

Phase 7: Integration & External Access

In this phase, we integrated Salesforce with external systems (e.g., SMS provider, Payment Gateway) using secure callouts, Named Credentials, and Custom Metadata. The goal was to enable real-time notifications, reminders, and payment processing while ensuring security and scalability for the Salesforce CRM implementation.

1) Named Credentials

Why: Simplifies external API authentication (avoids Remote Site Settings).

Setup:

1. Go to Setup → Quick Find → Named Credentials → New.
2. Configure:
 - Label: MySMSProvider
 - Name: MySMSProvider
 - URL: `https://api.smsprovider.com/v1`
 - Identity Type: Named Principal
 - Authentication Protocol: No Authentication (API key managed via Custom Metadata)

Result: Apex callouts use syntax: `callout:MySMSProvider/...`

2) Custom Metadata for API Keys

Why: Avoids hardcoding API keys in Apex.

Setup:

- Create Custom Metadata Type → `External_Provider__mdt`
- Add field: `API_Key__c` (Text, 255)
- Create record with `DeveloperName = MySMSProvider`, `API_Key__c = <test-key>`

Usage: Queried inside Apex callouts to add Authorization headers dynamically.

3) Web Services (REST Callouts)

SMS Integration using Queueable Apex:

`SmsQueueable.cls` (excerpt):

```
public class SmsQueueable implements Queueable, Database.AllowsCallouts {
    private String toPhone;
    private String messageBody;

    public SmsQueueable(String toPhone, String messageBody) {
        this.toPhone = toPhone;
        this.messageBody = messageBody;
    }
}
```

```

public void execute(QueueableContext ctx) {
    Http http = new Http();
    HttpRequest req = new HttpRequest();
    req.setEndpoint('callout:MySMSProvider/messages');
    req.setMethod('POST');
    req.setHeader('Content-Type', 'application/json');
    Map<String, Object> payload = new Map<String, Object>{
        'to' => toPhone,
        'from' => 'SENDERID',
        'body' => messageBody
    };
    req.setBody(JSON.serialize(payload));
    HttpResponse res = http.send(req);
    System.debug('SMS response: ' + res.getBody());
}
}

```

☞ Result: Counselors can send SMS reminders directly from Salesforce.

4) Callouts Test Framework

Created MockSmsResponse.cls and SmsQueueableTest.cls to simulate API responses.

Example:

```

@isTest
private class SmsQueueableTest {
    @isTest static void testSmsQueueable_callout() {
        Test.setMock(HttpCalloutMock.class, new MockSmsResponse());
        Test.startTest();
        Id jobId = System.enqueueJob(new SmsQueueable('+919800000000', 'Test SMS'));
        Test.stopTest();
        System.assertNotEquals(null, jobId);
    }
}

```

☞ Guarantees test coverage and safe deployment.

5) Payment Gateway Stub

PaymentService.cls → Handles POST /charges callouts.

Code Excerpt:

```

public with sharing class PaymentService {
    @AuraEnabled
    public static Map<String, Object> createCharge(String token, Decimal amount, String currencyCode) {
        HttpRequest req = new HttpRequest();
        req.setEndpoint('callout:PaymentProvider/charges');
    }
}

```

```

    req.setMethod('POST');
    req.setHeader('Content-Type', 'application/json');
    Map<String,Object> body = new Map<String,Object>{'token' => token,'amount' => amount,'currency'
=> currencyCode};
    req.setBody(JSON.serialize(body));
    HttpResponse res = new Http().send(req);
    return (Map<String,Object>) JSON.deserializeUntyped(res.getBody());
  }
}

```

ProviderWebhook.cls → Exposed as REST service to handle provider callbacks.

6) Platform Events & Change Data Capture (Future Scope)

Salesforce supports event-driven integration.

Example: Trigger event when Enrollment__c is created → notify external system.

7) Salesforce Connect (Future Scope)

Not implemented here.

Use case: Read-only access to external databases via OData without storing data in Salesforce.

8) API Limits & Authentication

- Callout Limits: Max 100 per transaction, 250k per day.
- Authentication: API key-based via Custom Metadata.
- Future: OAuth flows for payment providers like Stripe/Razorpay.

9) Remote Site Settings

Normally required for external URLs, but skipped here since Named Credentials handle it automatically.

✓ Outcome

Phase 7 enabled Salesforce to securely connect with external APIs. Counselors can send SMS reminders, payment gateway integration is stubbed and test-ready, API keys are stored securely, and test classes guarantee deployment readiness. The system is prepared for scaling with advanced integrations.