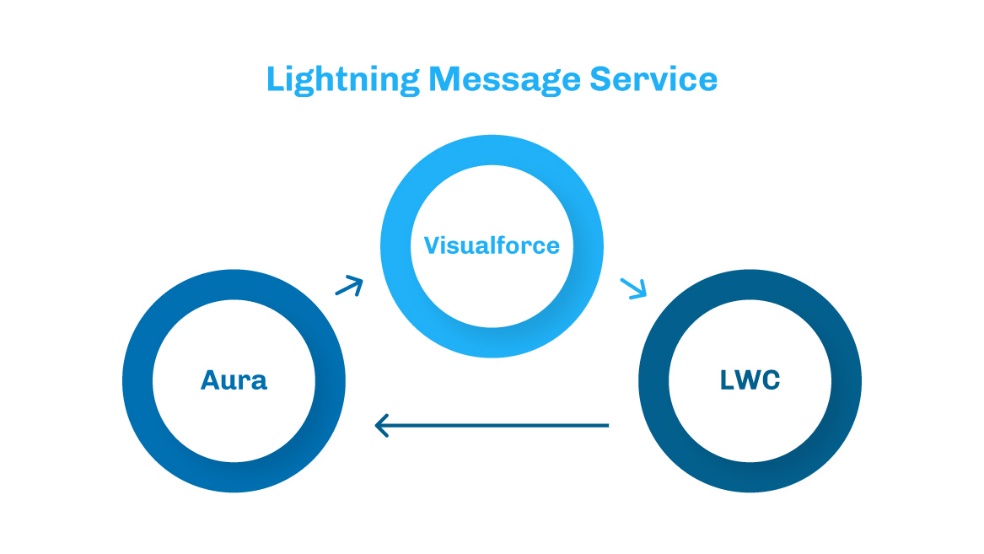
65. Lightning Messaging Service - 05 July 2022

Today we will learn:

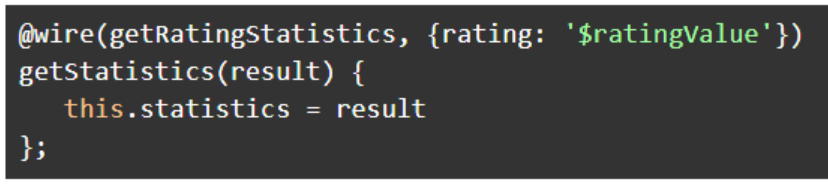
1] Lightning Message Service (LMS)

Lightning Messaging Service (LMS)



@Wire:

Apni Wali : This decorator is used in LMS MessageContext. And It is also used to call Apex Controller method directly without importing it (Apex Controller Method).



OR

Components use @wire in their JavaScript class to read data from one of the wire adapters in the lightning/ui\*Api modules and also to call the apex controller server-side methods using wire services.

LWC

LWC1

LWC2

Parent to Child

Child to Parent

Aura

Aura 1

Aura 2

Parent to Child

Child to Parent

VF

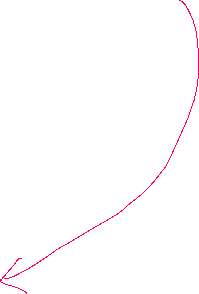
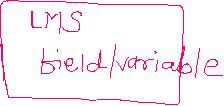
VF1 1

VF 2

Parent to Child

Child to Parent

Lightning Message Service



sendLWC

receiveLWC



Flow of LMS



**messageChannels**



**AccountDataChannel**.**messageChannel-meta.xml**

FieldName = accData



package.xml

Component A

1. Import channel
2. import messageService
3. publish the data

MYCHANNEL

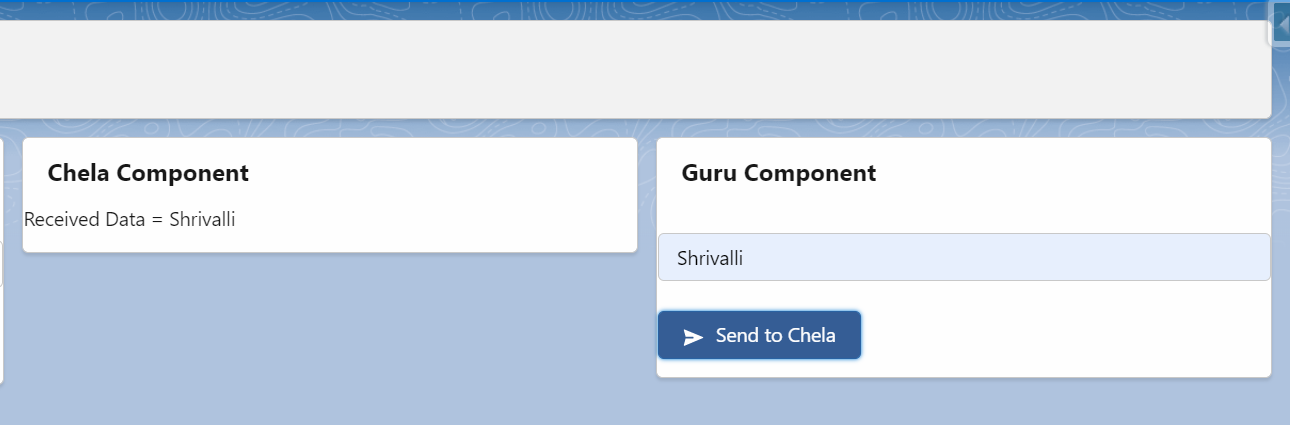
Component B

1. Import channel
2. import messageService
3. subscribe the data

MYCHANNEL

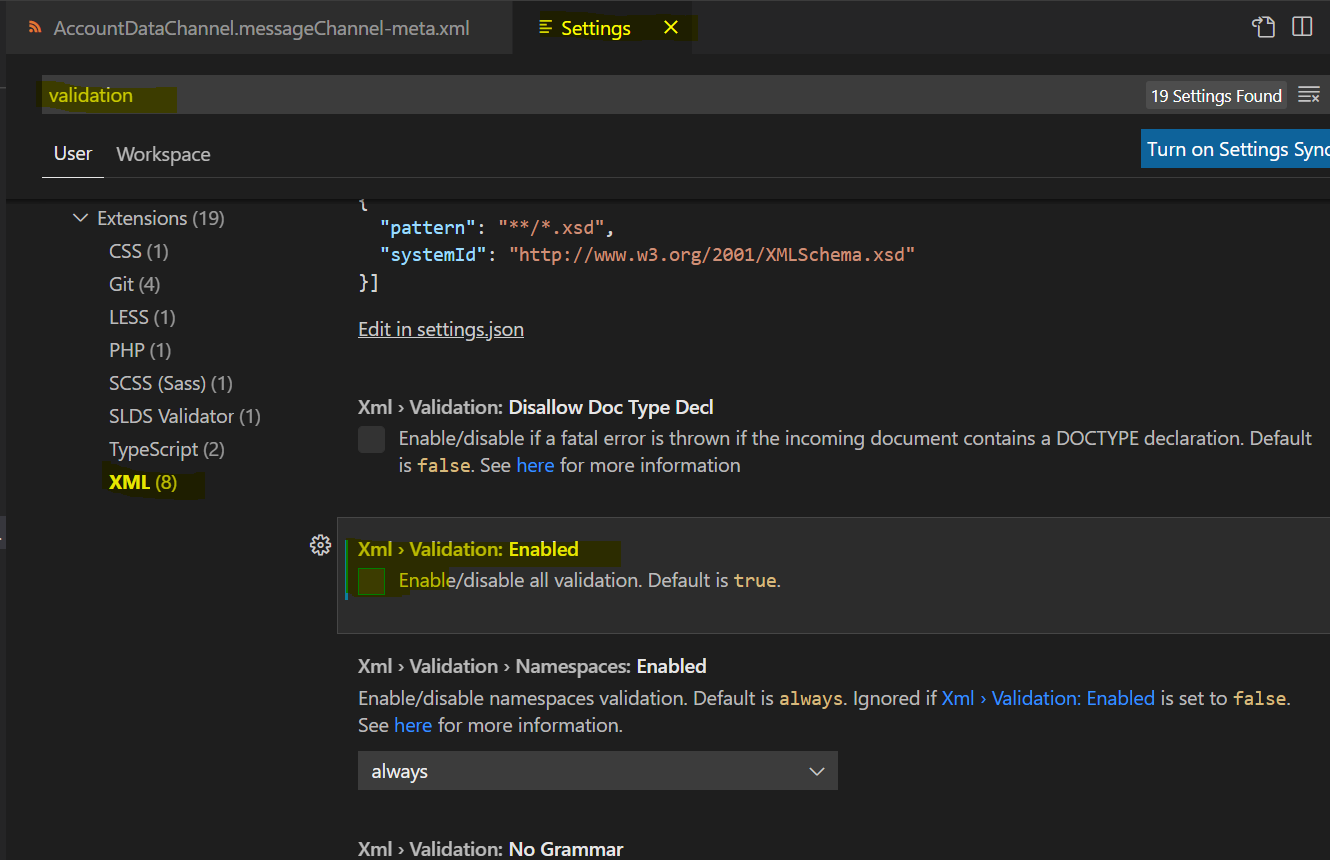


**User Story 1]**



Detailed STEPS

Setting : Open VS Code 🡪 File 🡪 Preferences 🡪 Settings 🡪 Type **Validation**





1. Create a *folder* **messageChannels** under the path force-app\main\default
2. Create a xml file ChannelName.messageChannel-meta.xml inside the above folder.

Example : ChannelName 🡪 **AccountDataChannel**.messageChannel-meta.xml

1. Add following XML definition to the message channel file above.

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

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

    <description>This is a sample Lightning Message Channel.</description>

    <isExposed>true</isExposed>

    <lightningMessageFields>

        <description>This is the data</description>

        <fieldName>accountName</fieldName>

    </lightningMessageFields>

    <masterLabel>SampleMessageChannel</masterLabel>

</LightningMessageChannel>

1. Update the Manifest/package.xml by adding the Lightning Message Channel

<types>

<members>\*</members>

<name>LightningMessageChannel</name>

</types>

1. The API version should be 47 and above.

**7] Sender Component:**

import MYCHANNEL from "@salesforce/messageChannel/AccountDataChannel\_\_c";

import {publish, MessageContext} from "lightning/messageService"

    @wire(MessageContext)

    context

sendDataHandler(){

        const message={

            accountName:{

                value:this.myName

            }

         }

         publish(this.context, MYCHANNEL, message);

    }

**8] Receiver Component:**

import MYCHANNEL from "@salesforce/messageChannel/AccountDataChannel\_\_c";

import {subscribe, MessageContext,APPLICATION\_SCOPE} from "lightning/messageService"

    @wire(MessageContext)

    Context

    connectedCallback(){

        this.receiveDataHandler()

}

receiveDataHandler(){

    subscribe(this.context, MYCHANNEL, (message)=>{this.handleMessage(message)}, {scope : APPLICATION\_SCOPE} )

}

handleMessage(message){

   this.receivedName = message.accountName.value

}

**FULL CODE:**

**LMS:**

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

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

    <description>This is a sample Lightning Message Channel.</description>

    <isExposed>true</isExposed>

    <lightningMessageFields>

        <description>This is the data</description>

        <fieldName>accountName</fieldName>

    </lightningMessageFields>

    <masterLabel>SampleMessageChannel</masterLabel>

</LightningMessageChannel>

**GURU Compo:**

<template>

    <lightning-card title="Guru 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 Chela" onclick={sendDataHandler}></lightning-button>

    </lightning-card>

</template>

import { LightningElement,wire } from 'lwc';

import MYCHANNEL from "@salesforce/messageChannel/AccountDataChannel\_\_c";

import {publish, MessageContext} from "lightning/messageService"

export default class GuruCompo extends LightningElement {

    @wire (MessageContext)

    context

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

    sendDataHandler(){

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

        const message={

            accountName:{

                value:this.accountObject.Name

            }

         }

         publish(this.context, MYCHANNEL, message);

      }

}

**Chela Compo:**

<template>

    <lightning-card title="Chela Component">

        Received Data = <lightning-formatted-text value={accName} ></lightning-formatted-text></p>

    </lightning-card>

</template>

import { LightningElement, wire } from 'lwc';

import MYCHANNEL from "@salesforce/messageChannel/AccountDataChannel\_\_c";

import {subscribe, MessageContext,APPLICATION\_SCOPE} from "lightning/messageService"

export default class ChelaCompo extends LightningElement {

    @wire(MessageContext)

    context

    accName='Wating...';

    connectedCallback(){

        this.receiveDataHandler();

    }

receiveDataHandler(){

    subscribe(this.context, MYCHANNEL, (message)=>{this.handleMessage(message)}, {scope : APPLICATION\_SCOPE} )

}

handleMessage(message){

    this.accName = message.accountName.value

   console.log('Received Data ' + this.accName);

}

}

**=======================================================================================**

**OBJECT WALA:**

**LMS:**

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

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

    <description>This is a sample Lightning Message Channel.</description>

    <isExposed>true</isExposed>

    <lightningMessageFields>

        <description>This is the data</description>

        <fieldName>objAccount</fieldName>

    </lightningMessageFields>

    <masterLabel>SampleMessageChannel</masterLabel>

</LightningMessageChannel>

<template>

    <lightning-card title="Guru 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 Chela" onclick={sendDataHandler}></lightning-button>

    </lightning-card>

</template>

import { LightningElement,wire } from 'lwc';

import MYCHANNEL from "@salesforce/messageChannel/AccountDataChannel\_\_c";

import {publish, MessageContext} from "lightning/messageService"

export default class GuruCompo extends LightningElement {

    @wire (MessageContext)

    context

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

    sendDataHandler(){

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

        const message={

            objAccount:{

                value:this.accountObject

            }

         }

         publish(this.context, MYCHANNEL, message);

      }

}

**Chela:**

<template>

    <lightning-card title="Chela Component">

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

    </lightning-card>

</template>

import { LightningElement, wire } from 'lwc';

import MYCHANNEL from "@salesforce/messageChannel/AccountDataChannel\_\_c";

import {subscribe, MessageContext,APPLICATION\_SCOPE} from "lightning/messageService"

export default class ChelaCompo extends LightningElement {

    @wire(MessageContext)

    context

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

    connectedCallback(){

        this.receiveDataHandler();

    }

receiveDataHandler(){

    subscribe(this.context, MYCHANNEL, (message)=>{this.handleMessage(message)}, {scope : APPLICATION\_SCOPE} )

}

handleMessage(message){

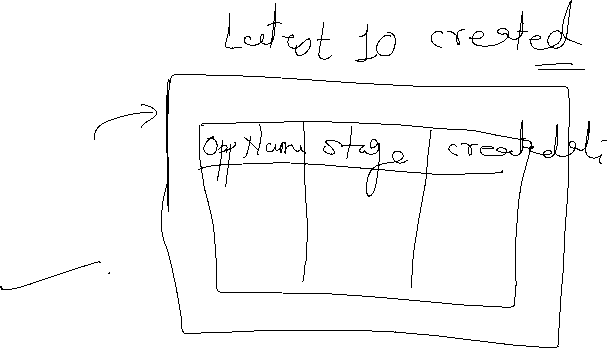
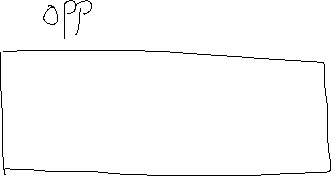
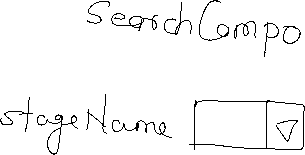
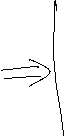
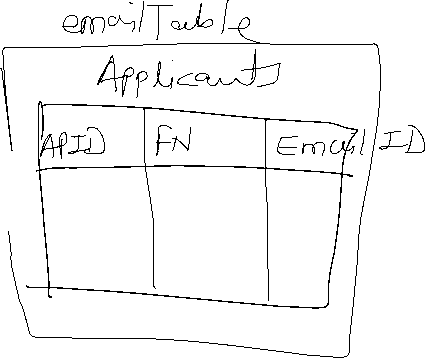
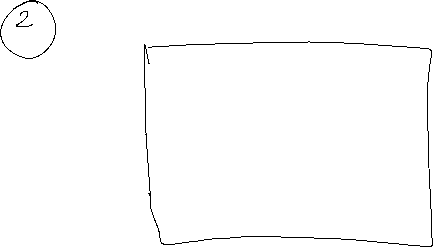
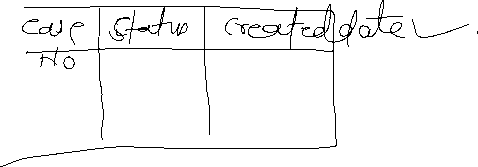
   this.accObject = message.objAccount.value

   console.log('Received Data ' + message.accountName.value);

}

}

**Assignment:**



**Google:**

**What the Scratch ORG and How to use it in VS Code Studio?**