

HOW TO ACCESS APEX CLASS USING LWC-

Lightning web components can import methods from Apex classes. The imported methods are functions that the component can call either via @wire or imperatively.

Call apex method Using Wire services.

Call the apex method imperatively.

Using Wire method:

To call the apex method in the lightning web component, First, we have to create the apex class and add the @AuraEnabled method at the first line, i.e., before starting the method. To call it from Wire Service, the method should be cacheable. Hence, add cacheable=true in @AuraEnabled. And in the LWC js controller, we need to write the import method using the @salesforce/apex/className.MethodName; at the start of the js controller and in the lightning, we need to write the @wire to get the records we provide in the apex class.

step1: we have to create the apex class

step2 : add the @AuraEnabled method at the first line, i.e., before starting the method.

step3: To call it from Wire Service, the method should be cacheable. Hence, add cacheable=true in @AuraEnabled.

step4: And in the LWC js controller, we need to write the import method using the @salesforce/apex/className.MethodName.

step5: at the start of the js controller and in the lightning, we need to write the @wire to get the records we provide in the apex class.

Example: Get all Contact records using wire method -

Apex Class: ContactController.cls

```
public with sharing class ContactController {

    @AuraEnabled(cacheable=true)
    public static List<Contact> getContacts(){
        List<Contact> cons = [select id,name,phone,department,email,AccountId,birthdate from
contact];
        return cons;
    }
    @AuraEnabled(cacheable=true)
    public static List<Contact> getContactDetails(string coid){
        return [select id,name,phone,department,email,AccountId,birthdate from contact where
id=:coid];
    }
}
```

Lightning Web Component: contactWireComponent - component

i)contactWireComponent.html

```
<template>
    <lightning-card title="Calling Apex using Wire Method">
        <div class="slds-box" style="background-color:rgb(0, 255, 221) ;">
            <template if:true={cat.data}>
                <div class="slds-grid slds-gutters slds-wrap">
                    <template for:each={cat.data} for:item="co">
                        <div key={co.Id} class="slds-col slds-size_1-of-3
slds-p-around_small">
                            <lightning-card title={co.Name}>
                                <p>Related Account : {co.AccountId}</p>
                                <p>  Phone      : {co.Phone}</p>
                                <p>Contact Email   : {co.Email}</p>
                                <p>Department   : {co.Department}</p>
                                <p>BirthDate    : {co.BirthDate}</p>
                            </lightning-card>
                        </div>
                    </template>
                </div>
            </template>
        </div>
    </template>
```

```
        <template if:false={cat.data}>
            <h1>error</h1>
        </template>
    </div>
</lightning-card>
</template>
```

ii)contactWireComponent.js

```
import { LightningElement , wire} from 'lwc';
import conList from '@salesforce/apex/ContactController.getContacts';
export default class ContactWireComponent extends LightningElement {
    @wire(conList) cat;
}
```

iii)contactWireComponent.meta-xml

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>55.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target> lightning__RecordPage </target>
        <target> lightning__HomePage </target>
        <target> lightning__AppPage </target>
        <target> lightning__Tab </target>
    </targets>
</LightningComponentBundle>
```

OUTPUT:

Calling Apex using Wire Method

Madhu11 Related Account : Phone : Contact Email : Department : Mech BirthDate :	Tarun33 Related Account : Phone : Contact Email : Department : BirthDate :	Tarun Kumar Related Account : Phone : Contact Email : Department : BirthDate :
Hardhik Related Account : 0015g00000ffenZAAQ Phone : Contact Email : Department : BirthDate :	Renuka Related Account : Phone : 91080055618 Contact Email : cse@gmail.com Department : CSE BirthDate :	Robin Related Account : Phone : 91080055618 Contact Email : cse@gmail.com Department : cse BirthDate :
Dhoni Related Account : Phone : Contact Email : Department : BirthDate :	Kumar Madhu Related Account : 0015g00000fflFPAAU Phone : Contact Email : Department : BirthDate :	Dhoni Related Account : Phone : Contact Email : cse@gmail.com Department : CsE BirthDate :
Bravo	madhu kumar	Mr.3

Using Imperatively method:

Calling the apex method in the lightning web component without using the wire method is more accessible than the wire method. Making the AuraEnabled method cacheable is not mandatory to call it imperatively. While using the wire method, we need to add the data, but there is no need for this in the imperatively method. It is the easiest way to call the apex method in the LWC.

When we have combo box -> onchange event -> =
Text box -> button -> onclick -> =

Text box -> onchange event => like operator

Apex Class : OpportunityController.cls

```
public with sharing class OpportunityController {
```

```

@AuraEnabled(cacheable=true)
public static List<Opportunity> getAllOpp(string accName){
    List<Opportunity> op;
    if(string.isNotBlank(accName)){
        string str='%'+accName+'%';
        op = [select
id,name,amount,StageName,closedate,LeadSource,Account.Name,description,isClosed,type,Ac
countId from Opportunity where Account.Name like : str ];

    }else{
        op = [select
id,name,amount,StageName,closedate,LeadSource,Account.Name,description,isClosed,type,Ac
countId from Opportunity ];

    }
    return op;
}

// the method needs to call imperatively so it doesnt require cacheable =true
@AuraEnabled
public static string closeOpp(String opportunityId){
    string msg;

    try {
        Opportunity op = new Opportunity(Id=opportunityId);
        op.StageName='Closed Won';
        update op;
        msg = 'Record updated successfully..!!!';
    } catch (Exception e) {
        msg=e.getMessage();
    }
    return msg;
}
}

```

Lightning Component -

a) accountRelatedOpportunityComponent.html

```

<template>
    <lightning-card title="Calling APEX class using Imperative Method">
        <lightning-input onchange={accountHandler} label="Account Name" placeholder="Seach
Account"> </lightning-input>
        <div class="slds-box">

```

```

        <template if:true={opportunities.data}>
            <div class="slds-grid slds-gutters slds-wrap slds-m-around_small"
style="background-color:rgb(4, 133, 0);">
                <template for:each={opportunities.data} for:item="oppoObj">
                    <div key={oppoObj.Id} class="slds-box slds-col slds-size-1-of-2
slds-p-around_small slds-p-around_large">
                        <lightning-card title={oppoObj.Id}>
                            <div slot="actions">
                                <template if:true={oppoObj.IsClosed}>
                                    <span style="color: blue;">
                                        Opportunity is Closed
                                    </span>
                                </template>
                                <template if:false={oppoObj.IsClosed}>
                                    <lightning-button label="Close Opportunity"
variant="success" value={oppoObj.Id} onclick={opportunityHandler}></lightning-button>
                                </template>
                            </div>
                        </lightning-card>

                        <p>Account Id : {oppoObj.AccountId}</p>

                        <p>Opportunity status : {oppoObj.IsClosed}</p>
                        <p>Opportunity Name : {oppoObj.Name}</p>
                        <p>Description : {oppoObj.Description}</p>
                        <p>Lead Source : {oppoObj.LeadSource}</p>
                        <p>Stage Name : {oppoObj.StageName}</p>
                        <p>Amount : {oppoObj.Amount}</p>
                        <p>Close Date : {oppoObj.CloseDate}</p>
                        <p>Type : {oppoObj.Type}</p>

                    </div>
                </template>
            </div>

        </template>
        <template if:false={opportunities.data}>
            <h1>Error</h1>
        </template>
    </div>
</lightning-card>

```

</template>

b) accountRelatedOpportunityComponent.js

```
import { LightningElement, wire } from 'lwc';
import getOppo from '@salesforce/apex/OpportunityController.getAllOpp';
import oppoClosed from '@salesforce/apex/OpportunityController.closeOpp';
import {refreshApex} from '@salesforce/apex';

export default class AccountRelatedOpportunityComponent extends LightningElement {
    oppoldj = "";
    acName = "";
    msg = "";

    @wire (getOppo,{accName : '$acName'}) opportunities ;

    accountHandler(event){
        this.acName = event.target.value;
    }

    opportunityHandler(event){
        this.oppoldj = event.target.value;
        oppoClosed({opportunityId:this.oppoldj})
        .then(result =>{
            this.msg = result;
            alert("message: "+this.msg);
            refreshApex(this.opportunities);
        })
        .catch(error=>{
            this.msg=error.getMessage;
        })
    }
}
```

c) accountRelatedOpportunityComponent.meta.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>55.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target> lightning__RecordPage </target>
        <target> lightning__HomePage </target>
        <target> lightning__AppPage </target>
    </targets>
</LightningComponentBundle>
```

```
<target> lightning__Tab </target>
</targets>
</LightningComponentBundle>
```

OUTPUT:

Calling APEX class using Imperative Method		
Account Name		
Search Account		
<div><div>0065g00000K8VNCaAN</div><div>Opportunity is Closed</div><div>Account Id : Opportunity status : true Opportunity Name : Madhu 23 Description : Lead Source : Stage Name : Closed Won Amount : Close Date : 2022-03-02 Type :</div></div>	<div><div>0065g00000ASHZmAAP</div><div>Opportunity is Closed</div><div>Account Id : 0015g00000HwEigAaF Opportunity status : true Opportunity Name : Edge Emergency Generator Description : Lead Source : Word of mouth Stage Name : Closed Won Amount : 75000 Close Date : 2021-11-19 Type : New Customer</div></div>	<div><div>0065g00000ASHZtAAP</div><div>Opportunity is Closed</div><div>Account Id : 0015g00000HwEigAaF Opportunity status : true Opportunity Name : Edge Installation Description : Lead Source : Word of mouth Stage Name : Closed Won Amount : 50000 Close Date : 2021-11-19 Type : Existing Customer - Upgrade</div></div>
<div><div>0065g00000ASHZuAAP</div><div>Opportunity is Closed</div><div>Account Id : 0015g00000HwEigAaF Opportunity status : true Opportunity Name : Edge SLA Description : Lead Source : Word of mouth Stage Name : Closed Won Amount : 60000 Close Date : 2021-11-19 Type : Existing Customer - Upgrade</div></div>	<div><div>0065g00000ASHa8AAH</div><div>Opportunity is Closed</div><div>Account Id : 0015g00000HwEigAaF Opportunity status : true Opportunity Name : Edge Emergency Generator Description : Lead Source : Stage Name : Closed Lost Amount : 35000 Close Date : 2021-11-19 Type : Existing Customer - Replacement</div></div>	<div><div>0065g00000ASHa1AAH</div><div>Opportunity is Closed</div><div>Account Id : 0015g00000HwEigAaF Opportunity status : true Opportunity Name : Burlington Textiles Weaving Plant Generator Description : Lead Source : Web Stage Name : Closed Won Amount : 235000 Close Date : 2021-11-19 Type : New Customer</div></div>