

## Phase 7: Integration & External Access

---

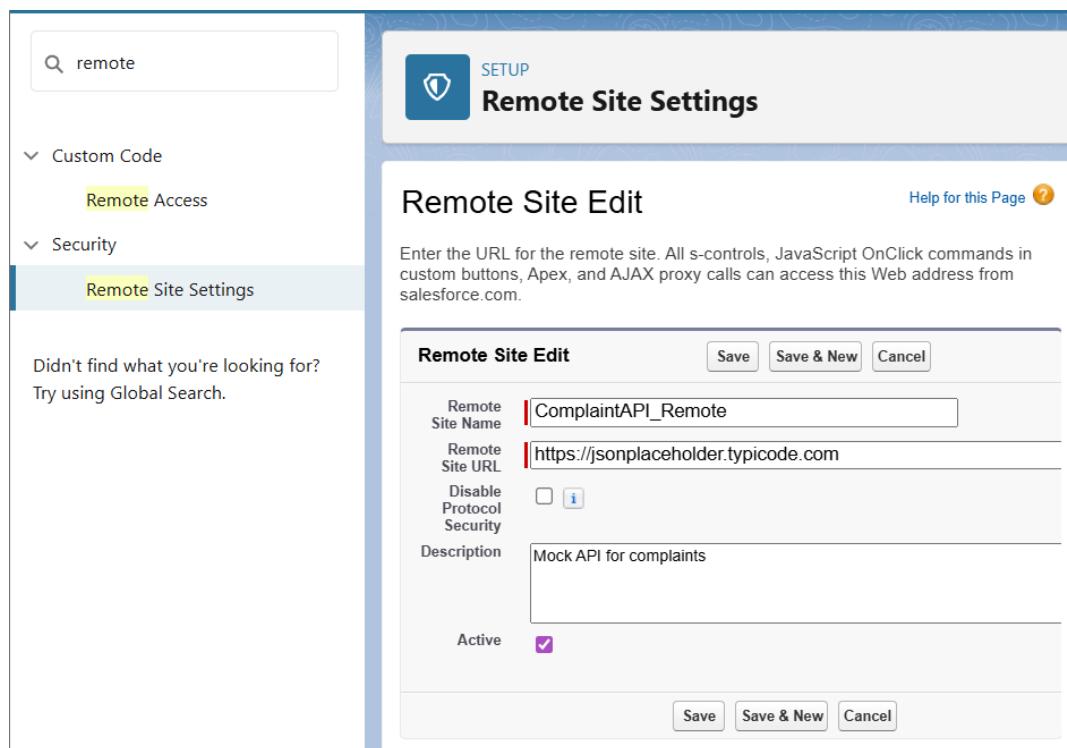
### 1. Remote Site Settings

#### Purpose:

Allows Salesforce to send outbound HTTP callouts to external APIs by whitelisting their domains.

#### Steps:

1. Setup → Quick Find → **Remote Site Settings** → New
2. Fill:
  - o Remote Site Name: ComplaintAPI\_Remote
  - o Remote Site URL: <https://jsonplaceholder.typicode.com>
  - o Description: Mock API for complaints
3. Save



---

### 2. Named Credentials

#### Purpose:

Store endpoint URLs and authentication in one place for callouts.

#### Steps:

1. Setup → Quick Find → **Named Credentials** → New
2. Fill:

- Label / Name: Complaint\_new
- URL: <https://jsonplaceholder.typicode.com/posts>
- Enabled for Callouts:
- Authentication: Link to **External Credential**

### 3. Save

#### Usage in Apex:

```
HttpRequest req = new HttpRequest();
req.setEndpoint('callout:Complaint_new');
req.setMethod('GET');
```

### New Named Credential

\* Label  
Complaint\_new

\* Name  
Complaint\_new

\* URL  
<https://jsonplaceholder.typicode.com/posts>

Enabled for Callouts

**Authentication**

\* External Credential

Select an Option

Client Certificate **NoAuthCredential**

Search Certifica

**Callout Options**

Generate Authorization Header

### 3. External Credentials

#### Purpose:

Handle authentication (OAuth, Basic, or No Auth) for Named Credentials.

**Steps:**

1. Setup → Quick Find → **External Credentials** → New
2. Fill:
  - Label: NoAuthCredential
  - Identity Type: Named Principal
  - Authentication Protocol: No Authentication
3. Assign access to **System Administrator Profile**
4. Save

The screenshot shows the 'External Credentials' setup page. At the top, there's a shield icon, the text 'SETUP > NAMED CREDENTIALS', the name 'NoAuthCredential', and 'Edit' and 'Delete' buttons. Below this, there are two sections: 'Label' (NoAuthCredential) and 'Name' (NoAuthCredential), separated by a horizontal line. Under 'Authentication Protocol', it says 'No Authentication'. A 'Managed Package Access' section is present. Below that, 'Created By Namespace' is listed with a help icon. At the bottom, there's a 'Related Named Credentials' section with a table showing one entry: 'ComplaintAPI' (Label), 'ComplaintAPI' (Name), and 'https://jsonplaceholder.typicode.com' (URL). Navigation arrows are at the bottom of this section.

---

**4. External Services**

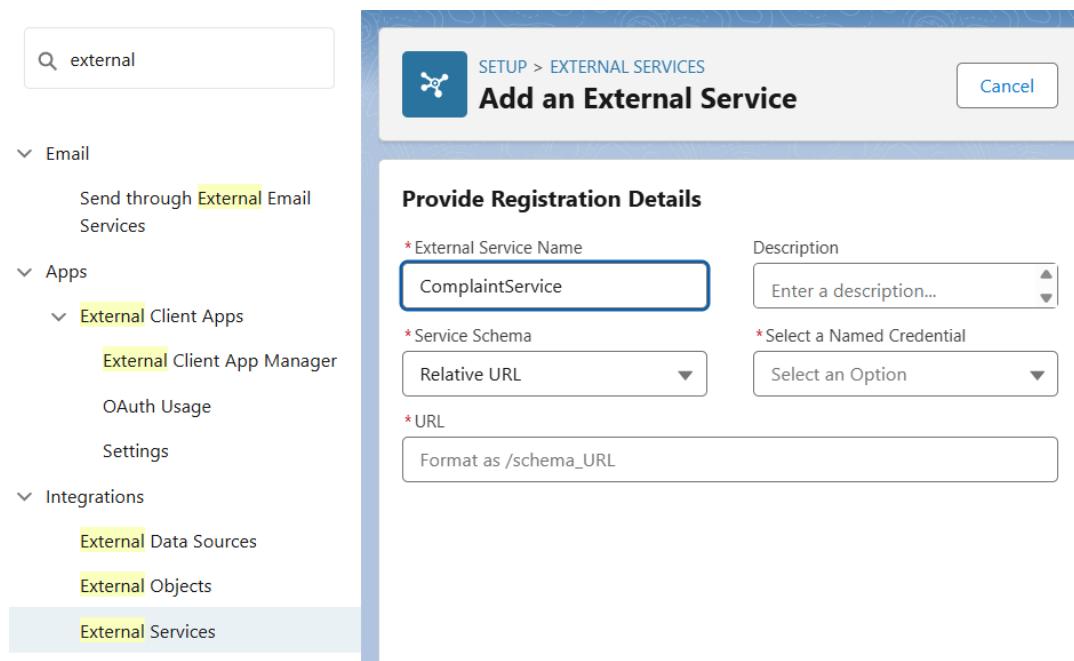
**Purpose:**

Expose APIs (REST/OpenAPI or SOAP/WSDL) in Salesforce without custom code.

**Steps:**

1. Setup → Quick Find → **External Services** → New
2. Register with:
  - Named Credential (endpoint)
  - OpenAPI schema (for REST) or WSDL (for SOAP)

3. Salesforce generates **Apex actions** that can be used in **Flow or Process Builder**.



## 5. Web Services (REST/SOAP)

### Purpose:

Expose Salesforce functionality as a service OR consume external services.

### REST Example (Expose Salesforce):

```
@RestResource(urlMapping='/students/*')
global with sharing class StudentAPI {
    @HttpGet
    global static Student__c getStudent() {
        RestRequest req = RestContext.request;
        String id = req.requestURI.substring(req.requestURI.lastIndexOf('/')+1);
        return [SELECT Id, Name FROM Student__c WHERE Id = :id];
    }
}
```

### SOAP Example (Expose Salesforce):

- Create an **Apex Class → Generate WSDL**.
- Consumers can call using SOAP clients.

## 6. Callouts

### Purpose:

Fetch or send data to external APIs from Salesforce.

### Example Apex:

```
public class ComplaintCallout {  
    public static void fetchComplaints() {  
        HttpRequest req = new HttpRequest();  
        req.setEndpoint('callout:Complaint_new');  
        req.setMethod('GET');  
        Http http = new Http();  
        HttpResponse res = http.send(req);  
        System.debug(res.getBody());  
    }  
}
```

### Steps to Run:

1. Developer Console → Debug → Execute Anonymous
2. Run: ComplaintCallout.fetchComplaints();
3. Check logs → USER\_DEBUG → See API response

---

```
public class ComplaintCallout {  
    public static void fetchComplaints() {  
        HttpRequest req = new HttpRequest();  
        req.setEndpoint('callout:Complaint_new'); /  
        req.setMethod('GET');  
  
        Http http = new Http();  
        HttpResponse res = http.send(req);  
  
        System.debug(res.getBody());  
    }  
}
```

---