2

64. Independent Component Communication, Pub-Sub Technique- 04 July 2022

Today we will learn:

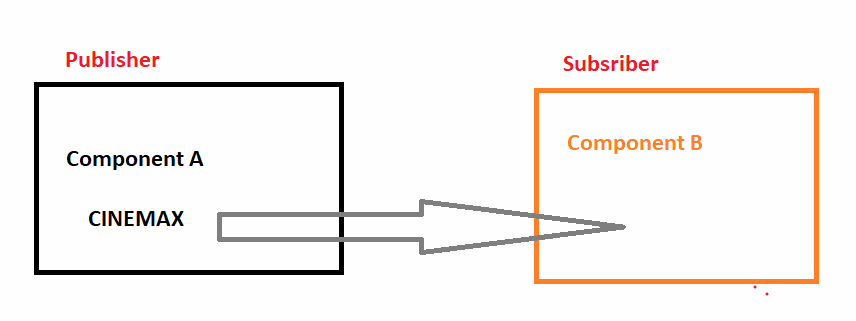
1] Independent Components Communication

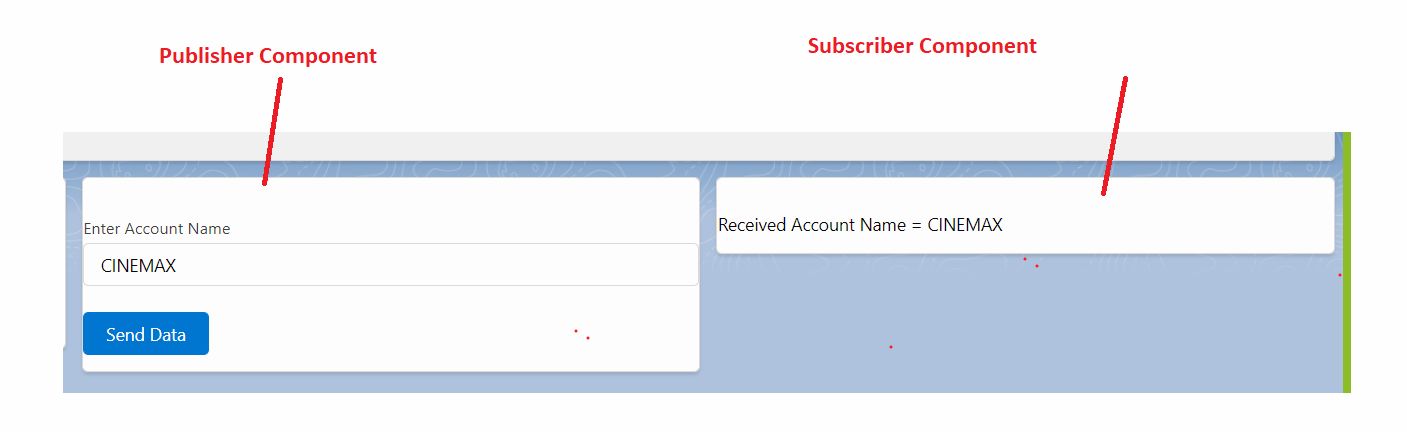
* Pub-Sub Technique
* LMS (Lightning Messaging Service)

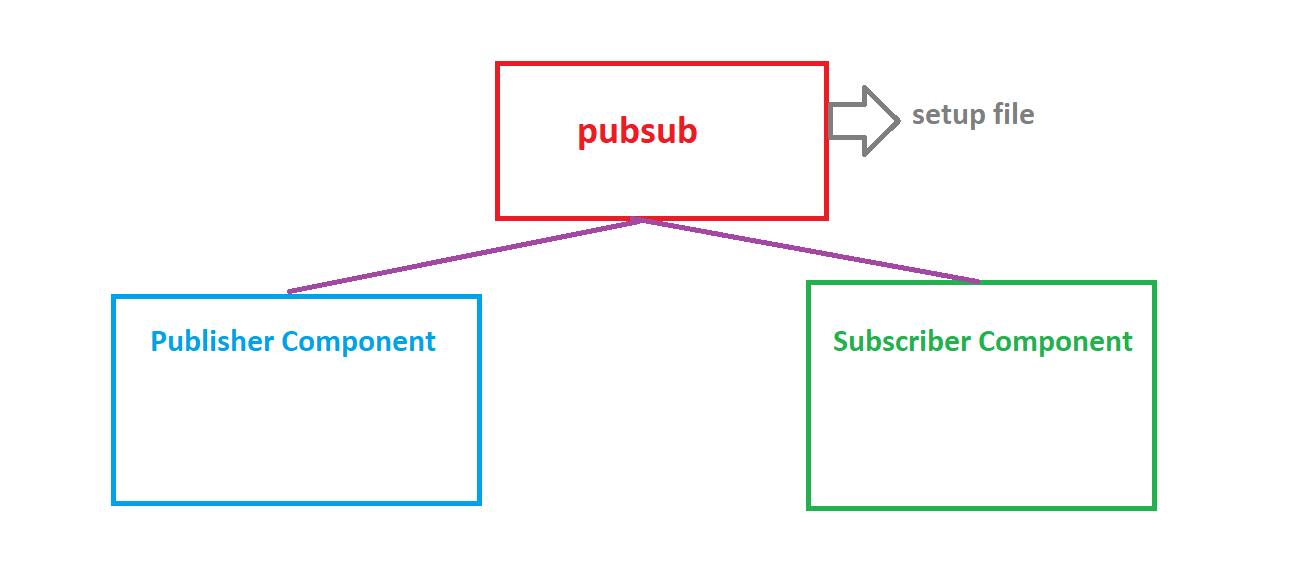
1] Pub-Sub Technique

Pub = Publisher (Sender)

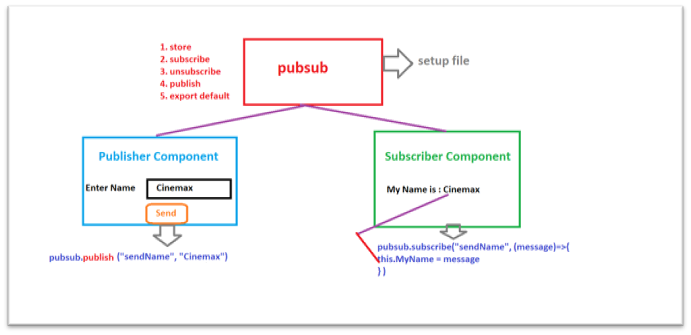
Sub = Subscriber (Receiver)

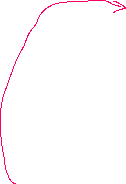
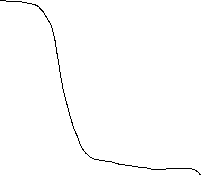
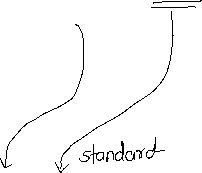












STEPS:

1] Create a pubsub setup file (Lightning Component). And rename “eventName”.

2] import pubsub file in sender and receiver components.

E.g. *import pubsub from ‘c/pubsub’;*

3] Call pubsub publish method from Publisher Component

E.g. *pubsub.publish(‘eventName’ , data);*

4] Call pubsub subscribe method from Subscriber Component on Component Load (connectCallBack)

*connectedCallback(){*

*this.receivedDataHandler()*

*}*

*receivedDataHandler(){*

*pubsub.subscribe("eventName" , (message) => {*

*this.receivedData = message*

*});*

*}*

Pubsub setup file.

const store={};

const subscribe = (eventName,callback) =>{

        if(!store[eventName]){

        store[eventName] = new Set();

        }

        store[eventName].add(callback);

};

const unsubscribe = (eventName,callback) =>{

        if(store[eventName]){

        store[eventName].delete(callback);

        }

};

const publish = (eventName, payLoad) =>{

        if(store[eventName]){

        store[eventName].forEach(callback => {

            try{

            callback(payLoad);

            }

            catch(error){

            console.log(error);

            }

});

}

};

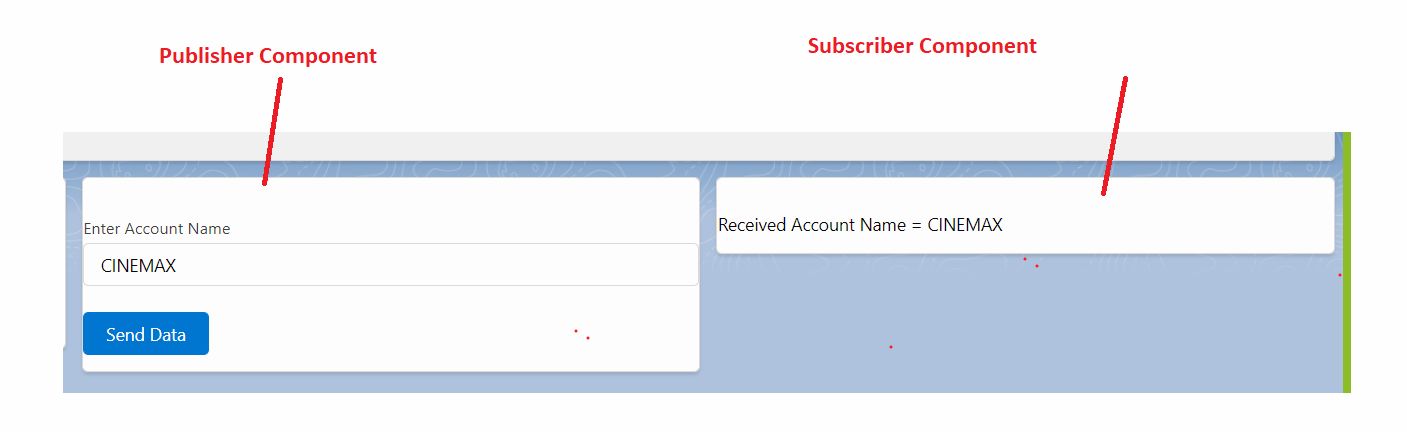
export default{

        subscribe,

        unsubscribe,

        publish

}



Publisher : modelPublisherCompo

Subscriber : carSubscriberCompo

Publisher: import { LightningElement } from 'lwc';

import pubsub from 'c/pubsub';

export default class ModelPublisherCompo extends LightningElement {

    accountObject = {'sObjectType' : 'Account'};

    sendDataHandler(){

        debugger;

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

        pubsub.publish('sendData' , this.accountObject);

    }

}

<template>

    <lightning-card title="Publisher Component">

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

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

    </lightning-card>

</template>

Subscriber:

<template>

    <lightning-card title="Subscriber Component">

        <p>

            Received Data = <lightning-formatted-text value={accObject.Name} ></lightning-formatted-text></p>

    </lightning-card>

</template>

import { LightningElement } from 'lwc';

import pubsub from 'c/pubsub';

export default class CarSubscriberCompo extends LightningElement {

    accObject = {'sObjectType' : 'Account'}

    connectedCallback(){ //It Runs Automatically on Component Load

        console.log('Aati kya khandala..!!!');

        this.receivedDataHandler();

    }

    receivedDataHandler(){

        pubsub.subscribe('sendData' , (message) => {

           this.accObject = message;

        });

    }

}

Pub Sub Model:

const store={};

const subscribe = (sendData,callback) =>{

        if(!store[sendData]){

        store[sendData] = new Set();

        }

        store[sendData].add(callback);

};

const unsubscribe = (sendData,callback) =>{

        if(store[sendData]){

        store[sendData].delete(callback);

        }

};

const publish = (sendData, payLoad) =>{

        if(store[sendData]){

        store[sendData].forEach(callback => {

            try{

            callback(payLoad);

            }

            catch(error){

            console.log(error);

            }

});

}

};

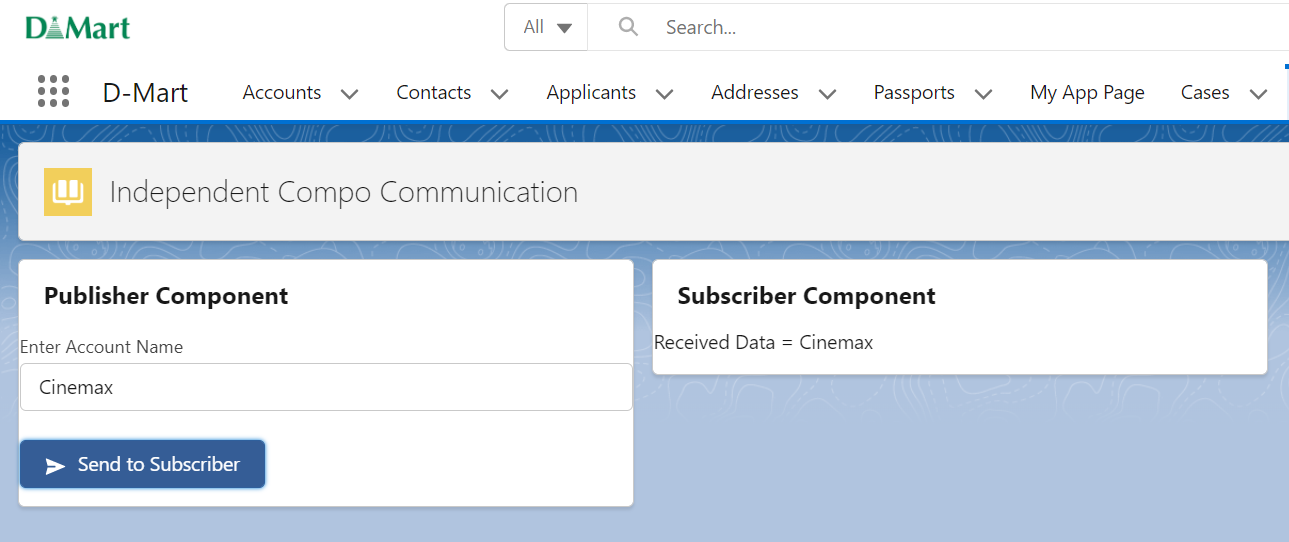
export default{

        subscribe,

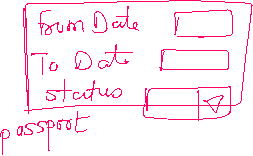
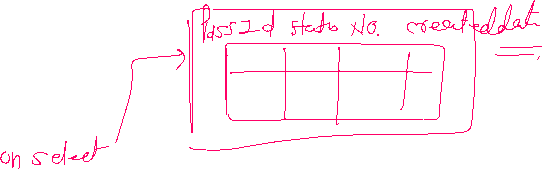
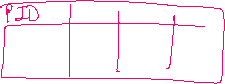
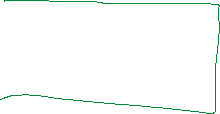
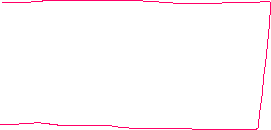
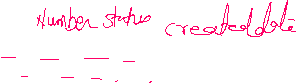
        unsubscribe,

        publish

}



Assignment:



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

IMP 05 July – 07 July Fee Payment

1] LWC 01 OR 02

2] Community: 02

3] Integration: 07 Days

4] Deployment: 02

5] Aura Lightning: 15 Days

6] Test Classes: (Video: 05)

Coding Test 🡺 Project (10 Days) 🡺 Mock Interview 🡺 Final Interview 🡺 Resume Up