66. Wire Decorator, get Current Record Id, get Current Fields, Call Apex Method - 06 July 2022

Today we will learn:

1] Get current Record Id

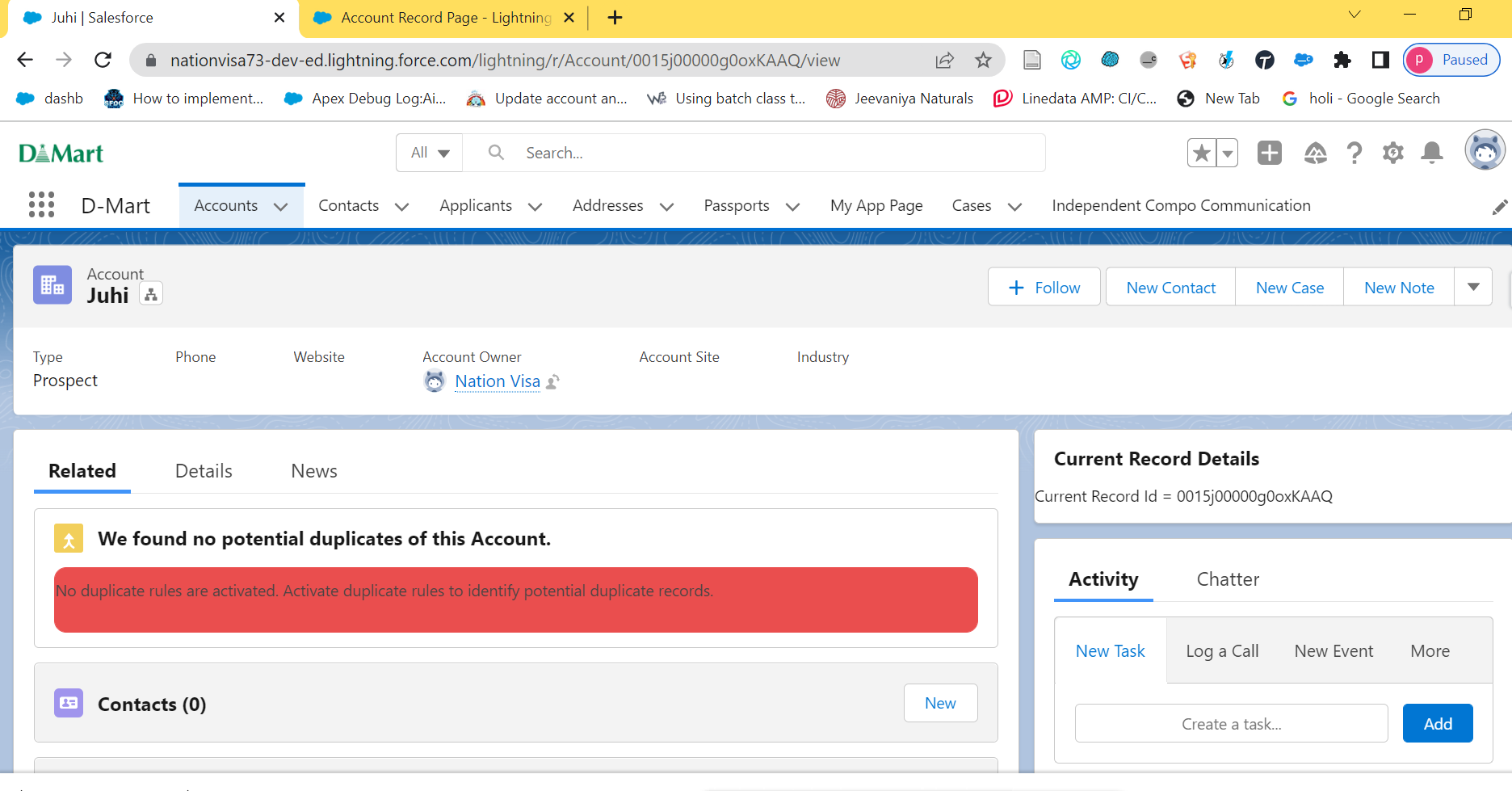
2] Get Fields of Current Record

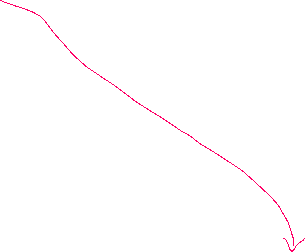
3] Call Apex Method

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

1] Get current Record Id

@api recordId;





<template>

    <lightning-card title="Current Record Details">

        Current Record Id = {recordId}

    </lightning-card>

</template>

import { LightningElement, api } from 'lwc';

export default class CurrentRecordIdCompo extends LightningElement {

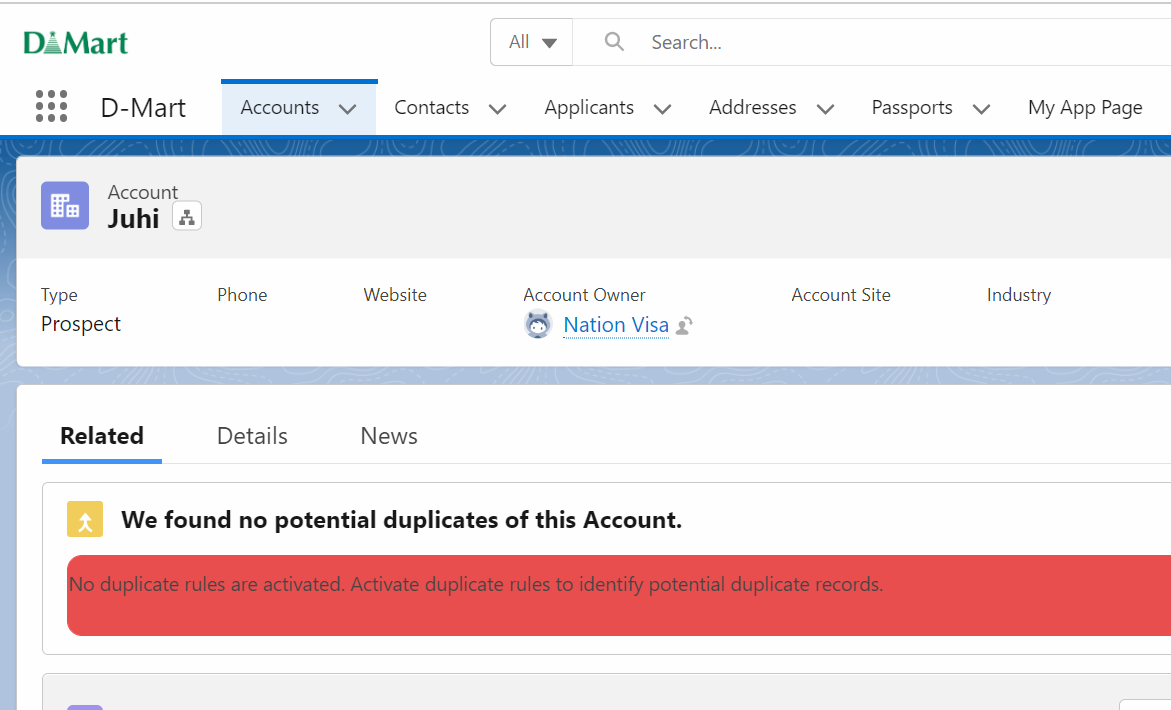
    @api recordId;

}

Assignment:

Create LWC component and Display it on Account Details Page Tab “Account Contacts Table” and show all the contacts of current Account.







2] Get Fields of Current Record

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

import { getRecord, getFieldValue } from 'lightning/uiRecordApi';

import CREATED\_FIELD from '@salesforce/schema/Account.CreatedDate';

import SLA\_FIELD from '@salesforce/schema/Account.SLA\_\_c';

const fields = [REVENUE\_FIELD, CREATED\_FIELD, EXP\_FIELD,SLA\_FIELD];

export default class WireTest extends LightningElement {

@api recordId;

@wire(getRecord, { recordId: '$recordId', fields })

account;

get created() {

return getFieldValue(this.account.data, CREATED\_FIELD);

}

get sla(){

return getFieldValue(this.account.data,SLA\_FIELD);

}

}

User Story 1] Get Account Type and SLA of Current Record.

<template>

    <lightning-card title="Current Record Details">

        <p><lightning-formatted-text value={type} ></lightning-formatted-text></p>

        <p><lightning-formatted-text value={sla} ></lightning-formatted-text></p>

    </lightning-card>

</template>

import { LightningElement, api, wire } from 'lwc';

import { getRecord, getFieldValue } from 'lightning/uiRecordApi';

import TYPE\_FIELD from '@salesforce/schema/Account.Type';

import SLA\_FIELD from '@salesforce/schema/Account.SLA\_\_c';

const fields = [TYPE\_FIELD, SLA\_FIELD];

export default class CurrentRecordIdCompo extends LightningElement {

    @api recordId;

    @wire(getRecord, { recordId: '$recordId', fields })

      account;

      get type() {

          return getFieldValue(this.account.data, TYPE\_FIELD);

      }

      get sla(){

          return getFieldValue(this.account.data,SLA\_FIELD);

      }

}

User Story 2] Print “In Progress” if Current Applicant Gender is “Male”. “Closed” for Gender Otherwise.

<template>

    <lightning-card title="Applicant Field">

        <p><lightning-formatted-text value={gender} ></lightning-formatted-text></p>

        <template if:true={genderFlag}>

            In Progress

        </template>

        <template if:false={genderFlag}>

            Closed

        </template>

    </lightning-card>

</template>

import { getRecord, getFieldValue } from 'lightning/uiRecordApi';

import { LightningElement, api, wire } from 'lwc';

import GENDER\_FIELD from '@salesforce/schema/Applicant\_\_c.Gender\_\_c';

const fields = [GENDER\_FIELD];

export default class ApplicantFieldCompo extends LightningElement {

 @api recordId;

 genderFlag = false;

 @wire(getRecord, { recordId: '$recordId', fields })

 applicant;

 get gender() {

    if(getFieldValue(this.applicant.data, GENDER\_FIELD) == 'Male'){

            this.genderFlag = true;

    }

    else

    {

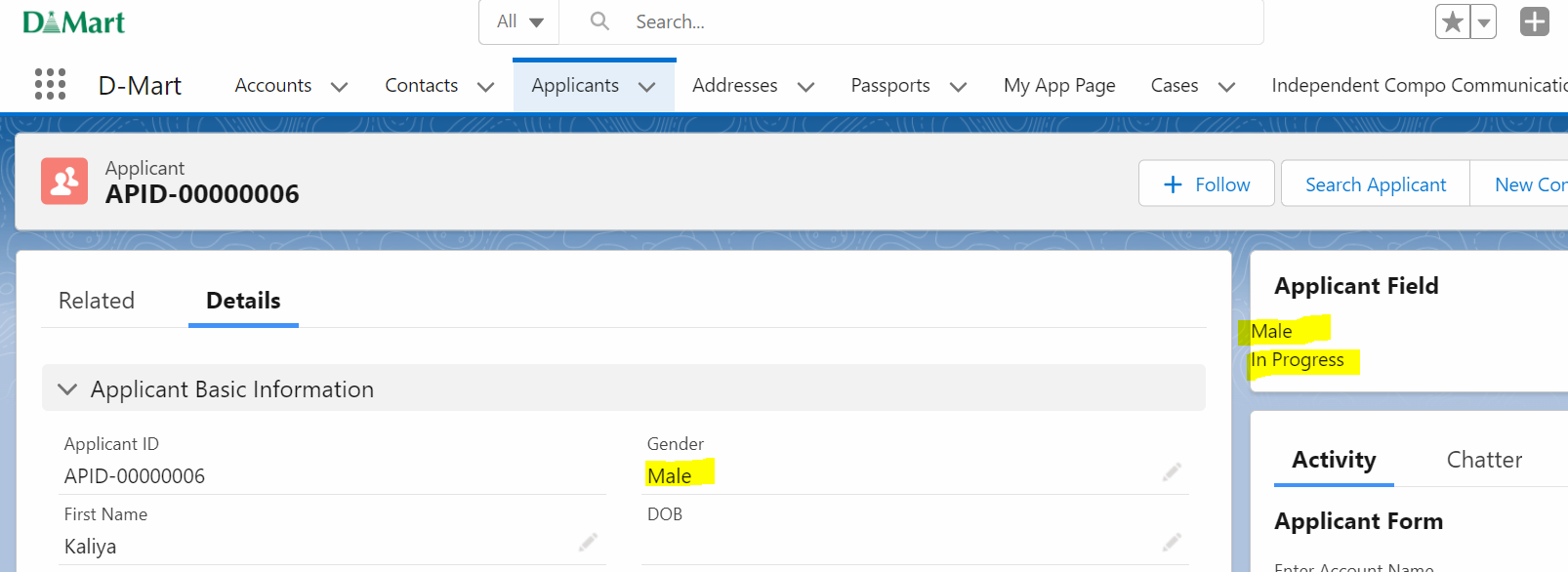
        this.genderFlag = false;

    }

     return getFieldValue(this.applicant.data, GENDER\_FIELD);

 }

}



User Story 3] Print “In Progress” if Current Account Type is “Prospect”. “Closed” for Type Otherwise.

3] Call Apex Method

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

import *apexMethodName* from '@salesforce/apex/*Namespace*.*Classname*.*apexMethodReference*';

@wire(*apexMethodName*, *{ apexMethodParams }*)

*propertyOrFunction*;

User Story 1] Fetch 10 Applicant Records

<template>

    <lightning-card title="ApexWireMethodToProperty" icon-name="standard:account">

        <div class="slds-m-around\_medium">

            <template if:true={applicantList.data}>

                <template for:each={applicantList.data} for:item="app">

                    <p key={app.Id}>{app.First\_Name\_\_c}</p>

                </template>

            </template>

        </div>

    </lightning-card>

</template>

import { LightningElement, api, wire } from 'lwc';

import getAllApplicants from '@salesforce/apex/ApplicantProvider.getAllApplicants';

export default class ApplicantFieldCompo extends LightningElement {

    @wire(getAllApplicants) applicantList;

}

 @AuraEnabled(cacheable=true)

   public static List<Applicant\_\_c> getAllApplicants(){

    try {

        return [select Id, First\_Name\_\_c from Applicant\_\_c LIMIT 10];

    } catch (Exception e) {

        throw new AuraHandledException(e.getMessage());

    }

   }

User Story 2] Fetch Applicant List Having provided Gender.

<template>

    <lightning-card title="ApexWireMethodToProperty" icon-name="standard:account">

        <div class="slds-m-around\_medium">

            <template if:true={applicantList.data}>

                <template for:each={applicantList.data} for:item="app">

                    <p key={app.Id}>{app.First\_Name\_\_c}</p>

                    <p key={app.Id}>{app.Police\_Verification\_\_c}</p> &nbsp;&nbsp;

                </template>

            </template>

        </div>

    </lightning-card>

</template>

import { LightningElement, api, wire } from 'lwc';

import getAllApplicants from '@salesforce/apex/ApplicantProvider.getAllApplicants';

export default class ApplicantFieldCompo extends LightningElement {

gender = 'Female';

policeVer = true;

@wire(getAllApplicants,{gender : '$gender' , police : '$policeVer'}) applicantList;

}

@AuraEnabled(cacheable=true)

   public static List<Applicant\_\_c> getAllApplicants(String gender, Boolean police){

    try {

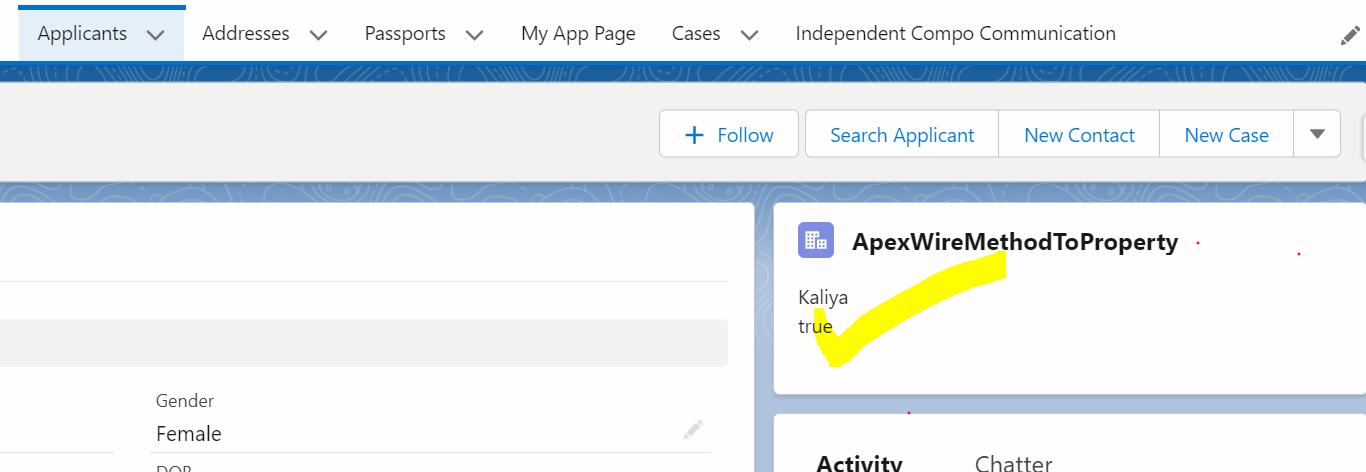
        System.debug('Gender = '+gender);

        return [select Id, First\_Name\_\_c, Gender\_\_c,Police\_Verification\_\_c from Applicant\_\_c where Gender\_\_c=:gender and Police\_Verification\_\_c= :police LIMIT 10];

    } catch (Exception e) {

        throw new AuraHandledException(e.getMessage());

    }



User Story 3] Fetch Account Record of ‘Juhi’.

User Story 4] Fetch Account Records such that Type = ‘Hot’ and Rating =’Cold’.