

Created a LWC named “shipmentTracking” and an Apex class named “ShipmentTrackingController”.

shipmentTracking.html

In this HTML file, we are showing the status of shipment by using a JS variable “trackingStatus”.

```
<template>

  <div class="slds-p-around_medium" style="background: white;">
    <div class="slds-text-heading_medium slds-text-align_center">
      Shipment Tracking
    </div>
    <div class="slds-m-top_medium">
      <b>Tracking Status:</b> {trackingStatus}
    </div>
  </div>

</template>
```

shipmentTracking.js

Using this JS, we are trying to fetch the status from Apex class method “fetchShipmentStatus”.

recordId: It will store the record id in which record page the LWC is placed.

objectApiName: It will store the object name in which record page the LWC is placed.

trackingFieldApi: It will store the value which we will provide when we place the LWC in record detail page.

trackingStatus: It is used to store the response which we get from fetchShipmentStatus method.

In connectCallback method we are calling the Apex method fetchShipmentStatus with the parameters as object name, field API name, record id. And the response is stored in trackingStatus variable which is showing in UI.

```
import { LightningElement, api } from 'lwc';
import fetchShipmentStatus from '@salesforce/apex/ShipmentTrackingController.fetchShipmentStatus';

export default class ShipmentTracking extends LightningElement {

  @api recordId;
  @api objectApiName;
  @api trackingFieldApi;

  trackingStatus;
```

```

connectedCallback() {

    fetchShipmentStatus({ objectName: this.objectApiName, fieldApi: this.trackingFieldApi, recordId: this.recordId })
        .then(result => {
            this.trackingStatus = result;
        })
        .catch(error => {
            console.log('Error in fetchShipmentStatus: ' + error);
        })
    }
}

```

shipmentTracking.js-meta.xml

lightning_RecordPage: To display our LWC in record detail page.

trackingFieldApi: To store the field name given by user when the LWC is placed in record detail page.

```

<?xml version="1.0"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>57.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__RecordPage</target>
    </targets>
    <targetConfigs>
        <targetConfig targets="lightning__RecordPage">
            <property name="trackingFieldApi" type="String" label="Provide tracking field API name"/>
        </targetConfig>
    </targetConfigs>
</LightningComponentBundle>

```

Apex class:

ShipmentTrackingController.apxc

fetchShipmentStatus(objectName, fieldApi, recordId): In this method we are fetching the record to get the tracking number. We are using querying the records and storing in recordsList. Next we are using HTTP callout to fetch the tracking status from the end-point. Finally we are returning the body which we get in HttpResponse.

```

public class ShipmentTrackingController {

```

```

    @AuraEnabled
    public static String fetchShipmentStatus(String objectName, String fieldApi, String recordId) {

        String query = 'SELECT ' + fieldApi + ' FROM ' + objectName + ' WHERE Id = :recordId';
        List<sObject> recordsList = Database.query(query);

        String endPointUrl = 'https://merzcommunities--mel.sandbox.my.salesforce-sites.com/services/apexrest/mockShipmentStatus';
        if(recordsList[0].get(fieldApi) != null && recordsList[0].get(fieldApi) != '') {
            endPointUrl += '?trackingNumber=' + recordsList[0].get(fieldApi);
        }

        HttpRequest req = new HttpRequest();
        req.setEndpoint(endPointUrl);
        req.setMethod('GET');

        HttpResponse res = new HttpResponse();
        res = new Http().send(req);

        return res.getBody();
    }
}

```

Test Classes:

1. ShipmentTrackingControllerMock

```

@isTest
public class ShipmentTrackingControllerMock implements HttpCalloutMock {

    public HttpResponse respond(HttpRequest req) {

        HttpResponse res = new HttpResponse();
        res.setBody('Shipped- On Time');
        res.setStatusCode(200);

        return res;
    }
}

```

2. ShipmentTrackingControllerTest:

@isTest

```
public class ShipmentTrackingControllerTest {
```

```
    @isTest
```

```
    static void fetchStatus() {
```

```
        Account acc = new Account();
```

```
        acc.Name = 'Test';
```

```
        insert acc;
```

```
        Test.setMock(HttpCalloutMock.class, new ShipmentTrackingControllerMock());
```

```
        Test.startTest();
```

```
        String res = ShipmentTrackingController.fetchShipmentStatus('Account',  
'Tracking_Number__c', acc.Id);
```

```
        Test.stopTest();
```

```
        System.assertEquals(res, 'Shipped- On Time');
```


```
    }
```

```
}
```

Adding the Site URL:

Goto Setup -> Remote Site Settings

Click on New and provide below details:

 **Remote Site Settings**

Remote Site Details

[Help for this Page](#)

Remote Site Detail		Edit	Delete	Clone
Remote Site Name	Merzcommunities	Modified By		Ajay Maguluri, 1/18/2024, 5:07 AM
Remote Site URL	https://merzcommunities--mel.sandbox.my.salesforce-sites.com			
Disable Protocol Security	<input type="checkbox"/>			
Description				
Active	<input checked="" type="checkbox"/>			
Created By	Ajay Maguluri, 1/18/2024, 5:07 AM			

[Edit](#) [Delete](#) [Clone](#)

Remote Site Name: Merzcommunities

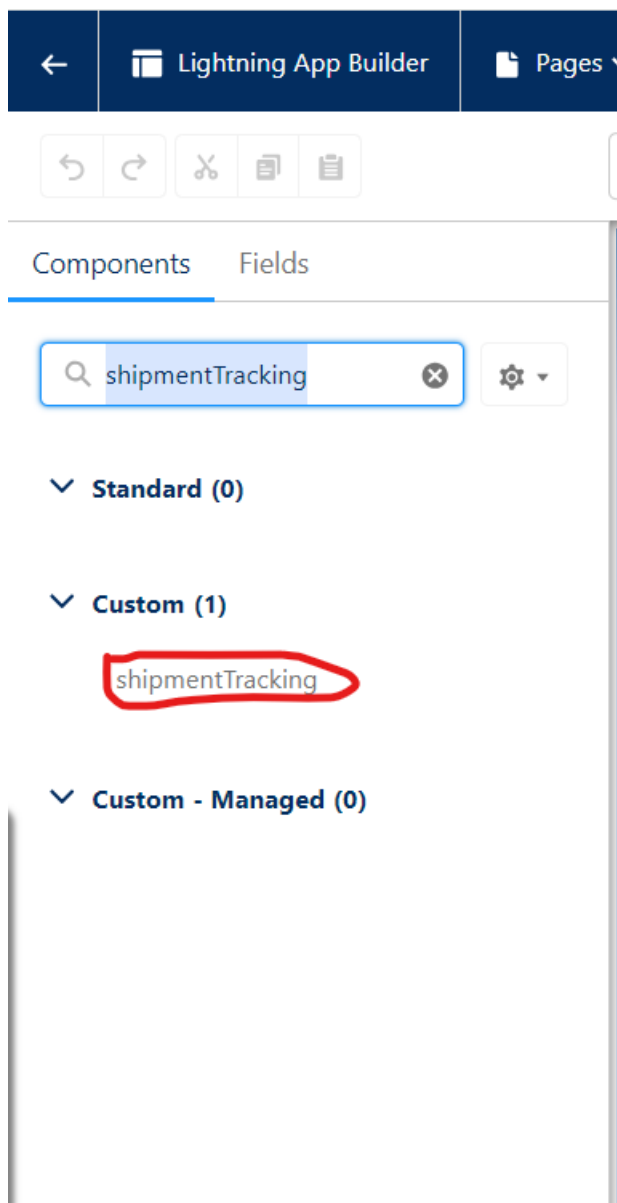
Remote Site URL: <https://merzcommunities--mel.sandbox.my.salesforce-sites.com>

How to place the LWC in record detail page

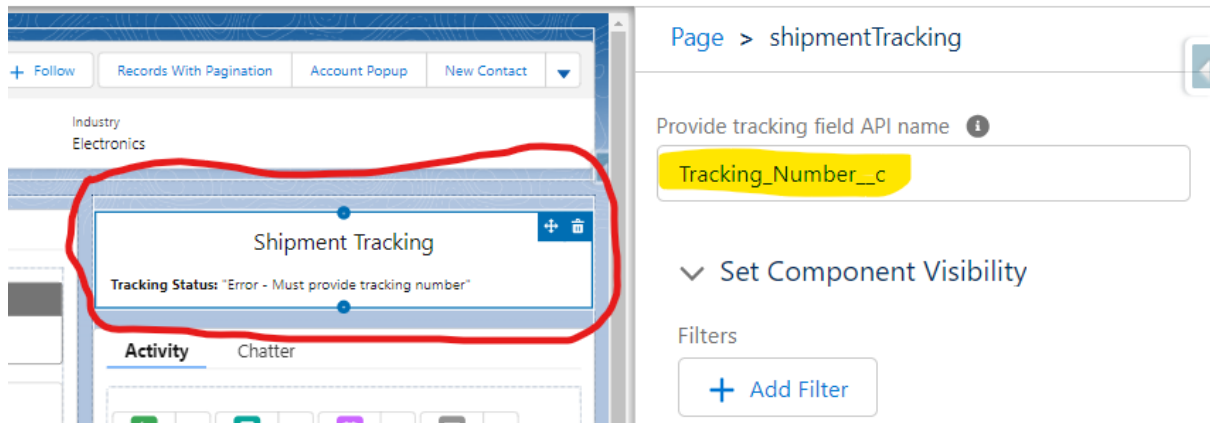
Open any record and click on Setup button.

Click on Edit Page.

In the Search box enter shipmentTracking and in Custom you can see the LWC component.



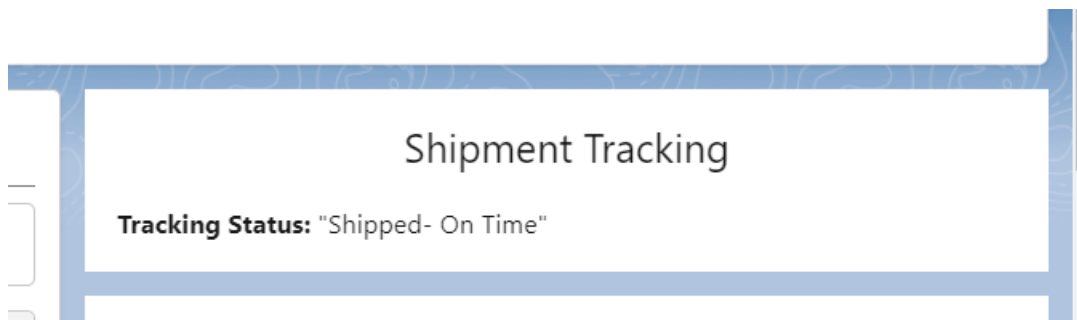
Drag and drop it on the record detail page.



In the right side, there is an input box to provide the tracking number field API. In that box provide the field API related to tracking number.

Click on Save and go back to the record detail page.

If the tracking field contains any value, the output will be like this:



Or if the tracking field value is empty, then output is:

