55. JS Controller Structure , Create New Records - 22 June 2022

Today we will learn:



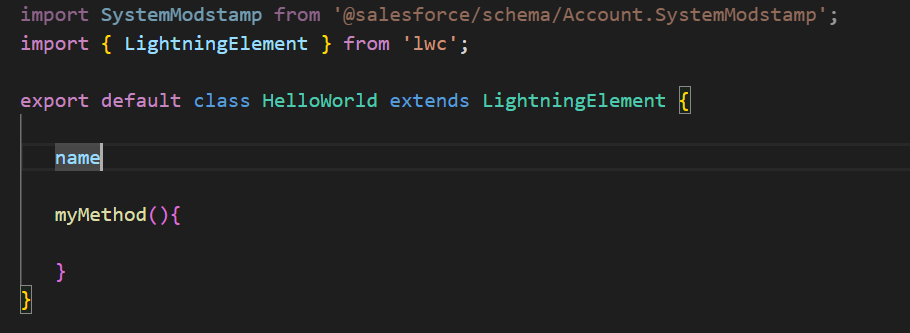
1] JS Controller Structure

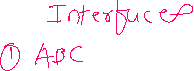
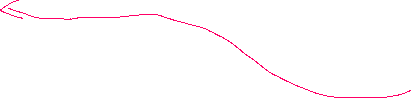


2] Create New Records and Calling Apex Method from JS Controller Method

3] data-formfield in Input Field

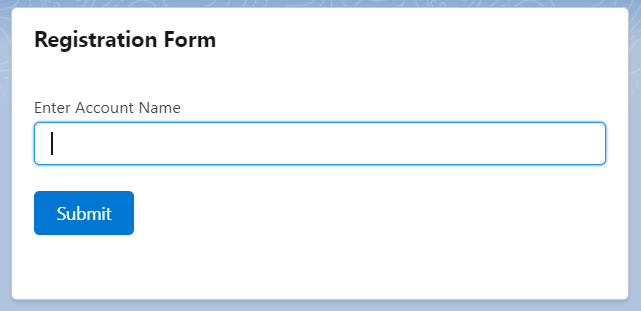
**1] JS Controller Structure**





**2] Create New Records and Calling Apex Method from JS Controller Method**

Create New Records



|  |  |
| --- | --- |
| HTML | |
|  | |
| *JS CONTROLLER* | **APEX CLASS** |

**STEPS :**

|  |  |
| --- | --- |
| 1] **Apex Controller** | Create / Reuse Apex Controller Method |
| 2] **HTML** | Create data-formfield (**Input Field**) |
| 3] **JS Controller** | A] Import Apex Controller Method  B] get input values and store into the object.  C] Call Apex Controller method inside of JS Controller Method |

Details STEPS

|  |  |  |
| --- | --- | --- |
| 1]**Apex Controller** | Create / Reuse Apex Controller Method | public class AccountProvider{  @AuraEnabled  public static void createRecord(Account objAcc){  insert objAcc;  }  } |
| 2] **HTML** | Create data-formfield (Input Field) | <lightning-input data-formfield="accountName" type="text" label="Enter Account Name" name="firstName"></lightning-input> |
| 3] **JS Controller** | A] Import Apex Controller Method  B] get input values and store into the object.  C] Call Apex Controller method inside of JS Controller Method | A] import createRecord from "@salesforce/apex/AccountProvider.createRecord";  B] this.accountObject.Name = this.template.querySelector('lightning-input[data-formfield="accountName"] ').value;  C] **showName(){**  this.accountObject.Name = this.template.querySelector('lightning-input[data-formfield="accountName"] ').value;  createAccount({ accName : this.accountName})  .then((result) => {  this.result = result;  this.error = undefined;  })  .catch((error) => {  this.error = error;  this.result = undefined;  }); |

<template>

    <lightning-card  title="Registration Form">

        <lightning-input type="text" label="Enter Account Name" name="accName" ></lightning-input>

        <lightning-button variant="brand" label="Submit" title="Submit" onclick={createAccountRecord} class="slds-m-left\_x-small" ></lightning-button>

    </lightning-card>

</template>

import SystemModstamp from '@salesforce/schema/Account.SystemModstamp';

import { LightningElement } from 'lwc';

import createAccount from '@salesforce/apex/AccountProvider.createAccount'

export default class HelloWorld extends LightningElement {

   accountName;

    createAccountRecord(){

       this.accountName =   this.template.querySelector('lightning-input').value; //Cinemax

       console.log(this.accountName);

       createAccount({ accName : this.accountName})

       .then((result) =>{

            console.log("Result = "+result);

       } )

       .catch((error) =>{

            console.log("Error ="+error);

       });

   }

}

public with sharing class AccountProvider {

@AuraEnabled

public static string createAccount(String accName){

    try {

                if(!String.isBlank(accName)){

                Account objAcc = new Account(Name = accName);

                insert objAcc;

                return 'Account Record Created successfully...';

    }

            else{

                return 'Locha..!!!';

            }

    } catch (Exception e) {

        throw new AuraHandledException(e.getMessage());

    }

}

}

Error Types:

1] Syntactic Error

2] Logic Error

Exception Handling:

Syntax:

try{

//Logic

}

catch(Exception ex){

//Exception Handle

}

try{

integer x=1;

integer y=0;

integer z= (x/y);

System.debug(z);

}

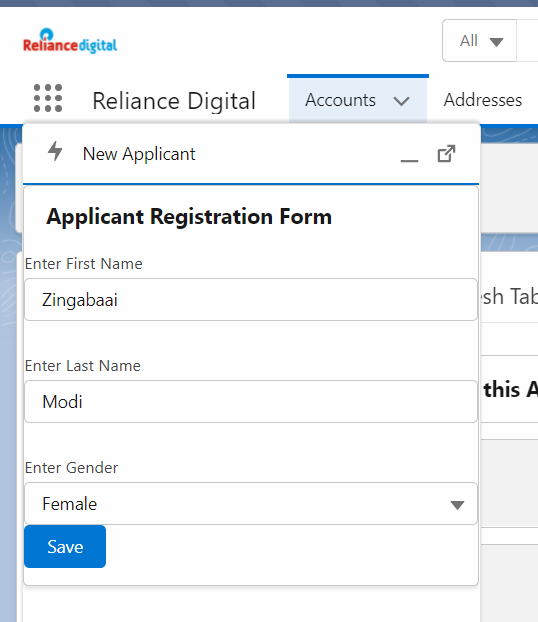
catch(Exception ex){

System.debug('Locha '+ex);

}

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

Create Applicant Form as mentioned below and show this form in Utility Bar.



**Input Field : HTML**

<lightning-input **data-formfield**="accountName" type="text" label="Enter Account Name" name="accName"> </lightning-input> <br />

**JS Controller:**

Get Input Data:

this.template.querySelector('lightning-input[data-formfield="accountName"]').value;

<template>

    <lightning-card title="Applicant Registration Form">

        <lightning-input  data-formfield="firstName"  type="text" label="Enter First Name" name="firstName"> </lightning-input> <br />

        <lightning-input  data-formfield="lastName"  type="text" label="Enter Last Name" name="lastName"> </lightning-input> <br />

        <lightning-combobox

            name="gender"

            label="Select Gender"

            value={gender}

            placeholder="Select Gender"

            options={options}

            onchange={handleChange} ></lightning-combobox> <br />

            <lightning-button label="Submit" onclick={submitRecord}></lightning-button>

    </lightning-card>

</template>

import { LightningElement } from 'lwc';

import createNewApplicantRecord from '@salesforce/apex/ApplicantProvider.createNewApplicantRecord';

export default class ApplicantRegistrationForm extends LightningElement {

     objApp = {'sObjectType' : 'Applicant\_\_c'}

   get options() {

        return [

            { label: 'Male', value: 'Male' },

            { label: 'Female', value: 'Female' },

            { label: 'Transgender', value: 'Transgender' },

        ];

    }

    handleChange(event){

      this.objApp.Gender\_\_c = event.detail.value;

    }

    submitRecord(){

        this.objApp.First\_Name\_\_c = this.template.querySelector('lightning-input[data-formfield="firstName"]').value;

        this.objApp.Last\_Name\_\_c = this.template.querySelector('lightning-input[data-formfield="lastName"]').value;

        //Calling Apex Method createNewApplicantRecord

        createNewApplicantRecord({objApplicant : this.objApp })

        .then((result) =>{

                console.log(result);

        })

        .catch((error) =>{

            console.log(error);

        });

    }

}

<?xml version="1.0" encoding="UTF-8"?>

<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">

    <apiVersion>54.0</apiVersion>

    <isExposed>true</isExposed>

<targets>

    <target>lightning\_\_UtilityBar</target>

</targets>

</LightningComponentBundle>

public with sharing class ApplicantProvider {

    @AuraEnabled

    public static string createNewApplicantRecord(Applicant\_\_c objApplicant){

        try {

            if(objApplicant != null ){

                    insert objApplicant;

                    return 'Applicant Record Created Successfully...!!!';

            }

            else{

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

            }

        } catch (Exception e) {

            throw new AuraHandledException(e.getMessage());

        }

    }

}

Multi Parameters:

 createNewApplicantRecord( { objApplicant : this.objApp , name : this.myName }  )

**Assignment:**

