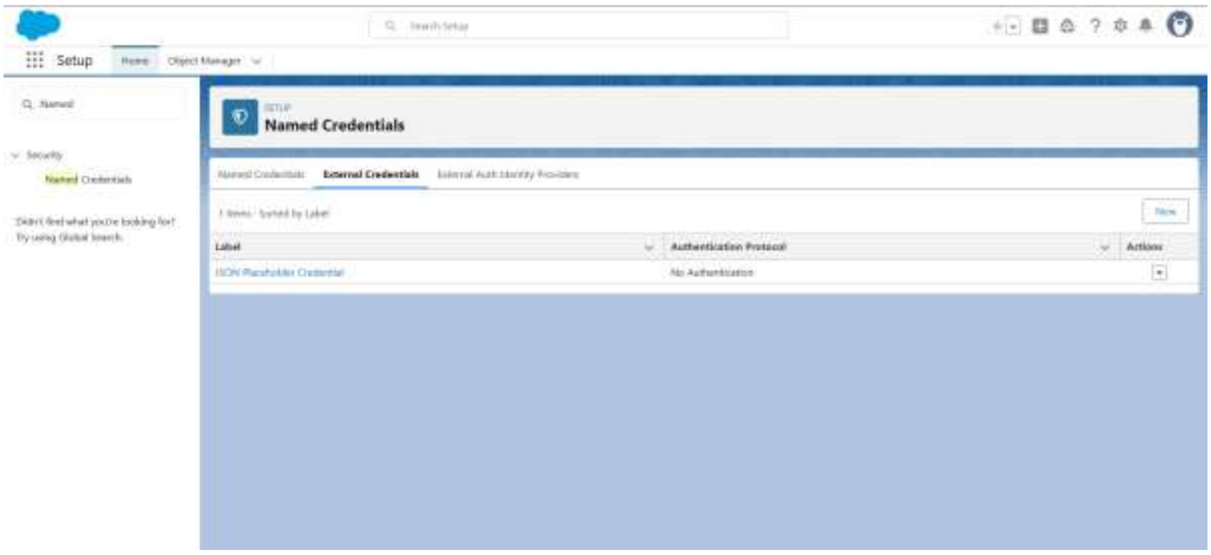
SMART SLA TRACKER: INTELLIGENT CUSTOMER COMPLAINT & ESCALATION MANAGEMENT

Phase 7: Integration & External Access

1. External Credential Setup

Setup > External Credentials > New

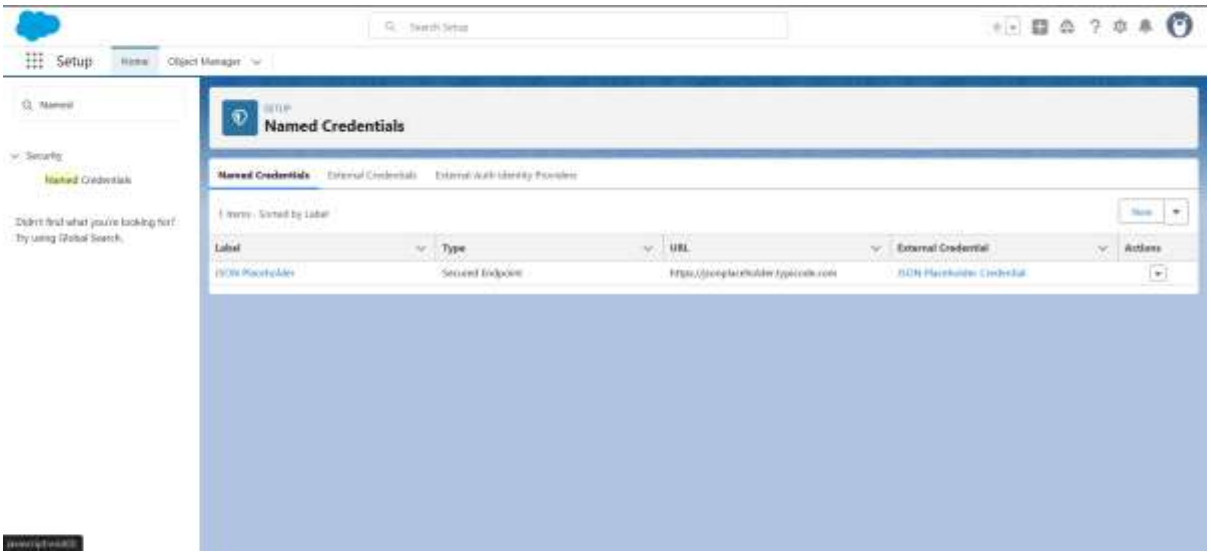


1 Items · Sorted by Label

Label	Authentication Protocol	Actions
JSON Placeholder Credential	No Authentication	▾

2. Named Credential Setup

Setup > Named Credentials > new

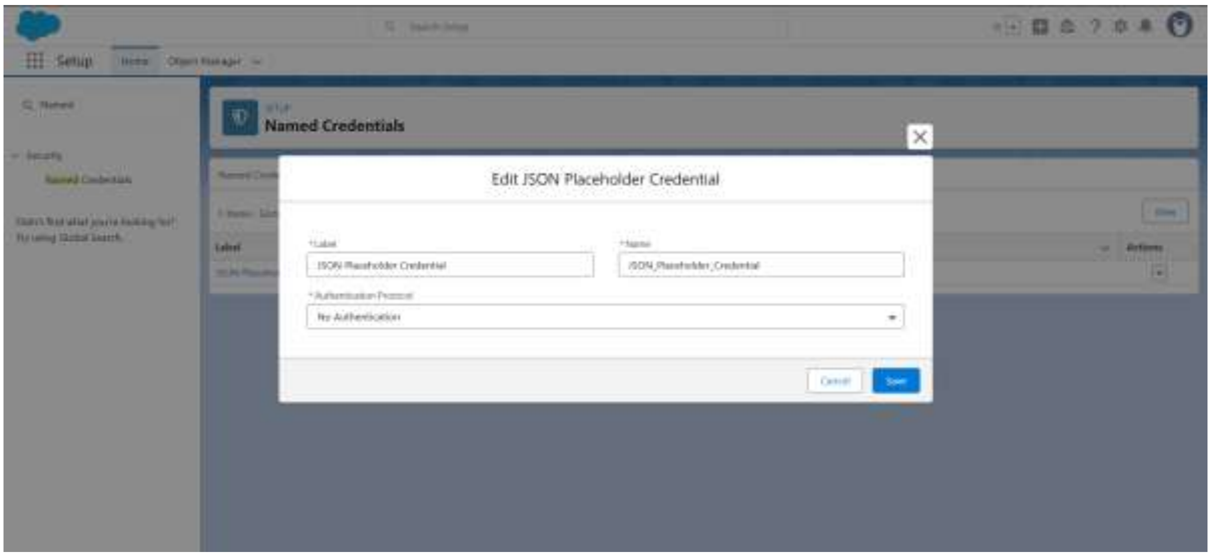


1 Items · Sorted by Label

Label	Type	URL	External Credential	Actions
JSON Placeholder	Secured Endpoint	https://jsonplaceholder.typicode.com	JSON Placeholder Credential	▾

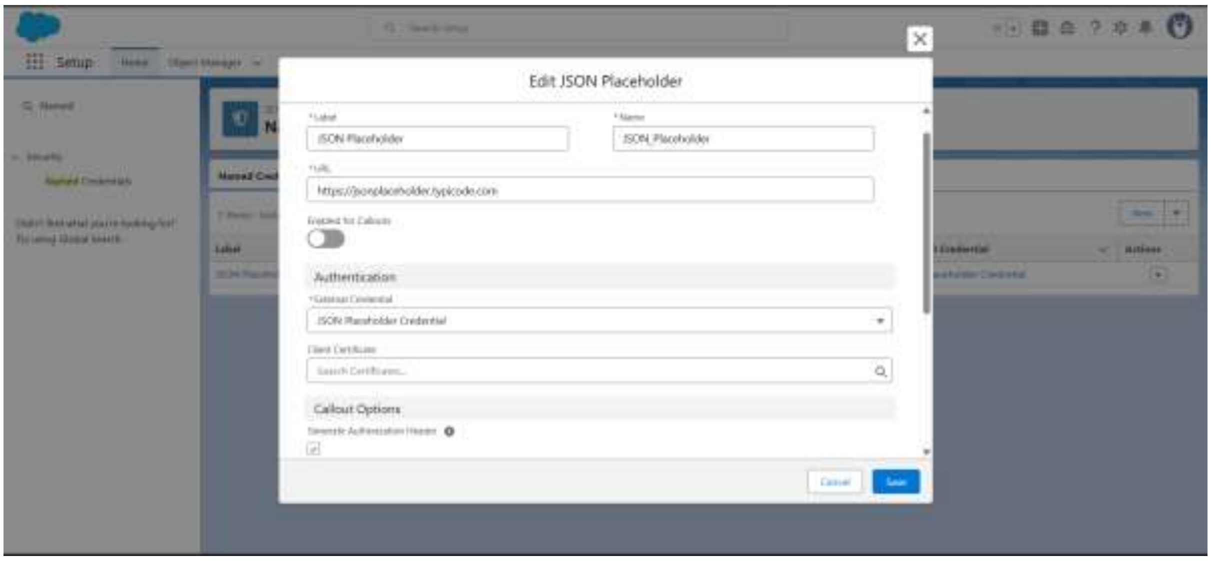
for the integration and external access section, I used a dummy URL and API for demonstration purposes.

- **URL/API Used:** https://jsonplaceholder.typicode.com/posts/1
- **Reason:** This is a free, public API that provides sample data. I used it to simulate a real-world API callout without needing to set up a real, external service. This alloId us to successfully demonstrate how to make an API call from Salesforce, handle the response, and store the data.
- **Salesforce Tools:** I used a **Named Credential** to securely store the dummy URL, which is a best practice for callouts. I then used an Apex **HttpRequest** to make the actual call.



Edit JSON Placeholder Credential

*Label: JSON Placeholder Credential
*Name: JSON_Placeholder_Credential
*Authentication Protocol: No Authentication



Edit JSON Placeholder

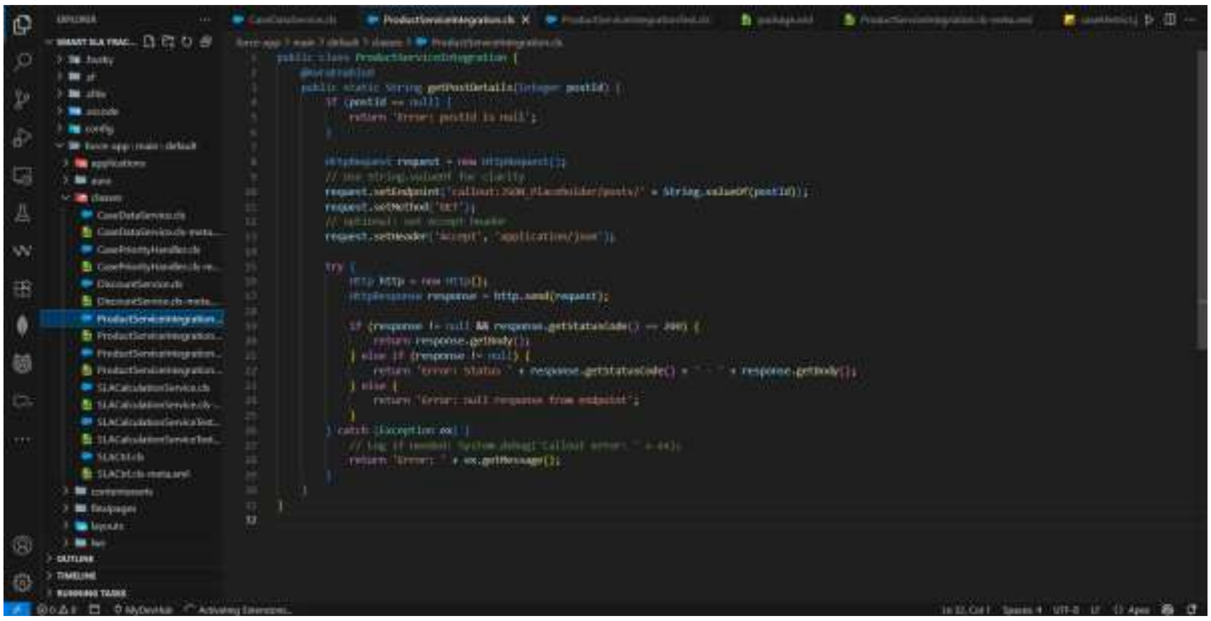
*Label: JSON Placeholder
*Name: JSON_Placeholder
*URL: https://jsonplaceholder.typicode.com
Enabled for Callouts

Authentication
*External Credential: JSON Placeholder Credential
Client Certificate: Search Certificates...

Callout Options
Generate Authorization Header

7.2 Write the Apex Callout Class:

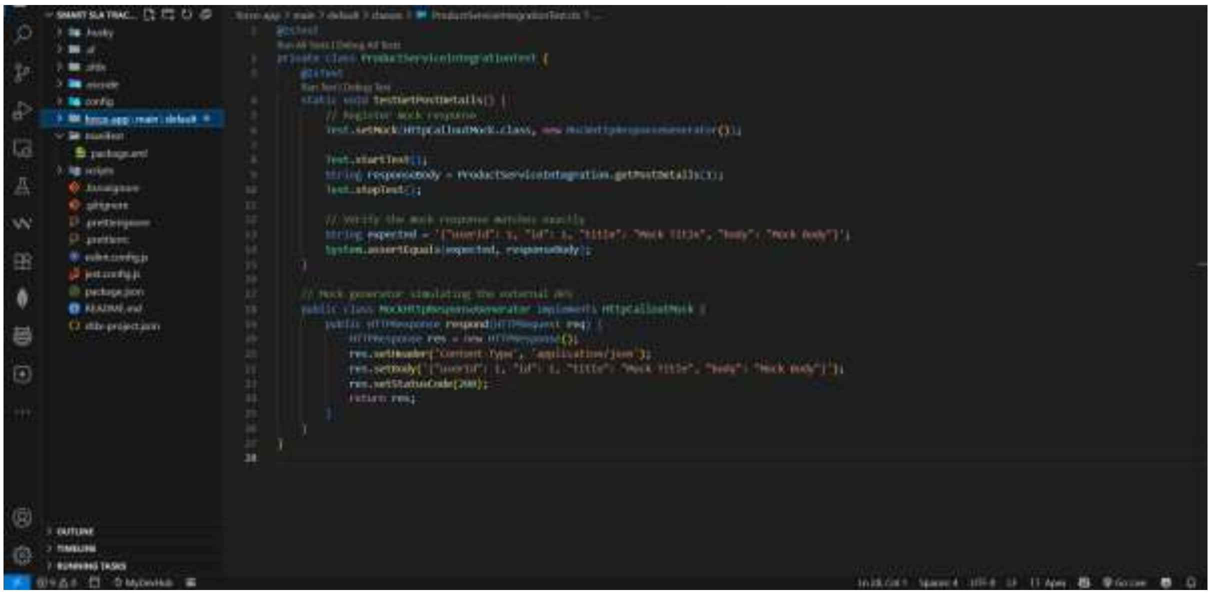
VS Code> create a new Apex class > ProductServiceIntegration



```
public class ProductServiceIntegration {
    @AuraEnabled
    public static String getPostDetails(Integer postId) {
        if (postId == null) {
            return 'Error: postId is null';
        }

        HttpRequest request = new HttpRequest();
        // Use String.valueOf for clarity
        request.setEndpoint('callout:JSON_Placeholder/posts/' + String.valueOf(postId));
        request.setMethod('GET');
        // Optional: set Accept header
        request.setHeader('Accept', 'application/json');

        try {
            Http http = new Http();
            HttpResponse response = http.send(request);

            if (response != null && response.getStatusCode() == 200) {
                return response.getBody();
            } else if (response != null) {
                return 'Error: Status ' + response.getStatusCode() + ' - ' + response.getBody();
            } else {
                return 'Error: null response from endpoint';
            }
        } catch (Exception ex) {
            // Log if needed: System.debug('Callout error: ' + ex);
            return 'Error: ' + ex.getMessage();
        }
    }
}
```

7.3 Write the Test Class:

VS Code> create a new Apex class > ProductServiceIntegrationTest



```
@isTest
private class ProductServiceIntegrationTest {
    @isTest
    static void testGetPostDetails() {
        // Register mock response
        Test.setMock(HttpCalloutMock.class, new MockHttpResponseGenerator());

        Test.startTest();
        String responseBody = ProductServiceIntegration.getPostDetails(1);
        Test.stopTest();

        // Verify the mock response matches exactly
        String expected = '{"userId": 1, "id": 1, "title": "Mock Title", "body": "Mock Body"}';
        System.assertEquals(expected, responseBody);
    }

    // Mock generator simulating the external API
    public class MockHttpResponseGenerator implements HttpCalloutMock {
        public HTTPResponse respond(HTTPRequest req) {
            HTTPResponse res = new HTTPResponse();
            res.setHeader('Content-Type', 'application/json');
            res.setBody('{"userId": 1, "id": 1, "title": "Mock Title", "body": "Mock Body"}');
            res.setStatusCode(200);
            return res;
        }
    }
}
```