

Phase 7: Integration & External Access

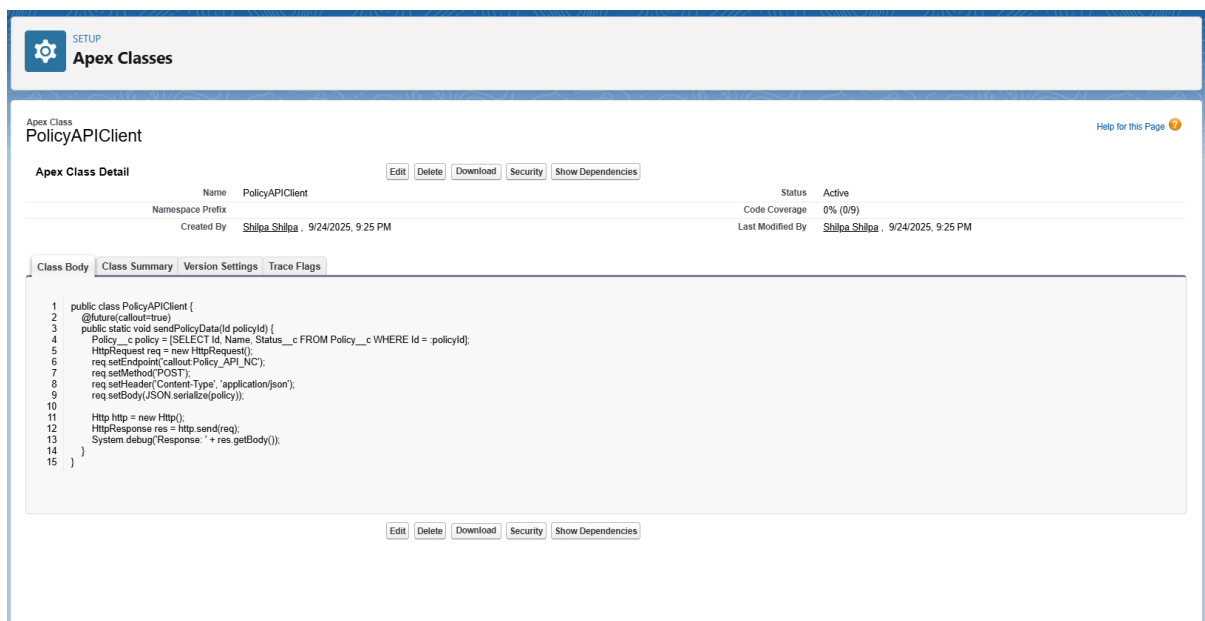
1. Named Credentials

Used to securely store authentication details for external systems that Salesforce needs to connect with. Named Credentials simplify callouts by handling authentication automatically.

2. Web Services (REST / SOAP)

Purpose:

Allow Salesforce to send or receive data to/from external systems. For this project, REST API is used to push policy or claim data.



The screenshot displays the Salesforce Apex Class editor for a class named `PolicyAPIClient`. The interface includes a top navigation bar with a 'SETUP' icon and the text 'Apex Classes'. Below this, the class name 'PolicyAPIClient' is shown with a 'Help for this Page' link. The 'Apex Class Detail' section provides metadata: Name (PolicyAPIClient), Namespace Prefix, Created By (Shilpa Shilpa), Created Date (9/24/2025, 9:25 PM), Status (Active), Code Coverage (0% (0/9)), and Last Modified By (Shilpa Shilpa, 9/24/2025, 9:25 PM). The 'Class Body' tab is selected, showing the following Apex code:

```
1 public class PolicyAPIClient {
2     @future(callout=true)
3     public static void sendPolicyData(Id policyId) {
4         Policy__c policy = (SELECT Id, Name, Status__c FROM Policy__c WHERE Id = :policyId);
5         HttpRequest req = new HttpRequest();
6         req.setEndpoint('callout:Policy_API_NC');
7         req.setMethod('POST');
8         req.setHeader('Content-Type', 'application/json');
9         req.setBody(JSON.serialize(policy));
10
11         Http http = new Http();
12         HttpResponse res = http.send(req);
13         System.debug('Response: ' + res.getBody());
14     }
15 }
```

3. Callouts

Purpose:

Callouts allow Salesforce to send HTTP requests to external systems and receive responses. Named Credentials handle authentication automatically.

Steps Performed:

1. Enabled **Callouts** in Apex by adding `@future(callout=true)` or in Scheduled/Queueable Apex.
2. Used **HttpRequest & HTTP classes** in Apex to send/receive data.
3. Validated successful callout execution by checking **Debug Logs** in Developer Console.