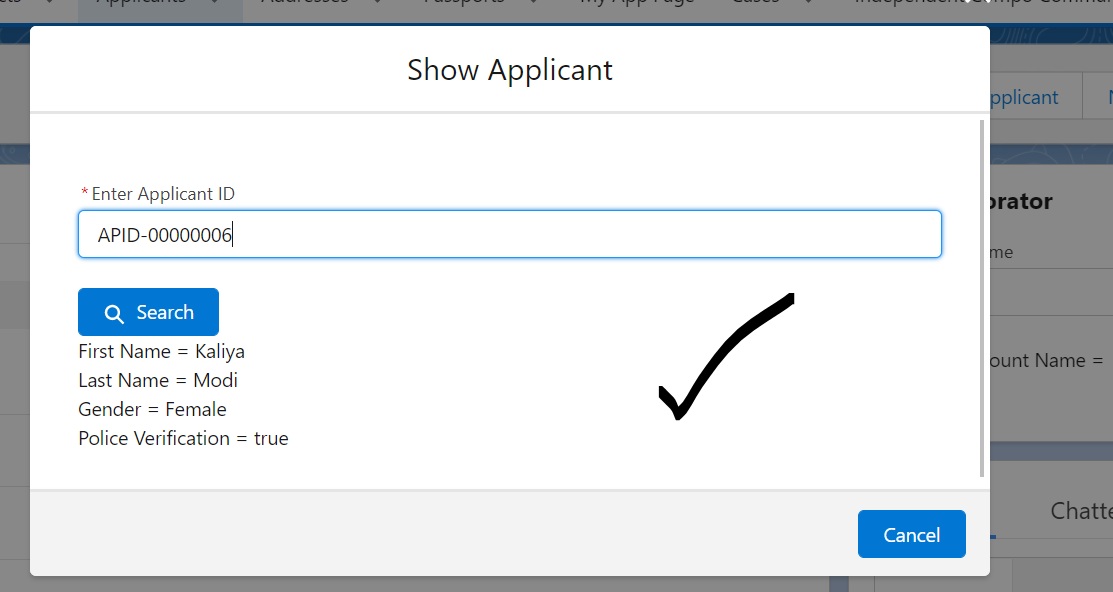
}

}

}

=============================================================================

2] Search Applicant Name and Show its First Name, Last Name, Gender.



COMPONENT:

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

<aura:attribute name="applicantObject" type="Applicant\_\_c" default="{'sObjectType' : 'Applicant\_\_c'}" />

<lightning:card>

<lightning:input type="text" name="appName" label="Enter Applicant ID" required="true" value="{!v.applicantObject.Name}"/> <br/>

<lightning:button variant="brand" iconName="utility:search" label="Search" title="Search" onclick="{!c.searchApplicantController}" /> <br/>

First Name = {!v.applicantObject.First\_Name\_\_c} <br/>

Last Name = {!v.applicantObject.Last\_Name\_\_c} <br />

Gender = {!v.applicantObject.Gender\_\_c} <br/>

Police Verification = {!v.applicantObject.Police\_Verification\_\_c}

</lightning:card>

</aura:component>

JS CONTROLLER:

({

searchApplicantController : function(component, event, helper) {

helper.searchApplicantHelper(component, event, helper);

}

})

JS HELPER:

({

searchApplicantHelper : function(component, event, helper) {

var action = component.get('c.searchApplicantRecord'); //1

action.setParams({'objApp' : component.get('v.applicantObject')});

$A.enqueueAction(action);

action.setCallback(this, function(response){

var state = response.getState();

if(state === 'SUCCESS'){

console.log('sucess '+ JSON.stringify(response.getReturnValue()));

component.set('v.applicantObject' , response.getReturnValue());

}

else{

console.log('error ');

}

});

}

})

Apex Controller:

public class ApplicantProvider {

@AuraEnabled

public static Applicant\_\_c searchApplicantRecord(Applicant\_\_c objApp){

try {

if(objApp != null){

return ([select Id, First\_Name\_\_c, Last\_Name\_\_c, Gender\_\_c, Police\_Verification\_\_c from Applicant\_\_c where Name=: objApp.Name LIMIT 1] );

}

else{

return null;

}

} catch (Exception e) {

throw new AuraHandledException(e.getMessage());

}

}

}

Assignment:

1] Create an Applicant Record. Show Component in the Utility Bar.