

## Phase 7: Integration & External Access

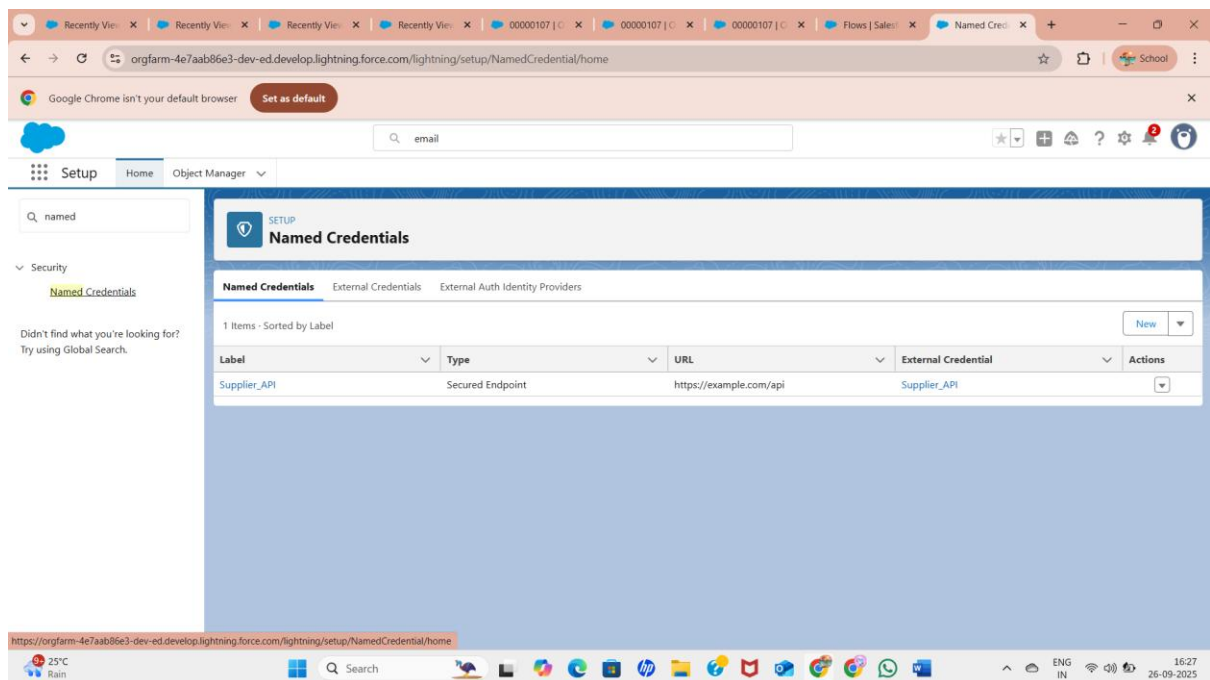
This phase covers how Salesforce interacts with external systems, APIs, and manages secure access.

### 1. Named Credentials

**Purpose:** Simplifies authentication when connecting to external systems.

**Steps to Create:**

1. Go to **Setup** → **Named Credentials** → **New Named Credential**.
2. Fill in:
  - **Label:** e.g., ExternalAPI\_Credential
  - **Name (API Name):** ExternalAPI\_Credential
  - **URL:** Base URL of external system, e.g., https://api.example.com
  - **Authentication:** Choose OAuth 2.0, Password, or External Credential
  - **Callout Options:** Enable “Generate Authorization Header” if required



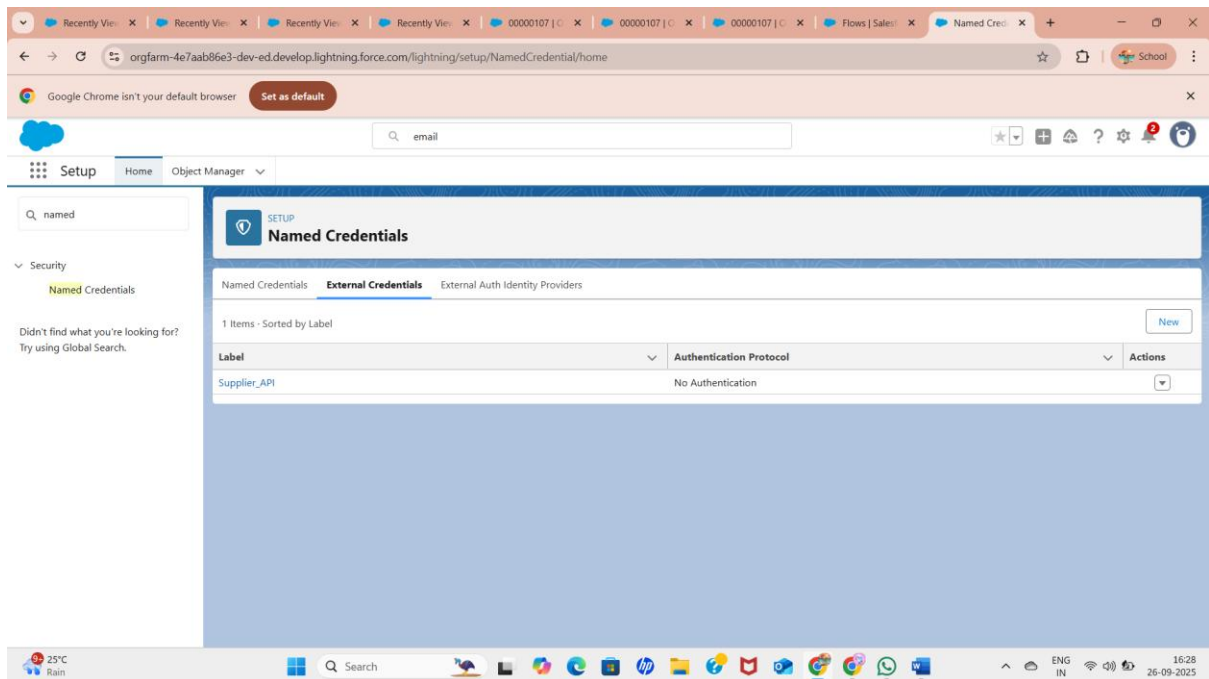
### 2. External Services

**Purpose:** Connect Salesforce to external APIs declaratively.

**Steps:**

- **Name:** InventoryAPI
- **Schema URL or Named Credential:** Connect via previously created Named Credential

2. Salesforce automatically generates Apex actions for flow use.

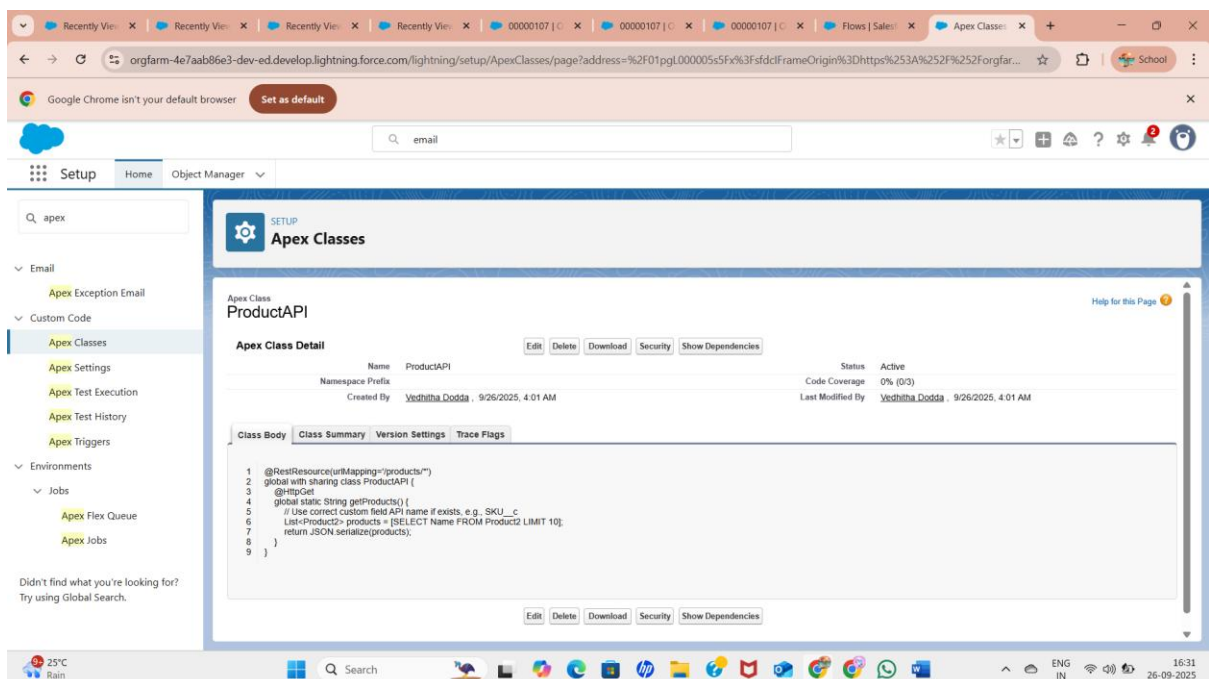


### 3. Web Services (REST/SOAP)

**Purpose:** Salesforce can both call external APIs and expose its own data via REST/SOAP.

### Key Points:

- **REST:** Lightweight, JSON-based, used for modern integrations
- **SOAP:** XML-based, often for enterprise systems
- **Setup:** Setup → Apex Classes → `@RestResource` for REST endpoints



## 4. Callouts

**Purpose:** Invoke external services from Apex or Flow.

**Steps:**

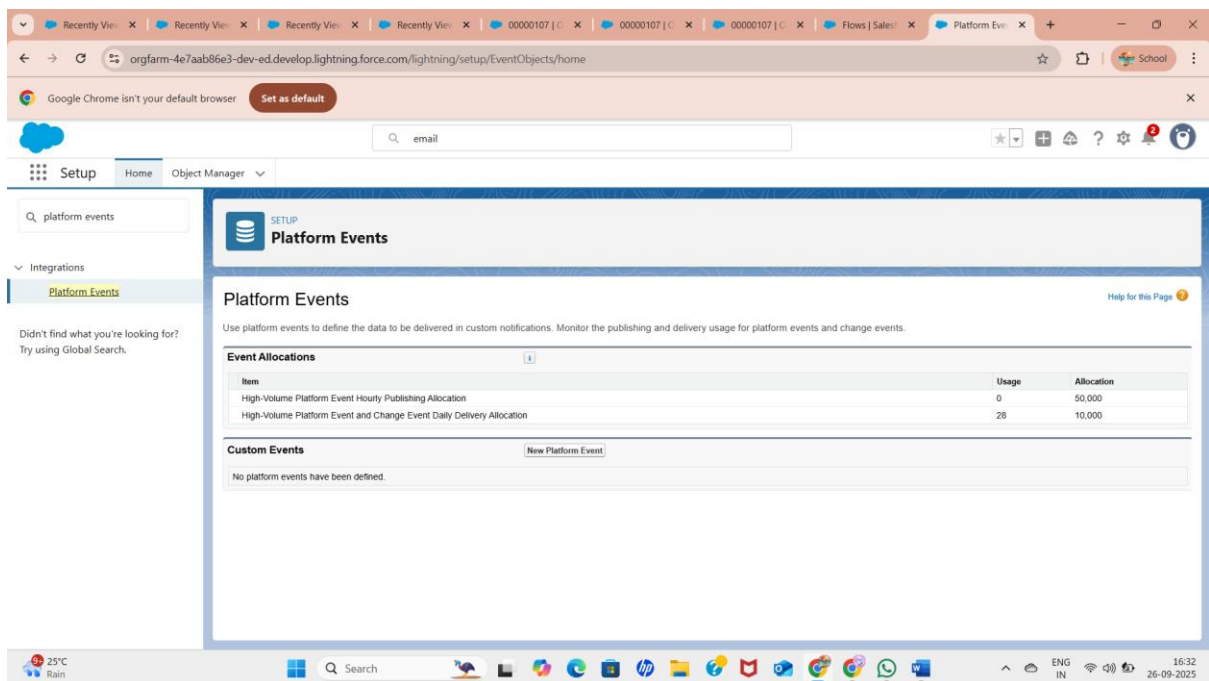
1. Create **Named Credential**
2. Use Apex **HTTP Callout** class or **Flow HTTP Request** action
3. Example in Apex:

```
HttpRequest req = new HttpRequest();  
req.setEndpoint('callout:ExternalAPI_Credential/products');  
req.setMethod('GET');  
Http http = new Http();  
HttpResponse res = http.send(req);  
System.debug(res.getBody());
```

## 5. Platform Events

**Purpose:** Real-time messaging between Salesforce and external apps.

- Define in **Setup** → **Platform Events** → **New Platform Event**
- Fields can include ProductId, Quantity\_\_c, EventType\_\_c
- Subscribers (internal/external) listen for these events



The screenshot shows the Salesforce Setup page for Platform Events. The browser address bar indicates the URL is `orgfarm-4e7aab863-dev-ed.develop.lightning.force.com/lightning/setup/EventObjects/home`. The page title is "Platform Events". Below the title, there is a description: "Use platform events to define the data to be delivered in custom notifications. Monitor the publishing and delivery usage for platform events and change events." The page is divided into two main sections: "Event Allocations" and "Custom Events".

Item	Usage	Allocation
High-Volume Platform Event Hourly Publishing Allocation	0	50,000
High-Volume Platform Event and Change Event Daily Delivery Allocation	28	10,000

Below the "Custom Events" section, there is a button labeled "New Platform Event" and a message stating "No platform events have been defined."

## 6. Change Data Capture (CDC)

**Purpose:** Track changes in Salesforce records in real-time.

- Go to **Setup** → **Change Data Capture** → **Select Objects**
- External systems or Flows can subscribe to updates for syncing

## 7. Salesforce Connect

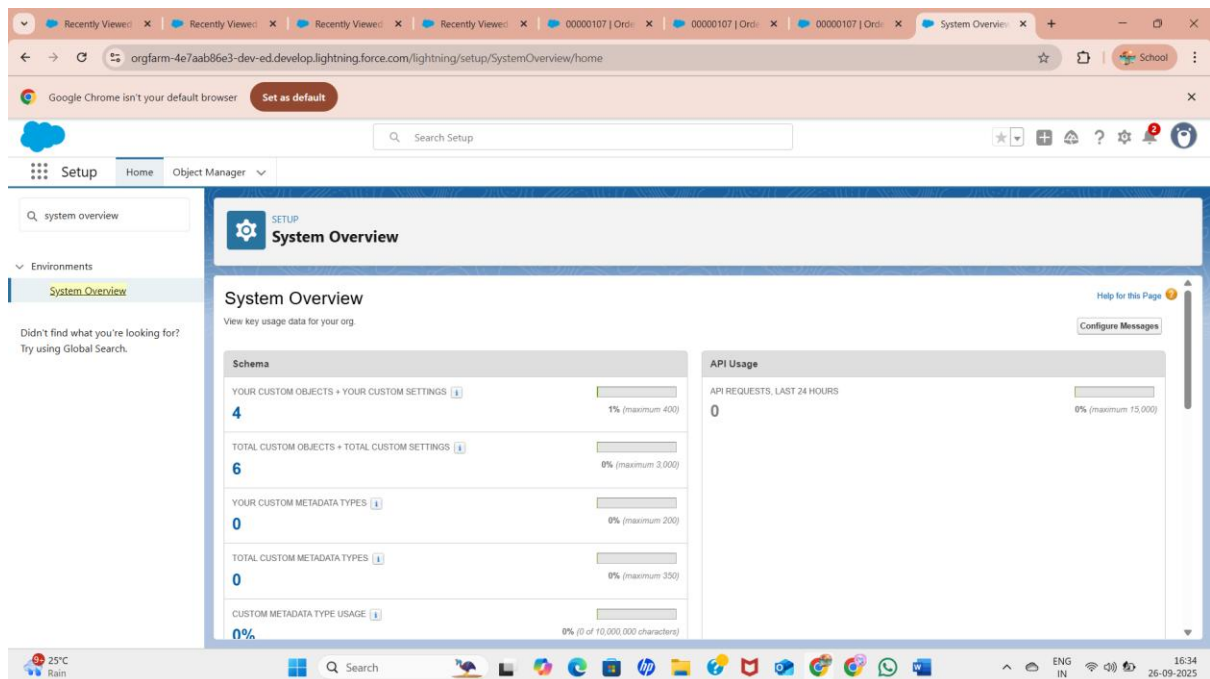
**Purpose:** Access external objects in real-time without storing data in Salesforce.

- Go to **Setup** → **External Data Sources** → **New External Data Source**
- Define OData endpoint, authentication, and create **External Objects**

## 8. API Limits

**Purpose:** Salesforce limits API calls per 24 hours to prevent overload.

- Monitor under **Setup** → **System Overview** → **API Usage**
- Optimize integrations to avoid hitting limits



## 9. OAuth & Authentication

**Purpose:** Securely connect Salesforce with external apps.

- Configure **Connected Apps** in **Setup** → **App Manager** → **New Connected App**
- Choose OAuth scopes, callback URL, and policies

## 10. Remote Site Settings

**Purpose:** Whitelist external URLs for Apex callouts.

- Go to **Setup** → **Remote Site Settings** → **New Remote Site**
- Enter **Remote Site Name** and **URL**
- Required for all Apex HTTP callouts