

PHASE - 7

Project: AI-Powered HR & Employee Management Bot

Remote Site Settings (Foundation – MUST DO FIRST)

Salesforce **blocks all external callouts by default.**

Step-by-step

1. Go to **Setup → Remote Site Settings**
2. Click **New Remote Site**
3. Fill:
 - **Remote Site Name:** HR_External_API
 - **Remote Site URL:**
Example:
 - https://api.hr-external.com
4. Save

📌 Why needed?

Any Apex callout (REST, AI API, attendance device) will fail without this.

Named Credentials (Secure API Access – MOST IMPORTANT)

Instead of hardcoding URLs, tokens, usernames → use **Named Credentials**.

Why Named Credentials?

- No credentials in Apex
- Auto token refresh
- Required for real projects & interviews

Step-by-step (Basic API Auth Example)

1. **Setup → Named Credentials**
2. Click **New Legacy** (or New → External Credential if enabled)
3. Fill:
 - **Label:** HR External API
 - **Name:** HR_External_API
 - **URL:**
 - https://api.hr-external.com
 - **Authentication:** Password / OAuth (choose based on API)

4. Save

✅ Salesforce now securely stores credentials.

Callouts from Apex (REST Call)

Use case (Realistic)

Fetch employee verification status from an external HR system.

Apex Example: REST Callout

```
public with sharing class HRExternalService {  
  
    public static String verifyEmployee(String employeeId) {  
  
        HttpRequest req = new HttpRequest();  
  
        req.setEndpoint('callout:HR_External_API/verifyEmployee?id=' + employeeId);  
  
        req.setMethod('GET');  
  
  
        Http http = new Http();  
  
        HttpResponse res = http.send(req);  
  
  
        if (res.getStatusCode() == 200) {  
  
            return res.getBody();  
  
        } else {  
  
            throw new CalloutException('Failed to verify employee');  
  
        }  
  
    }  
  
}
```

🔴 Uses **Named Credential** automatically

🔴 No token or URL in code

External Services (No-Code API Integration)

If an external system provides **OpenAPI / Swagger JSON**, Salesforce can auto-generate Apex.

Step-by-step

1. **Setup → External Services**
2. New External Service
3. Upload **Swagger/OpenAPI file**
4. Salesforce auto-creates:
 - Apex classes
 - Methods for callouts

📌 Useful if HR system provides official API docs

📌 Not mandatory for your project (optional)

Web Services (Expose Salesforce as REST API)

Now you expose **Salesforce data to outside apps** (Mobile App, AI Bot).

REST API Example – Leave Requests

@RestResource(urlMapping='/leave/*')

global with sharing class LeaveRESTService {

@HttpGet

global static List<Leave__c> getLeaves() {

return [

SELECT Id, Employee__c, Start_Date__c, End_Date__c, Approval_Status__c

FROM Leave__c

LIMIT 50

];

}

@HttpPost

global static String createLeave(Leave__c leave) {

insert leave;

```
        return 'Leave Created Successfully';  
    }  
}
```

Endpoint:

/services/apexrest/leave

📌 Used by:

- Mobile App
- AI Chatbot
- External HR portals

OAuth & Authentication (Secure External Access)

External apps **must authenticate** before accessing Salesforce APIs.

Step-by-step

1. **Setup → App Manager**
2. New **Connected App**
3. Enable **OAuth Settings**
4. Select scopes:
 - Access Salesforce APIs
 - Perform requests on your behalf
5. Save → Get:
 - **Client ID**
 - **Client Secret**

📌 Required for:

- Mobile apps
- Node / Python / PHP backend
- AI agents

Platform Events (Real-Time Notifications)

Used for **event-driven architecture**.

Use case

Notify external system when Leave is Approved.

Step 1: Create Platform Event

1. Setup → Platform Events → New

2. Name: **Leave_Approved_Event__e**

3. Fields:

- Employee_Id__c (Text)
- Leave_Id__c (Text)
- Status__c (Text)

Step 2: Publish Event from Apex

```
Leave_Approved_Event__e event = new Leave_Approved_Event__e(
```

```
    Employee_Id__c = leave.Employee__c,
```

```
    Leave_Id__c = leave.Id,
```

```
    Status__c = 'Approved'
```

```
);
```

```
EventBus.publish(event);
```

📌 External systems can subscribe via CometD.

Change Data Capture (CDC)

Automatically tracks **data changes** without triggers.

Step-by-step

1. Setup → Change Data Capture

2. Enable for:

- Employee__c
- Leave__c
- Attendance__c

3. Save

📌 Now Salesforce emits events on:

- Insert
- Update
- Delete

Used for:

- Sync external databases
- Analytics pipelines

Salesforce Connect (External Objects)

Access **external DB data** without storing in Salesforce.

- 📌 Optional for your project
- 📌 Example: Attendance from biometric device DB

Steps (High level):

1. Setup → Salesforce Connect
2. Create External Data Source
3. Create External Objects

10 API Limits (Very Important for Viva & Interviews)

Salesforce has **daily API limits**.

Check Limits

1. Setup → System Overview
2. See:
 - API Requests Used
 - Daily Limit

📌 Use:

- @AuraEnabled(cacheable=true)
 - Batch Apex
 - Platform Events
- to reduce API usage