67. @track Decorator, Record Form in LDS, CSS - 07 July 2022

Today we will learn:

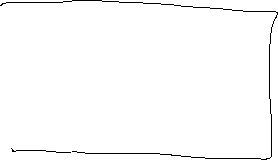
1] @track Decorator

2] Record Form

3] CSS

1] @track Decorator

It is used to make a variable reactive.



<template>

    <lightning-card title="Track Decorator">

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

            You Entered Account Name = {accObject.Name} <br/>

        <lightning-button variant="brand" icon-name="utility:send" label="Show" onclick={accountNameHandler}></lightning-button>

    </lightning-card>

</template>

import { LightningElement, track } from 'lwc';

export default class AccountTrackCompo extends LightningElement {

    @track accObject = {'sObjectType' : 'Account'}; //Reactive

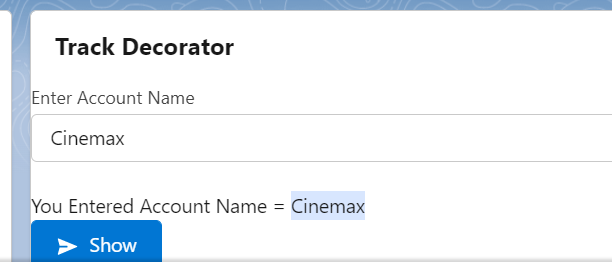
    accountNameHandler(){

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

        console.log(this.accObject.Name);

    }

}



03 Decorators

|  |  |
| --- | --- |
| Decorator Name | Purpose |
| @api | It is used to expose / public properties of JS Controller. |
| @wire | It is used in LMS (MessageContext) and also used to access the Apex Controller Method and Fields Directly. |
| @track | It is used to make a variable reactive and private. |

2] Record Form (LDS = Lightning Data Services)

LDS is used to perform CRUD operation without writing single line of apex code.

<https://developer.salesforce.com/docs/component-library/documentation/en/lwc/data_create_record>

CRUD = Create, Read, Update and Delete

A] Create a Record

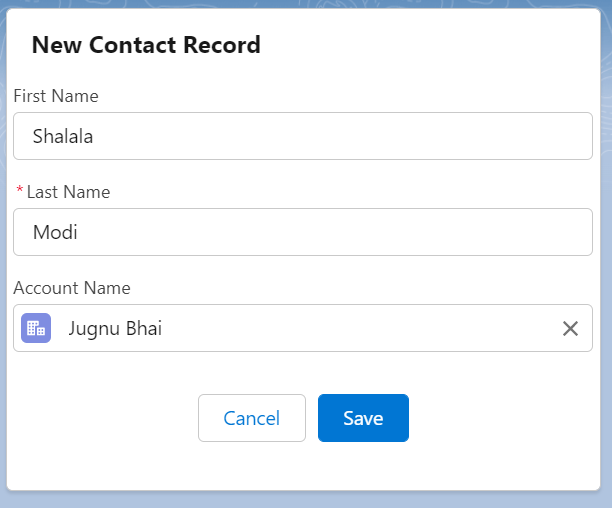
B] Edit a Record

C] View a Record

D] Delete a Record

E] To show lookup field

A] Create a Contact Record



<template>

    <lightning-card title="New Contact Record">

    <lightning-record-form

        object-api-name={contactObject}

        fields={myFields}

        onsuccess={handleContactCreated}>

    </lightning-record-form>

</lightning-card>

</template>

import { LightningElement } from 'lwc';

import CONTACT\_OBJECT from '@salesforce/schema/Contact';

import FIRST\_NAME\_FIELD from '@salesforce/schema/Contact.FirstName';

import LAST\_NAME\_FIELD from '@salesforce/schema/Contact.LastName';

import ACCOUNT\_FIELD from '@salesforce/schema/Contact.AccountId';

export default class NewContactRecordCompo extends LightningElement {

    contactObject = CONTACT\_OBJECT;

    myFields = [FIRST\_NAME\_FIELD, LAST\_NAME\_FIELD,ACCOUNT\_FIELD];

    handleContactCreated(){

        console.log('Created'); //ToastMessage

    }

}

After Record Created

onsuccess={handleContactCreated}>

Before Record Created and Call the Apex Controller Method

onclick={handleContactCreated}>

Assignment:

1] Create a New Passport Record using LDS.

2] Create a New Passport Record using LDS. Show Trigger Error when Status is ‘Hold’ on Front-End.

Pura Code Nahi hai ye, Call Apex Controller Method and then show error in TOAST MESSAGE

<template>

    <lightning-record-edit-form object-api-name={passportObject} onclick={handlePassportRecord}>

        <lightning-messages></lightning-messages>

        <div class="slds-grid">

            <div class="slds-col slds-size\_1-of-2">

                <lightning-input-field field-name={passNumber} onchange={handlerPassNumberChange}></lightning-input-field>

                <lightning-input-field field-name={passStatus} onchange={handlerPassStatusChange}></lightning-input-field>

                <lightning-input-field field-name={applicant} onchange={handlerApplicantChange}></lightning-input-field>

            </div>

         </div>

        <lightning-button type="submit" variant="brand" label="Create Passport"></lightning-button>

    </lightning-record-edit-form>

</template>

import { LightningElement } from 'lwc';

import PASSPORT\_OBJECT from '@salesforce/schema/Passport\_\_c';

import PASSPORT\_NUMBER\_FIELD from '@salesforce/schema/Passport\_\_c.Passport\_Number\_\_c';

import STATUS\_FIELD from '@salesforce/schema/Passport\_\_c.Status\_\_c';

import ISSUE\_DATE\_FIELD from '@salesforce/schema/Passport\_\_c.Issue\_Date\_\_c';

import APPLICANT\_FIELD from '@salesforce/schema/Passport\_\_c.Applicant\_\_c';

export default class NewContactRecordCompo extends LightningElement {

    passportObject = PASSPORT\_OBJECT;

    passNumber = PASSPORT\_NUMBER\_FIELD;

    passStatus = STATUS\_FIELD;

    applicant = APPLICANT\_FIELD;

    objPassport={'sObjectType' : 'Passport\_\_c'};

    handlerApplicantChange(event){

       this.objPassport.Applicant\_\_c =  event.target.value;

       console.log(event.target.value);

    }

    handlerPassStatusChange(event){

        this.objPassport.Status\_\_c =    event.target.value;

    }

    handlerPassNumberChange(event){

        this.objPassport.Passport\_Number\_\_c =    event.target.value;

    }

    handlePassportRecord(){

    }

}

Load Record

<template>

    <lightning-record-form

        record-id={recordId}

        object-api-name={objectApiName}

        fields={fields}>

    </lightning-record-form>

</template>

import { LightningElement, api } from 'lwc';

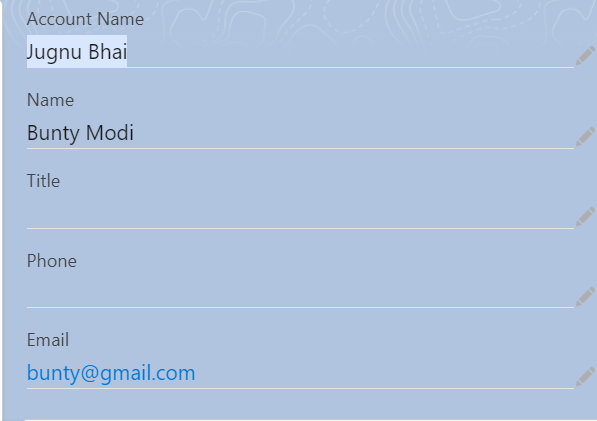
export default class NewContactRecordCompo extends LightningElement {

    @api recordId;

    @api objectApiName;

    fields = ['AccountId', 'Name', 'Title', 'Phone', 'Email'];

}



3] CSS : Cascade Style Sheet

<https://developer.salesforce.com/docs/component-library/documentation/en/lwc/lwc.create_components_css>

<https://www.santanuatonline.com/how-to-use-css-in-lightning-web-component/>

h1 {

    font-size: 50px;

    color: green;

    text-align: center;

    margin-top: 100px;

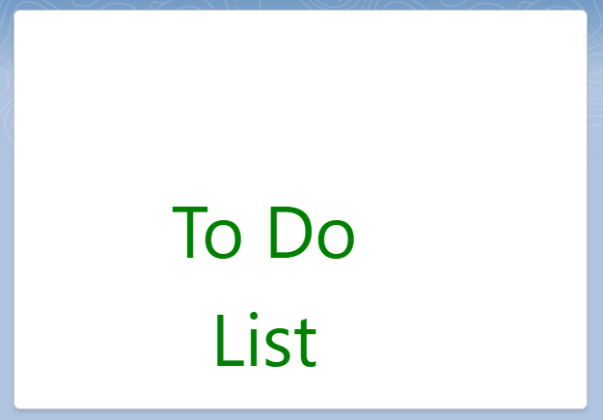
    margin-bottom: 10px;

    margin-right: 150px;

    margin-left: 100px;

}

======================END====================



Assignment:

Pagination in LWC

<https://www.mstsolutions.com/technical/datatable-and-pagination-using-lwc/#:~:text=Paginations%20in%20LWC%3A,page%20numbers%20and%20total%20records>.

<https://salesforce.stackexchange.com/questions/374880/how-to-paginate-a-html-table-in-lwc>

<https://www.biswajeetsamal.com/blog/salesforce-lwc-custom-datatable-pagination/>

<https://www.salesforcetroop.com/client_side_pagination_in_lwc>

EXTRAS:

https://developer.salesforce.com/docs/component-library/bundle/lightning:progressIndicator/example