

# Phase 7: Integration & External Access

## Flight Reservation & Scheduling System

### Salesforce-Based Flight Operations and Scheduling System

This phase focuses on connecting Salesforce with external systems (APIs, web services, third-party apps). It includes REST/SOAP callouts, Named Credentials, authentication, and real-time data exchange using Platform Events and Change Data Capture.

#### Step 1: Named Credentials

- **Purpose:** Simplifies authentication for callouts to external systems.
- **How to do it:**
- Go to **Setup** → **Named Credentials** → **New Named Credential**.
- Enter **Label, Name, URL of external service**.
- Choose **Identity Type** (Named Principal / Per User).
- Set **Authentication Protocol** (Password Authentication, OAuth, etc.).
- Save.

#### Step 2: External Services

- **Purpose:** Connect Salesforce to external APIs declaratively.
- **How to do it:**
- Go to **Setup** → **External Services** → **Register External Service**.
- Provide **API specification** (OpenAPI/Swagger file URL or JSON).
- Salesforce generates **invocable actions** to use in Flow or Apex.

#### Step 3: Web Services (REST/SOAP)

- **Purpose:** Expose Salesforce data or consume external services.
- **How to do it:**
- **REST API (Salesforce as server):**
- Create **Apex REST class**:
  - `@RestResource(urlMapping='/Flights/*')` - global with sharing class
  - FlightService {
  - `@HttpGet`
  - global static Flight\_\_c getFlight() { ... }
  - }
- **SOAP API (Salesforce as server):**
- Create **Apex SOAP class** and expose via WSDL.
- External system consumes WSDL.

- **REST/SOAP Callouts (Salesforce as client):**
- Add endpoint in **Remote Site Settings**.
- Use **HttpRequest** in Apex for callouts.

## Step 4: Callouts

- **Purpose:** Salesforce communicates with external services.
- **How to do it:**
- Go to **Setup → Remote Site Settings → New Remote Site**.
- Enter **Name & URL** of external service.
- In Apex, create **HTTP request**:
- `HttpRequest req = new HttpRequest();`
- `req.setEndpoint('https://example.com/api');`
- `req.setMethod('GET');`
- `HttpResponse res = new Http().send(req);`

## Step 5: Platform Events

- **Purpose:** Publish/Subscribe architecture for real-time integrations.
- **How to do it:**
- Go to **Setup → Platform Events → New Platform Event**.
- Add **custom fields** to event.
- Publish events in Apex or Flow.
- Subscribers receive events via **Apex triggers, CometD, or external systems**.

## Step 6: Change Data Capture (CDC)

- **Purpose:** Track Salesforce record changes in real-time.
- **How to do it:**
- Go to **Setup → Change Data Capture**.
- Select objects (e.g., `Booking__c`, `Flight__c`).
- External systems subscribe via **CometD or Platform Events**.

## Step 7: Salesforce Connect

- **Purpose:** Access external data **in real-time** without storing in Salesforce.
- **How to do it:**
- Go to **Setup → External Data Sources → New External Data Source**.
- Choose **Salesforce Connect type** (OData 2.0 / 4.0).
- Configure authentication (Named Credentials).
- Create **External Objects** to view data like standard Salesforce objects.

## Step 8: API Limits

- **Purpose:** Monitor Salesforce API usage to avoid hitting limits.
- **How to do it:**

- Go to **Setup** → **System Overview** → **API Usage**.
- Check **API calls remaining, last 24 hours**.
- Optimize callouts using **Bulk API** or **batching requests**.

### Step 9: OAuth & Authentication

- **Purpose:** Secure access for external apps using Salesforce APIs.
- **How to do it:**
- Go to **Setup** → **App Manager** → **New Connected App**.
- Enable **OAuth Settings**.
- Select **Scopes** (Full Access, API Access, etc.).
- Use **Consumer Key & Secret** for authentication in external apps.

### Step 10: Remote Site Settings

- **Purpose:** Allow Salesforce to call external services.
- **How to do it:**
- Go to **Setup** → **Remote Site Settings** → **New Remote Site**.
- Enter **Name & URL** of the external system.
- Save.