

Project Title: Pet Adoption Tracker

Phase 7: Integration & External Access

Objective: The goal of this phase was to extend the functionality of the Pet Adoption Tracker by integrating it with external systems (Microchip Registry, Communication Services) and implementing real-time event-driven architecture within Salesforce. This ensures data security for external calls, real-time updates for critical adoption changes, and adherence to platform best practices.

1. **Named Credential:** Created a Named Credential named **MicrochipRegistry**.

- **Configuration:**

- **URL:** Set to the base URL of the external service (<https://api.externalpetservice.com/v1/>).
- **Identity Type:** Named Principal (for demonstration simplicity).
- **Access:** Apex callouts use the format `callout:MicrochipRegistry` to securely proxy the request without exposing the full endpoint URL or API keys in the code.
- .

The screenshot shows the Salesforce 'Named Credentials' configuration page. The breadcrumb trail is 'SETUP > NAMED CREDENTIALS'. The page title is 'MicrochipRegistry'. There are 'Edit' and 'Delete' buttons in the top right corner. The configuration details are as follows:

Label	Name
MicrochipRegistry	MicrochipRegistry

URL: <https://api.externalpetservice.com/v1/microchips/>

Enabled for Callouts: ☒

Authentication

External Credential: [ExternalPetAPI](#)

Client Certificate:

Callout Options

Generate Authorization Header: ☒

Allow Formulas in HTTP Header: ☒

Allow Formulas in HTTP Body: ☒

Outbound Network Connection: ☒

2. External Services & Callouts

Purpose: Call external REST/SOAP services.

Use Case in Your Project:

- Retrieve pet data from an external database.

- **Send adoption request data to a partner organization.**

ExternalPetAPI

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

Label

ExternalPetAPI

Name

PawTrack

Authentication Protocol

OAuth 2.0

Authentication Flow Type

Browser Flow

Scope

Identity Provider

Authentication Provider

PetAPIAuthProvider

Callout Options

Additional Status Codes for Token Refresh

Managed Package Access

Created By Namespace

Remote Site Settings (For Legacy/General Compliance)

Although Named Credentials are the preferred method, a Remote Site Setting was configured to ensure compatibility and document the requirement for allowing server-side communication to an external domain.

- **Configured:** Remote Site Setting MicrochipLookupSite was created.
- **URL:** Configured with the root domain of the external Microchip API.
- **Purpose:** Ensures Salesforce can send callouts to this trusted endpoint without security restrictions.

Remote Site Settings

[Help for this Page](#)

All Remote Sites

Below is the list of Web addresses that your organization can invoke from salesforce.com. To add another Web address, click New Remote Site.

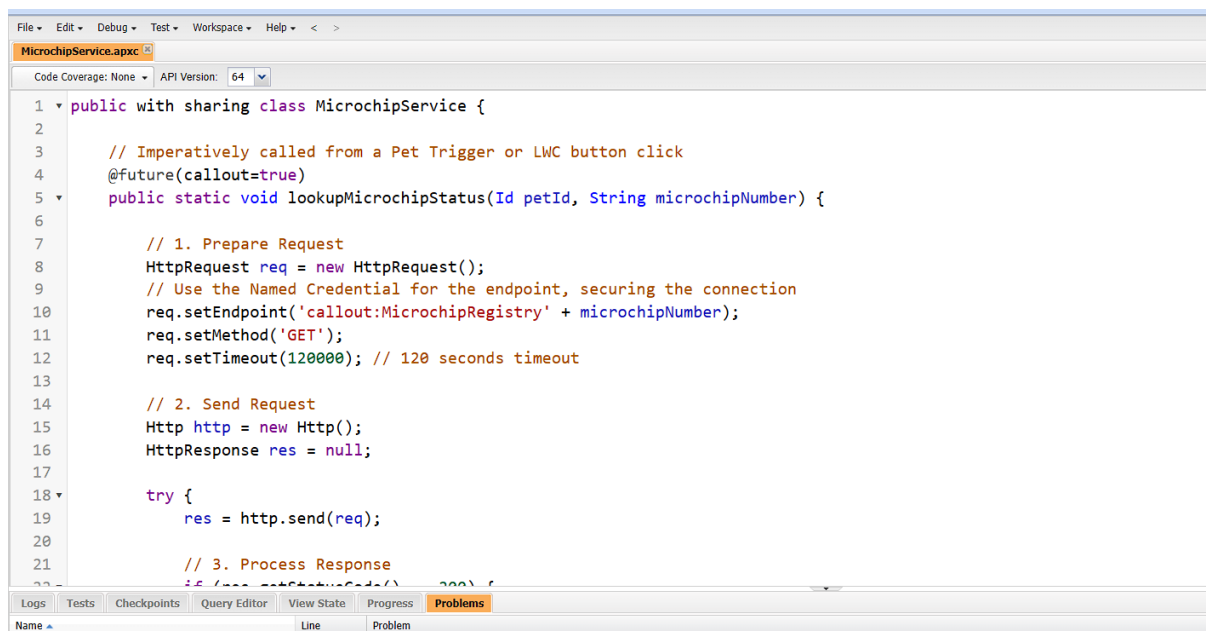
View: [All Remote Sites](#) [Create New View](#)

Action	Remote Site Name ↑	Namespace Prefix	Remote Site URL	Active	Created By	Created Date	Last Modified By	Last Modified Date
Edit Del	ApexDevNet	-	http://www.apexdevnet.com	✓	Shelter Admin	7/17/2025, 12:13 PM	Shelter Admin	7/17/2025, 12:13 PM
Edit Del	MicrochipLookupSite	-	https://api.externalpetservice.com	✓	HARSHA VARDHAN_SURLA	9/25/2025, 11:19 AM	HARSHA VARDHAN_SURLA	9/25/2025, 11:19 AM

3. Apex Callouts (Microchip Verification)

An asynchronous Apex class was created to perform the external lookup, ensuring the operation does not consume the user's transaction time and adheres to callout limits.

- **Apex Class: MicrochipService.cls**
- **Method: lookupMicrochipStatus(Id petId, String microchipNumber)**
- **Mechanism:** Uses the `@future(callout=true)` annotation to execute the REST callout asynchronously.
- **Integration:** Performs an HTTP GET callout using the `callout:MicrochipRegistry` endpoint and updates the `Pet__c.External_Registration_Status__c` field based on the API response (e.g., 'Verified' or 'Unverified').



```
1 public with sharing class MicrochipService {
2
3     // Imperatively called from a Pet Trigger or LWC button click
4     @future(callout=true)
5     public static void lookupMicrochipStatus(Id petId, String microchipNumber) {
6
7         // 1. Prepare Request
8         HttpRequest req = new HttpRequest();
9         // Use the Named Credential for the endpoint, securing the connection
10        req.setEndpoint('callout:MicrochipRegistry' + microchipNumber);
11        req.setMethod('GET');
12        req.setTimeout(120000); // 120 seconds timeout
13
14        // 2. Send Request
15        Http http = new Http();
16        HttpResponse res = null;
17
18        try {
19            res = http.send(req);
20
21            // 3. Process Response
22            if (res.getStatusCode() == 200) {
23                // Parse JSON response
24                String response = res.getBody();
25                // ... (code continues) ...
26            }
27        } catch (Exception e) {
28            // Handle exception
29        }
30    }
31 }
```

4. Platform Events (Real-Time Adoption Status)

Platform Events were implemented to create a decoupled, real-time mechanism for notifying internal systems (e.g., Dashboards, Staff) immediately upon a critical status change.

- **Platform Event:** Created a custom Platform Event named **AdoptionApprovedEvent__e**.
- **Fields:** Includes key data points like `Pet_Name__c` and `Adopter_Email__c`.
- **Publishing:** The existing **AdoptionRequestTriggerHandler.cls** was modified to publish this event upon a status change from 'Pending' to 'Approved'. This ensures event publication happens only after the database transaction is successfully committed (**Publish After Commit**).

Platform Events

Object Name: AdoptionApprovedEvent
 API Name: AdoptionApprovedEvent_e
 Event Type: High Volume
 Publish Behavior: Publish After Commit
 Created By: SURLA HARSHA VARDHAN 9/25/2025, 11:33 AM
 Modified By: SURLA HARSHA VARDHAN 9/25/2025, 11:33 AM

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

Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Edt Del	Adopter_Email__c	Adopter_Email__c	Text(255)			SURLA HARSHA VARDHAN 9/25/2025, 11:50 AM
Edt Del	Adoption_Request_ID__c	Adoption_Request_ID__c	Text(10)			SURLA HARSHA VARDHAN 9/25/2025, 11:50 AM
Edt Del	Pet_Name__c	Pet_Name__c	Text(255)			SURLA HARSHA VARDHAN 9/25/2025, 11:50 AM

Triggers
No triggers defined

Subscriptions
Subscriber: Last Processed Id, Last Published Id, State, Batch Size, User

```

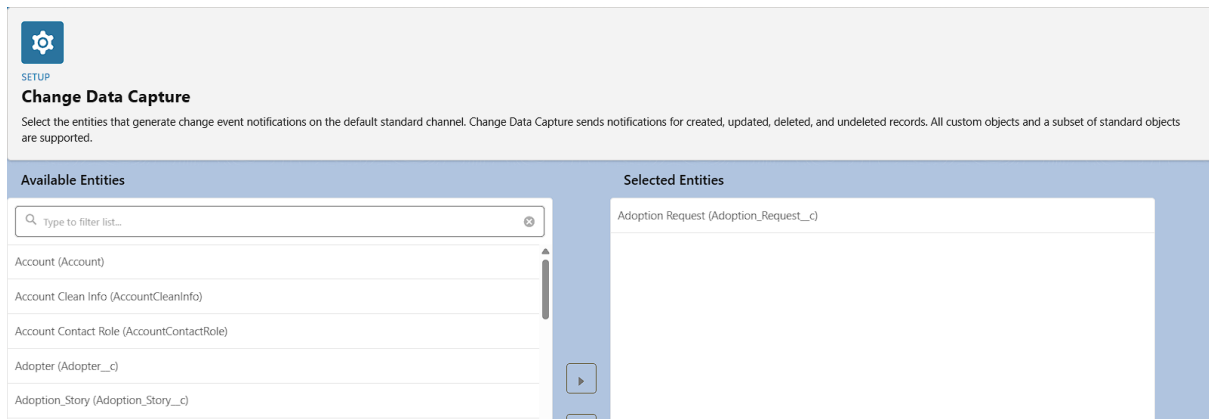
1 public class AdoptionRequestTriggerHandler extends TriggerHandler {
2
3     // Before Insert
4     public override void beforeInsert(List<SObject> newList) {
5         List<Adoption_Request__c> newRequests = (List<Adoption_Request__c>) newList;
6         validateAdoptionRequests(newRequests);
7         setDefaultValues(newRequests);
8     }
9
10    // Before Update
11    public override void beforeUpdate(List<SObject> newList, List<SObject> oldList, Map<Id, SObject> newMap, Map<Id, SObject> oldMap) {
12        List<Adoption_Request__c> newRequests = (List<Adoption_Request__c>) newList;
13        Map<Id, Adoption_Request__c> oldRequests = (Map<Id, Adoption_Request__c>) oldMap;
14        validateStatusChanges(newRequests, oldRequests);
15    }
16
17    // After Insert
18    public override void afterInsert(List<SObject> newList, Map<Id, SObject> newMap) {
19        List<Adoption_Request__c> newRequests = (List<Adoption_Request__c>) newList;
20        handleNewAdoptionRequests(newRequests);
21    }
22

```

5. Change Data Capture (CDC)

CDC was enabled on key transactional objects to allow external systems to reliably subscribe to record changes, ensuring data synchronization outside of Salesforce.

- **Enabled CDC:** Activated for the **Adoption_Request__c** object.
- **Purpose:** This allows an external partner (e.g., a custom website or mobile app) to subscribe via a Salesforce Streaming API client (like CometD) and receive notifications whenever an adoption request record is created, updated, or deleted, ensuring the external view of adoption status is always current.



6. API Limits & Asynchronous Processing

Architectural decisions were made to manage Salesforce Governor Limits and API Callout Limits.

- **Asynchronous Processing:** All external callouts are handled via **@future(callout=true)** methods, ensuring they run in separate transactions and do not block the user interface.
- **Limit Management:** The use of Platform Events further ensures that high-volume internal notifications (like updating dashboards) are decoupled from DML transactions, preserving system stability and performance.