Phase 7: Integration & External Access

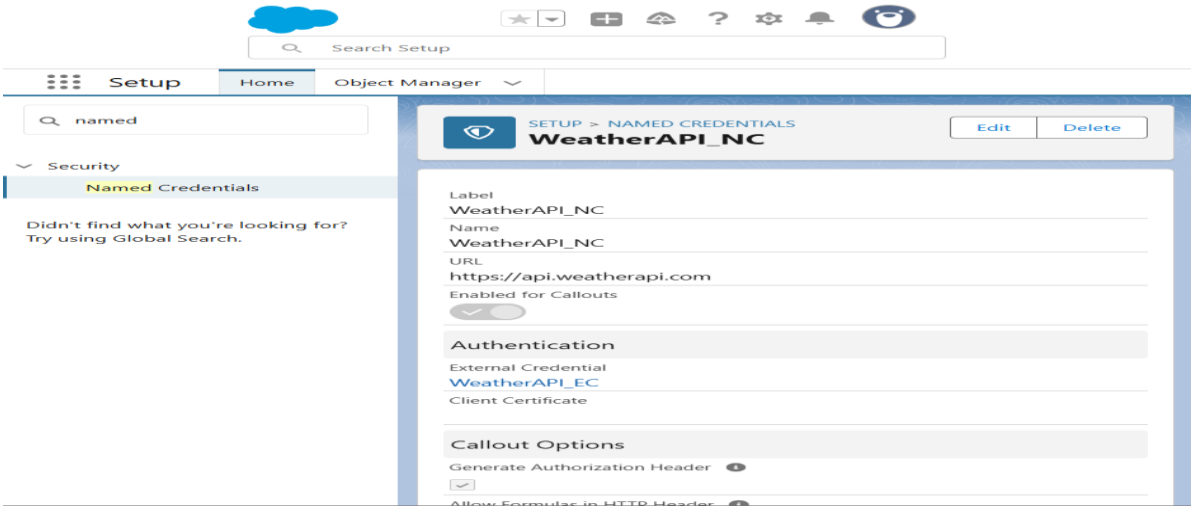
1. Named Credentials

Purpose

- Securely store authentication information for external APIs.
- Simplifies API callouts by avoiding hardcoding credentials in Apex code.
- Manages authentication protocols such as OAuth 2.0, Password Authentication, or Named Principal.
- Enhances security by controlling access centrally.

Key Components

- **Label & Name:** User-friendly label and unique API name for reference in Apex.
- **URL:** Base URL of the external service.
- **Identity Type:** Defines who is making the callout:
  - `Named Principal` – All users share the same authentication.
  - `Per User` – Each Salesforce user uses their own credentials.
- **Authentication Protocol:** Type of authentication used for the API:
  - `Password Authentication`
  - `OAuth 2.0`
  - `AWS Signature`
  - `JWT Bearer Token`
- **External Credential (Optional):** Can link to external credentials stored securely.
- **Callout Enabled:** Must be checked to allow callouts using this credential.
- **Usage in Apex**: HttpRequest req = new HttpRequest();
  req.setEndpoint('callout:WeatherAPI_NC/v1/current.json?q=London');
  req.setMethod('GET'); HttpResponse res = new Http().send(req);
  System.debug(res.getBody());



2. External Services

Purpose

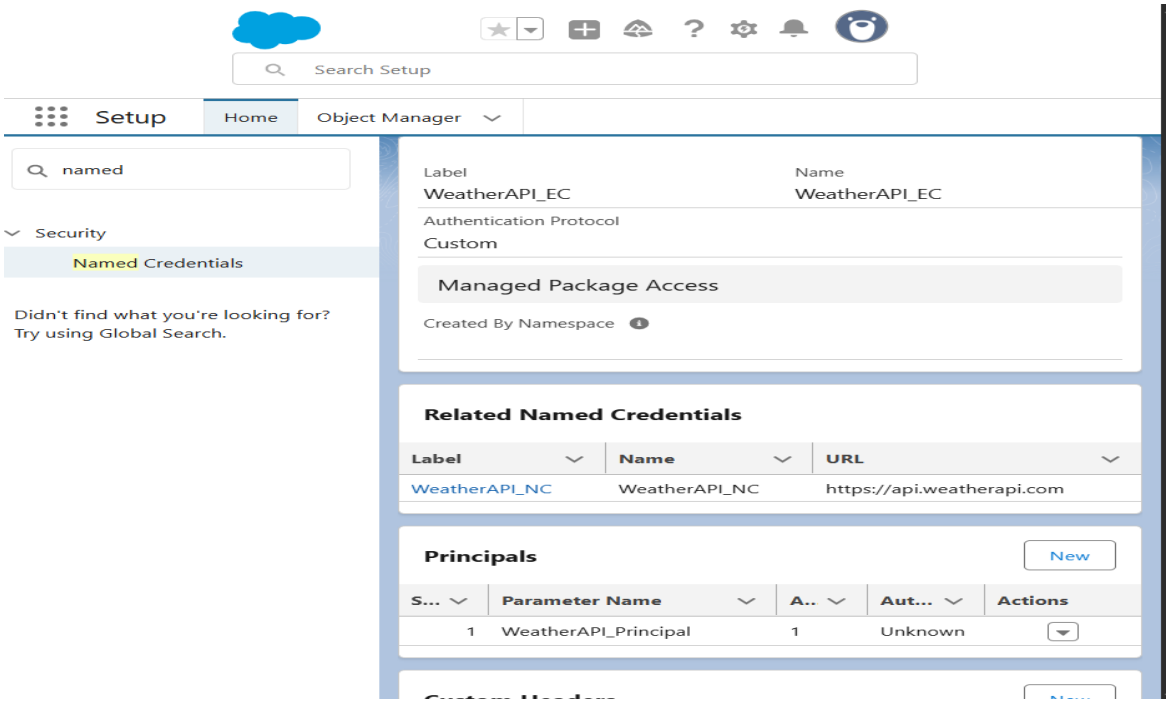
- Connect Salesforce to external APIs **declaratively**.
- Automatically generate **actions for Flows**.
- Simplifies integration for admins and non-developers.
- Supports both **REST and SOAP APIs**.

Key Features

- Declarative API integration with no Apex required.
- Uses **Swagger/OpenAPI schema** to define API structure.
- Actions are **automatically created** in Salesforce.
- Works with **Named Credentials** for secure authentication.
- Can be used in **Screen Flows, Scheduled Flows, and Record-Triggered Flows**.
- Supports reusability and governance of API calls.

Steps:

- Setup → External Services → New External Service
- Provide Name, Description, Service Schema (Swagger/OpenAPI URL)
- Select Named Credential
- Salesforce auto-generates actions usable in Flow



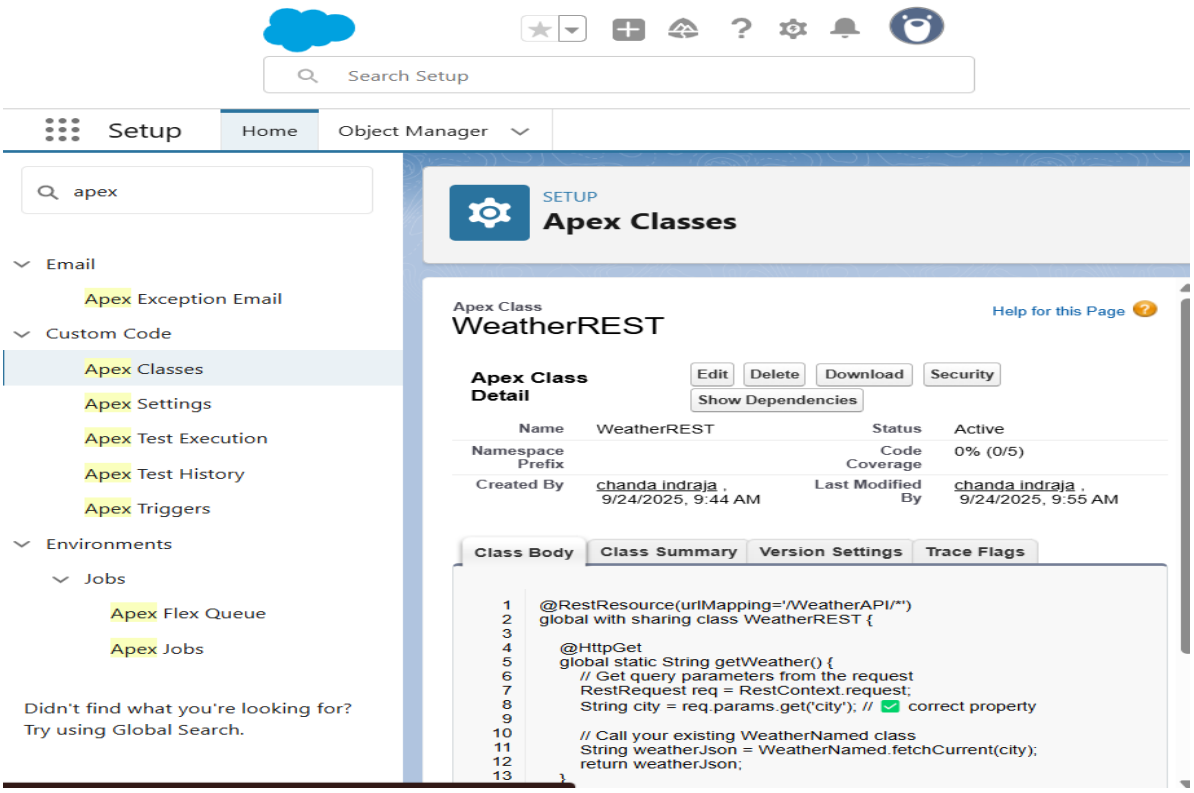
3. Web Services (REST/SOAP)

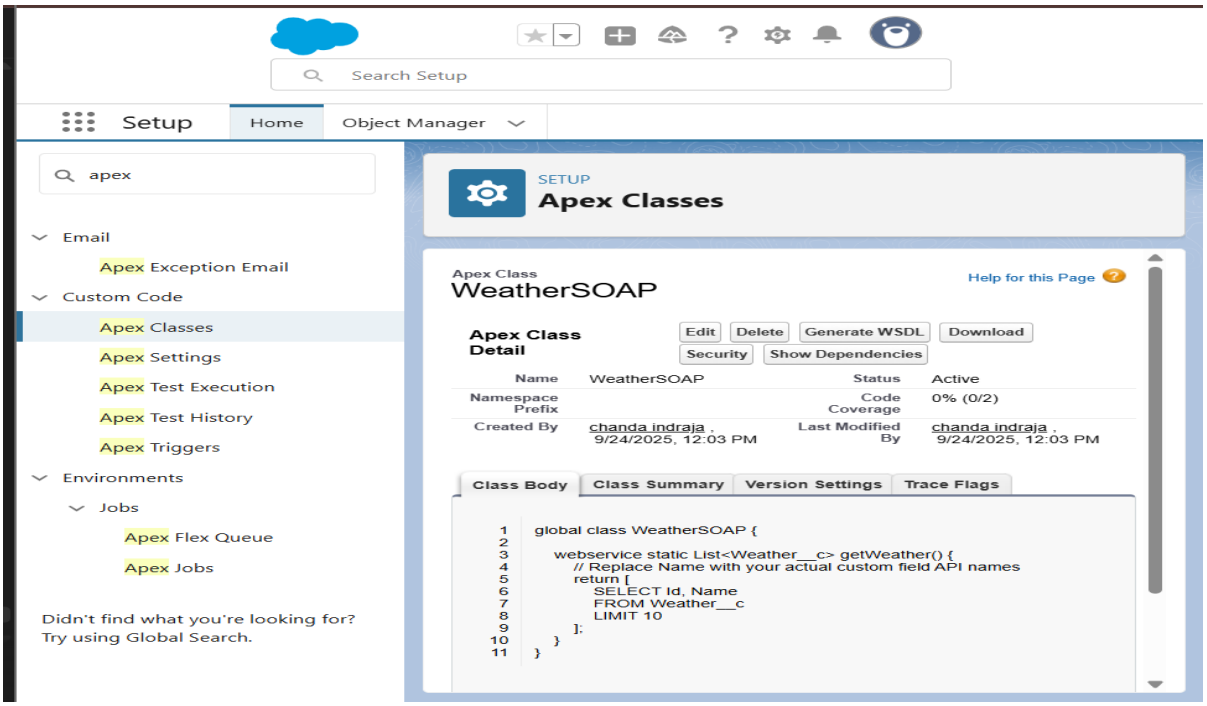
Purpose

- Expose Salesforce data to external systems.
- Consume external web services within Salesforce.
- Enable integration with external applications using standard protocols.
- Supports both **REST (lightweight)** and **SOAP (structured)** protocols.

Key Features

- **REST Services**
  - Lightweight, stateless, uses HTTP methods (GET, POST, PUT, DELETE).
  - Supports JSON and XML formats.
  - Easier to use for modern web and mobile apps.
- **SOAP Services**
  - Standardized protocol using XML.
  - Strongly typed operations, supports WSDL.
  - Preferred for enterprise-level integrations.





4. Callouts

Purpose

- Enable Salesforce to **communicate with external APIs** or web services.
- Retrieve or send data from/to external systems.
- Used in integrations, third-party services, or external data synchronization.

Key Features

- Supports **HTTP methods**: GET, POST, PUT, DELETE.
- Can use **Named Credentials** for secure authentication.
- Can also use **Remote Site Settings** for older setups.
- Supports **synchronous** (immediate response) and **asynchronous** (future methods, batch Apex) callouts.
- Can handle **REST and SOAP APIs**.
- Use **Named Credentials** instead of Remote Site Settings for security.
- Always handle exceptions with `try-catch`.

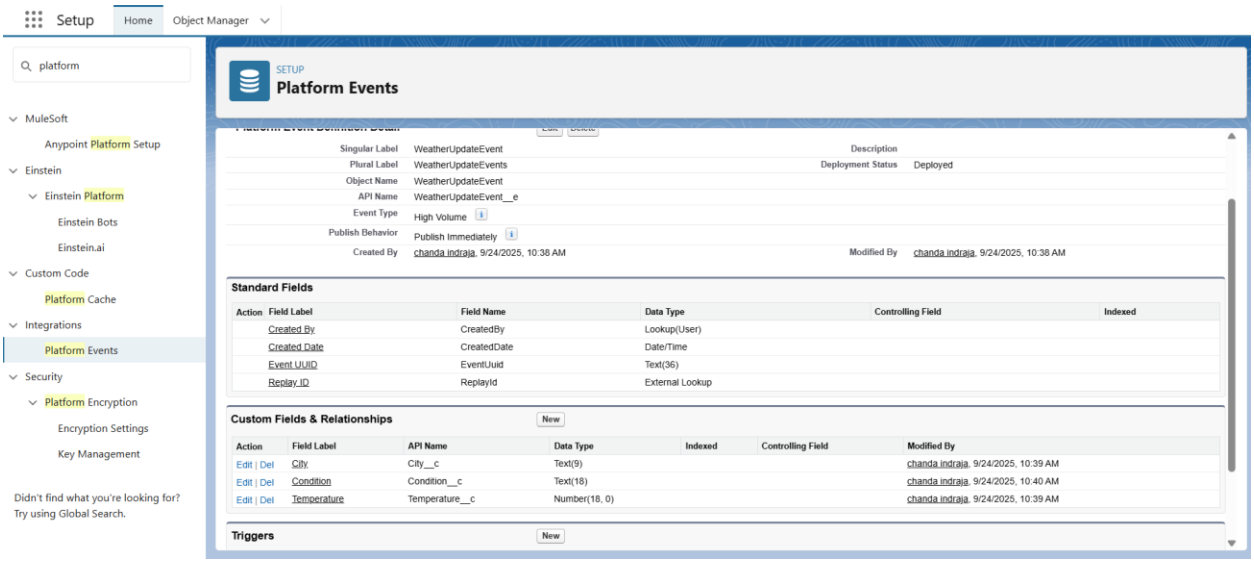
5. Platform Events in Salesforce

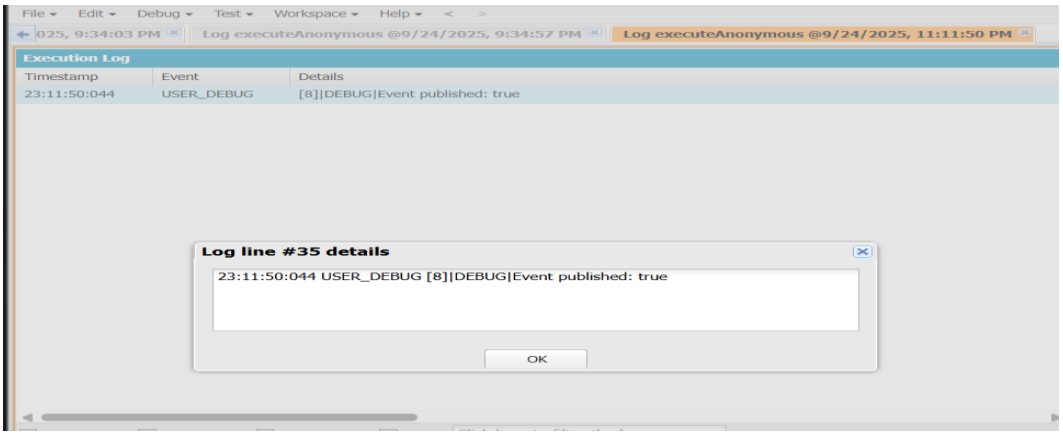
Purpose

- Enable **event-driven architecture** inside Salesforce.
- Send and receive custom notifications between Salesforce processes or external systems.
- Useful for real-time updates and decoupled integrations.

Key Features

- **Custom Event Objects** with fields like any other object.
- Events are **published** and **subscribed** asynchronously.
- Supports **Publish-Subscribe model**:
  - **Publisher**: Creates and publishes events.
  - **Subscriber**: Triggers, Flows, or external systems that react to events.
- Can be used in **Apex, Flows, and External Systems (Streaming API)**.
- Durable events: Can retain events for up to **24 hours**.





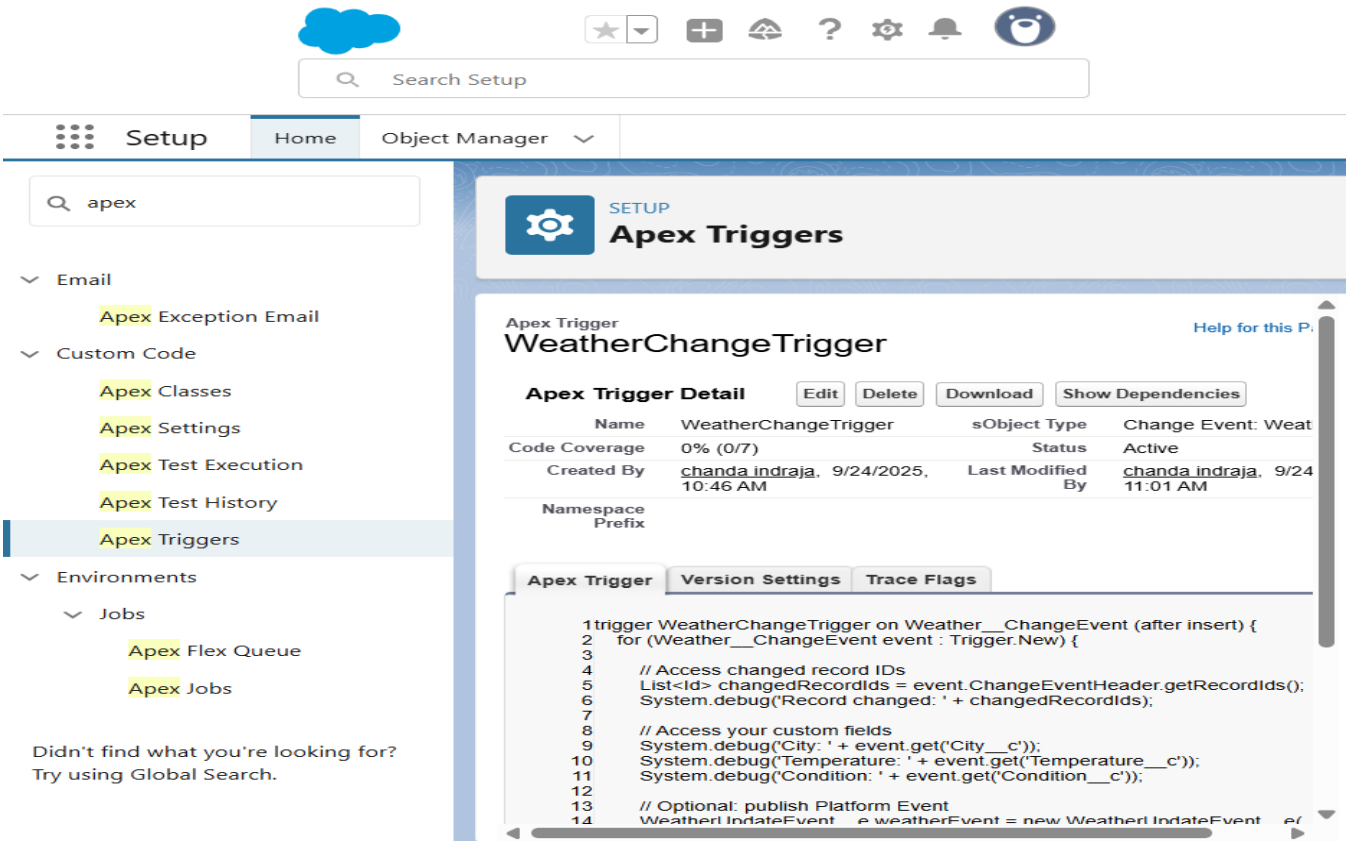
6. Change Data Capture (CDC)

Purpose

- Track **create, update, delete, and undelete events** on Salesforce records in real time.
- Enable **asynchronous integration** with internal or external systems.
- Helps in **synchronizing Salesforce data** with other systems.
- Works with **any standard or custom object**, including `Weather__c`.

Key Features

- Real-time tracking of **record changes**.
- Automatically generates **Change Event objects** for tracked objects.
- Supports **Apex Triggers, Flows, and External Subscribers**.
- Captures **all field changes** automatically.
- Can be integrated with **Streaming API** or external systems.
- Works with **standard objects** and **custom objects**.



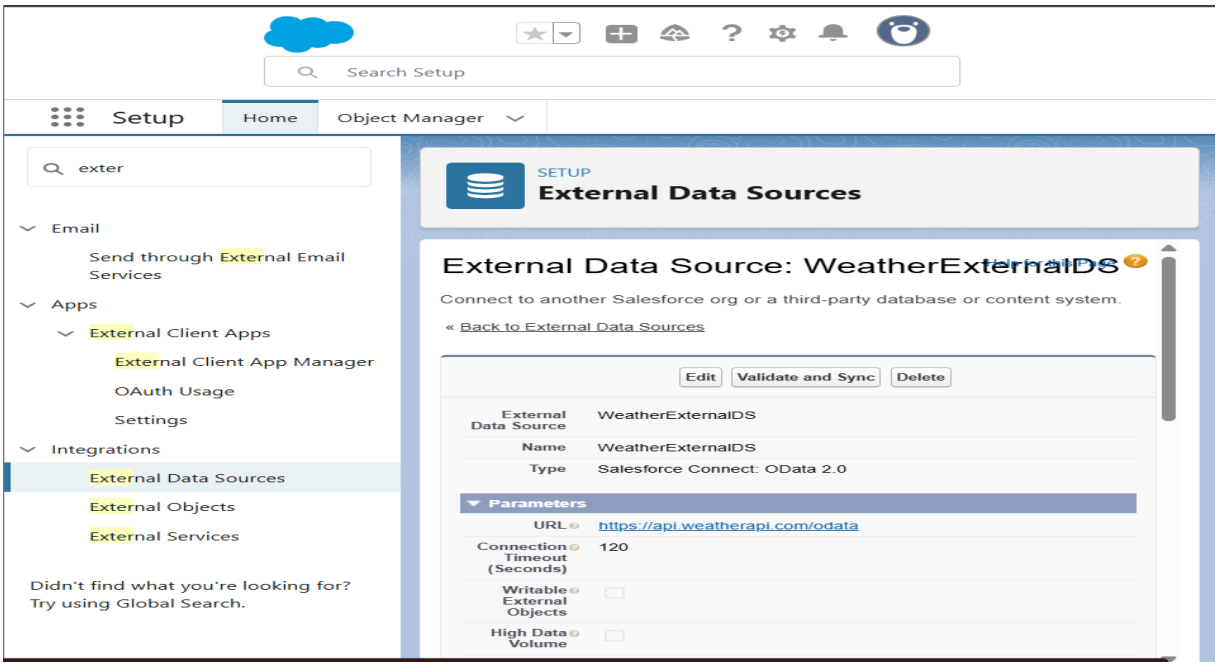
7. Salesforce Connect

Purpose

- Integrate external data sources directly into Salesforce without storing the data in Salesforce.
- Enables **real-time access** to external objects via **OData, REST, or custom adapters**.
- Ideal for **large datasets** or external systems where replication is not practical.

Key Features

- Supports **OData 2.0 and 4.0** protocols.
- Creates **External Objects** in Salesforce representing external tables.
- Provides **read, create, update, and delete** access depending on external system capabilities.
- Works with **Lightning, Visualforce, and Apex**.
- Can combine **external objects with standard Salesforce objects** using **relationship**



8. API Limits

Purpose

- Salesforce enforces **limits on API calls** to protect system resources.
- Helps track and manage usage for integrations and automation.

Key Points

- **API Request Limits** vary by **edition, license type, and user**.

- Types of API calls:
  - REST API
  - SOAP API
  - Bulk API
  - Streaming API
  - Tooling API
- Monitor usage via: **Setup → System Overview → API Usage**, or **REST `Limits` resource**.
- **Exceeded limits** return `REQUEST_LIMIT_EXCEEDED` errors.

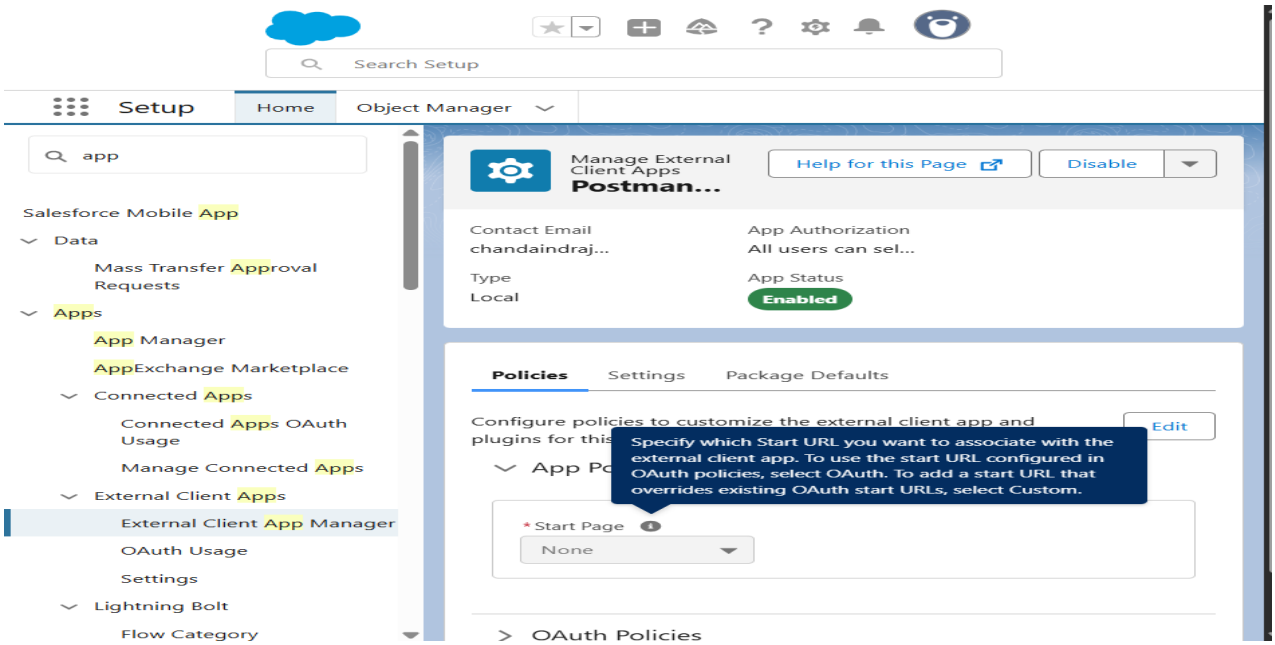
9. OAuth & Authentication

Purpose

- Secure integration with Salesforce or external systems.
- Provides **token-based authentication** without storing passwords.
- Supports **single sign-on (SSO) and API access**.

Key Features

- Supports **OAuth 2.0 flows**:
  - Authorization Code
  - Username-Password
  - JWT Bearer
  - Refresh Token
- Provides **access token** for API calls.
- Tokens can expire; refresh tokens allow re-authentication without user intervention**.**



10.Remote Site Settings

Purpose

- Allows Salesforce to **perform HTTP or HTTPS callouts** to external systems.
- Required when using **Apex HTTP callouts** to URLs that are **not secured via Named Credentials**.
- Ensures **security by allowing only trusted URLs** to be accessed from Salesforce.

Key Features

- Lets Salesforce **call external web services or APIs**.
- Provides a **centralized place** to manage external URLs.
- Required for **REST/SOAP API integration**, webhooks, or external system data fetch.
- Works with **all Apex HTTP callouts**, including GET, POST, PUT, DELETE.
- Supports **HTTPS** endpoints for secure communication.
- Can be used in combination with **Named Credentials** for advanced authentication.