SafeCity Salesforce Project:

Phase 7: Integration & External Access

In this phase, we focused on enabling Salesforce to interact with external systems in a **secure and controlled manner**. The key components implemented were **Remote Site Settings** and **Auth Provider**.

---

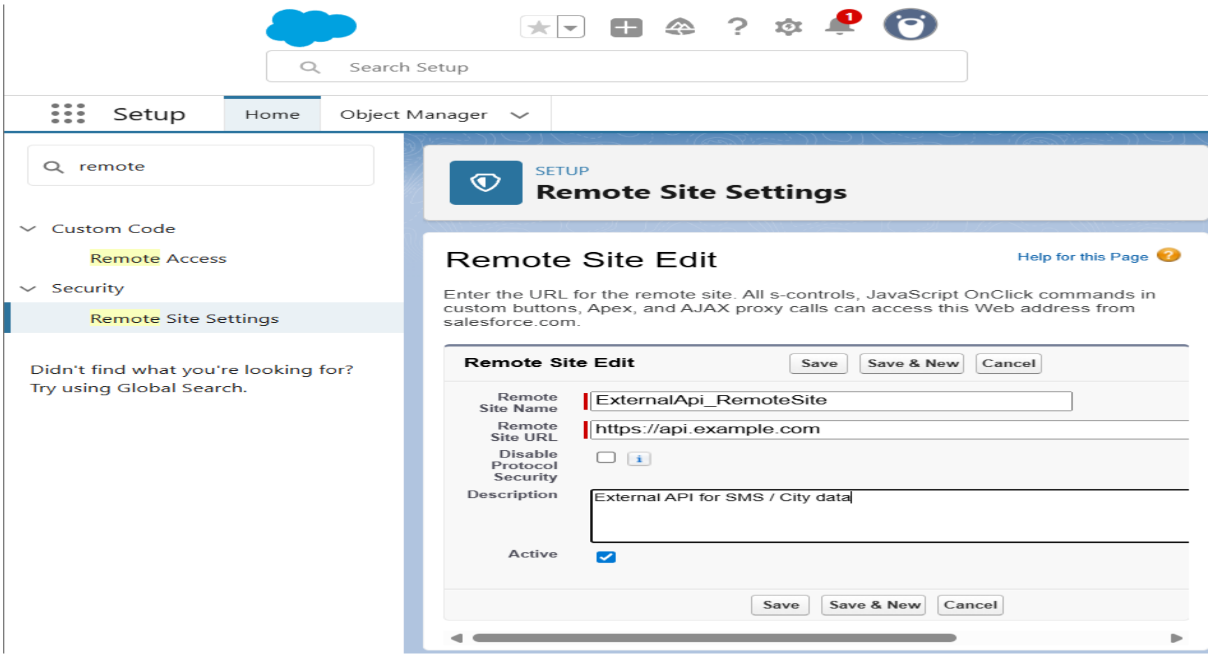
1. Remote Site Settings

Purpose:
Salesforce enforces strict security policies and does not allow callouts to external endpoints unless the URL is explicitly registered. Remote Site Settings allow us to **whitelist external domains** so Salesforce can safely send and receive data.

Configuration:

- Created a new Remote Site called **SafeCity_Remote**.
- Added the dummy URL: https://api.example.com.
- Marked the site as **Active**.

Outcome:
Now, Salesforce is authorized to perform HTTP callouts to the whitelisted domain. This is essential for future integrations (e.g., syncing complaint data with municipal or police systems).



Didn't find what you're looking for? Try using Global Search.

Remote Site Edit

Enter the URL for the remote site. All s-controls, JavaScript OnClick commands in custom buttons, Apex, and AJAX proxy calls can access this Web address from salesforce.com.

Remote Site Edit

Remote Site Name: ExternalApi_RemoteSite
Remote Site URL: https://api.example.com
Disable Protocol Security: □
Description: External API for SMS / City data
Active: ☑

---

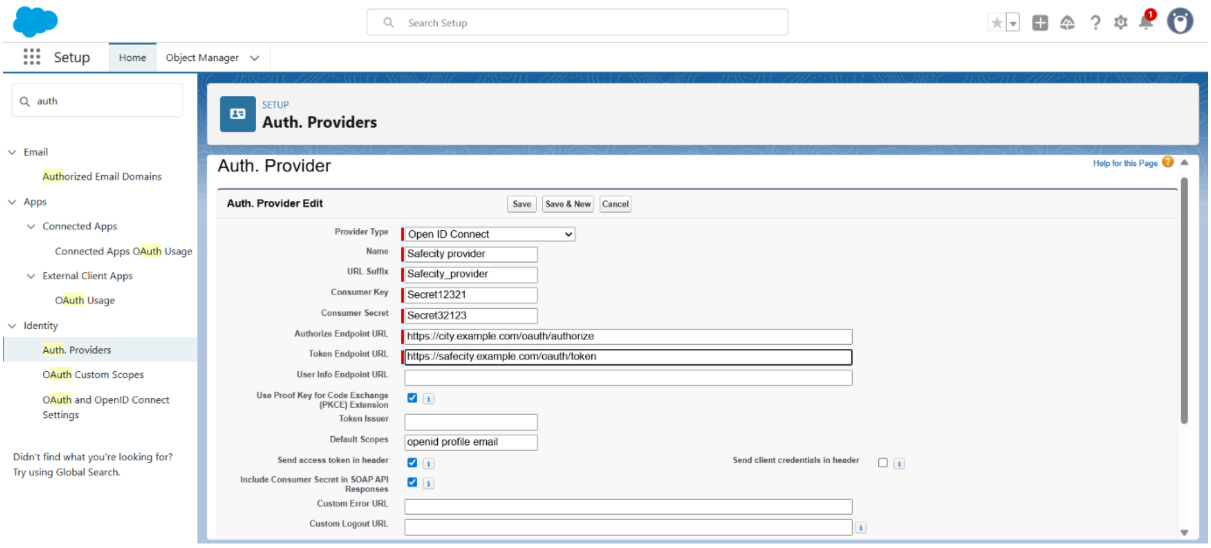
2. Auth Provider

Purpose:
Auth Providers in Salesforce define how the platform can securely **authenticate with external systems** using industry-standard protocols like OAuth 2.0. This avoids hardcoding usernames and passwords.

Configuration:

- Created a new **Auth Provider** (OpenID Connect type for demonstration).
- Entered placeholder **Consumer Key** and **Consumer Secret** values.
- Used the auto-generated Salesforce **Callback URL** to complete the setup.

Outcome:
This simulates an OAuth handshake, enabling SafeCity to integrate with external services (for example, authenticating citizens through a third-party identity provider).



Auth. Provider

Auth. Provider Edit

Provider Type	Open ID Connect
Name	Safecity provider
URL Suffix	Safecity_provider
Consumer Key	Secret12321
Consumer Secret	Secret32123
Authorize Endpoint URL	https://city.example.com/oauth/authorize
Token Endpoint URL	https://safecity.example.com/oauth/token
User Info Endpoint URL	
Use Proof Key for Code Exchange (PKCE) Extension	☑
Token Issuer	
Default Scopes	openid profile email
Send access token in header	☑
Send client credentials in header	□
Include Consumer Secret in SOAP API Responses	☑
Custom Error URL	
Custom Logout URL	

3. Named Credentials (Skipped)

Note:
Although Named Credentials are generally recommended to combine authentication + endpoint in one place, for this project we limited our scope to **Remote Site Settings** and **Auth Provider** only. Named Credentials can be added in the future if real API callouts are required.