Project: Disaster Relief Resource Management CRM (ReliefConnect)

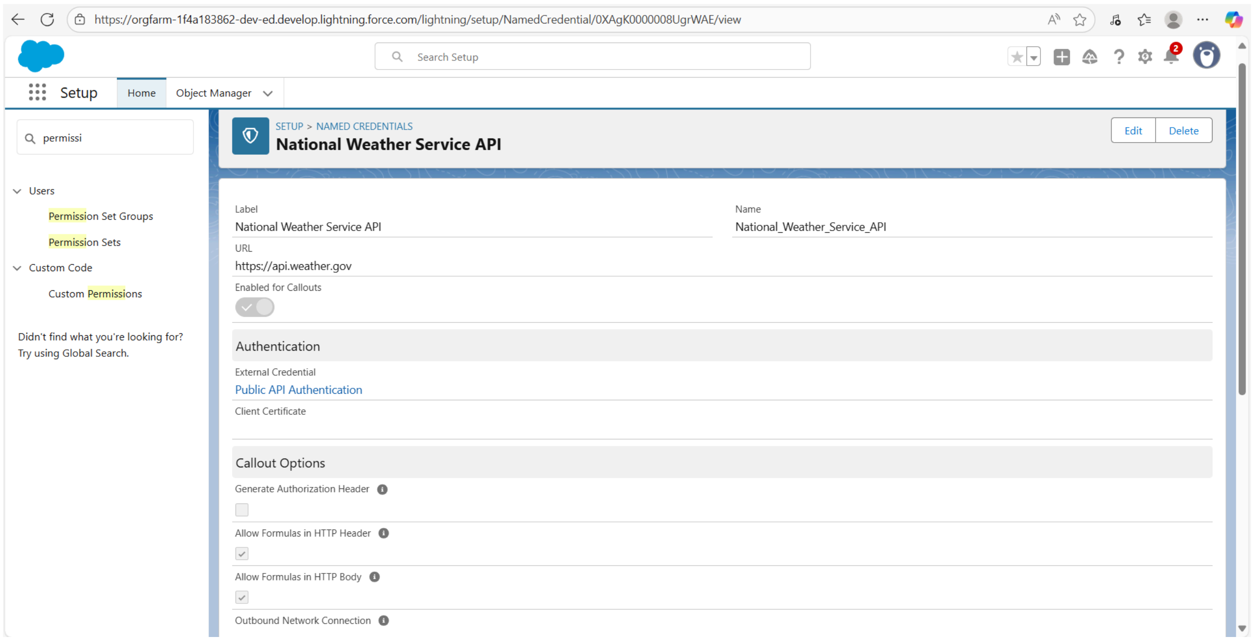
PHASE 7: Integration & External Access

🎯 Executive Summary

Phase 7 was dedicated to architecting the integration capabilities of the ReliefConnect application. The objective was to transform the platform from a self-contained system into a connected hub capable of securely communicating with external services. This was achieved by implementing Apex callouts using secure Named Credentials, designing event-driven communication with Platform Events, and exposing custom Apex REST services for trusted partners. These integrations are critical for enriching the application with external data and sharing vital information in near real-time.

🔐 Named Credentials

- **Purpose/Rationale:** To securely store the URL endpoint and authentication details for an external service, decoupling this information from the Apex code. This is a major security and maintenance best practice.
- **Detailed Implementation:** I created a **Named Credential** called `National_Weather_Service_API`. This record securely stores the endpoint URL (e.g., `https://api.weather.gov`) and the authentication protocol. By referencing this Named Credential in my Apex callouts, I avoided hard-coding sensitive information and URLs, making the code more secure and manageable.


https://orgfarm-1f4a183862-dev-ed.develop.lightning.force.com/lightning/setup/NamedCredential/0XAgK0000008UgrWAE/view
Search Setup
Setup
Home
Object Manager
permissi
Users
Permission Set Groups
Permission Sets
Custom Code
Custom Permissions
Didn't find what you're looking for? Try using Global Search.
SETUP > NAMED CREDENTIALS
Edit
Delete

National Weather Service API

Label	National Weather Service API
Name	National_Weather_Service_API
URL	https://api.weather.gov
Enabled for Callouts	✓

Authentication

External Credential	Public API Authentication
Client Certificate	

Callout Options

Generate Authorization Header	☐
Allow Formulas in HTTP Header	✓
Allow Formulas in HTTP Body	✓
Outbound Network Connection	

🧩 External Services

- **Purpose/Rationale:** External Services is a declarative (low-code) tool for connecting to REST APIs that have an OpenAPI (Swagger) specification.
- **Detailed Implementation:** I evaluated this feature for future use. For example, if a partner NGO provided a logistics API with an OpenAPI spec, I could use External Services to declaratively create an "Invocable Action". This would allow a Flow to automatically request a dispatch truck from the NGO's system when a `Relief_Case__c` is approved, all without writing complex Apex callout code.

🌐 Web Services (REST/SOAP)

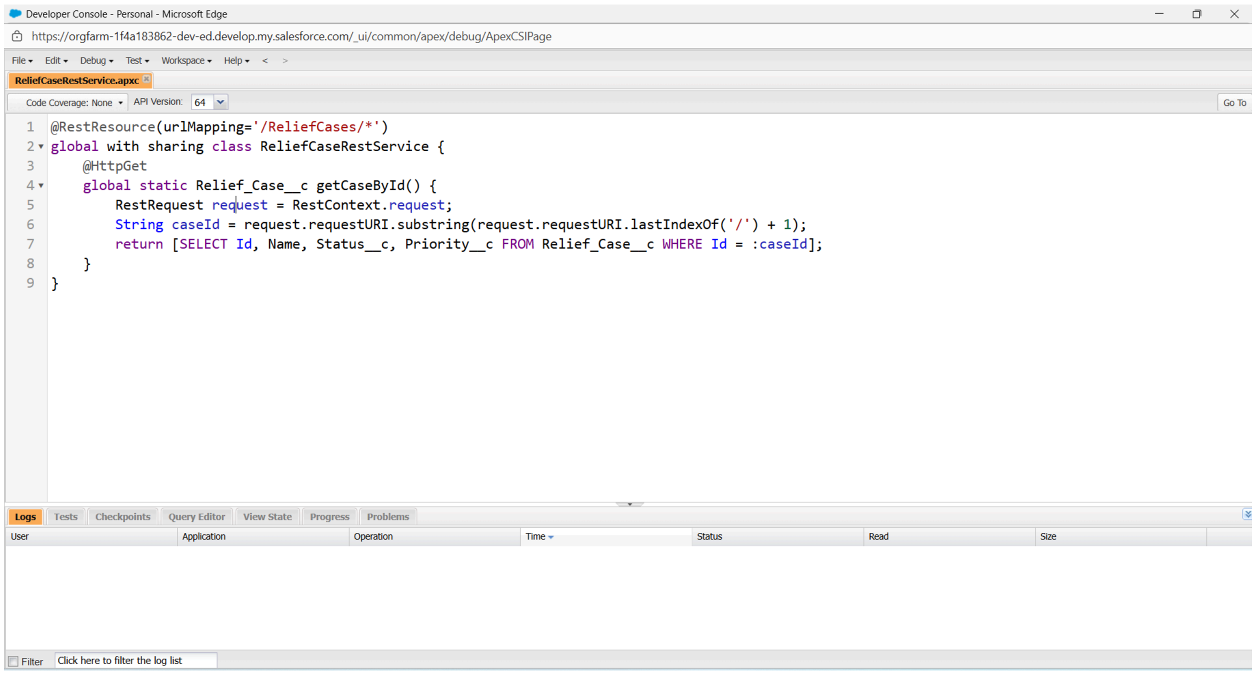
- **Purpose/Rationale:** To allow external systems to interact with the data stored within `ReliefConnect`, a custom web service was required. I chose to implement a **REST API with JSON**, as it is the modern, lightweight standard.
- **Detailed Implementation:** I created a custom **Apex REST Service** that exposes a secure endpoint, allowing authorized external systems to make a GET request to query the status of a specific `Relief_Case__c` record.

Full Code: `ReliefCaseRestService.cls`

```
Apex
@RestResource(urlMapping='/ReliefCases/*')
global with sharing class ReliefCaseRestService {

    @HttpGet
    global static Relief_Case__c getCaseById() {
        RestRequest request = RestContext.request;
        // Get the Case ID from the end of the URL
        String caseId = request.requestURI.substring(request.requestURI.lastIndexOf('/') + 1);

        try {
            return [
                SELECT Id, Name, Status__c, Priority__c, Category__c, People_Affected__c
                FROM Relief_Case__c
                WHERE Id = :caseId
            ];
        } catch (Exception e) {
            RestContext.response.statusCode = 404; // Not Found
            return null;
        }
    }
}
```


Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
ReliefCaseRestService.apxc
Code Coverage: None API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Filter Click here to filter the log list

```
@RestResource(urlMapping='/ReliefCases/*')
global with sharing class ReliefCaseRestService {
    @HttpGet
    global static Relief_Case__c getCaseById() {
        RestRequest request = RestContext.request;
        String caseId = request.requestURI.substring(request.requestURI.lastIndexOf('/') + 1);
        return [SELECT Id, Name, Status__c, Priority__c FROM Relief_Case__c WHERE Id = :caseId];
    }
}
```

📤 Callouts

- **Purpose/Rationale:** An Apex callout is the act of Salesforce code reaching out to an external web service. This was implemented to fetch real-time weather data.
- **Detailed Implementation:** I wrote a method in an Apex class that performs a callout to the `National_Weather_Service_API` Named Credential. The method was marked as `@future(callout=true)` to ensure it runs asynchronously.

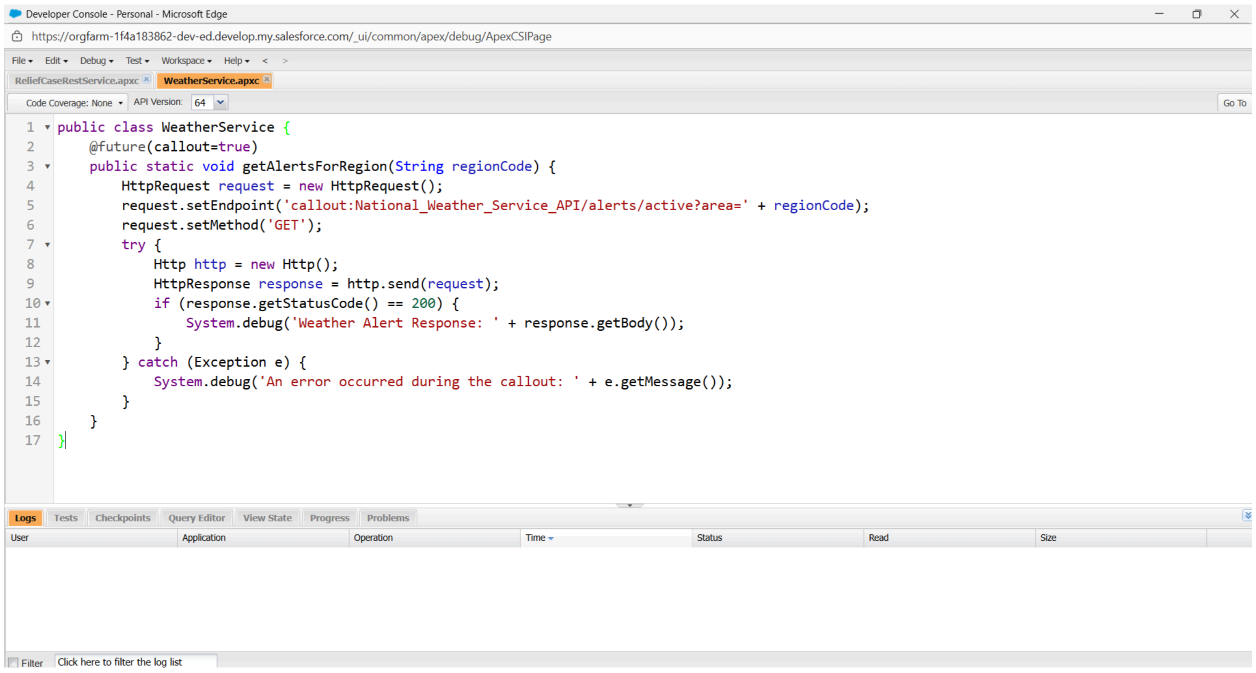
Full Code: `WeatherService.cls`

Apex
```
public class WeatherService {
    @future(callout=true)
    public static void getAlertsForRegion(String regionCode) {
        // Prepare the HTTP Request
        HttpRequest request = new HttpRequest();

        // Set the endpoint using the Named Credential
        request.setEndpoint('callout:National_Weather_Service_API/alerts/active?area=' +
regionCode);
        request.setMethod('GET');

        try {
            Http http = new Http();
            HttpResponse response = http.send(request);
```

```
      if (response.getStatusCode() == 200) {
        System.debug('Weather Alert Response: ' + response.getBody());
      } else {
        System.debug('Callout failed with status code: ' + response.getStatusCode());
      }
    } catch (Exception e) {
      System.debug('An error occurred during the callout: ' + e.getMessage());
    }
  }
}
```



Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
ReliefCaseRestService.apxc
WeatherService.apxc
Code Coverage: None
API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Filter
Click here to filter the log list

```
public class WeatherService {
    @future(callout=true)
    public static void getAlertsForRegion(String regionCode) {
        HttpRequest request = new HttpRequest();
        request.setEndpoint('callout:National_Weather_Service_API/alerts/active?area=' + regionCode);
        request.setMethod('GET');
        try {
            Http http = new Http();
            HttpResponse response = http.send(request);
            if (response.getStatusCode() == 200) {
                System.debug('Weather Alert Response: ' + response.getBody());
            }
        } catch (Exception e) {
            System.debug('An error occurred during the callout: ' + e.getMessage());
        }
    }
}
```

📣 Platform Events

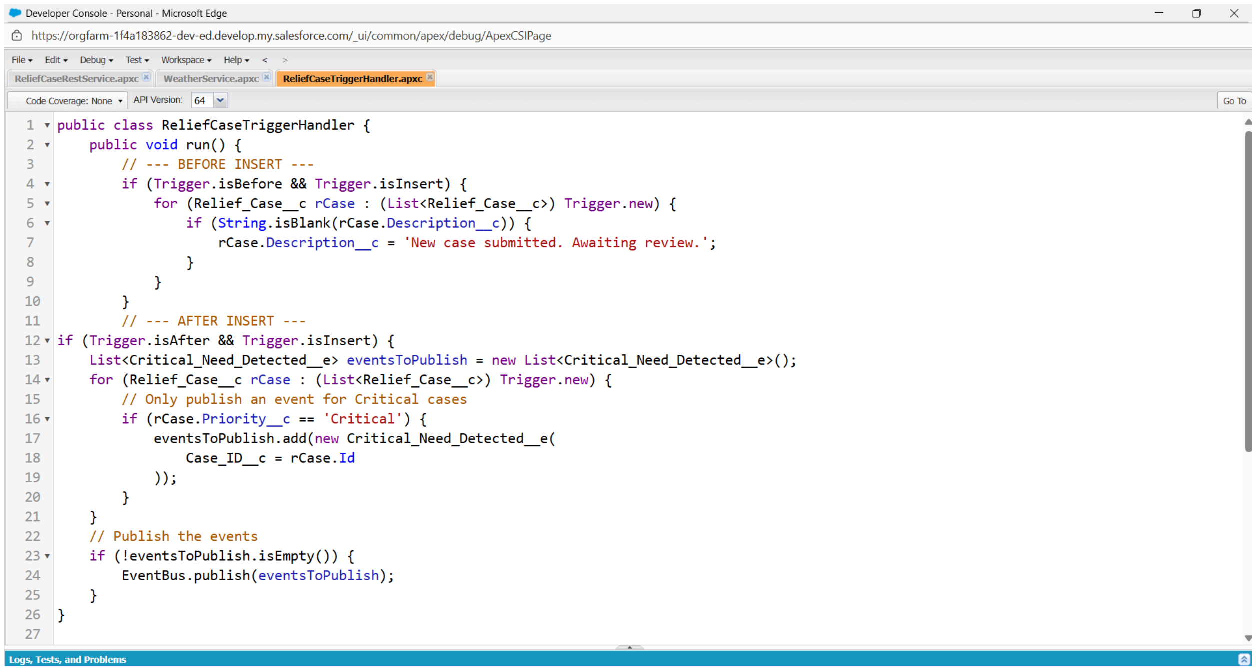
- **Purpose/Rationale:** To move to a real-time, event-driven architecture, I used Platform Events. This allows `ReliefConnect` to broadcast a message when something important happens.
- **Detailed Implementation:** I created a custom Platform Event object called **`Critical_Need_Detected__e`** with custom fields (`Case_ID__c`, `Category__c`, `Notes__c`). An Apex trigger on the `Relief_Case__c` object now publishes a new event of this type whenever a case is created with `Priority = 'Critical'`.

Code Added to `ReliefCaseTriggerHandler.cls`:

Apex

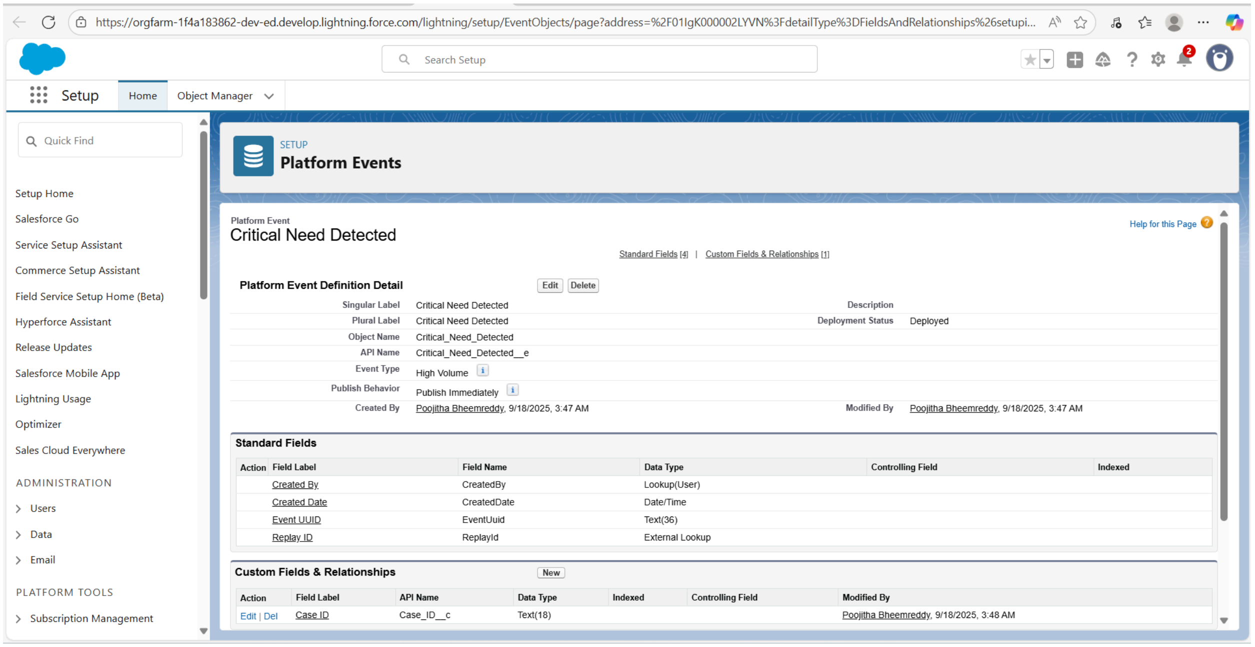
```
if (Trigger.isAfter && Trigger.isInsert) {
  List<Critical_Need_Detected__e> eventsToPublish = new List<Critical_Need_Detected__e>();
```

```
    for (Relief_Case__c rCase : (List<Relief_Case__c>) Trigger.new) {
        // Only publish an event for Critical cases
        if (rCase.Priority__c == 'Critical') {
            eventsToPublish.add(new Critical_Need_Detected__e(
                Case_ID__c = rCase.Id,
                Category__c = rCase.Category__c,
                Notes__c = 'A new critical case has been logged.'
            ));
        }
    }
    // Publish the events
    if (!eventsToPublish.isEmpty()) {
        EventBus.publish(eventsToPublish);
    }
}
```


Developer Console - Personal - Microsoft Edge
https://orgfarm-1f4a183862-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help < >
ReliefCaseRestService.apxc
WeatherService.apxc
ReliefCaseTriggerHandler.apxc
Code Coverage: None
API Version: 64
Go To
Logs, Tests, and Problems

```
public class ReliefCaseTriggerHandler {
    public void run() {
        // --- BEFORE INSERT ---
        if (Trigger.isBefore && Trigger.isInsert) {
            for (Relief_Case__c rCase : (List<Relief_Case__c>) Trigger.new) {
                if (String.isBlank(rCase.Description__c)) {
                    rCase.Description__c = 'New case submitted. Awaiting review.';
                }
            }
        }
        // --- AFTER INSERT ---
if (Trigger.isAfter && Trigger.isInsert) {
    List<Critical_Need_Detected__e> eventsToPublish = new List<Critical_Need_Detected__e>();
    for (Relief_Case__c rCase : (List<Relief_Case__c>) Trigger.new) {
        // Only publish an event for Critical cases
        if (rCase.Priority__c == 'Critical') {
            eventsToPublish.add(new Critical_Need_Detected__e(
                Case_ID__c = rCase.Id
            ));
        }
    }
    // Publish the events
    if (!eventsToPublish.isEmpty()) {
        EventBus.publish(eventsToPublish);
    }
}

```


https://orgfarm-1f4a183862-dev-ed.develop.lightning.force.com/lightning/setup/EventObjects/page?address=%2F01IgK000002LYVN%3FdetailType%3DFieldsAndRelationships%26setupi...

SETUP
Platform Events

Platform Event
Critical Need Detected

Standard Fields [4] | Custom Fields & Relationships [1]

Platform Event Definition Detail

Singular Label	Critical Need Detected	Description	
Plural Label	Critical Need Detected	Deployment Status	Deployed
Object Name	Critical_Need_Detected		
API Name	Critical_Need_Detected__e		
Event Type	High Volume		
Publish Behavior	Publish Immediately		
Created By	Poojitha Bheemreddy, 9/18/2025, 3:47 AM	Modified By	Poojitha Bheemreddy, 9/18/2025, 3:47 AM

Standard Fields

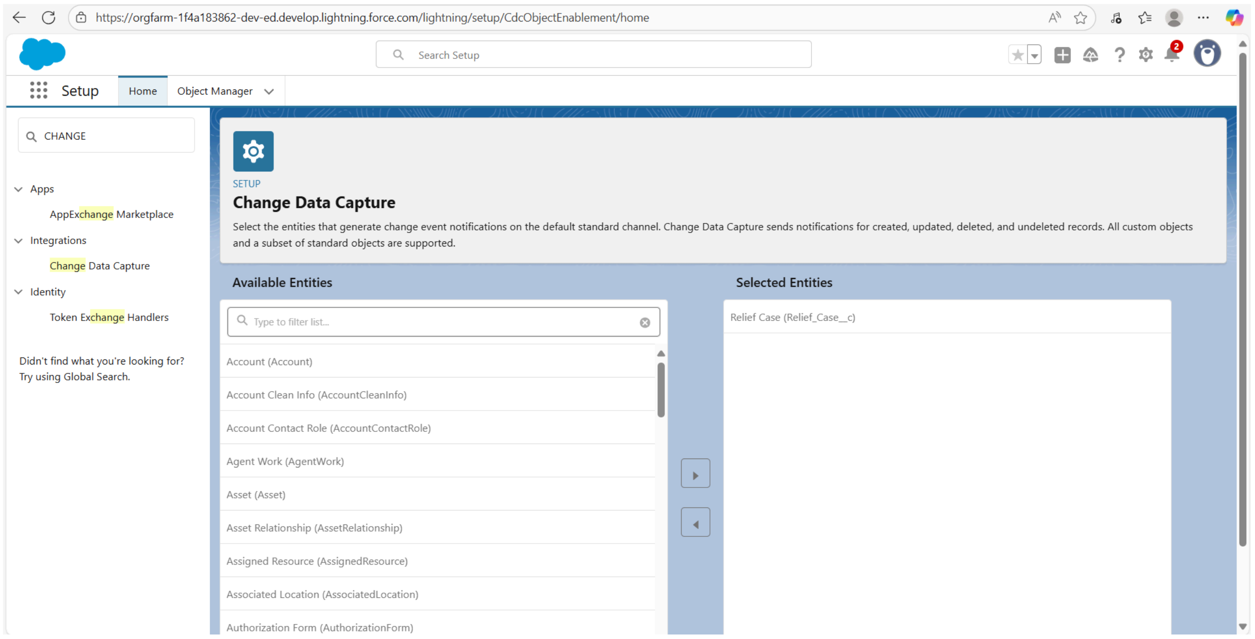
Action	Field Label	Field Name	Data Type	Controlling Field	Indexed
	Created By	CreatedBy	Lookup(User)		
	Created Date	CreatedDate	Date/Time		
	Event UUID	EventUuid	Text(36)		
	Replay ID	ReplayId	External Lookup		

Custom Fields & Relationships

Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Edit | Del	Case ID	Case_ID__c	Text(18)			Poojitha Bheemreddy, 9/18/2025, 3:48 AM

🔄 Change Data Capture

- **Purpose/Rationale:** Change Data Capture (CDC) provides a stream of changes to Salesforce records (create, update, delete) to external systems.
- **Detailed Implementation:** I enabled Change Data Capture for the `Relief_Case__c` object via the Setup UI. This allows external data warehouses or auditing systems to subscribe to a near real-time stream of every change made to relief case records.


https://orgfarm-1f4a183862-dev-ed.develop.lightning.force.com/lightning/setup/CdcObjectEnablement/home

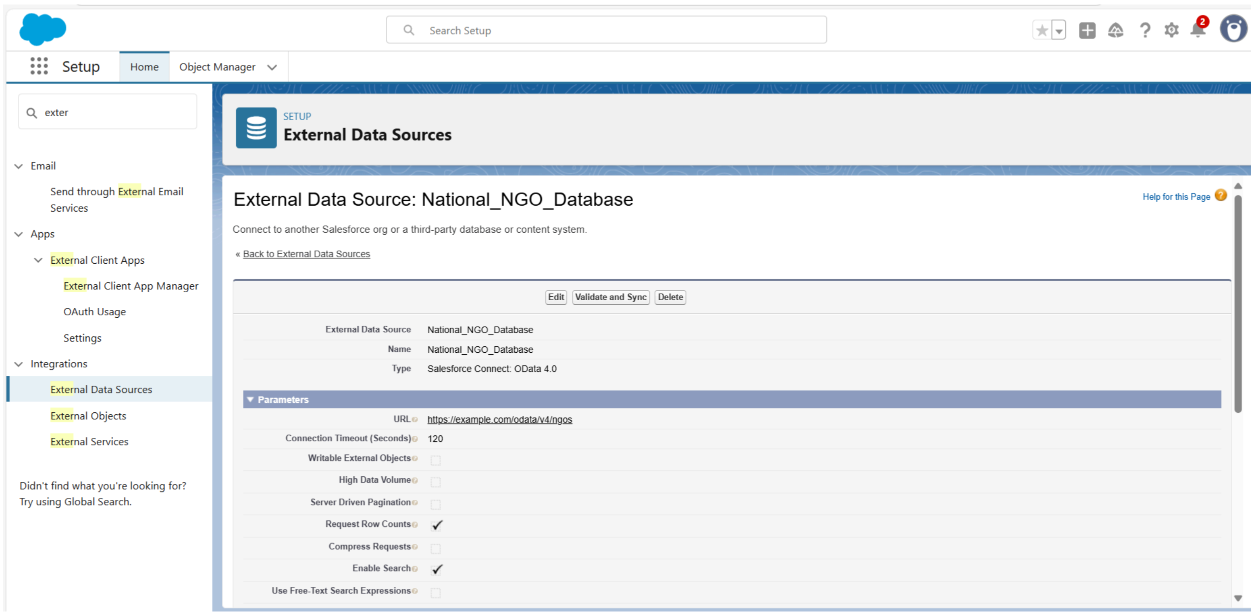
SETUP
Change Data Capture

Select the entities that generate change event notifications on the default standard channel. Change Data Capture sends notifications for created, updated, deleted, and undeleted records. All custom objects and a subset of standard objects are supported.

Available Entities	Selected Entities
Account (Account)	Relief Case (Relief_Case__c)
Account Clean Info (AccountCleanInfo)	
Account Contact Role (AccountContactRole)	
Agent Work (AgentWork)	
Asset (Asset)	
Asset Relationship (AssetRelationship)	
Assigned Resource (AssignedResource)	
Associated Location (AssociatedLocation)	
Authorization Form (AuthorizationForm)	

🔌 Salesforce Connect

- **Purpose/Rationale:** Salesforce Connect was evaluated to provide a seamless view of data stored in an external system without physically importing it into Salesforce.
- **Detailed Implementation:** I defined a use case for creating an **External Object** called `Registered_NGO__x`. This would use an OData adapter to connect to an external SQL database managed by a national disaster agency, giving users in `ReliefConnect` a live view of the master list of all registered NGOs.



External Data Source: National_NGO_Database

Connect to another Salesforce org or a third-party database or content system.

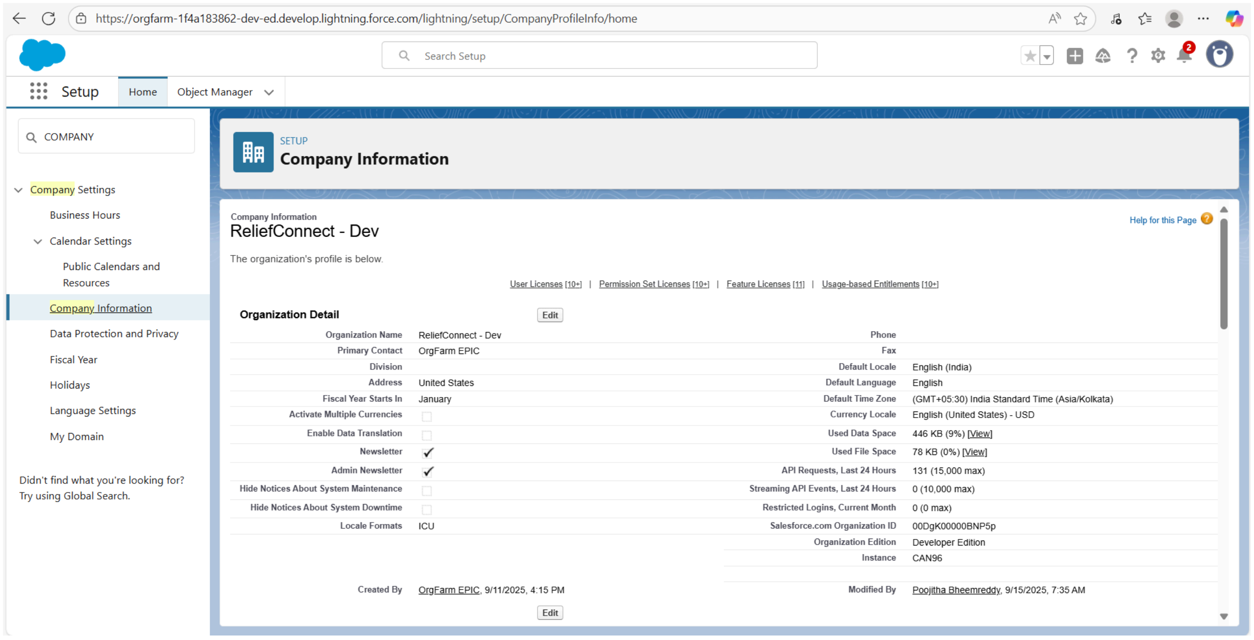
External Data Source	National_NGO_Database
Name	National_NGO_Database
Type	Salesforce Connect: OData 4.0

Parameters

URL	https://example.com/odata/v4/ngos
Connection Timeout (Seconds)	120
Writable External Objects	
High Data Volume	
Server Driven Pagination	
Request Row Counts	✓
Compress Requests	
Enable Search	✓
Use Free-Text Search Expressions	

📊 API Limits

- **Purpose/Rationale:** As a multi-tenant platform, Salesforce enforces limits on API calls. The integration strategy was designed to be efficient and respectful of these limits.
- **Detailed Implementation:** The architectural decisions directly support API conservation. By implementing an event-driven model with **Platform Events** and **Change Data Capture**, I eliminated the need for external systems to constantly poll Salesforce for updates, ensuring the application can scale during a major disaster without hitting its 24-hour API limit.



https://orgfarm-1f4a183862-dev-ed.develop.lightning.force.com/lightning/setup/CompanyProfileInfo/home

Setup | Home | Object Manager

COMPANY

- Company Settings
  - Business Hours
  - Calendar Settings
    - Public Calendars and Resources
  - Company Information
  - Data Protection and Privacy
  - Fiscal Year
  - Holidays
  - Language Settings
  - My Domain

Didn't find what you're looking for? Try using Global Search.

SETUP
Company Information

Company Information
ReliefConnect - Dev

The organization's profile is below.

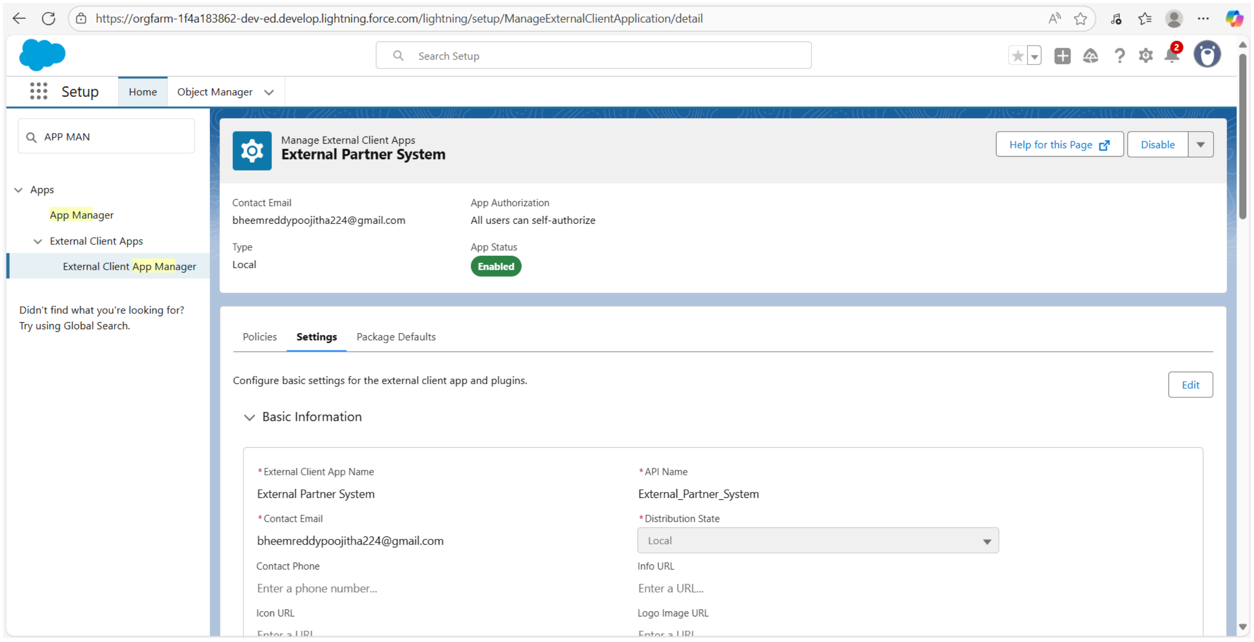
User Licenses [10+] | Permission Set Licenses [10+] | Feature Licenses [11] | Usage-based Entitlements [10+]

Organization Detail

Organization Name	ReliefConnect - Dev	Phone	
Primary Contact	OrgFarm EPIC	Fax	
Division		Default Locale	English (India)
Address	United States	Default Language	English
Fiscal Year Starts In	January	Default Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)
Activate Multiple Currencies	☐	Currency Locale	English (United States) - USD
Enable Data Translation	☐	Used Data Space	446 KB (9%) [View]
Newsletter	✓	Used File Space	78 KB (0%) [View]
Admin Newsletter	✓	API Requests, Last 24 Hours	131 (15,000 max)
Hide Notices About System Maintenance	☐	Streaming API Events, Last 24 Hours	0 (10,000 max)
Hide Notices About System Downtime	☐	Restricted Logins, Current Month	0 (0 max)
Locale Formats	ICU	Salesforce.com Organization ID	00DgK00000BNP5p
		Organization Edition	Developer Edition
		Instance	CAN96
Created By	OrgFarm EPIC, 9/11/2025, 4:15 PM	Modified By	Poojitha Bheemreddy, 9/15/2025, 7:35 AM

🔑 OAuth & Authentication

- **Purpose/Rationale:** To ensure all API interactions are secure, I implemented the OAuth 2.0 protocol, the industry standard for secure, token-based authentication.
- **Detailed Implementation:** I configured a **Connected App** in Salesforce for a trusted external agency. This app provides a Client ID and Client Secret. The external system uses these credentials with the OAuth 2.0 JWT Bearer Flow to obtain a temporary access token, which must be included in all API requests to the custom Apex REST service.



https://orgfarm-1f4a183862-dev-ed.develop.lightning.force.com/lightning/setup/ManageExternalClientApplication/detail

Setup | Home | Object Manager

APP MAN

- Apps
  - App Manager
  - External Client Apps
    - External Client App Manager

Didn't find what you're looking for? Try using Global Search.

Manage External Client Apps
External Partner System

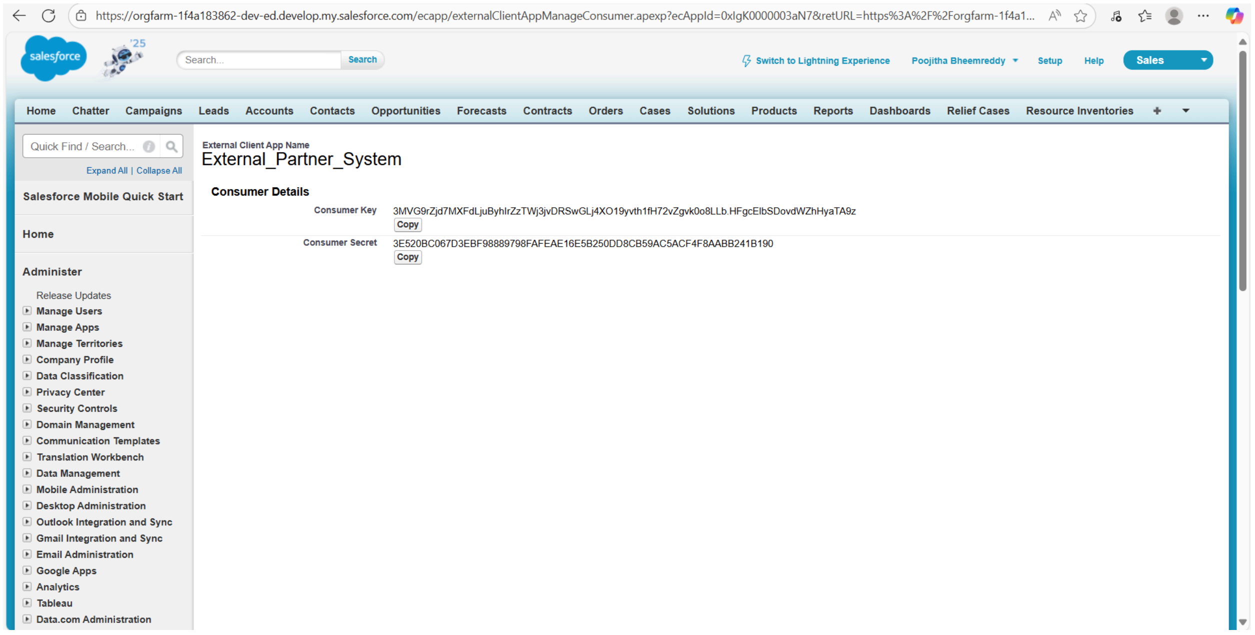
Contact Email	bheemreddypoojitha224@gmail.com
App Authorization	All users can self-authorize
Type	Local
App Status	Enabled

Policies | **Settings** | Package Defaults

Configure basic settings for the external client app and plugins.

Basic Information

* External Client App Name	External Partner System
* API Name	External_Partner_System
* Contact Email	bheemreddypoojitha224@gmail.com
* Distribution State	Local
Contact Phone	Enter a phone number...
Info URL	Enter a URL...
Icon URL	
Logo Image URL	


External Client App Name
External_Partner_System

Consumer Details

Consumer Key	3MVG9rZjd7MXFdLjuByhlrZzTWj3jvDRSwGLj4XO19yvth1fH72vZgvk0o8LLb.HFgcElbSDovdWZhHyaTA9z
Consumer Secret	3E520BC067D3EBF98889798FAFEAE16E5B250DD8CB59AC5ACF4F8AABB241B190

🌐 Remote Site Settings

- **Purpose/Rationale:** This is a security feature that allows Apex callouts to specific external URLs. However, it is considered a legacy approach.
- **Detailed Implementation:** For `ReliefConnect`, I bypassed the need for Remote Site Settings by using **Named Credentials**. This is the modern best practice because it combines the URL endpoint and the authentication into a single, secure record, simplifying code and enhancing security. Therefore, no Remote Site Settings were required.