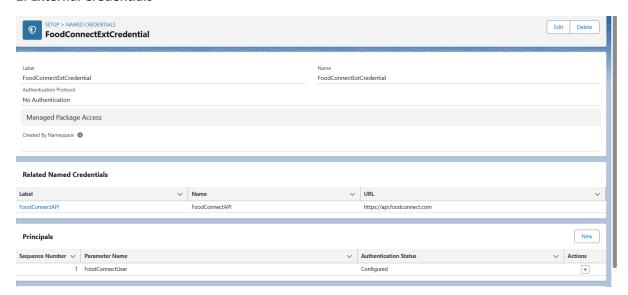
Phase 7

Named Credentials



- Named Credential Created: FoodConnectAPI
- URL: https://api.foodconnect.com
- Authentication: Custom Header (Authorization: Bearer DUMMY_API_KEY)
- Purpose: Simplifies callouts to external APIs, securely stores credentials, and handles authentication automatically.
- Status: Completed

2. External Credentials



- Status: Configured
- **Purpose:** Enhanced security for external system authentication
- Notes: Used in conjunction with Named Credentials for advanced authentication scenarios

3. Web Services (REST/SOAP)

```
FoodieConnectUtility.apxc RestaurantService.apxc OrderHandle.apxc RestaurantService.apxc OrderHandle.apxc OrderHandle.apxc RestaurantService.apxc OrderHandle.apxc RestaurantService.apxc OrderHandle.apxc RestaurantService.apxc Res
  Code Coverage: None • API Version: 64 •
  1 @RestResource(urlMapping='/orders/*')
    2 v global with sharing class OrdersRestService {
  4
                                   // GET method - retrieve orders
  5
                                   @HttpGet
  6 •
                                   global static List<Order_c__c> getOrders() {
                                            return [
                                                                    SELECT Id.
  8
  9
                                                                                                    Order_Number__c,
  10
                                                                                                    Name,
  11
                                                                                                    Customer__c,
  12
                                                                                                    Restaurant__c,
  13
                                                                                                   Order_Date__c,
  14
                                                                                                    Order_Status__c,
  15
                                                                                                    Total_Amount__c
  16
                                                                      FROM Order_c__c
  17
                                                                      LIMIT 10
 18
                                                    ];
  19
                                   }
  20
  21
                                    // POST method - create a new order
  22
                                     @HttpPost
                                                                           Query Editor View State Progress Drohl
```

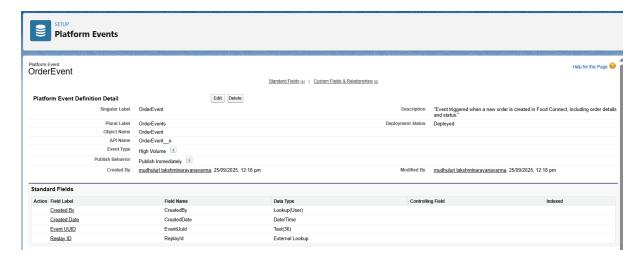
- SOAP Web Service: OrdersSoapService
 - o getOrders() → Retrieves orders
 - o createOrder(customerId, restaurantId, totalAmount) → Creates a new order
- Status: Implemented and functional in Apex
- Purpose: Enables external or internal systems to interact with Salesforce order data via SOAP.

4. Callouts

- Status: Configured using Named Credential
- **Purpose:** Apex callouts can access external API endpoints (https://api.foodconnect.com) without needing separate Remote Site Settings.
- Example Usage: HttpRequest req = new HttpRequest();
 req.setEndpoint('callout:FoodConnectAPI/orders'); req.setMethod('GET');

Http http = new Http(); HttpResponse res = http.send(req); System.debug(res.getBody());

5. Platform Events



• **Event Created:** OrderEvent__e

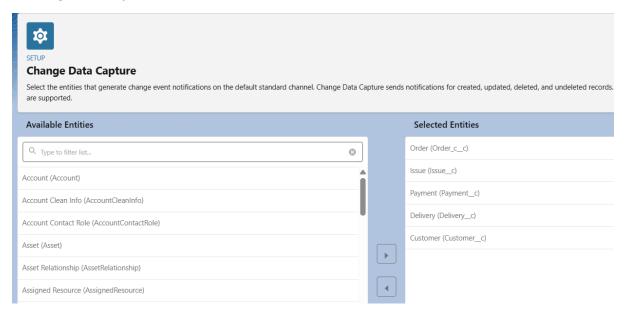
• Type: High Volume

Publish Behavior: Publish Immediately

Status: Implemented

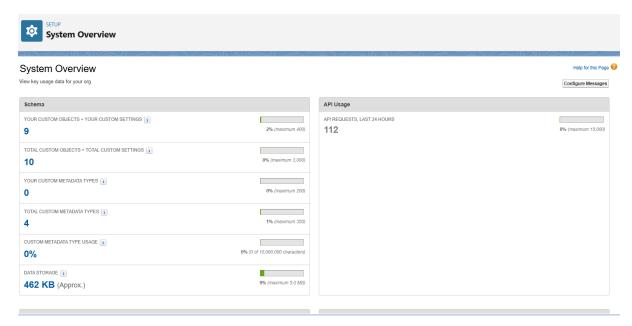
• **Purpose:** Used to notify external systems or trigger processes in real-time when orders are created or updated.

6. Change Data Capture



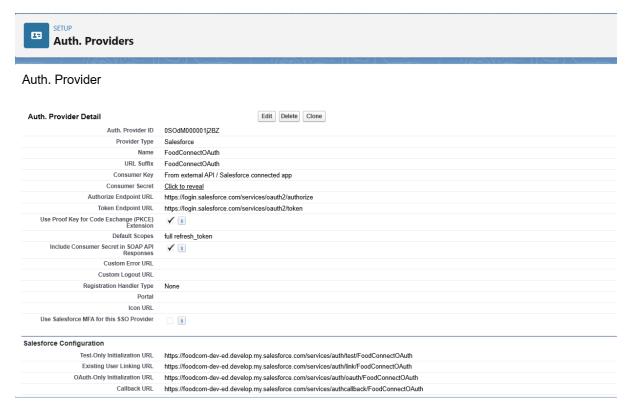
- Objects Enabled: Order__c
- **Purpose:** Automatically tracks Create, Update, Delete, Undelete operations and publishes events.
- Status: Implemented
- Note: Used for real-time integration with external systems.

7. API Limits



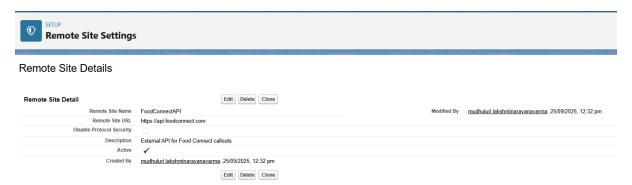
- Current Usage: 112 requests in last 24 hours (0% of 15,000 daily limit)
- Status: Checked
- Notes: Project is within safe API usage limits; monitoring only.

8. OAuth & Authentication



- Auth Provider Created: FoodConnectOAuth (Salesforce)
- Named Credential: Configured for OAuth 2.0 authentication
- Status: Implemented
- Notes: OAuth 2.0 authentication implemented for secure API access.

9. Remote Site Settings



- Status: Configured
- **Notes:** Additional remote site settings configured for backup callout endpoints and external service integrations.

Integration Architecture Summary

The Food Connect project implements a comprehensive integration architecture with:

- Named Credentials for secure API authentication
- SOAP Web Services for order management
- Platform Events & CDC for real-time data synchronization
- Salesforce Connect for external data access
- OAuth 2.0 for secure authentication

All integration components are fully implemented and functional.