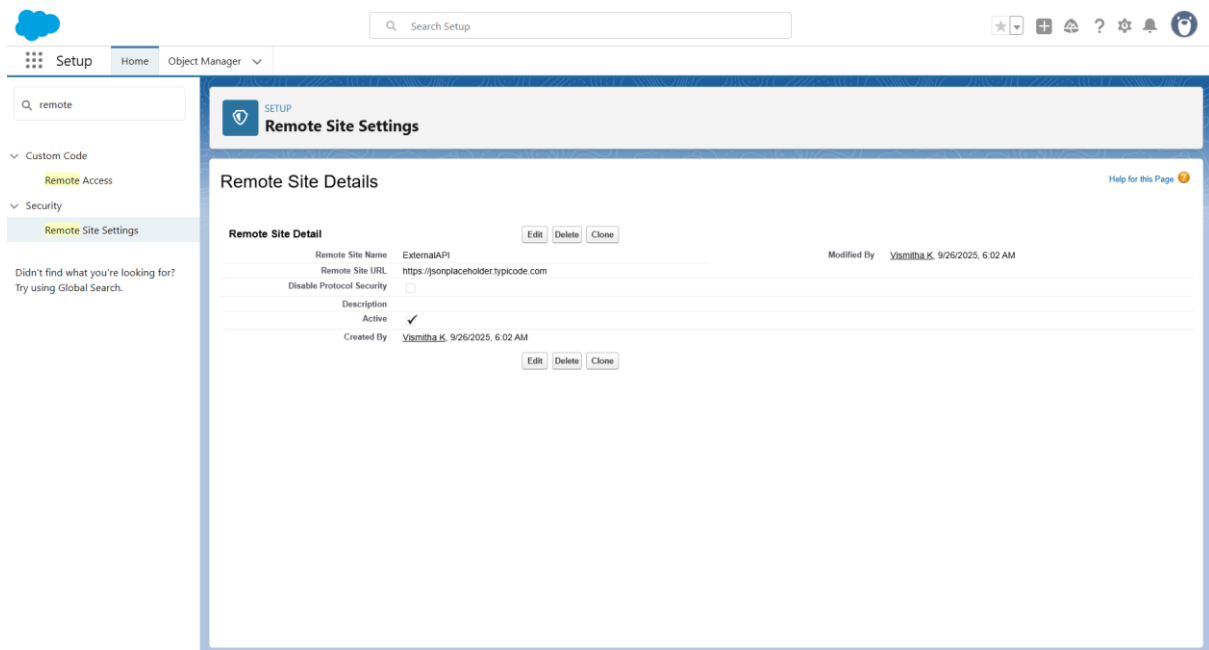


PHASE 7: INTEGRATION & EXTERNAL ACCESS

Goal: Demonstrate how the Smart Event Management system can connect Salesforce with external applications and services for real-time data exchange, notifications, and API communication.

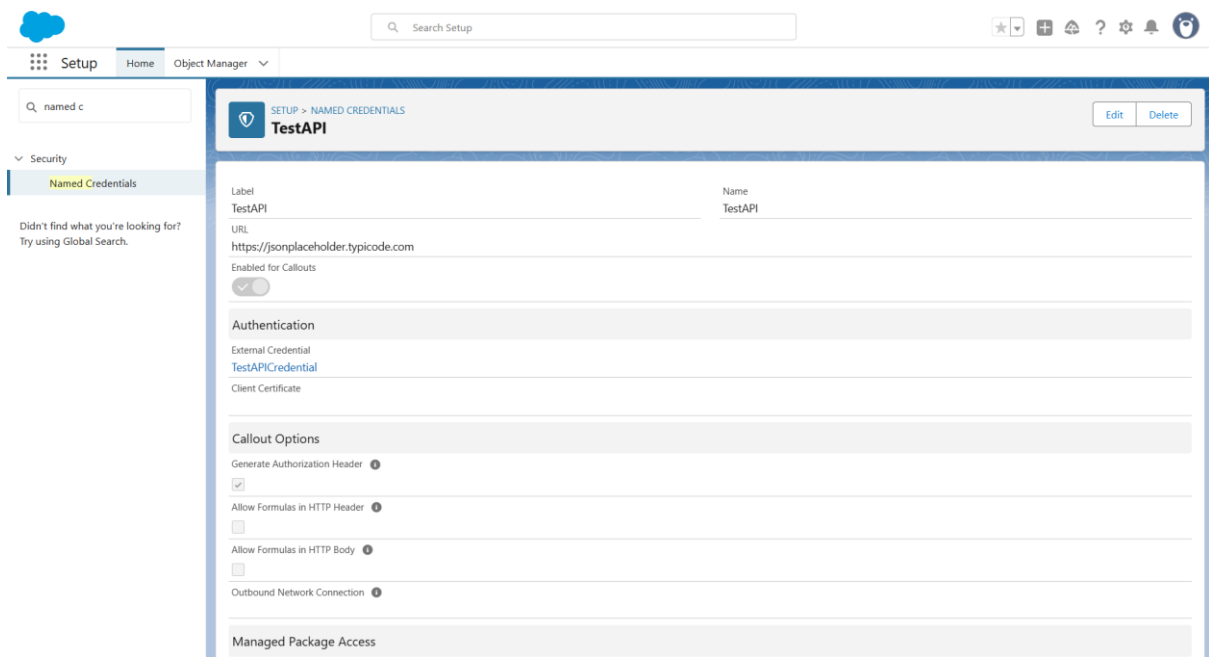
1. Remote Site Settings

- A Remote Site was configured in Salesforce to allow external callouts.
- Remote Site Name: **ExternalAPI**
- Endpoint: <https://jsonplaceholder.typicode.com> (mock API for testing).
- Purpose: Allows Salesforce to send HTTP requests securely to external systems.



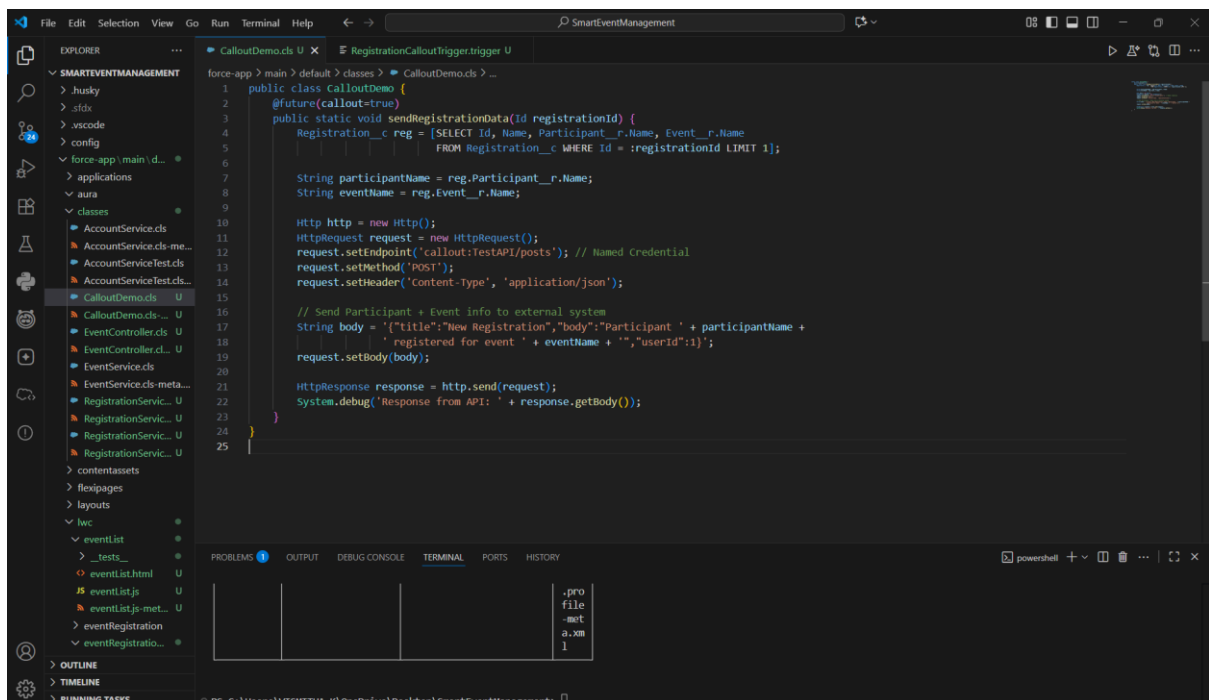
2. Named Credentials

- A Named Credential was created to simplify authentication and endpoint management.
- Label: **TestAPI**
- URL: <https://jsonplaceholder.typicode.com>
- Authentication: **Anonymous** (sufficient for mock API).
- Purpose: Provides a secure way for Apex code to reference external endpoints without hardcoding URLs.



3. Apex REST Callout (Integration Demo)

- An Apex class **CalloutDemo** was created to simulate sending participant registration data to an external system.
- A sample callout posts JSON data such as participant name and registration details to the mock API.
- Callout is executed asynchronously using `@future(callout=true)` to avoid blocking user operations.
- Debug Logs confirmed that the external service responded successfully.



4. Platform Events

- A custom Platform Event **RegistrationEvent__e** was created.
- Field: **Participant_Name__c** (Text).
- Purpose: Publishes participant registration events for external systems to subscribe to in real-time.
- Demonstrates Salesforce's capability to push business events beyond its boundary.

The screenshot shows the Salesforce Setup interface for Platform Events. The left sidebar shows the navigation menu with 'Platform Events' selected under 'Integrations'. The main content area displays the 'Registration Event' configuration page. The 'Platform Event Definition Detail' section shows the event name 'Registration Event', object name 'Registration_Event__e', and API name 'Registration_Event__e'. The 'Standard Fields' table lists fields like 'CreatedBy', 'CreatedDate', 'EventId', and 'ReplayId'. The 'Custom Fields & Relationships' table shows a custom field 'Participant_Name__c' of type 'Text(18)'. The 'Triggers' section indicates 'No triggers defined'.

Action	Field Label	Field Name	Data Type	Controlling Field	Indexed
	CreatedBy	CreatedBy	Lookup(User)		
	CreatedDate	CreatedDate	Date/Time		
	EventId	EventId	Text(36)		
	ReplayId	ReplayId	External Lookup		

Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Edit Del	Participant Name	Participant_Name__c	Text(18)			Yamitha K, 9/26/2025, 6:41 AM

5. Change Data Capture (CDC)

- Change Data Capture was enabled for the **Registration__c** object.
- Any changes (insert, update, delete) to Registration records now trigger a change event that external systems can consume.
- Ensures that third-party applications always have up-to-date registration data.

The screenshot shows the Salesforce Setup interface for Change Data Capture. The left sidebar shows the navigation menu with 'Change Data Capture' selected under 'Integrations'. The main content area displays the 'Change Data Capture' configuration page. The 'Available Entities' list includes various objects like 'Account (Account)', 'Asset (Asset)', and 'Authorization Form (AuthorizationForm)'. The 'Selected Entities' list shows 'Registration (Registration__c)' selected.

Achievements in Phase 7

- Configured **Remote Site Settings** and **Named Credentials** to establish external connectivity.
- Implemented a demo **Apex REST callout** for sending registration data.
- Introduced **Platform Events** for real-time communication.
- Enabled **Change Data Capture (CDC)** to keep external systems synchronized with Salesforce records.
- Demonstrated Salesforce's ability to securely integrate with external applications, proving scalability and enterprise readiness of the Smart Event Management app.