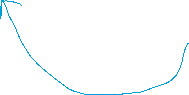
63. Child to Parent Communication - 01 July 2022

1] Child to Parent Communication

2] Pass Data from Child to Parent Component

1] Child to Parent Communication



<c-child-compo **oneventname={X1}**> </c-child-compo>

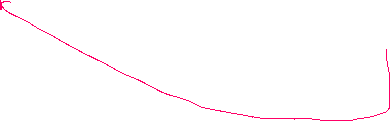


const myEvent = new **CustomEvent**(“eventname”, {

detail : “Data”

});

**this.dispatchEvent(myEvent);**



X1(event){

var childData = event.detail

}



Child JS Method:



const myEvent = new CustomEvent(“eventname”, {



detail : “Data”

});

this.dispatchEvent(myEvent);

Parent JS Method :

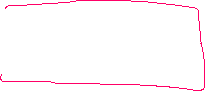
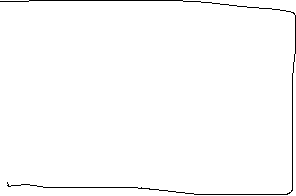
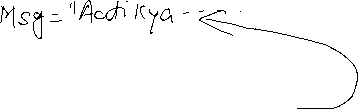


JSMethod(event){

var childData = event.detail

}

User Story 1]



Parent

<template>

    <lightning-card title="Mall Parent Compo">

        Message from Child Compo = {receivedMessage} <br />

        <!--Calling shop Child Compo-->

            <c-shop-child-compo oneventname={receiveMessageHandler}></c-shop-child-compo>

    </lightning-card>

</template>

import { LightningElement } from 'lwc';

export default class MallParentCompo extends LightningElement {

    receivedMessage = 'Waiting....!!!!';

    receiveMessageHandler(event){

        console.log(event.detail);

        this.receivedMessage = event.detail;

    }

}

Child

<template>

    <lightning-card title="Shop Child Compo">

        Child Compo Message = {childMessage} <br/>

        <lightning-button variant="success" icon-name="utility:send" label="Send to Parent" onclick={sendDataHandler}></lightning-button>

    </lightning-card>

</template>

import { LightningElement } from 'lwc';

export default class ShopChildCompo extends LightningElement {

    childMessage = 'Aati Kya Khandala...!!!';

    sendDataHandler(){

        const myEvent = new CustomEvent('eventname' , { detail : this.childMessage});

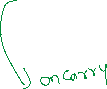
        this.dispatchEvent(myEvent);

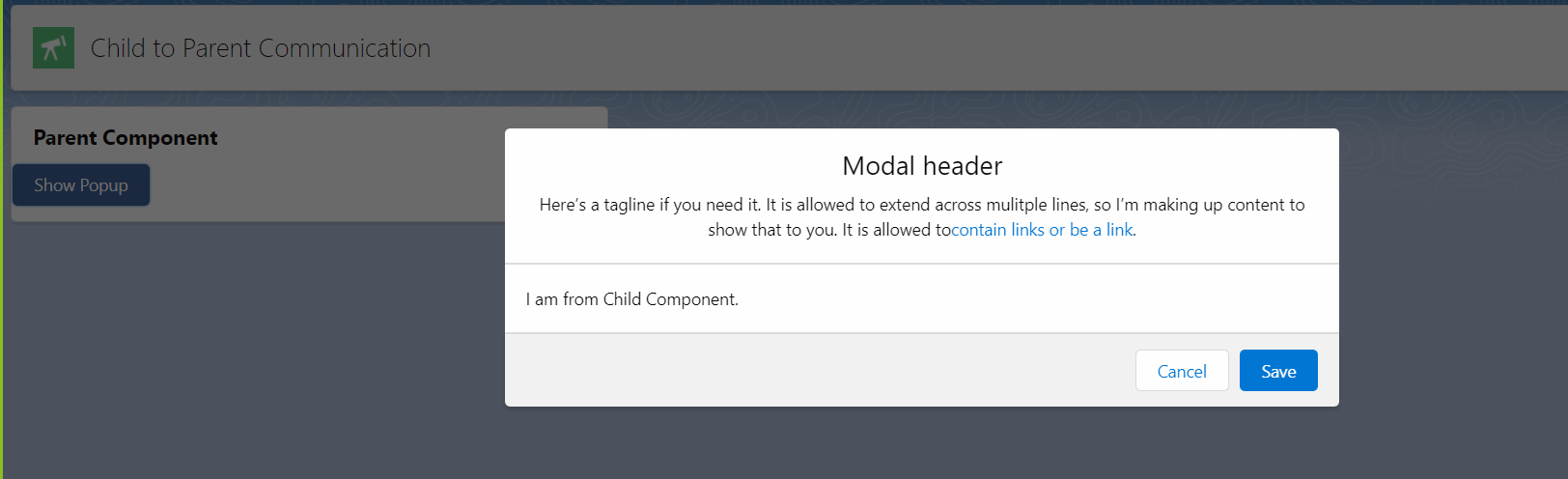
    }

}

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

User Story 2]

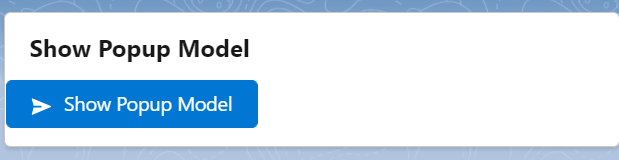


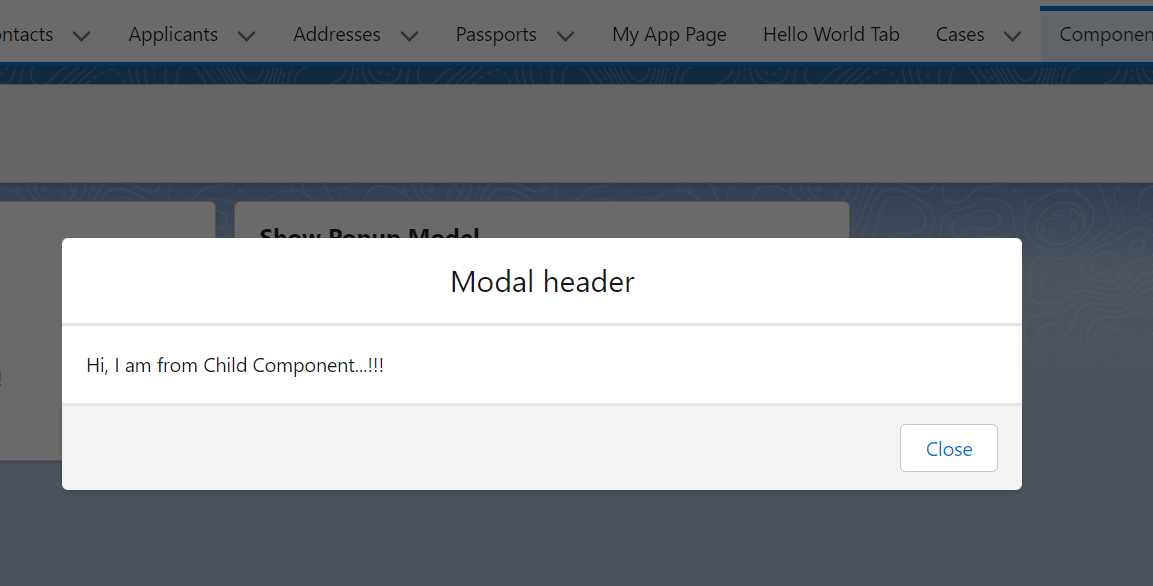


Parent : showPopupModal

Child : popUpMessage

Modal Link : <https://www.lightningdesignsystem.com/components/modals/>





Parent:

<template>

    <lightning-card title="Show Popup Model">

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

        <!--Calling Popup Message-->

            <template if:true={showPopupModalFlag}>

                <c-pop-up-message onclosemodal={closePopupModalHandler}></c-pop-up-message>

            </template>

    </lightning-card>

</template>

import { LightningElement } from 'lwc';

export default class ShowPopupModal extends LightningElement {

    showPopupModalFlag = false;

    showPopupHandler(){

        this.showPopupModalFlag = true;

    }

    closePopupModalHandler(event){

        this.showPopupModalFlag = event.detail; //false

    }

}

Child :

<template>

    <lightning-card title="Popup Modal">

        <section role="dialog" tabindex="-1" aria-modal="true" aria-labelledby="modal-heading-01" class="slds-modal slds-fade-in-open">

            <div class="slds-modal\_\_container">

            <button class="slds-button slds-button\_icon slds-modal\_\_close slds-button\_icon-inverse">

            <svg class="slds-button\_\_icon slds-button\_\_icon\_large" aria-hidden="true">

            <use xlink:href="/assets/icons/utility-sprite/svg/symbols.svg#close"></use>

            </svg>

            <span class="slds-assistive-text">Cancel and close</span>

            </button>

            <div class="slds-modal\_\_header">

            <h1 id="modal-heading-01" class="slds-modal\_\_title slds-hyphenate">Modal header</h1>

            </div>

            <div class="slds-modal\_\_content slds-p-around\_medium" id="modal-content-id-1">

            <p>Hi, I am from Child Component...!!!</p>

            </div>

            <div class="slds-modal\_\_footer">

            <button class="slds-button slds-button\_neutral" aria-label="Cancel and close" onclick={closeButtonHandler}>Close</button>

            </div>

            </div>

            </section>

            <div class="slds-backdrop slds-backdrop\_open" role="presentation"></div>

    </lightning-card>

</template>

import { LightningElement } from 'lwc';

export default class PopUpMessage extends LightningElement {

    closeButtonHandler(){

        const closePoupModalEvent = new CustomEvent('closemodal', {detail : false});

        this.dispatchEvent(closePoupModalEvent);

    }

}

Assignment:

