

## Phase 7: Integration & External Access

### Objective:

Enable Revaa to securely communicate with external systems, perform API callouts, handle events, and manage data integration efficiently.

---

#### 1 Named Credentials

**Purpose:** Store authentication settings for external systems, so you don't have to hardcode passwords, tokens, or endpoints in Apex or Flows.

##### Steps to Configure:

1. Navigate to **Setup → Named Credentials**.
2. Click **New Named Credential**.
3. Fill in the details:
  - o **Label:** RevaaLocalEndpoint
  - o **Name:** RevaaLocalEndpoint
  - o **URL:** `https://<ngrok-generated-url>` (*Example: https://chelsea-relevantly.ngrok-free.dev*)
  - o **Identity Type:** Anonymous or Named Principal (depending on authentication needs)
  - o **Authentication Protocol:** Choose as per your external service (e.g., OAuth 2.0 or Password Authentication)
4. Click **Save**.

SETUP > NAMED CREDENTIALS

**NgrokLocalService**

**Label:** NgrokLocalService

**Name:** NgrokLocalService

**URL:** https://chelsea-repand-relevantly.ngrok-free.dev

**Enabled for Callouts:**

**Authentication:**

- External Credential: LocalNgrokAuth
- Client Certificate

**Callout Options:**

- Generate Authorization Header:
- Allow Formulas in HTTP Header:
- Allow Formulas in HTTP Body:

**Outbound Network Connection:**

### Critical Notes:

- Named Credential must exist before configuring **External Services** or using Apex callouts.
- Do **not** type the URL manually in callouts; always reference the Named Credential.

## 2 External Services

**Purpose:** Import OpenAPI (Swagger) schemas into Salesforce to create invocable actions for Flows or Apex.

### Steps to Configure:

1. Navigate to **Setup → External Services**.
2. Click **New External Service**.
3. Provide the details:
  - **Name:** RevaaExternalAPI
  - **Named Credential:** Select RevaaLocalEndpoint (mandatory)
  - **Schema Type:** OpenAPI 2.0/Swagger
  - **Schema URL:** Paste your Swagger/OpenAPI JSON file URL or upload the schema.
4. Click **Save**.

Operation Name ↑	Description	Input parameters	Output parameters
createTransaction	Create a Transaction	body	responseCode, default, 201
getInvestor	Get Investor by ID	id	responseCode, 200, default
getTransactions	Get all Transactions		responseCode, default, 200

### Critical Notes:

- You **cannot proceed** if no Named Credential exists; ensure Step 1 is complete.
- After import, all API actions are available as **invocable actions in Flow**.

## 3 Web Services (REST/SOAP) & Callouts

**Purpose:** Enable Revaa to make callouts to external APIs.

### Steps for Apex REST Callout:

1. Ensure the endpoint domain is included in **Setup → Remote Site Settings**.
2. Create Apex class for callout:

```
public with sharing class RevaaAPIService {
    @future(callout=true)
    public static void callExternalAPI(String investorId) {
        Http http = new Http();
        HttpRequest request = new HttpRequest();
        request.setEndpoint('callout:RevaaLocalEndpoint/api/portfolio?investorId=' + investorId);
        request.setMethod('GET');
        HttpResponse response = http.send(request);
        System.debug('Response: ' + response.getBody());
    }
}
```

```
1  {
2    "openapi": "3.0.0",
3    "info": {
4      "title": "Revaa Local API",
5      "description": "OpenAPI schema for Revaa integration via ngrok",
6      "version": "1.0.0"
7    },
8    "servers": [
9      {
10        "url": "https://chelsea-repand-relevantly.ngrok-free.dev",
11        "description": "Ngrok tunnel for local development"
12      }
13    ],
14    "paths": {
15      "/transactions": {
16        "get": {
17          "summary": "Get all Transactions",
18          "operationId": "getTransactions",
19          "responses": {
20            "200": [
21              {
22                "description": "List of transactions",
23                "content": {
24                  "application/json": {
25                    "schema": {
26                      "type": "array",
27                      "items": {
28                        "$ref": "#/components/schemas/Transaction"
29                      }
29                    }
29                  }
29                }
29              ]
30            }
31          }
32        },
33      },
34      "post": {
35        "summary": "Create a Transaction",
36        "operationId": "createTransaction",
37        "requestBody": {
38          "required": true,
39          "content": {
40            "application/json": {
41              "schema": {
42                "$ref": "#/components/schemas/Transaction"
43              }
44            }
45          }
46        },
47        "responses": {
48          "201": {
49            "description": "Transaction created"
50          }
51        }
52      }
53    },
54    "/investors/{id)": {
55      "get": {
56        "summary": "Get Investor by ID",
57        "operationId": "getInvestor",
58        "parameters": [
59          {
60            "name": "id",
61            "in": "path",
62            "required": true,
63            "schema": {
64              "type": "string"
65            }
66          }
67        ]
68      }
69    }
70  }
71
72  <Live Share>
```

## Critical Notes:

- Use callout:<NamedCredentialName> to avoid hardcoding URLs.
- Methods making callouts must be annotated `@future(callout=true)` or run in asynchronous context.

## 4 Platform Events

**Purpose:** Enable real-time event-driven communication within Salesforce.

**Steps to Configure:**

1. Navigate to **Setup → Platform Events → New Platform Event**.
2. Create a Platform Event:
  - o **Label:** Transaction\_Approved
  - o **API Name:** Transaction\_Approved\_e
  - o **Fields:** Investor\_\_c (Lookup), Transaction\_\_c (Lookup), Amount\_\_c (Number)
3. Click **Save**.

**Use Case:** Trigger updates or notifications when a transaction is approved.

The screenshot shows the 'Platform Event Definition Detail' page for a 'Reva Transaction Event'. The top section displays basic details: Singular Label (Reva Transaction Event), Plural Label (Reva TransactionEvents), Object Name (Reva\_Transaction\_Event), API Name (Reva\_Transaction\_Event\_e), Event Type (High Volume), Publish Behavior (Publish Immediately), and Created By (ANIKET KAKADE). The Deployment Status is Deployed. Below this, the 'Standard Fields' section lists fields like Created By, Created Date, Event UUID, and Replay ID with their corresponding data types (Lookup(User), Date/Time, Text(36), External Lookup) and controlling fields. The 'Custom Fields & Relationships' section shows a single custom field named 'Investor Name' with API name 'Investor\_Name\_\_c' and data type 'Text(18)'. The page also includes links for 'Standard Fields' and 'Custom Fields & Relationships'.

## 5 Change Data Capture (CDC)

**Purpose:** Track real-time changes to Salesforce records for integration with external systems.

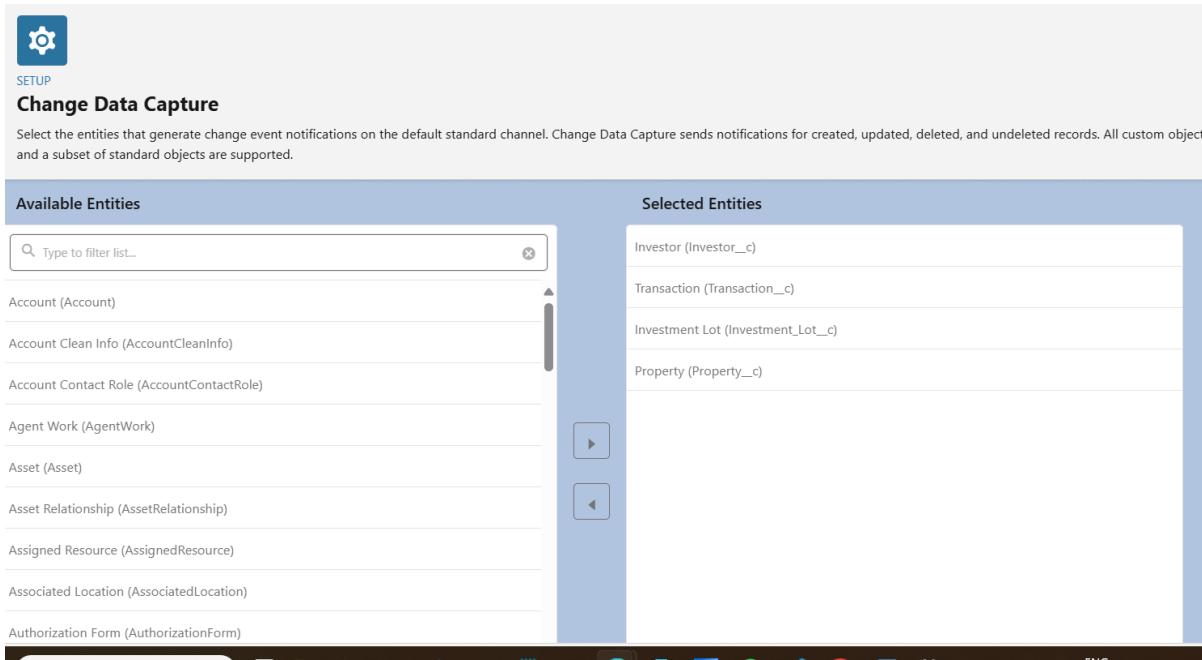
**Steps to Configure:**

1. Navigate to **Setup → Change Data Capture → Select Objects**.  
(Screenshot: *Change Data Capture page*)
2. Select relevant objects:
  - o Transaction\_\_c

- Investor\_\_c
- Property\_\_c

3. Enable events and click **Save**.

**Use Case:** External systems can subscribe to these events for real-time updates.



## 6 Salesforce Connect

**Purpose:** Access external data sources (ODBC, REST, SOAP) without importing data.

**Steps to Configure:**

1. Navigate to **Setup** → **External Data Sources** → **New External Data Source**.  
*(Screenshot: External Data Sources page)*

2. Fill in:

- **Label:** RevaaPropertyService
- **Name:** RevaaPropertyService
- **Type:** OData 2.0/3.0 or other relevant type
- **URL:** Your ngrok endpoint or API URL
- **Authentication:** Named Credential (RevaaLocalEndpoint)

3. Click **Save and Validate**.

4. Click **Sync** to bring in external objects.
- 

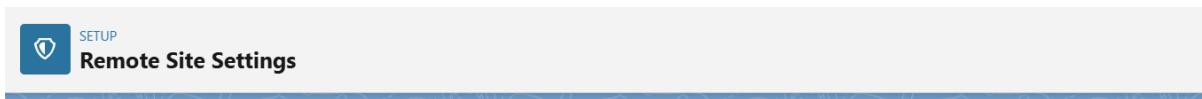
## 7 API Limits & Authentication

- Monitor API consumption at **Setup** → **System Overview** → **API Usage**.
  - For OAuth or token-based authentication, always create a **Connected App** if required.
  - Named Credential handles authentication for Flows, Apex, and External Services.
- 

## 8 Remote Site Settings

**Purpose:** Required for direct Apex callouts if **Named Credentials** are not used.

1. Navigate to **Setup** → **Remote Site Settings** → **New Remote Site**.
2. Fill in:
  - **Remote Site Name:** RevaaLocalHost
  - **Remote Site URL:** <https://chelsea-repand-relevantly.ngrok-free.dev>
3. Click **Save**.



### Remote Site Details

Remote Site Detail		<a href="#">Edit</a>	<a href="#">Delete</a>	<a href="#">Clone</a>	Modified By	ANIKET KAKADE, 9/28/2025, 4:30 PM
Remote Site Name	Revaa_External_API					
Remote Site URL	<a href="https://chelsea-repand-relevantly.ngrok-free.dev">https://chelsea-repand-relevantly.ngrok-free.dev</a>					
Disable Protocol Security	<input type="checkbox"/>					
Description						
Active	<input checked="" type="checkbox"/>					
Created By	ANIKET KAKADE, 9/28/2025, 4:30 PM					

[Edit](#) [Delete](#) [Clone](#)

## Phase 7 Summary

- **Named Credentials:** Secure authentication.
- **External Services:** Integrate external APIs as invocable actions.
- **Callouts:** Perform REST/SOAP communication.
- **Platform Events & CDC:** Real-time event handling.
- **Salesforce Connect:** Access external objects without importing.
- **Remote Site Settings:** Required for Apex callouts without Named Credentials.