AgriTrust Connect- Phase 7: Integration & External Access (Developer) Report

Project Title: AgriTrust Connect - Sustainable Agriculture CRM
Phase: 7 - Integration & External Access
Date: 24 September 2025
Prepared By: SRI VAISHNAVI B

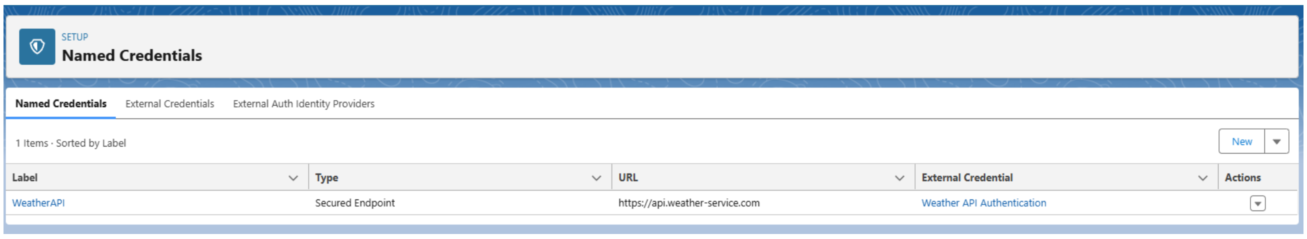
1. Introduction

Phase 7 focused on connecting the AgriTrust Connect application with the "outside world." The primary objective was to establish secure, scalable, and real-time integration patterns for both outbound (Salesforce to external systems) and inbound (external systems to Salesforce) data exchange. This phase transformed the application from a standalone system into a connected hub for agricultural data.

2. Outbound Integrations (Salesforce → External)

Purpose: To push data and notifications from Salesforce to external services and applications.

Technology Used	Implementation Details
Named Credentials	A Named Credential named WeatherAPI was created to securely store the endpoint URL and authentication details for an external weather service. This eliminates hardcoded credentials from the codebase.
Apex Callouts	The WeatherService.cls was implemented with an @future(callout=true) method. This class uses the WeatherAPI Named Credential to make a REST GET request, fetch weather forecast data, and automatically create Advisory__c records based on the response.
Platform Events	A Platform Event object, Procurement_Order_Event__e, was defined with a custom data payload. An Apex trigger on the Procurement__c object now publishes an event whenever a new order is created, allowing external ERP or supply chain systems to subscribe and receive notifications in near real-time.

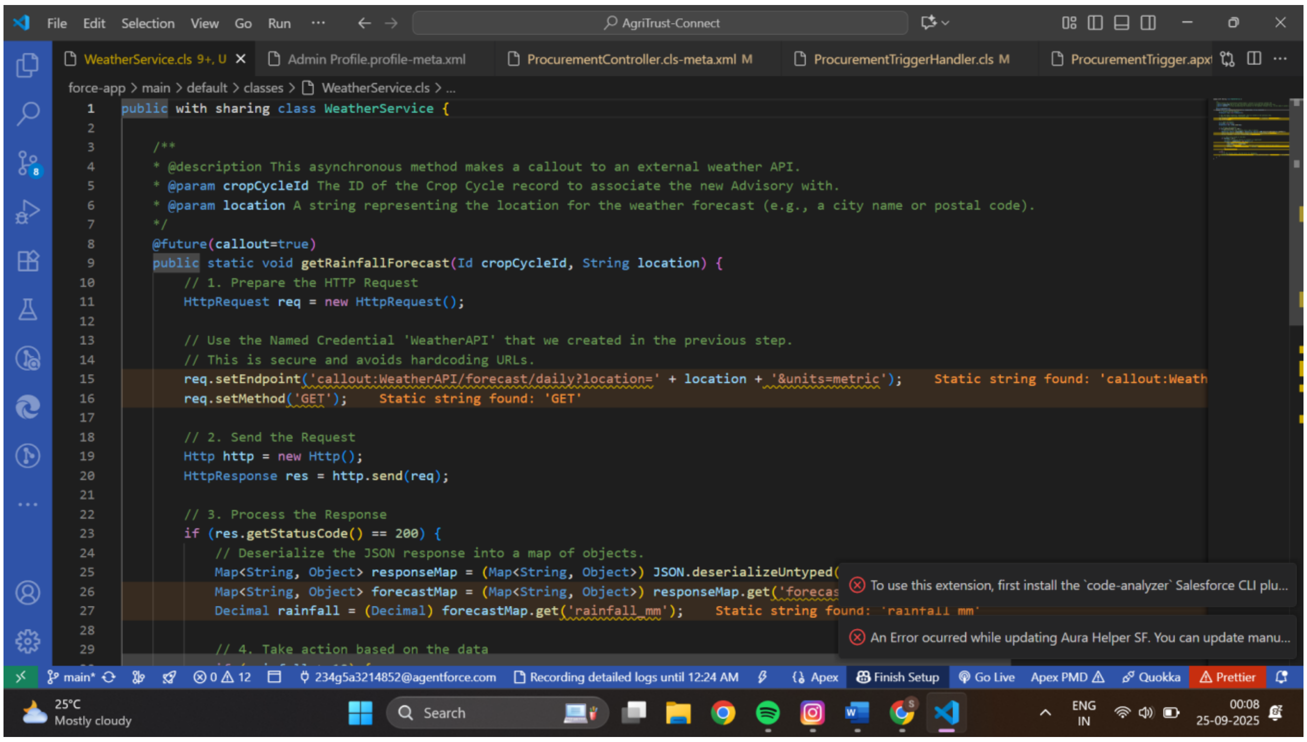

SETUP
Named Credentials

Named Credentials | External Credentials | External Auth Identity Providers

1 Items · Sorted by Label

New

Label	Type	URL	External Credential	Actions
WeatherAPI	Secured Endpoint	https://api.weather-service.com	Weather API Authentication	



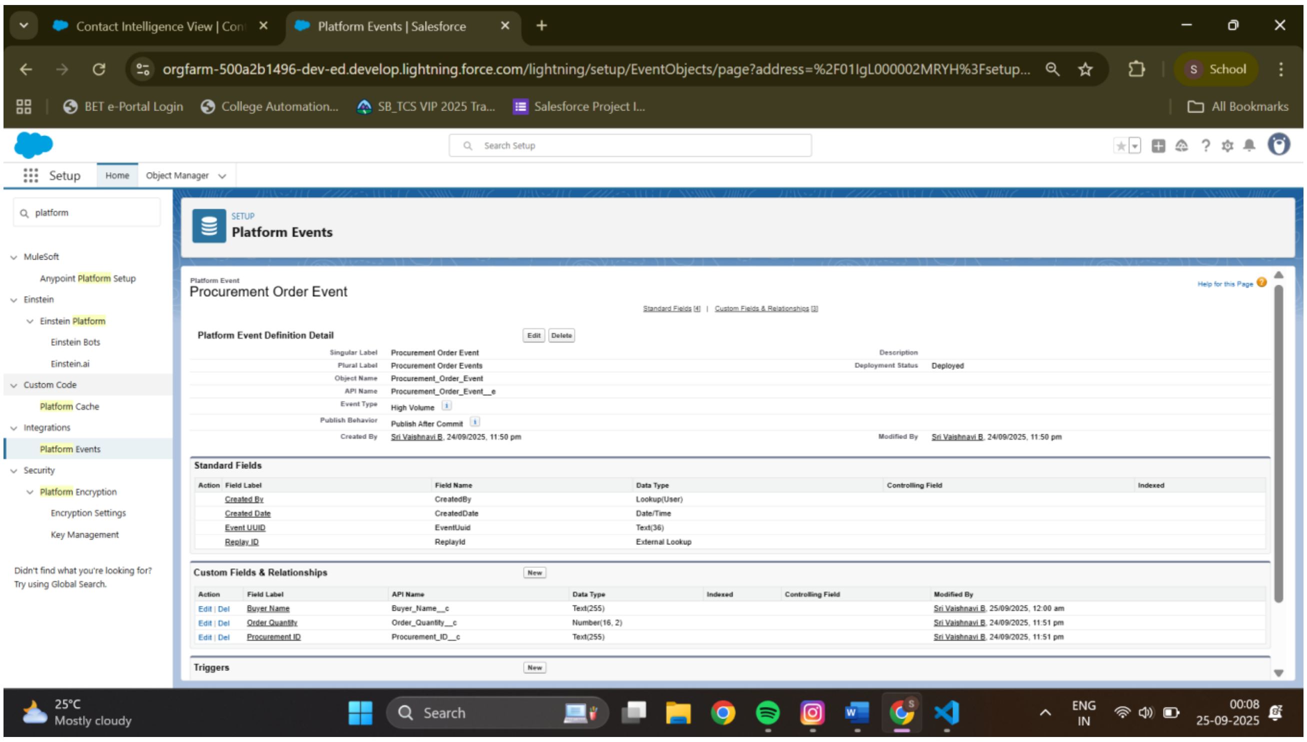
WeatherService.cls 9+, U

force-app > main > default > classes > WeatherService.cls > ...

```
public with sharing class WeatherService {

    /**
    * @description This asynchronous method makes a callout to an external weather API.
    * @param cropCycleId The ID of the Crop Cycle record to associate the new Advisory with.
    * @param location A string representing the location for the weather forecast (e.g., a city name or postal code).
    */
    @future(callout=true)
    public static void getRainfallForecast(Id cropCycleId, String location) {
        // 1. Prepare the HTTP Request
        HttpRequest req = new HttpRequest();

        // Use the Named Credential 'WeatherAPI' that we created in the previous step.
        // This is secure and avoids hardcoding URLs.
        req.setEndpoint('callout:WeatherAPI/forecast/daily?location=' + location + '&units=metric');    Static string found: 'callout:Weath
        req.setMethod('GET');    Static string found: 'GET'

        // 2. Send the Request
        Http http = new Http();
        HttpResponse res = http.send(req);

        // 3. Process the Response
        if (res.getStatusCode() == 200) {
            // Deserialize the JSON response into a map of objects.
            Map<String, Object> responseMap = (Map<String, Object>) JSON.deserializeUntyped(
            Map<String, Object> forecastMap = (Map<String, Object>) responseMap.get('forecas
            Decimal rainfall = (Decimal) forecastMap.get('rainfall_mm');    Static string found: 'rainfall_mm'

            // 4. Take action based on the data
```

To use this extension, first install the `code-analyzer` Salesforce CLI plu...

An Error ocurred while updating Aura Helper SF. You can update manu...



orgfarm-500a2b1496-dev-ed.develop.lightning.force.com/lightning/setup/EventObjects/page?address=%2F01IgL000002MRYH%3Fsetup...

SETUP
Platform Events

Platform Event
Procurement Order Event

Standard Fields [4] | Custom Fields & Relationships [3]

Platform Event Definition Detail

Singular Label	Procurement Order Event	Description	
Plural Label	Procurement Order Events	Deployment Status	Deployed
Object Name	Procurement_Order_Event		
API Name	Procurement_Order_Event__e		
Event Type	High Volume		
Publish Behavior	Publish After Commit		
Created By	Sri Vaishnavi B, 24/09/2025, 11:50 pm	Modified By	Sri Vaishnavi B, 24/09/2025, 11:50 pm

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

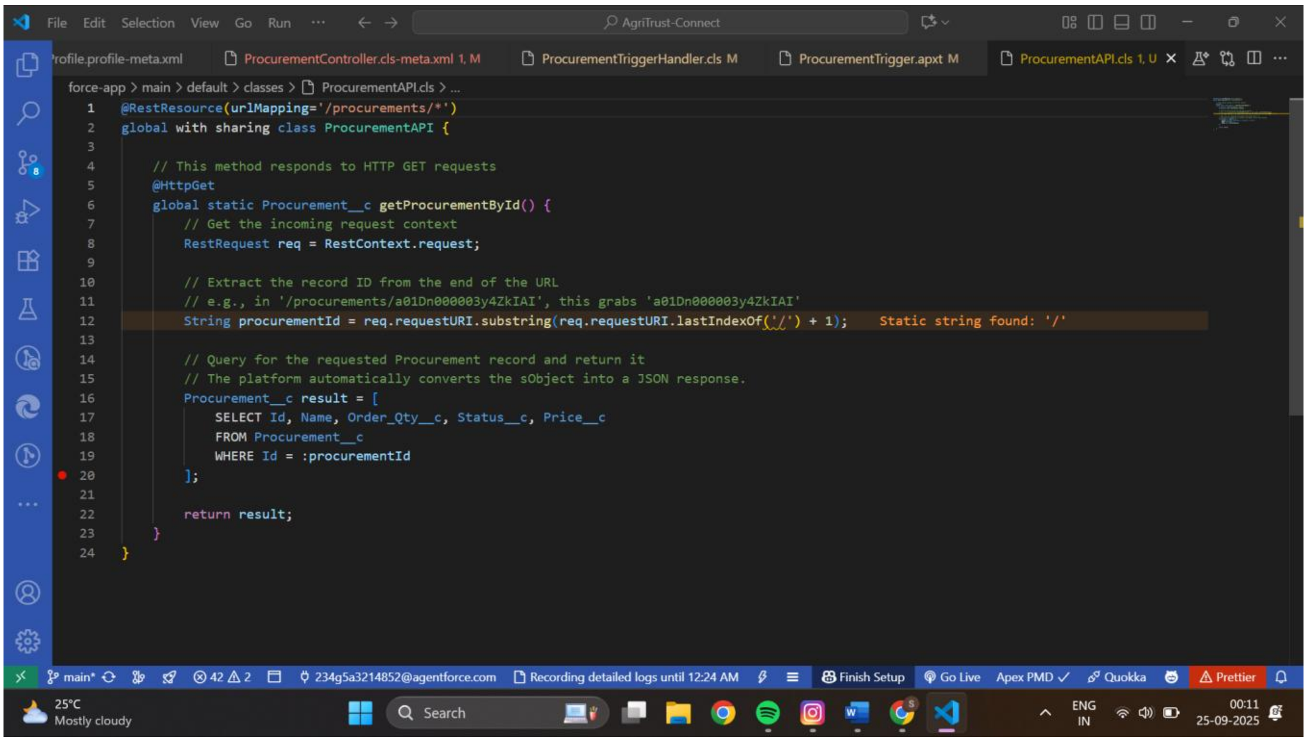
Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Edit | Del	Buyer Name	Buyer_Name__c	Text(255)			Sri Vaishnavi B, 25/09/2025, 12:00 am
Edit | Del	Order Quantity	Order_Quantity__c	Number(16, 2)			Sri Vaishnavi B, 24/09/2025, 11:51 pm
Edit | Del	Procurement ID	Procurement_ID__c	Text(255)			Sri Vaishnavi B, 24/09/2025, 11:51 pm

Triggers

3. Inbound Integrations (External → Salesforce)

Purpose: To allow external applications to securely access and retrieve data from Salesforce.

Technology Used	Implementation Details
Apex REST Service	An Apex class, ProcurementAPI.cls, was created with the @RestResource annotation. This exposes a custom endpoint (/services/apexrest/procurements/*) that allows authenticated external systems to make an HTTP GET request with a record ID to fetch specific Procurement__c records in JSON format.


File Edit Selection View Go Run
AgriTrust-Connect
rofile.profile-meta.xml
ProcurementController.cls-meta.xml 1, M
ProcurementTriggerHandler.cls M
ProcurementTrigger.apxt M
ProcurementAPI.cls 1, U
force-app > main > default > classes > ProcurementAPI.cls > ...

```
@RestResource(urlMapping='/procurements/*')
global with sharing class ProcurementAPI {

    // This method responds to HTTP GET requests
    @HttpGet
    global static Procurement__c getProcurementById() {
        // Get the incoming request context
        RestRequest req = RestContext.request;

        // Extract the record ID from the end of the URL
        // e.g., in '/procurements/a01Dn000003y4ZkIAI', this grabs 'a01Dn000003y4ZkIAI'
        String procurementId = req.requestURI.substring(req.requestURI.lastIndexOf('/') + 1);    Static string found: '/'

        // Query for the requested Procurement record and return it
        // The platform automatically converts the sObject into a JSON response.
        Procurement__c result = [
            SELECT Id, Name, Order_Qty__c, Status__c, Price__c
            FROM Procurement__c
            WHERE Id = :procurementId
        ];

        return result;
    }
}
```

4. Security & Governance

Purpose: To ensure all integrations are secure, authenticated, and comply with platform limits.

Component	Description
Authentication	The **OAuth 2.0** protocol was established as the standard for server-to-server authentication. **Named Credentials** were used to manage the storage and application of these authentication details for all callouts.
Authorization	**Remote Site Settings** were configured to explicitly whitelist the domains of external APIs, providing a foundational layer of security for outbound callouts.
API Limits	The implementation considered Salesforce governor limits, particularly the daily API call limits. All callouts and inbound requests were designed to be efficient and scalable, with monitoring mechanisms planned to prevent service disruptions.

5. Alternative Patterns Explored

In addition to the implemented patterns, the following technologies were evaluated for future use:

- **Change Data Capture (CDC):** Assessed as a high-volume alternative to Platform Events for broadcasting a stream of record changes to external systems.
- **Salesforce Connect:** Researched as a solution for data virtualization, with a potential use case to access a government subsidy database as an External Object without replicating the data inside Salesforce.

6. Phase 7 Outcome

- Successfully implemented secure outbound and inbound API integrations.
- Automated the creation of advisory records based on real-time external data.
- Enabled real-time synchronization with external systems via Platform Events.
- Established a robust security framework for all data exchange using Named Credentials and OAuth.
- AgriTrust Connect is now a fully integrated application, capable of acting as both a consumer and a provider of data in a wider technology landscape.