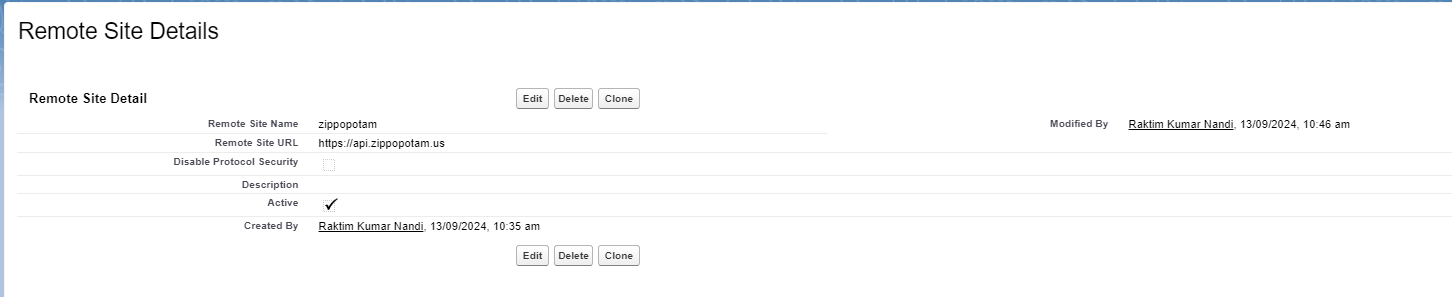
Use Case 1:

1. Create a lwc (lightning web component) that takes zip code as an input and makes an api call. You can use the free zip code api (http://api.zippopotam.us/us/). 2) If the country in response is “US”, response should be shown in another lightning component (No need to save data) but if the country value is anything else, response should be stored in a custom object. 3) Use api url dynamically e.g. if country is Mexico, use http://api.zippopotam.us/MX/01000 4) We should be able to see the non-US data in a different screen. 5) Achieve the same using flow with proper exception handling. For this create a separate UI experience. 6) Make sure to add test classes to adhere with Salesforce best practices.

>> Create Remote Site settings for the Endpoint URL



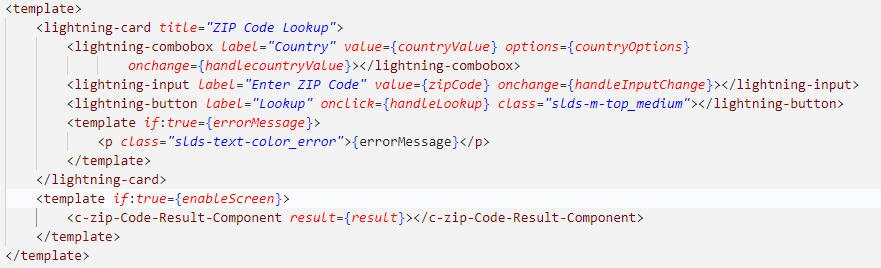
>> Create a custom object named **Custom\_Zipcode\_\_c** with the following fields:

* **ZipCode\_\_c (Text)**
* **Country\_\_c (Text)**
* **Place\_Details\_\_c (Long Text Area)**

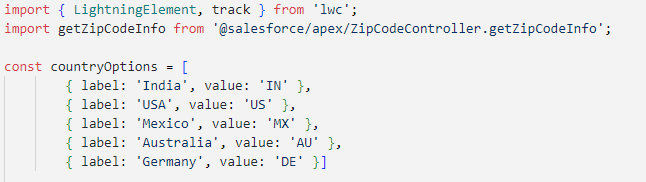
>> Create the LWC to Accept ZIP Code Input and Make API Call

Create an LWC component named **zipCodeComponent**.

**zipCodeComponent.html**



**zipCodeComponent.js**



****

Create an Apex controller **ZipCodeController** to handle API calls for the ZIP code lookup:



}

}

**3. Create Custom Object Custom\_Zipcode\_\_c**

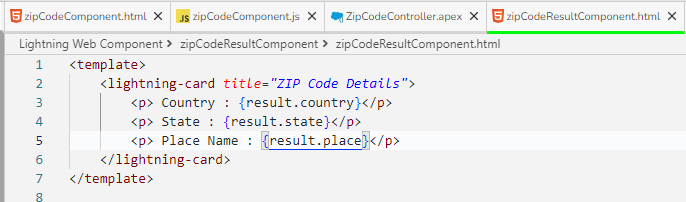
Create a custom object named Custom\_Zipcode\_\_c with the following fields:

* ZipCode\_\_c (Text)
* Country\_\_c (Text)

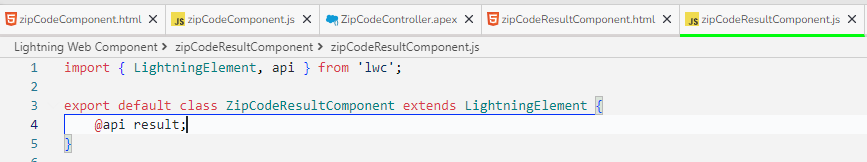
**Display Response in Another Component**

Create a new component to display the result. We'll trigger this component when the ZIP code belongs to the US.

**zipCodeResultComponent.html**

****

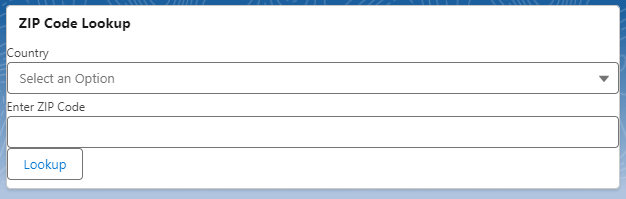
**zipCodeResultComponent.js**

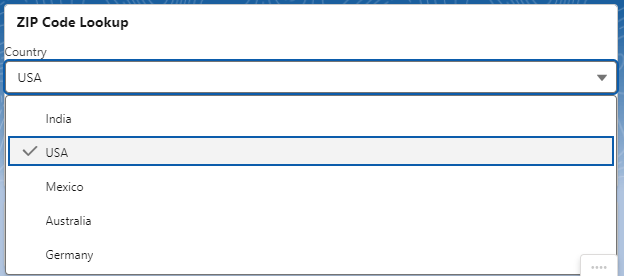


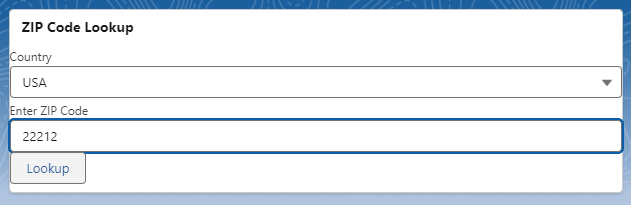
**Testing :---------------------------------------------------------------------------**

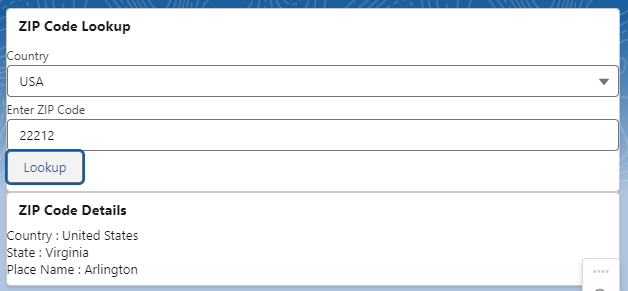
**Example :** [**https://api.zippopotam.us/IN/713206**](https://api.zippopotam.us/IN/713206)

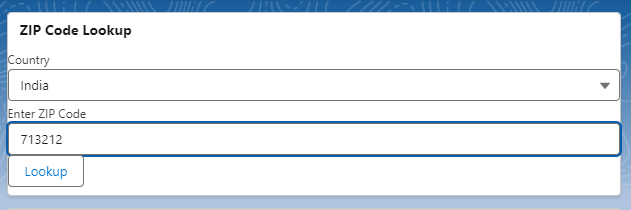
**Result :** {"post code": "713206", "country": "India", "country abbreviation": "IN", "places": [{"place name": "A B V Township Durgapur", "longitude": "87.5753", "state": "West Bengal", "state abbreviation": "WB", "latitude": "23.4621"}]}

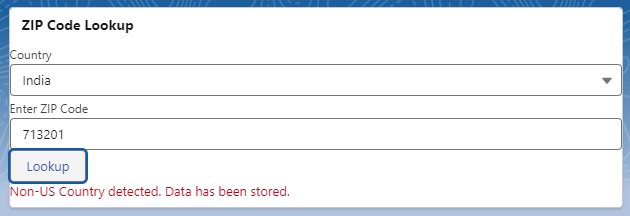


1. US Zipcode   




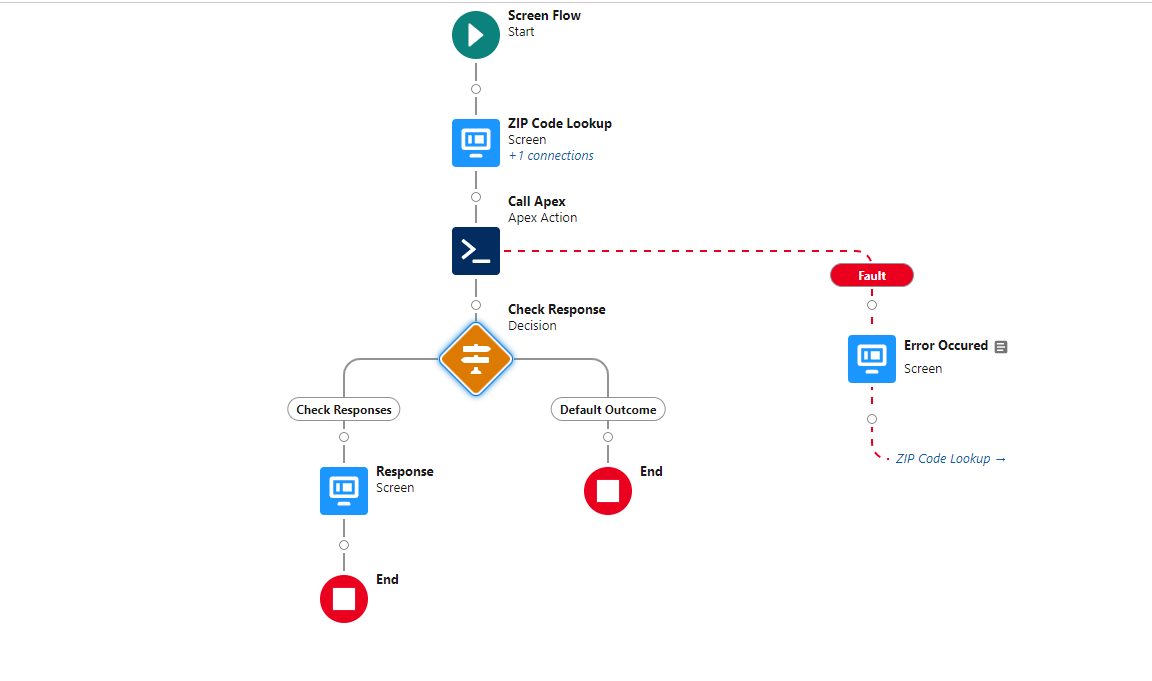


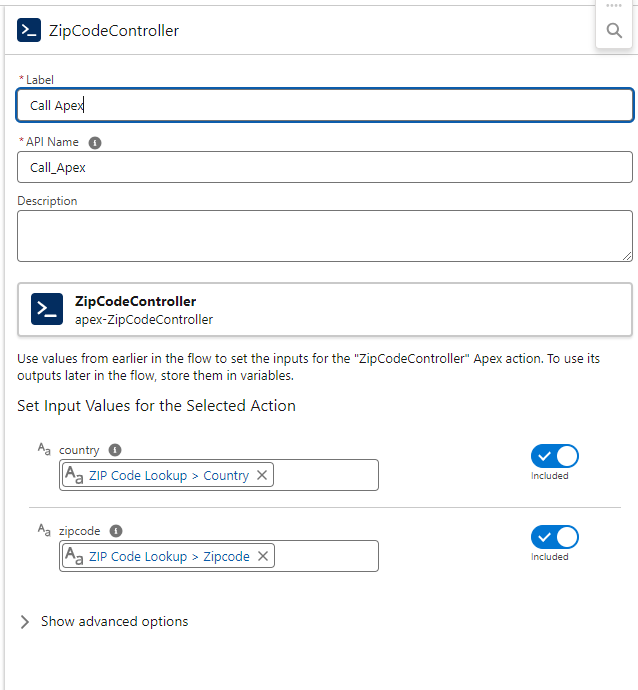
2. NON-US Zipcode  
  




**Salesforce Flow Implementation**

>> Create a Screen Flow

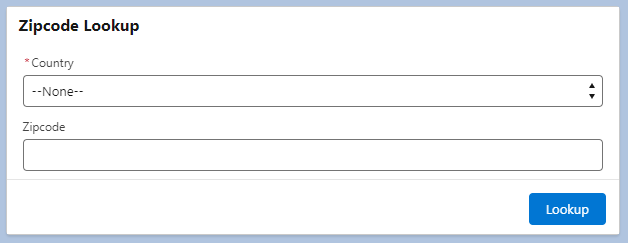


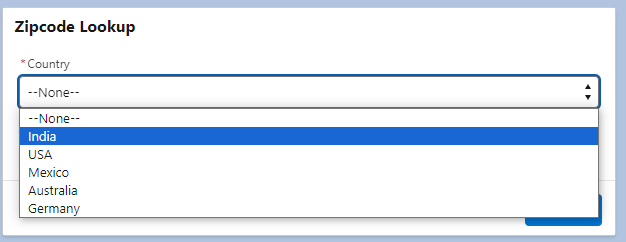


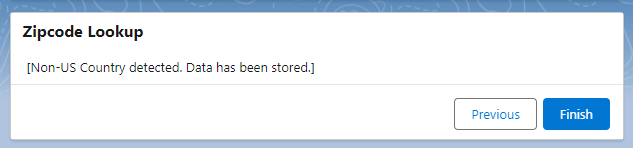
>> Invocable method to perform callout.

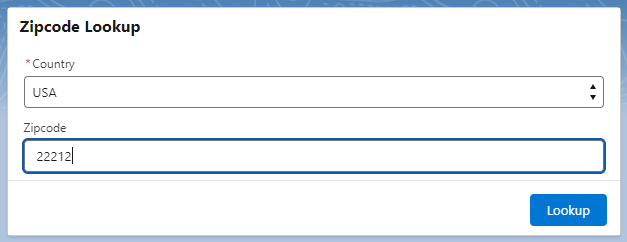


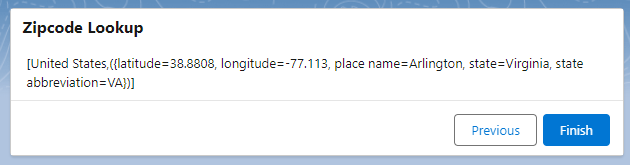
Alternative : can be done by Out of Box HTTP Callout from Flow , with Named and External Credential   
(Reason for not following this approach : auth error in my sandbox)

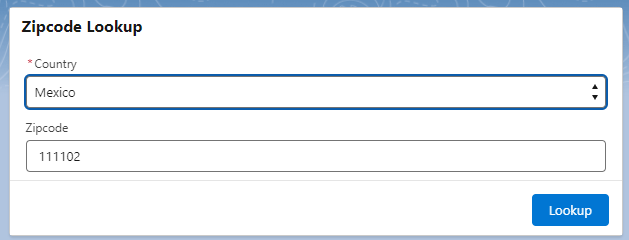
Results :  
  


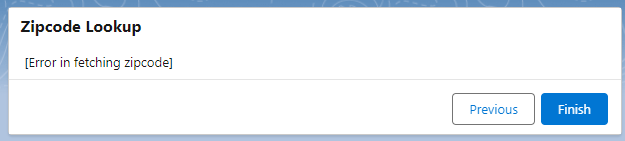


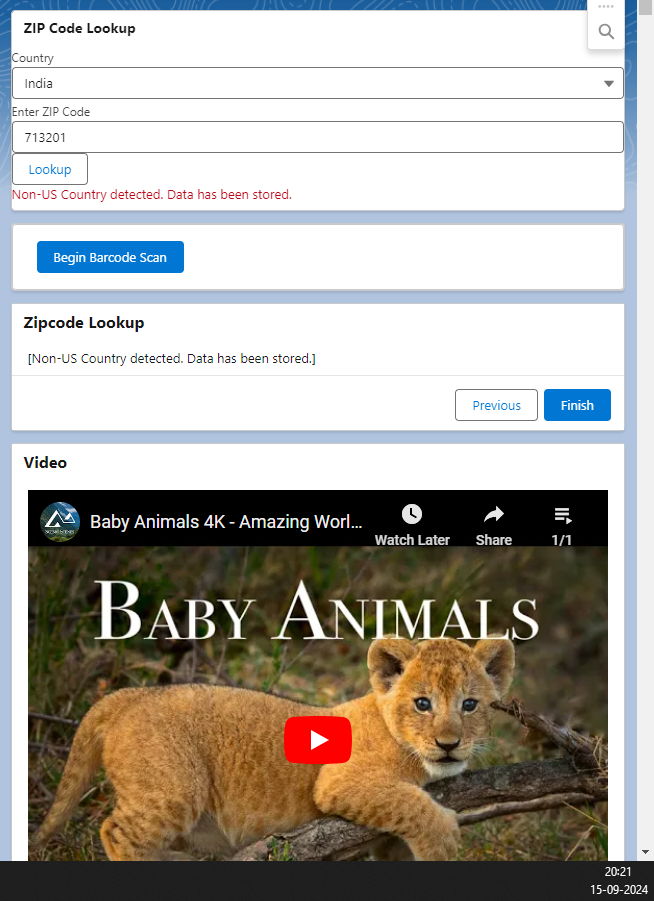


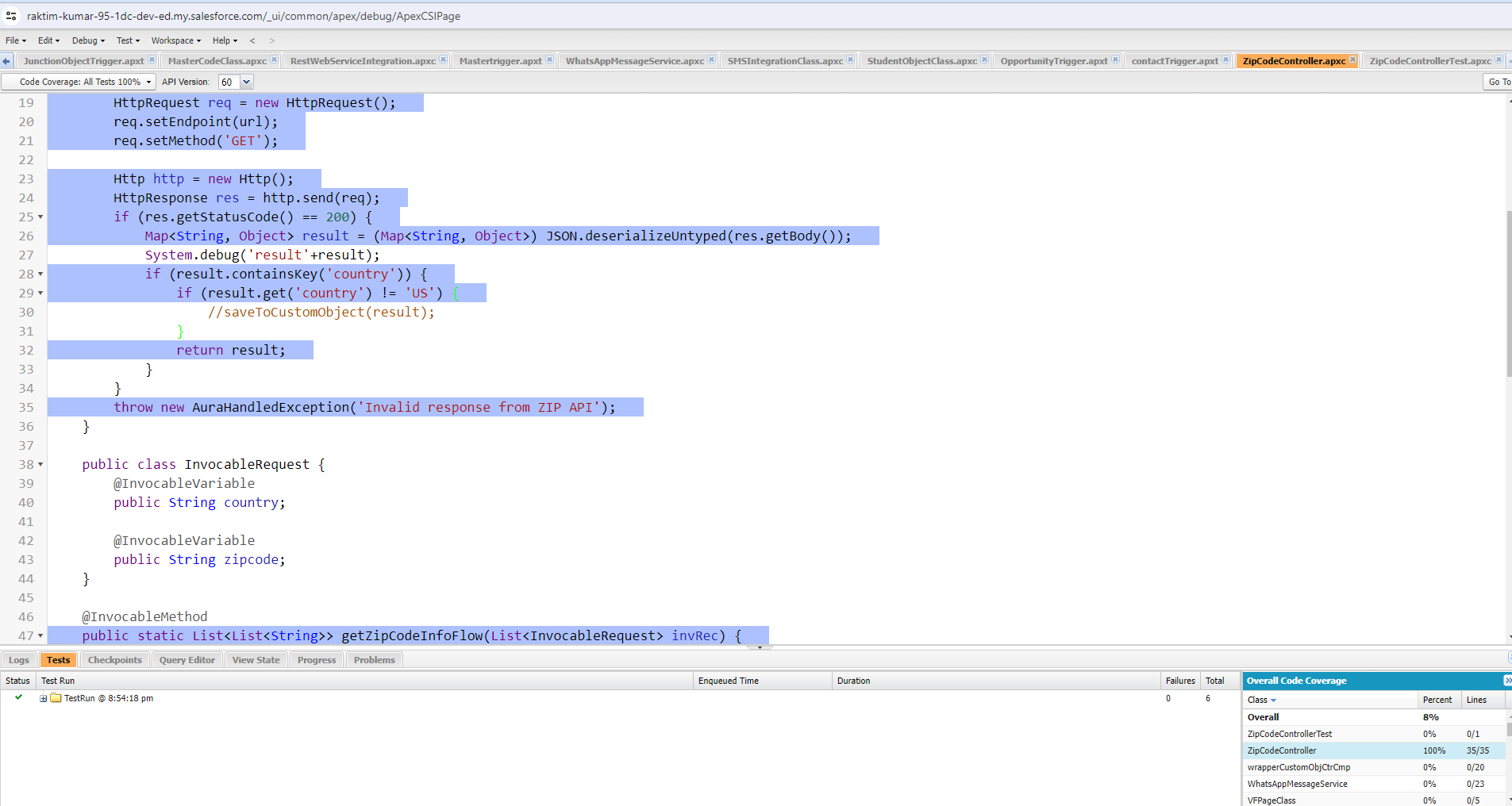




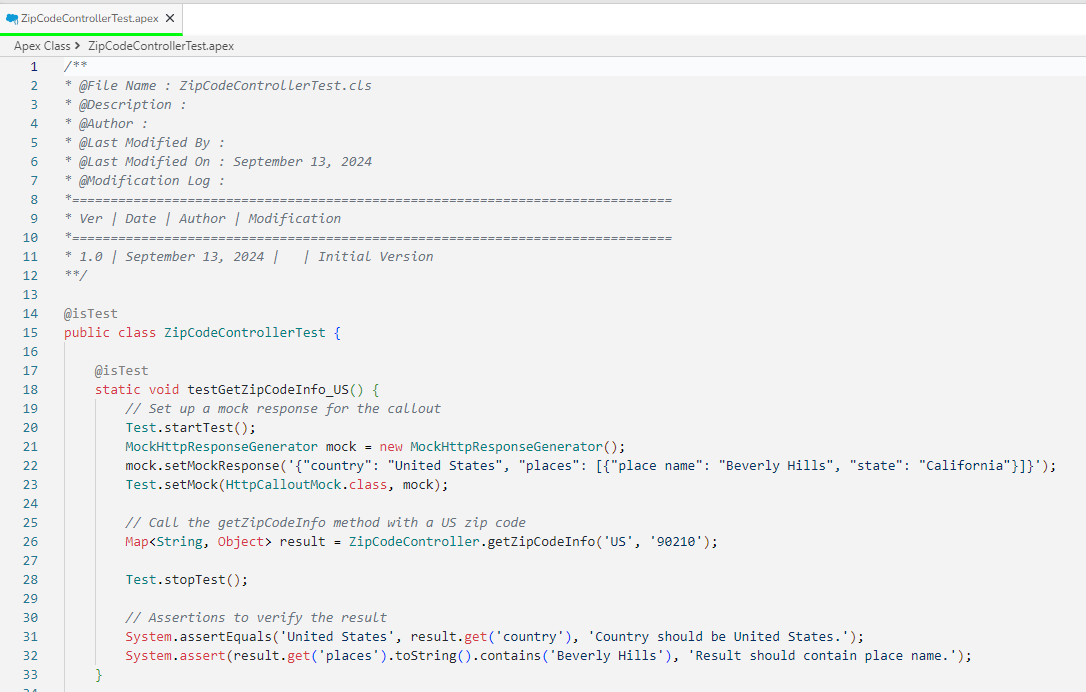




Both LWC and Flow :  


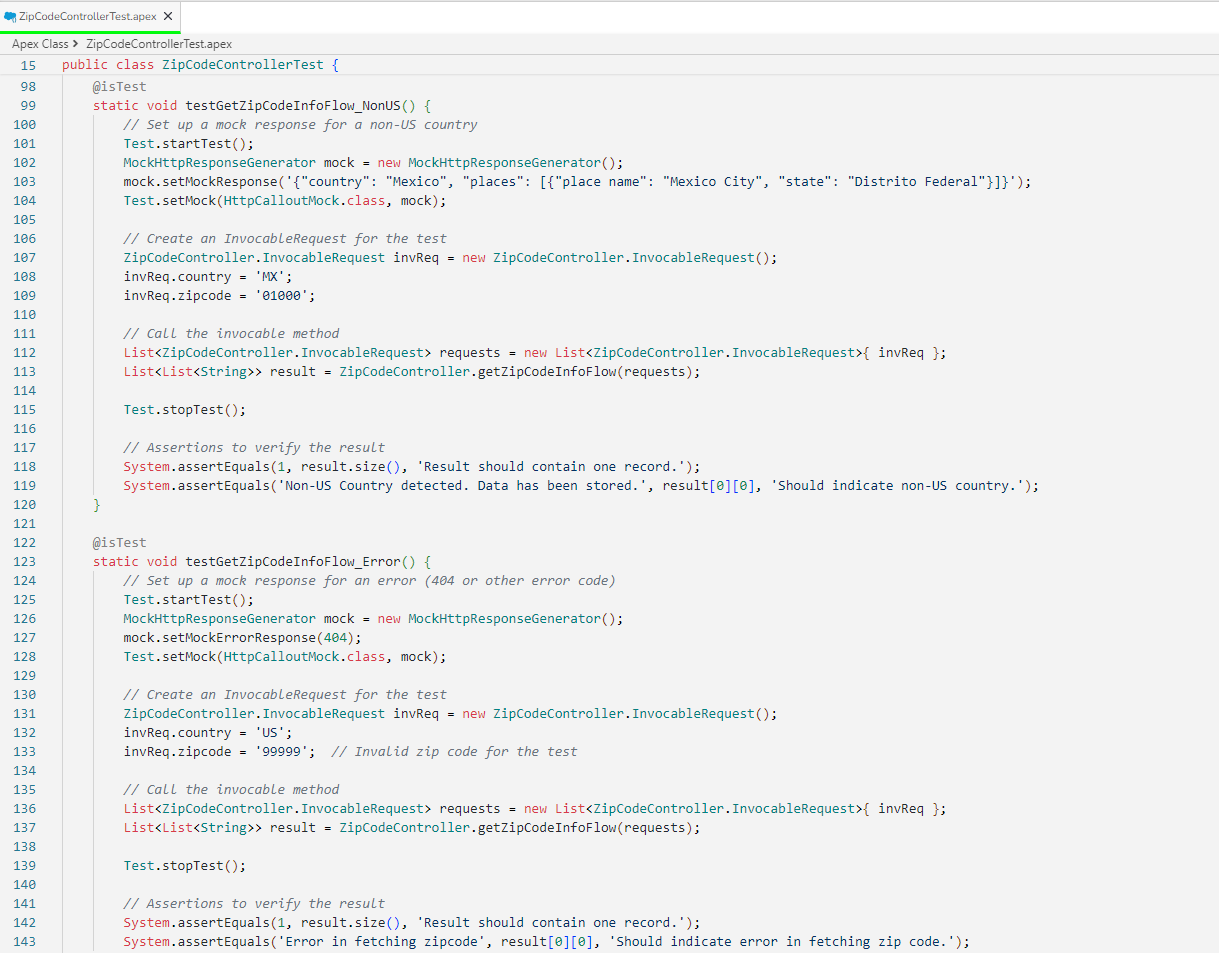
**Test Classes : 100% coverage  
**

**>> Code**









Use Case 2:

1) Create a custom field called “Risk” with values as High, Low, Medium.

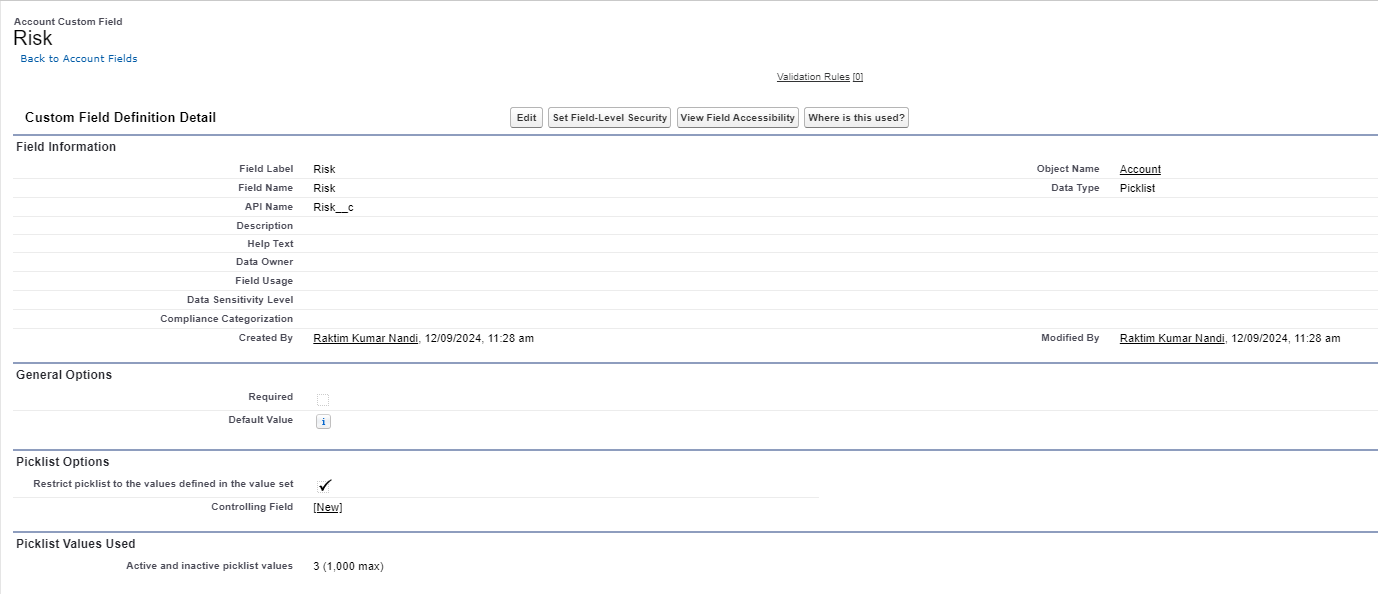
2) When the value “High” is selected, trigger a platform event.

3) Platform event should be subscribed, and case should be created. Case Owner should be a user different than account own

Soln :

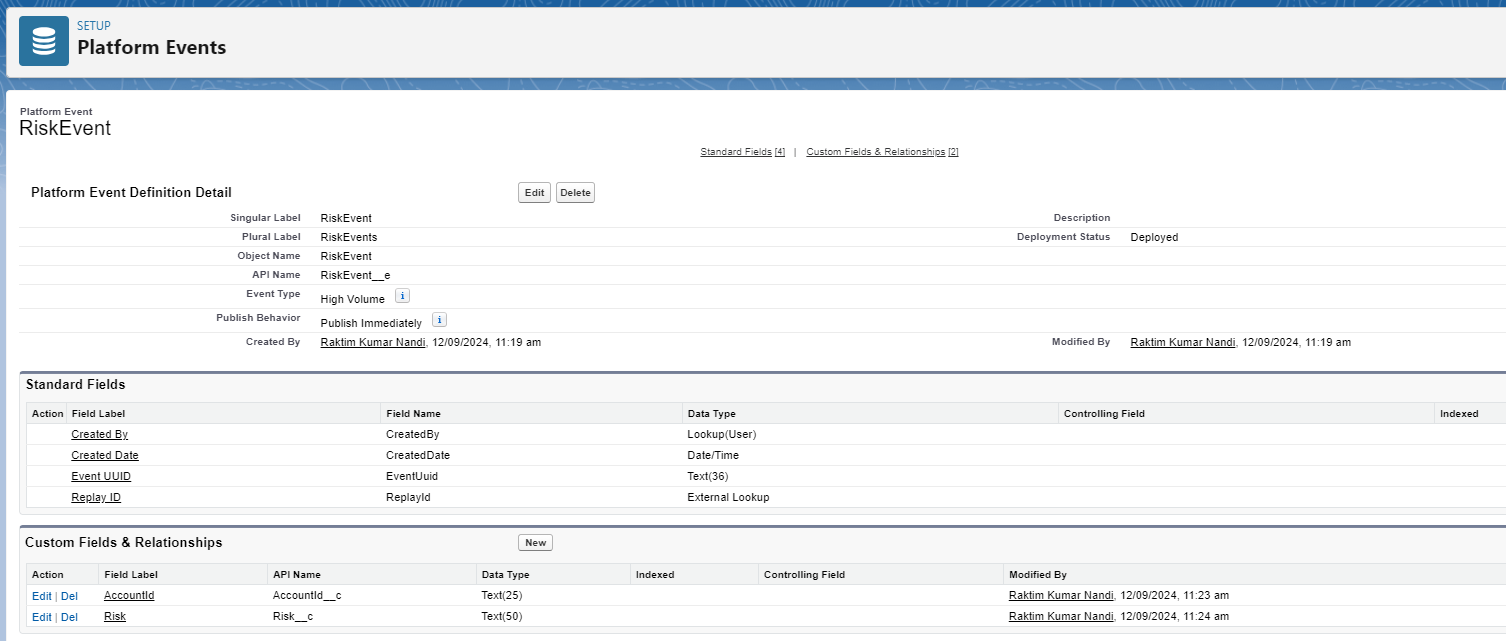
Create Custom Field "Risk"

* + Type : Picklist
  + Field Label: Risk.
  + Values: High, Medium, and Low



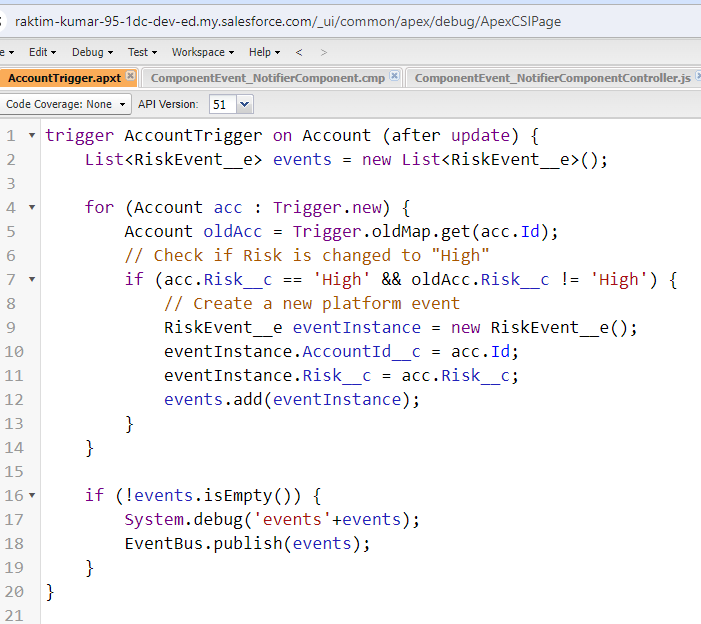
>> Create the Platform Event:

* Name: Risk Event.
* Select Publish Immediately.
* Create fields in the platform event, such as:
  + AccountId(Text)
  + Risk(Text)

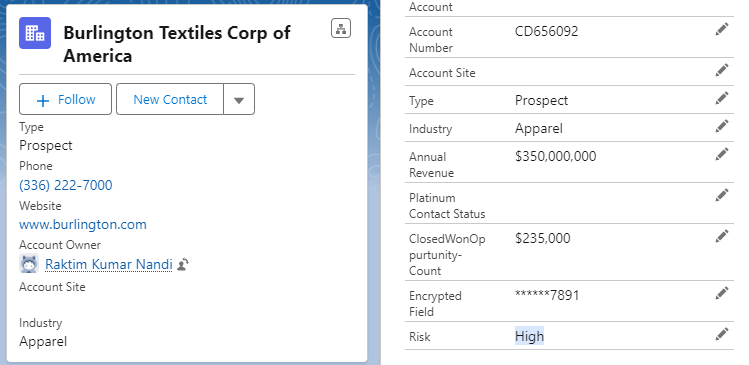


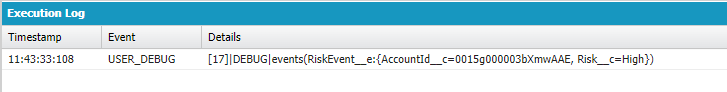
### Create Apex Trigger to Fire Platform Event

Create an Apex trigger those fires when the Risk field is updated to "High". This trigger will publish the platform event.

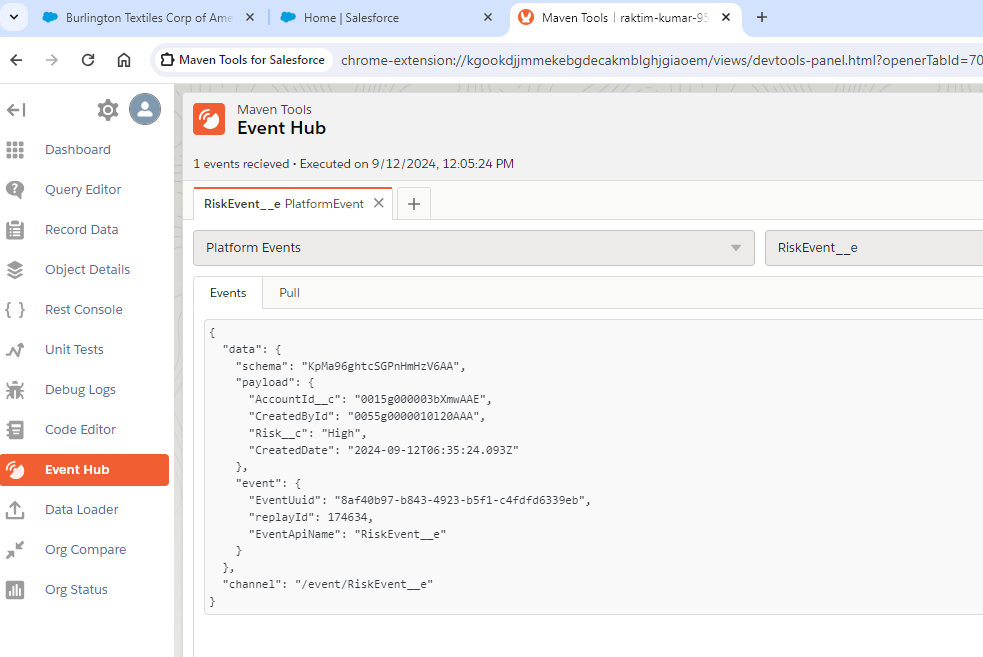


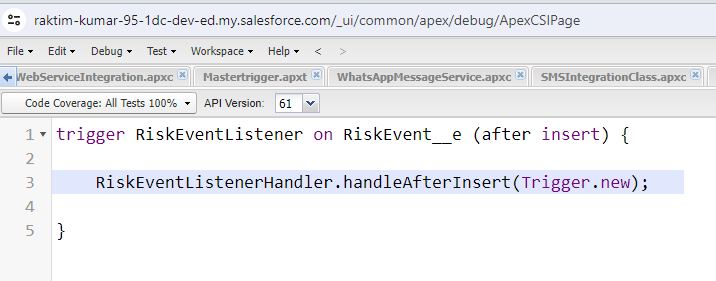
Testing :

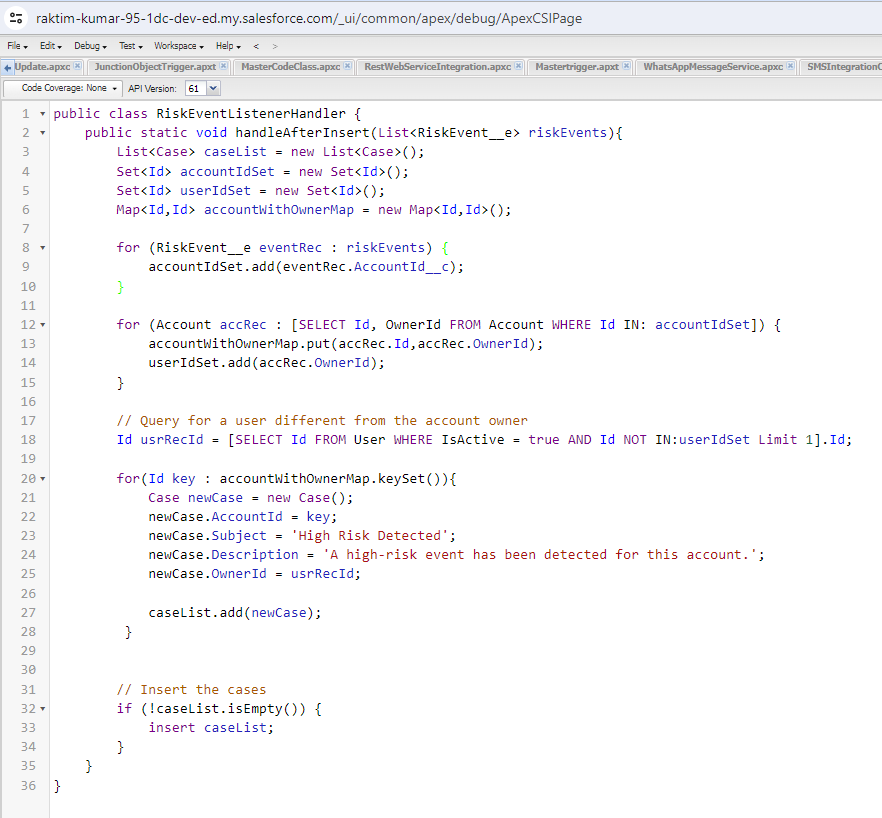
Publish the event satisfying Trigger Condition.  


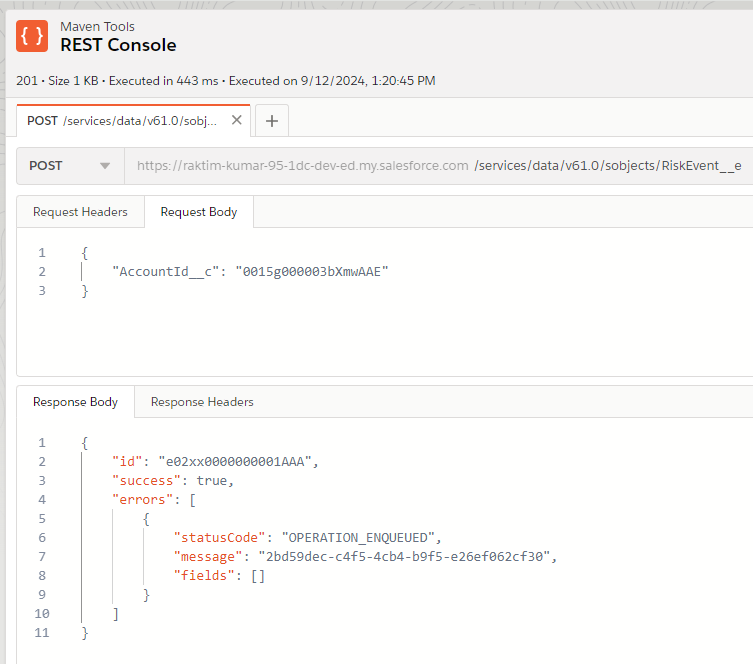


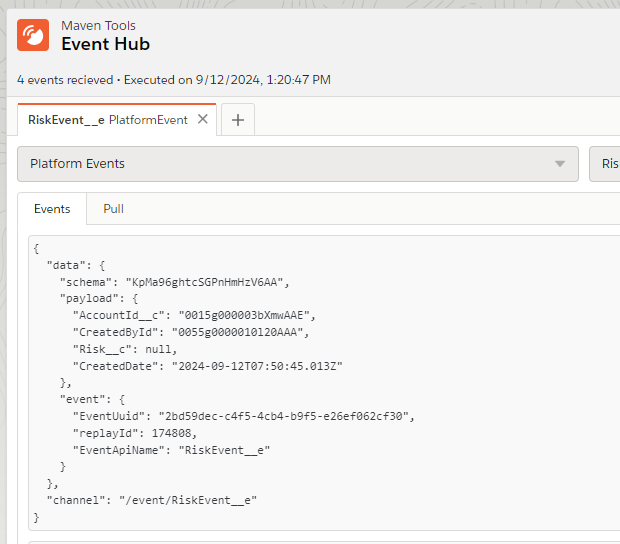
Subscribing the Event in External System from Salesforce:



>>> Subscribe the Event in Salesforce :  
Create another trigger that listens to the platform event, creates a case, and assigns it to a user different than the account owner  
  




Testing :   




Test Class to test the Publish and Subscribe Event :

