SALESFORCE PROJECT

PROJECT TITTLE :RealEstate CRM (Property & Agent Management)

PROBLEM: Real estate firms struggle with property listings, agent performance, and client leads. Solution

Phase 7: Integration & External Access

A. Named Credentials

Purpose: Securely store authentication info for external services so Apex or Flows can call them.

Steps:

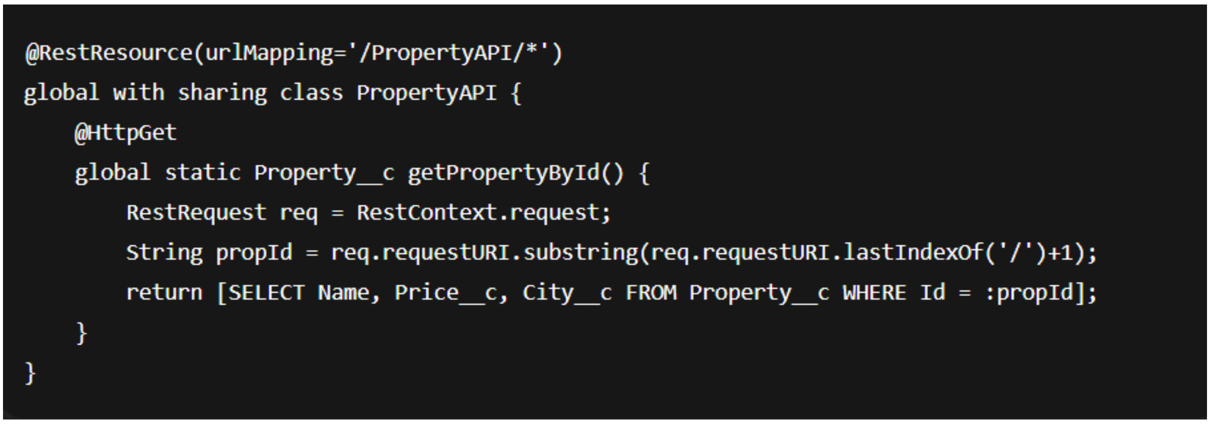
1. **Setup → Named Credentials → New**
    - **Label:** GoogleMapsAPI
    - **URL:** https://maps.googleapis.com
    - **Identity Type:** Named Principal
    - **Authentication:** OAuth 2.0 or API Key (depends on service)
    - **Generate Authorization Header:** Enabled (if OAuth)

B. External Services

- **Purpose:** Declaratively call external REST APIs without Apex.
- **Example:** DocuSign REST API to send contracts for signature.
- **Steps:**
    1. Setup → External Services → New External Service
    2. Provide **OpenAPI/Swagger JSON** from DocuSign
    3. Auto-generate Apex actions available in Flows or LWC

C. Web Services (REST / SOAP)

- **Inbound (Exposing Salesforce):**
    - Create **Apex REST classes** to expose endpoints for mobile apps or portals.



```
@RestResource(urlMapping='/PropertyAPI/*')
global with sharing class PropertyAPI {
    @HttpGet
    global static Property__c getPropertyById() {
        RestRequest req = RestContext.request;
        String propId = req.requestURI.substring(req.requestURI.lastIndexOf('/')+1);
        return [SELECT Name, Price__c, City__c FROM Property__c WHERE Id = :propId];
    }
}
```

Outbound (Calling External APIs):

- Use Apex HTTP class with **Named Credentials**
- Ensure remote site or Named Credential URL is allowed

D. Callouts

- **HTTP Callouts:** Retrieve external data (e.g., Google Maps geocoding, MLS feeds)
- **Considerations:**
    - Asynchronous callouts if needed in trigger context (@future(callout=true))
    - Respect **Salesforce governor limits**: 100 synchronous callouts per transaction

E. Platform Events

- **Purpose:** Publish/subscribe to system-wide events in real-time
- **Use Case:**
    - Example: “Property Updated” event triggers notification in external portal or mobile app
- **Steps:**

1. Setup → Platform Events → New Platform Event (e.g., Property_Updated__e)
2. Fields: PropertyId, Status, UpdatedBy
3. Subscribers: Apex Trigger, Flow, External System via CometD

F. Change Data Capture

- **Purpose:** Listen to record changes in Salesforce for external integrations
- **Example:** Send updates to ERP or mobile app whenever Deal__c changes stage.
- **Steps:**
    1. Setup → Change Data Capture → Enable on Deal__c, Property__c
    2. Subscribe via **CometD** or **Platform Event Trigger**

G. Salesforce Connect (External Objects)

- **Purpose: Access external data (like MLS listings) in Salesforce without storing it internally**
- **Setup:**
    1. **Create External Data Source (OData / REST / SOAP)**
    2. **Create External Objects that map to external tables**
    3. **Configure relationships to Salesforce objects (e.g., Deal__c → External Property)**

H. API Limits

- **Considerations:**
    - **Google Maps API: Free tier = 25,000 requests/month (adjust for heavy usage)**
    - **Salesforce: 100 synchronous callouts per transaction**
    - **Monitor limits via Setup → System Overview → API Usage**

I. OAuth & Authentication

- **Best Practice:**
    - **Use OAuth 2.0 for secure integrations**
    - **Store credentials in Named Credentials**
    - **Avoid hardcoding API keys in Apex or Flows**

J. Remote Site Settings

- **Required only if you use Apex callouts without Named Credentials**
    - **Setup → Remote Site Settings → Add URL**
    - **Example: https://maps.googleapis.com**