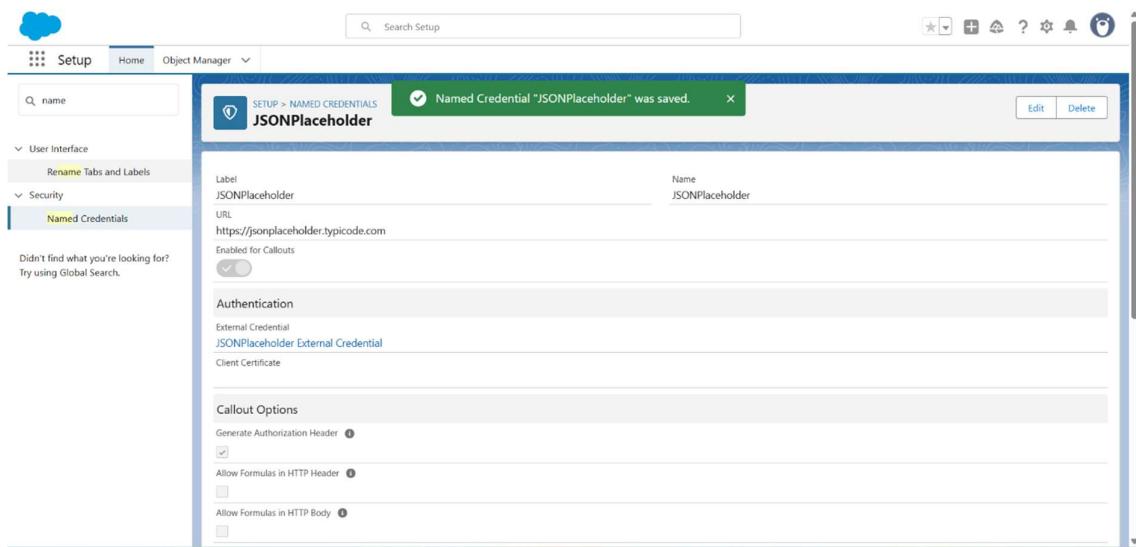


Phase 7: Integration & External Access

1. Named Credentials

- **Where:** Setup → Named Credentials → New.
- Label: JSONPlaceholder.
- URL: <https://jsonplaceholder.typicode.com>.
- Identity Type: Anonymous.
- Save.



2. REST Web Service

- Class: ComplaintRestService.
- Annotated with @RestResource(urlMapping='/complaints/*').
- Test in Postman → <https://yourorg.my.salesforce.com/services/apexrest/complaints/{id}>.
- Auth: OAuth token from Connected App.

workbench  info queries data migration utilities

REST Explorer

JANU NAGESH AT SJB INSTITUTE OF TECHNOLOGY ON API 62.0

Try the Salesforce APIs for Postman.

Choose an HTTP method to perform on the REST API service URI below:

GET POST PUT PATCH DELETE HEAD Headers Reset Up

/services/apexrest/complaints/a02gK000005phxhQAA Execute

[Expand All](#) | [Collapse All](#) | [Show Raw Response](#)

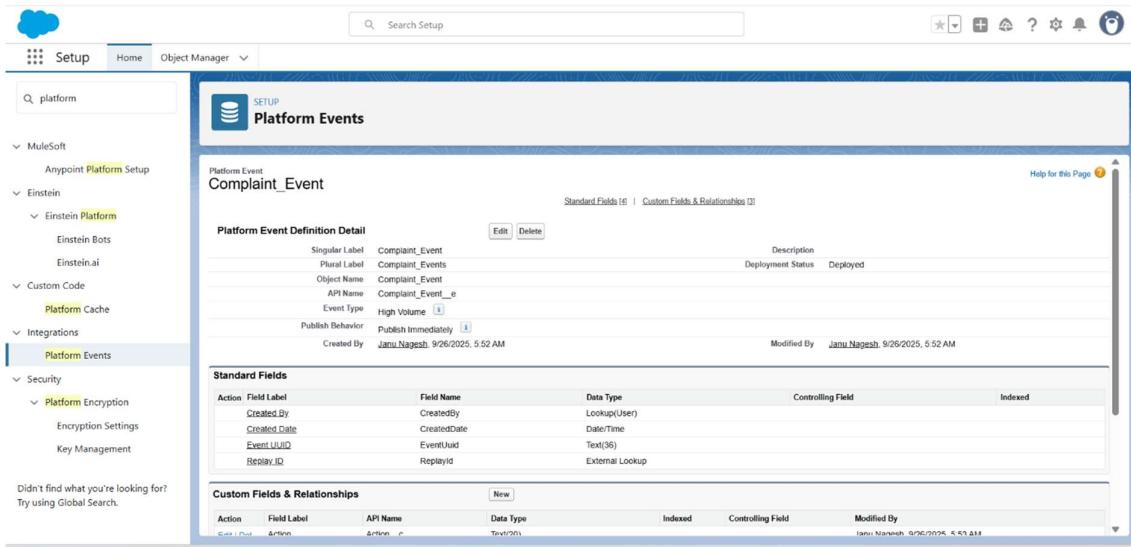
attributes

- ❖ Id: **a02gK000005phxhQAA**
- ❖ Citizen_Name__c: **Pragathi Ram**
- ❖ Department__c: **a03gK00000AS89IQAD**
- ❖ Status__c: **In Progress**

Requested in 0.109 sec
Workbench 62.0.0

3. Platform Events

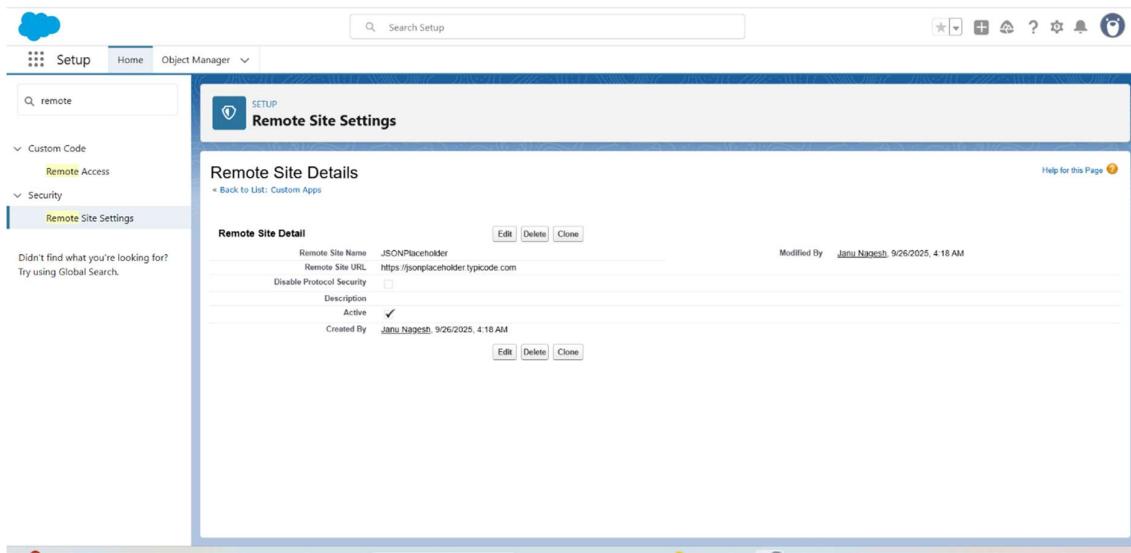
- **Where:** Setup → Platform Events → New.
- **Name:** Complaint_Event.
- **Field:** ComplaintId (Lookup), Status (Text).
- Publish from Apex → subscribe via CometD or Flow.



The screenshot shows the Salesforce Setup interface. On the left, the navigation sidebar includes categories like MuleSoft, Einstein, Custom Code, Integrations, Security, and Platform Encryption. The 'Platform Events' option under Integrations is selected and highlighted in blue. The main content area displays the 'Platform Events' page for the 'Complaint_Event' definition. The 'Platform Event Definition Detail' section shows the event name 'Complaint_Event' and its properties: Singular Label (Complaint_Event), Plural Label (Complaint_Events), Object Name (Complaint_Event), API Name (Complaint_Event_e), Event Type (High Volume), Publish Behavior (Publish Immediately), and Created By (Janu Nagesh). The 'Standard Fields & Relationships' section lists fields such as Created By, Created Date, Event UUID, and Replay ID. The 'Custom Fields & Relationships' section is currently empty.

4. Remote Site Settings

- **Where:** Setup → Remote Site Settings → New.
- Name: JSONPlaceholder.
- URL: <https://jsonplaceholder.typicode.com>.
- Active: .

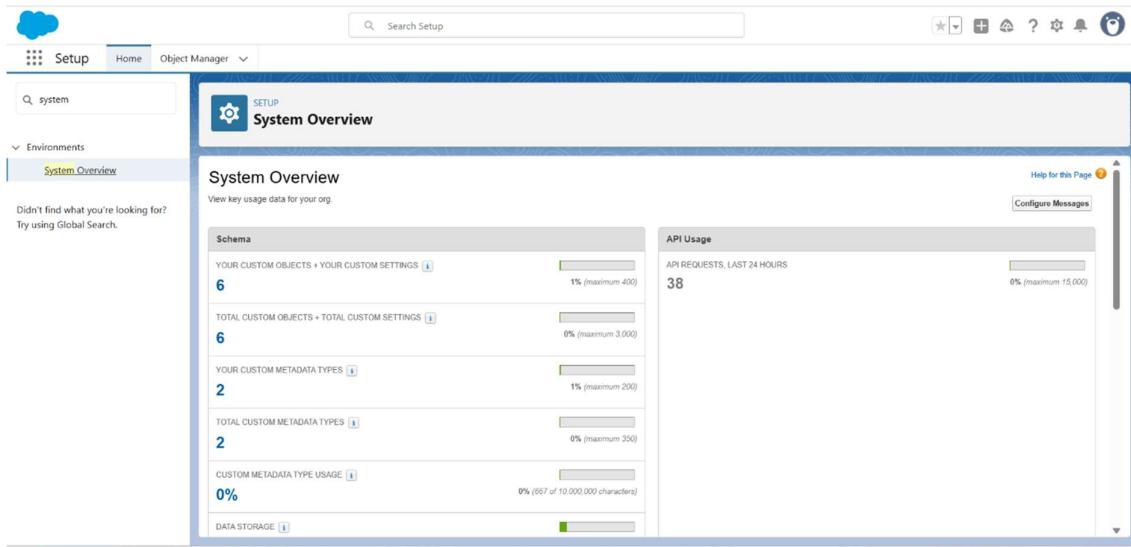


5. API Limits

This isn't setup, it's more of **monitoring**.

Steps

1. Setup → Quick Find → **System Overview**.
 - Shows API calls used today.



6. OAuth & Authentication

We'll create a **Connected App** so external systems (like Postman or a mobile app) can log in.

Steps

1. Setup → Quick Find → **App Manager** → New Connected App.
2. Fill in:
 - **Name:** SmartCity_OAuth_App
 - **API Name:** SmartCity_OAuth_App
 - **Contact Email:** your email.
 - **Check Enable OAuth Settings.**
 - **Callback URL:** For Postman testing, use:
https://oauth.pstmn.io/v1/callback
 - **Selected OAuth Scopes:**
 - Full Access (full)
 - Perform requests on your behalf at any time (refresh_token, offline_access)
 - Access and manage your data (api)
 - Save → you'll get **Consumer Key** and **Consumer Secret**.
3. **Test in Postman**
 - Auth → OAuth 2.0 → Get New Access Token.
 - Token URL:

- <https://login.salesforce.com/services/oauth2/token>
- Client ID = Consumer Key.
- Client Secret = Consumer Secret.

The screenshot shows the Salesforce Setup interface with the following details:

- Page Header:** Search Setup, Home, Object Manager
- Section:** Connected App Name: SmartCityIntegration
- Buttons:** Edit, Delete, Manage, Migrate to External Client App
- Information:**
 - Version: 1.0
 - API Name: SmartCityIntegration
 - Created Date: 9/26/2025, 5:40 AM
 - By: Janu Nagash
 - Contact Email: jahnavigowda912@gmail.com
 - Contact Phone: (Not visible)
 - Last Modified Date: 9/26/2025, 5:40 AM
 - By: Janu Nagash
 - Description: Info URL
- API (Enable OAuth Settings):**
 - Consumer Key and Secret: Manage Consumer Details
 - Selected OAuth Scopes: Full access (full)
 - Callback URL: https://login.salesforce.com/services/oauth2/success
 - Enable for Device Flow:
 - Require Proof Key for Code Exchange (PKCE) Extension for Supported Authorization Flows:
 - Require Secret for Web Server Flow: