Phase 7 – Integration & External Access

Project: Event Management Portal

This phase focuses on enabling external integrations and API access for the Event Management Portal. Only the features implemented for this project are documented here.

---

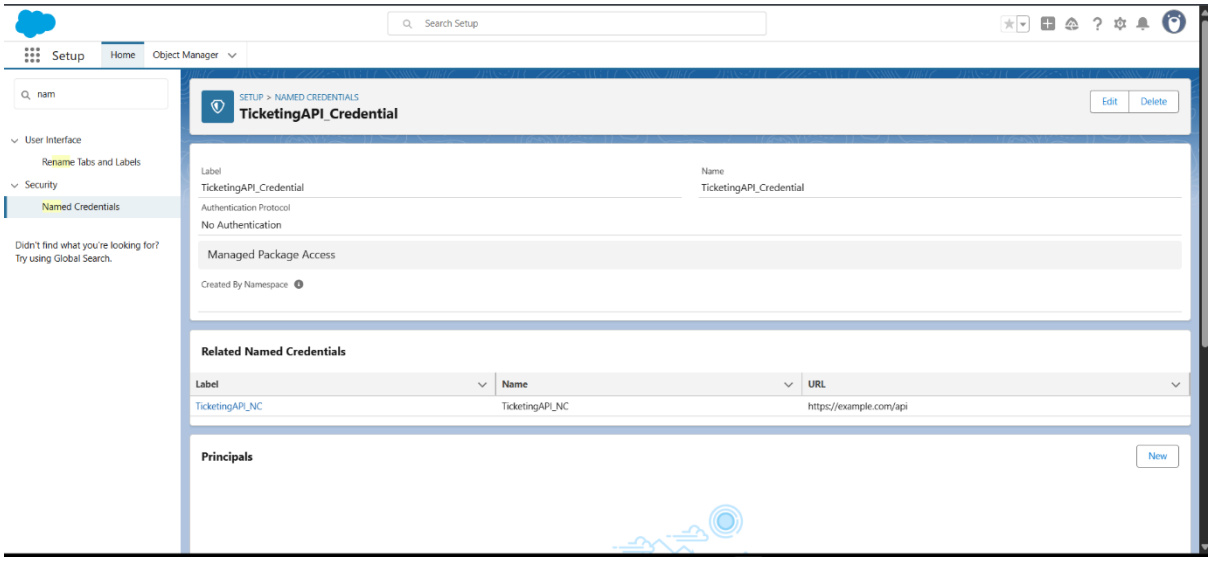
1. External Credential

Purpose:
Defines authentication settings for external API callouts.

Steps Performed:

1. Navigated to **Setup → External Credentials → New**.
2. Entered the following:
    - **Label:** TicketingAPI_Credential
    - **Name:** TicketingAPI_Credential (auto-filled)
    - **Authentication Protocol:** No Authentication
3. Saved the credential.

Screenshot :



---

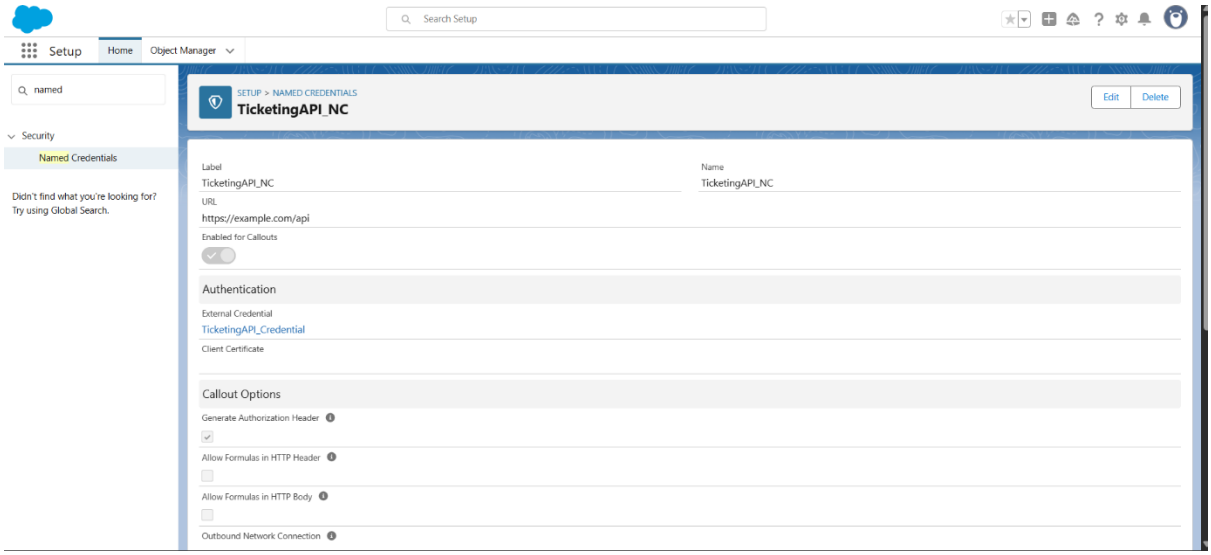
2. Named Credential

Purpose:
Provides the endpoint URL and links to the External Credential for making API callouts from Salesforce.

Steps Performed:

1. Navigated to **Setup → Named Credentials → New**.
2. Entered the following:
    - **Label:** TicketingAPI_NC
    - **Name:** TicketingAPI_NC
    - **URL:** https://example.com/api
    - **External Credential:** TicketingAPI_Credential
3. Saved the Named Credential.

Screenshot :



---

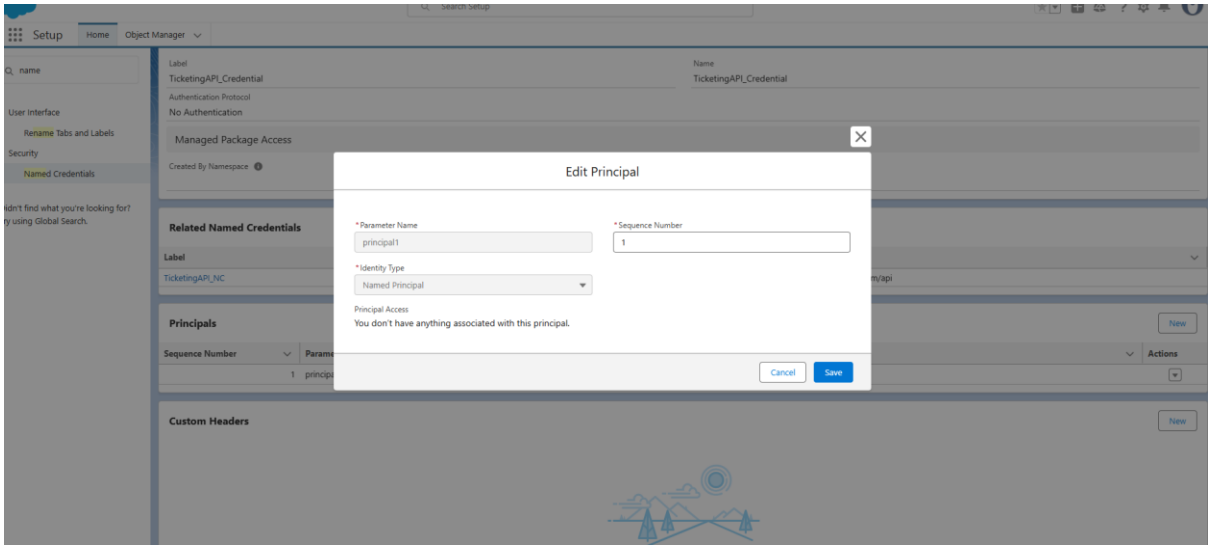
3. Principal Setup

Purpose:
Configures access for the Named Credential using the External Credential.

Steps Performed:

1. In the External Credential setup, clicked **Create Principal**.
2. Entered the following:
    - **Parameter Name:** principal1
    - **Sequence Number:** 1
    - **Identity Type:** Named Principal
3. Saved the Principal configuration.

Screenshot :



---

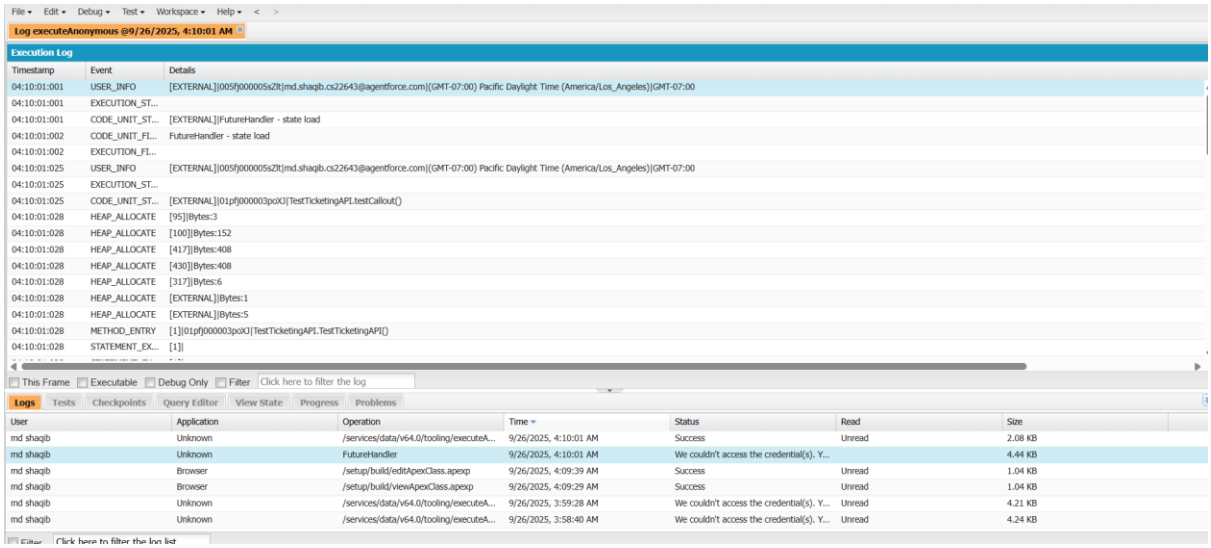
4. Callout Test

Purpose:
To ensure the Named Credential and External Credential work together and allow external API communication.

Steps Performed:

1. Created a test Apex class to call the external API using the Named Credential.
2. Verified the callout executed successfully in **Developer Console → Execute Anonymous** or **Debug Logs**.

Screenshot :



---

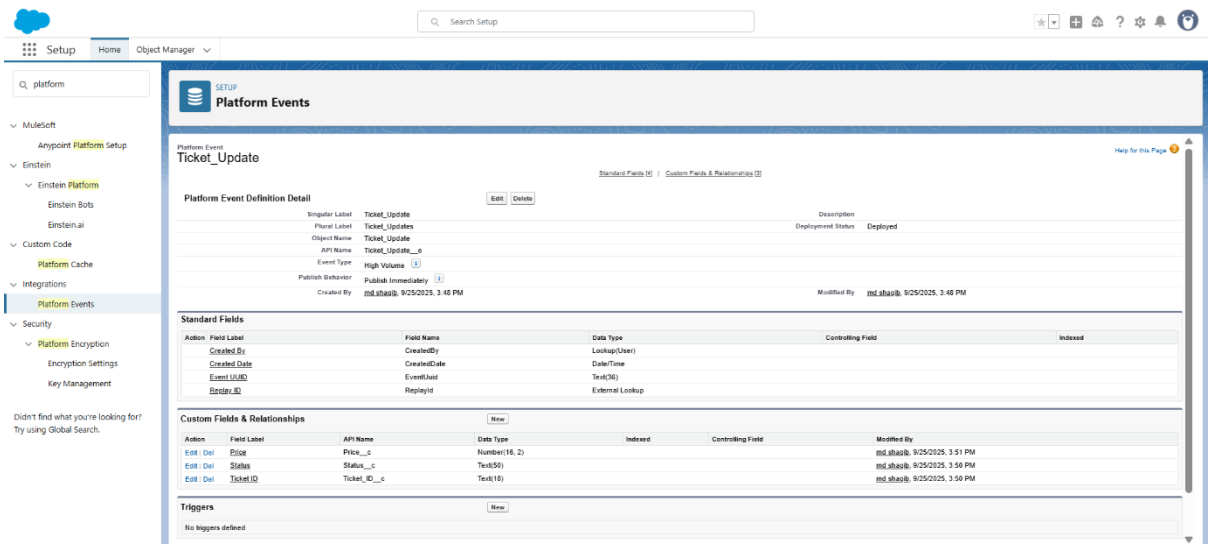
5. Platform Event

Purpose:
To notify external systems or other Salesforce processes when a Ticket record is updated.

Steps Performed:

1. Navigated to **Setup → Platform Events → New Platform Event**.
2. Created the event with the following properties:
    - **Label:** Ticket_Updates
    - **Plural Label:** Ticket_Updates
    - **Object Name:** Ticket_Updates
    - **Publish Behavior:** Publish Immediately
3. Added custom fields to track ticket details:
    - **Ticket ID:** Text(18)
    - **Status:** Text(50)
    - **Price:** Number(16,2)

Screenshot :



---

6. Apex Test – Publish Platform Event

Purpose:
Tested the publishing of Platform Events using Apex to ensure proper event delivery.

Steps Performed:

1. Created a test Apex class TestTicketEvent with a method publishTicketUpdate() to publish sample events.

2. Executed the method and verified events were published successfully in **Developer Console → Logs**.

Screenshot :