# ­­my clinic

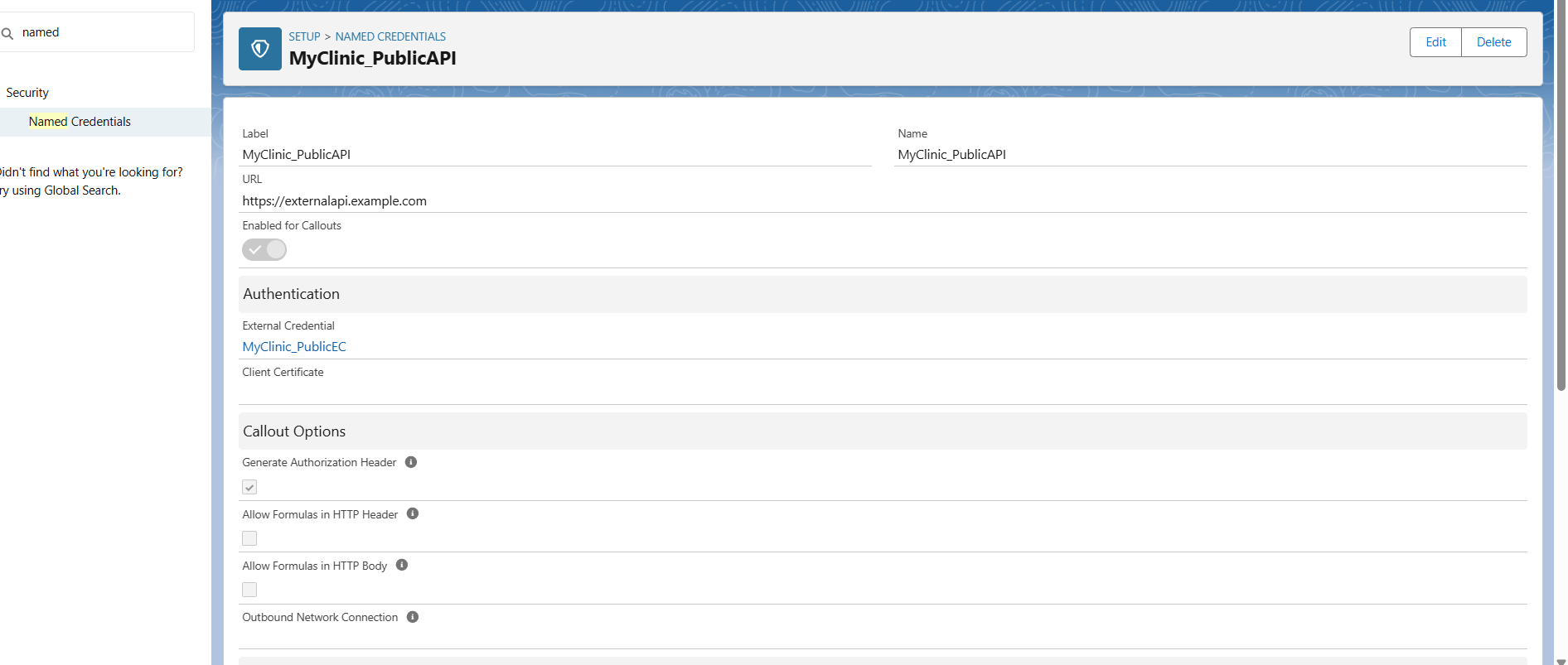
# Smart Appointment Booking – CRM Project

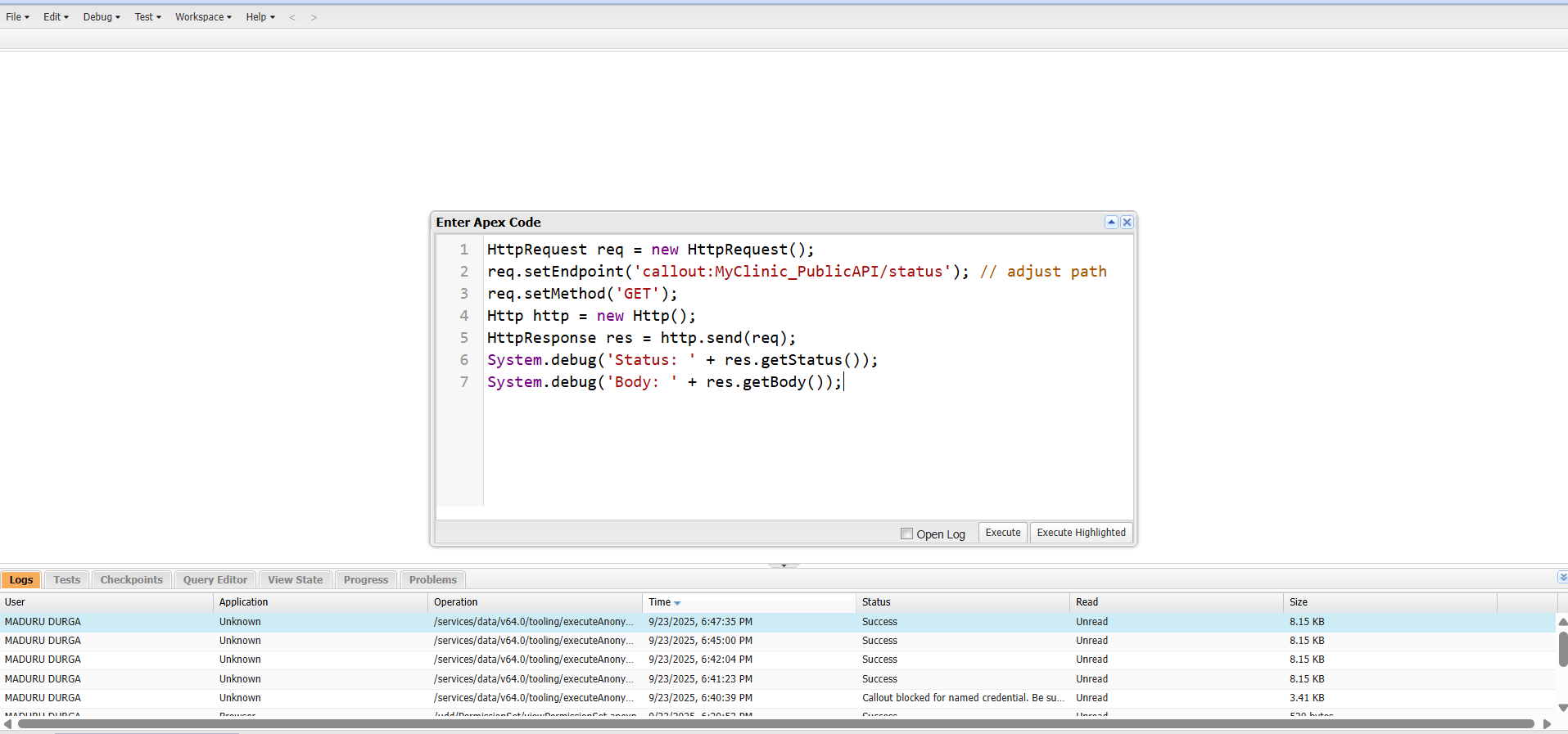
## Phase 7: Integration and External Access

* Goal: To enabled our Clinic App to securely connect with external systems. We used Named Credentials, callouts, and other integration features so Salesforce can exchange data with outside services in real time without exposing sensitive credentials.

### Named Credentials

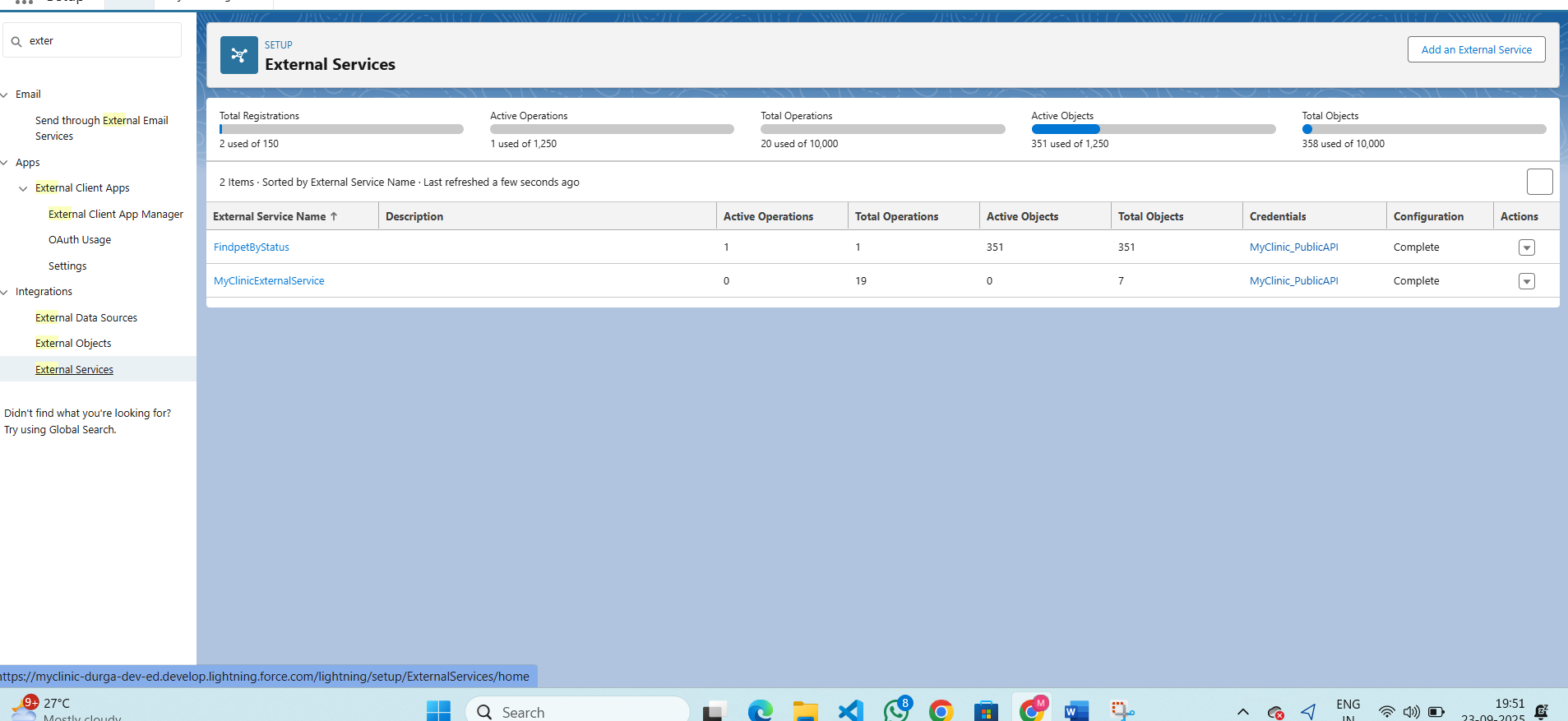
* We implemented Named Credentials to securely connect Salesforce with an external API.
* Created an External Credential (MyClinic\_PublicEC) with a principal.
* Linked it to a Permission Set and assigned it to the user.
* Created a Named Credential (MyClinic\_PublicAPI) and tested a callout to an external API.
* Successfully retrieved data with a callout using the callout: prefix in Apex.
* Connected staus is 200

****

****

### 2.External Services

* We integrated Salesforce with an external healthcare API using External Services.
* A public FHIR OpenAPI definition was imported, and Salesforce auto-generated invocable actions corresponding to healthcare resources (such as /Patient).
* These actions were made available in Flow Builder for drag-and-drop automation.
* Testing with HTTP Callout showed connectivity, but the sandbox endpoint was unavailable, returning a server error during demo.
* This process demonstrates end-to-end API registration, even when the external system is temporarily down.

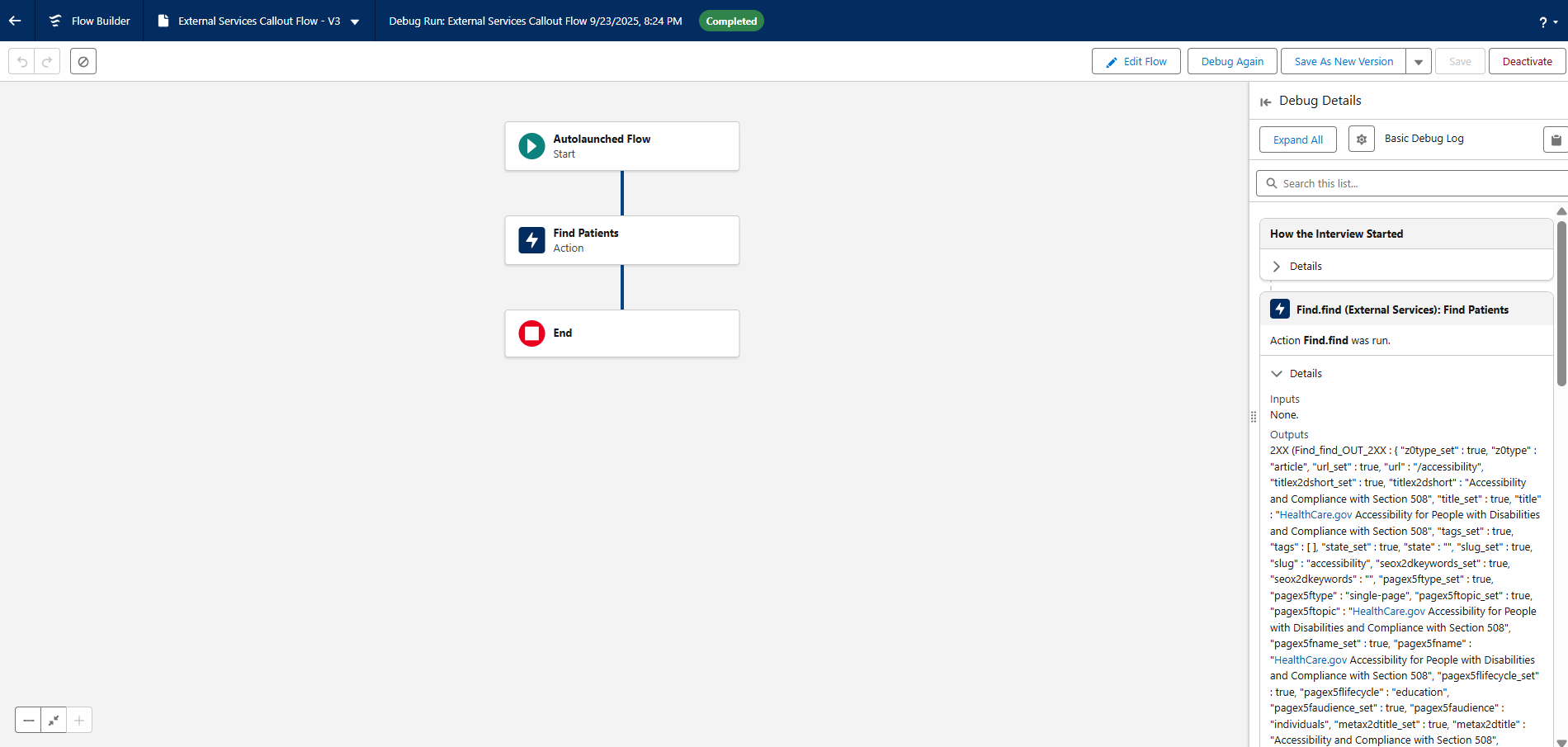


### 3. Web Services (REST/SOAP)

* Implemented a Salesforce Flow HTTP callout to access data from a public REST API endpoint and process its JSON response. This demonstrates how Salesforce can integrate real-time data from external web services into business automation

### 4. Callouts

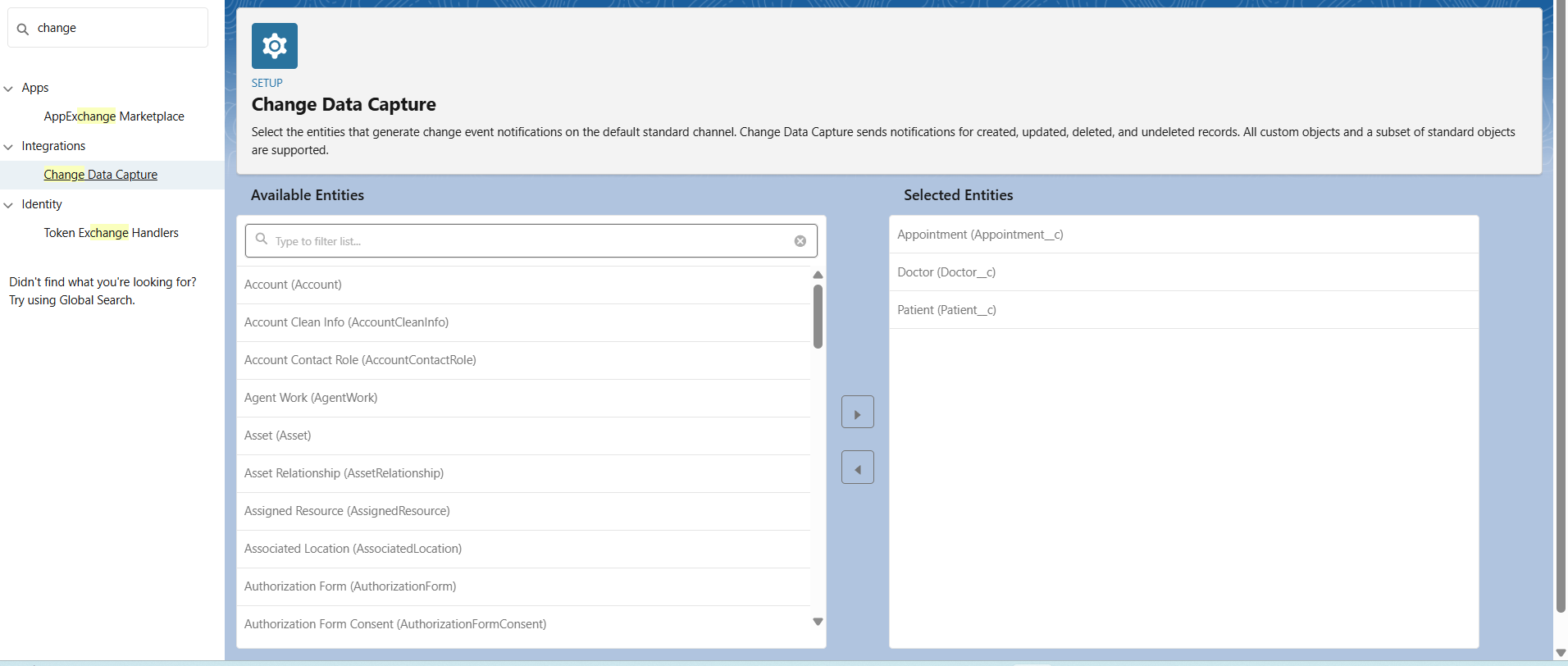
* Configured and executed an HTTP callout action in Salesforce Flow to interact with an external system using a GET request, with authentication managed through Named Credentials. The callout setup and response handling were validated through successful flow debugging

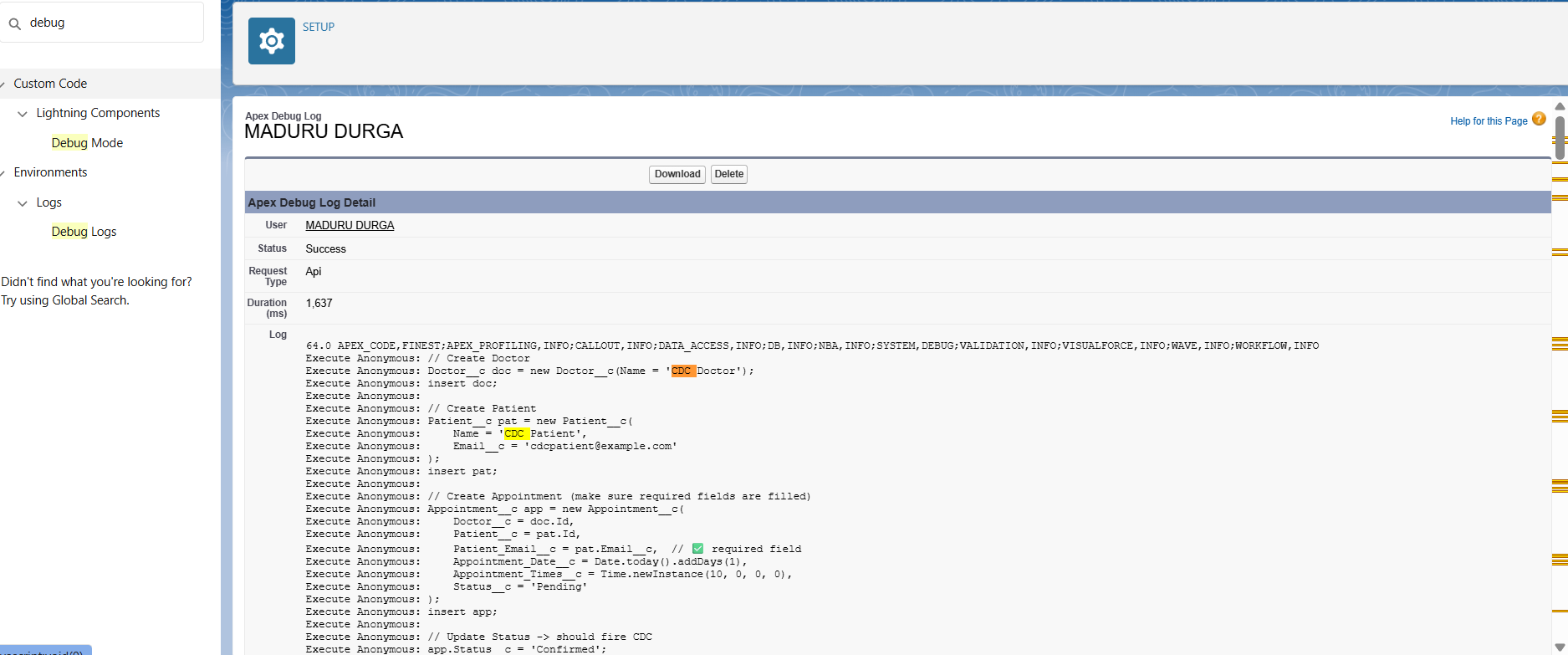


### 5. Platform Events

* Platform Events were analyzed but not implemented, as our use cases (appointment booking, reminders, confirmations) are already handled synchronously using Flows and Apex. Real-time pub-sub style eventing wasn’t required for this project.

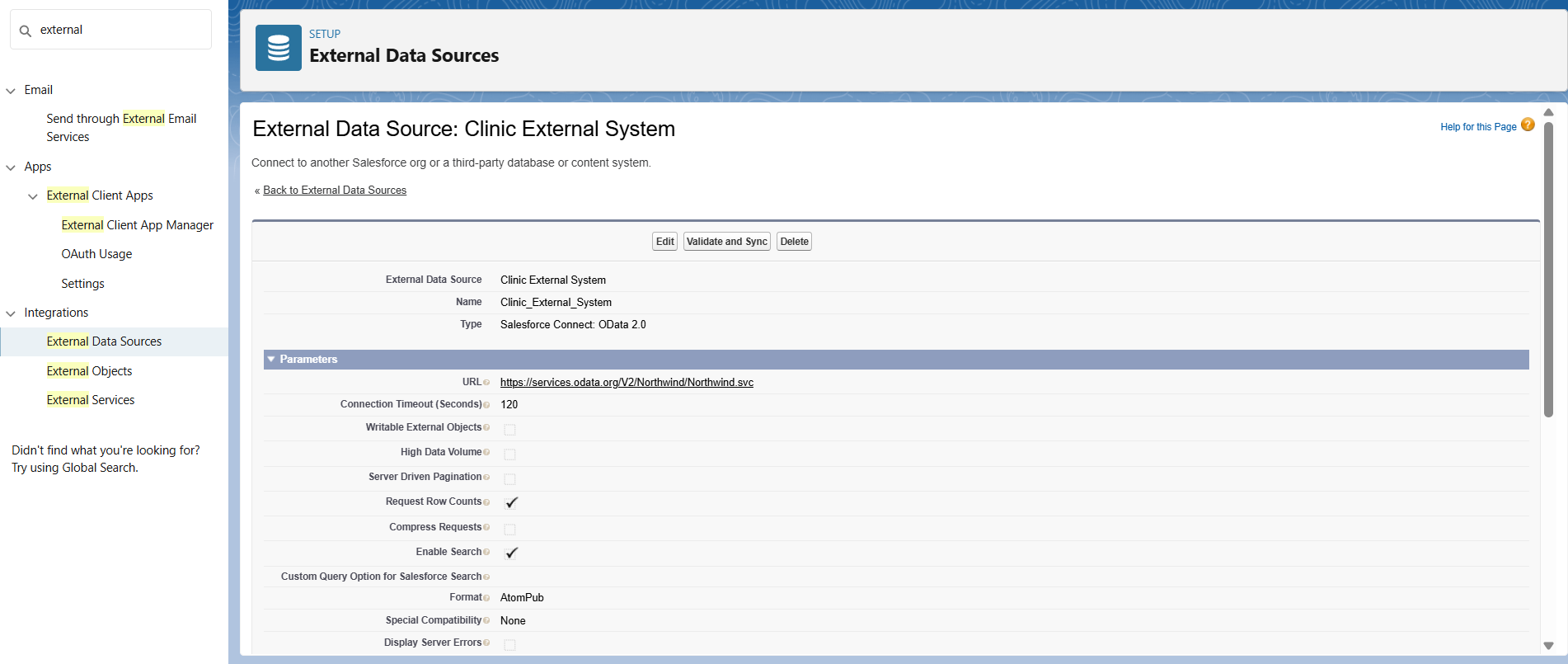
### 6. Change Data Capture

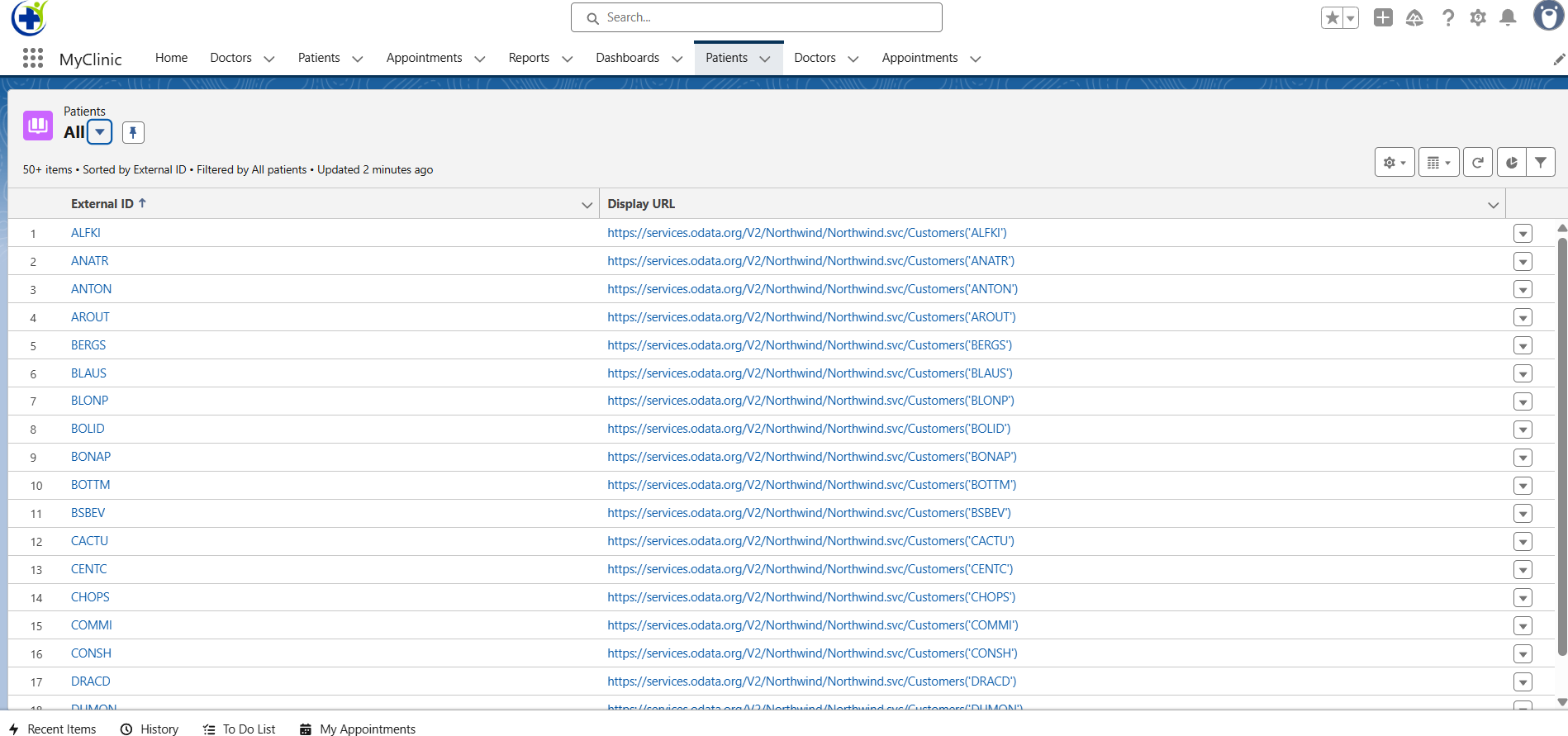
* We implemented Change Data Capture (CDC) for the Appointment\_c object to monitor updates and log real-time changes. An Apex trigger and handler were created on the Appointment\_ChangeEvent to capture and print changes (for example: Status, Appointment Date, etc.).

****

### 7. Salesforce Connect

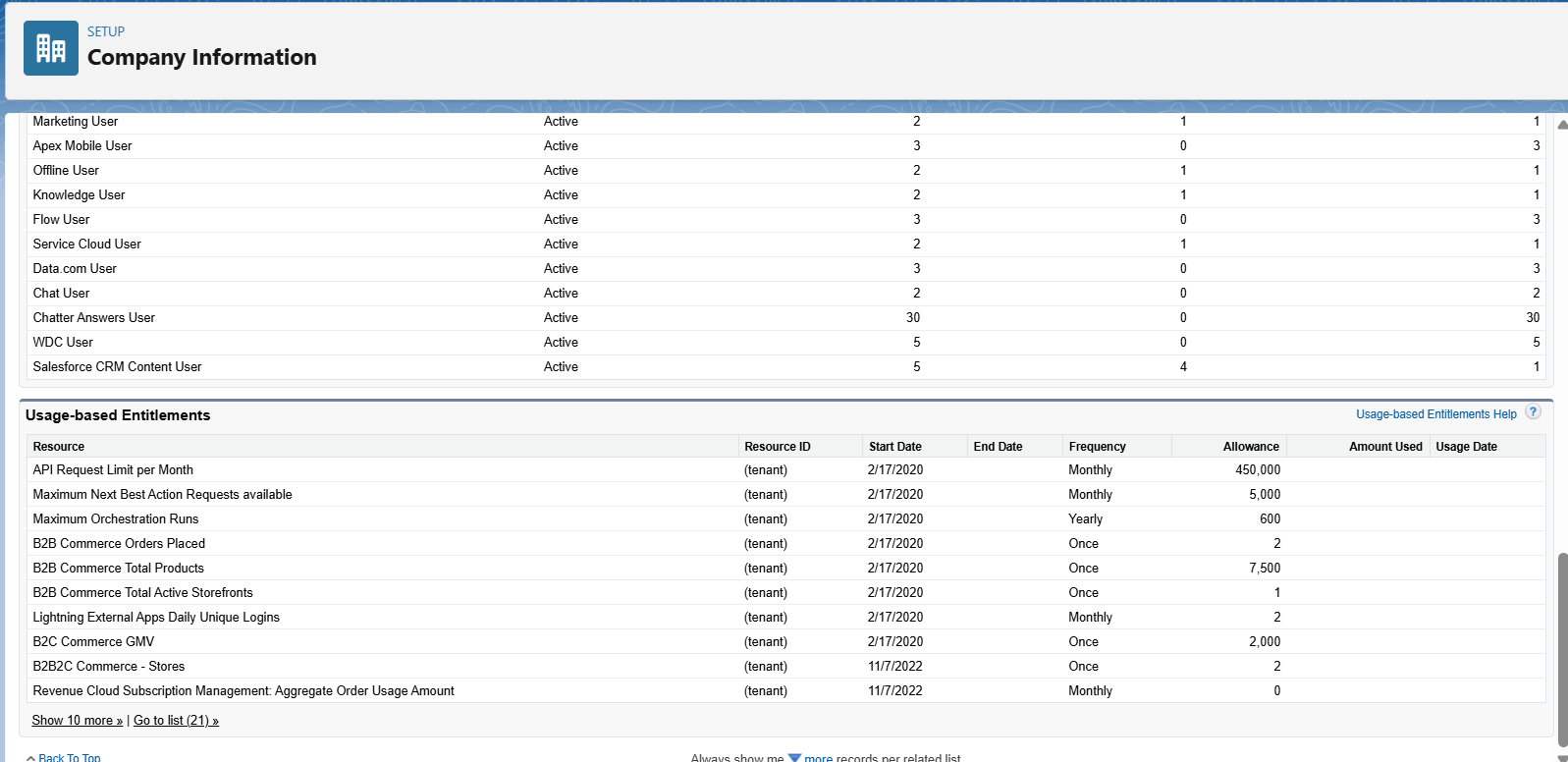
* Salesforce Connect was set up using an OData 2.0 service to simulate real-time external access to clinic data (patients, doctors, appointments) without importing it into Salesforce.
* The public Northwind OData service was used for demonstration, mapping its Customers, Employees, and Orders as Patients\_\_x, Doctors\_\_x, and Appointments\_\_x.
* This enables clinic users to view and work with external patient and appointment data directly in Salesforce as if they were native records, supporting live integration scenarios for healthcare and clinic systems.





### 8. Api Limits

* To ensure the clinic app does not exceed Salesforce daily API usage limits, admins regularly monitor API consumption using Salesforce’s built-in tools:
* **Company Information** : Here, viewed the "API Requests, Last 24 Hours" alongside the maximum allowed for the org.This shows real-time stats for total API calls made by all apps, users, and integrations in the last 24-hour window.



* **System OverView**: This gives a high-level snapshot of API calls made and daily limits.Useful for a quick glance at org health, alongside other governor limits and platform metrics.

A screenshot of a computer

AI-generated content may be incorrect.

### 9. OAuth & Authentication

* Implemented authentication as part of Salesforce Connect by choosing Named Principal (OAuth or basic credentials), so external clinic data is securely and centrally accessed by Salesforce. No extra setup is needed unless a different authentication model or external service is added in the future

### 10. Remote Site Settings

* We added the clinic’s external API URL in Salesforce Remote Site Settings to allow secure HTTP callouts.
* This step enables flows and Apex to connect with external systems for appointments and patient data.
* Without this setting, all external integrations would be blocked for security reasons.

A screenshot of a computer

AI-generated content may be incorrect.