Phase 7: Integration & External Access

PROJECT TITLE:-

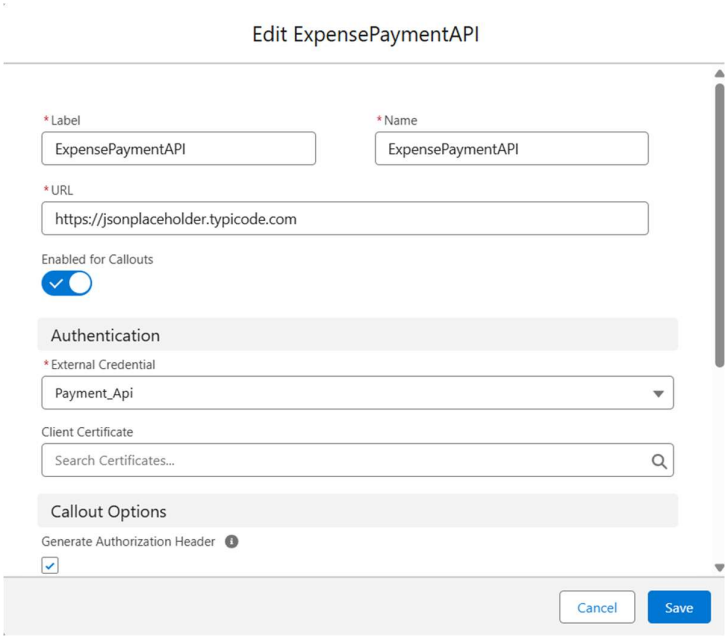
Expense On a Page: An expense approval & insight system.
Industry: Finance / Corporate Expense Management.
Target User: Employees, Managers, and Finance Teams.

NAMED CREDENTIALS:-

The project *Expense on a Page* is not connected to any external API, so there isn't any real use of Named Credentials.
Making a Named Credentials just for academic purpose that will **store the external Payment website Login credential.**

Creating a Named Credential:

Label: ExpensePaymentAPI
Name: ExpensePaymentAPI
URL: https://jsonplaceholder.typicode.com (Testing URL)
External Credential: Label- Payment_Api
Name- Payment_Api
Authentication Protocol- Basic Authentication
SaveCallout Options: Generate Authorization Header
Save.

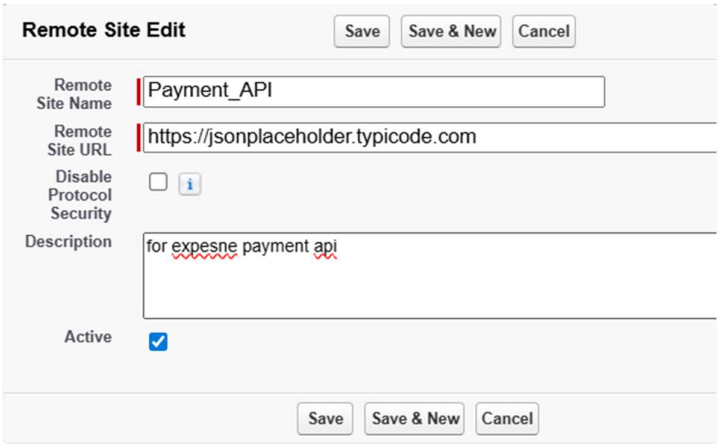


REMOTE SITE SETTINGS:-

As we are working with the Named Credential which are used as callouts, *we might all have to make remote site setting so it can call payment Api.*

Creating the Remote Site Setting:

Remote Site Name: Payment_API
Remote Site URL: https://jsonplaceholder.typicode.com
Save



CALLOUTS:-

Writing the final apex callout code for the Named Credential and Remote Site Settings we made-

Apex Anonymous Window Code:

```
Http http = new Http();
HttpRequest req = new HttpRequest();


req.setEndpoint('https://jsonplaceholder.typicode.com/posts/1');
req.setMethod('GET');

HttpResponse res = http.send(req);
System.debug('Response: ' + res.getBody());
```

Enter Apex Code

```
1  Http http = new Http();
2  HttpRequest req = new HttpRequest();
3
4  // Use Named Credential instead of full URL
5  req.setEndpoint('https://jsonplaceholder.typicode.com/posts/1');
6  req.setMethod('GET');
7
8  HttpResponse res = http.send(req);
9  System.debug('Response: ' + res.getBody());
10
```

Apex Anonymous Window Output:

Response: {
“userId” : 1,
“id” : 1,
“title” : "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
"body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
}

Timestamp	Event	Details
01:08:57:097	USER_DEBUG	[9]|DEBUG|Response: {
01:08:57:000	USER_DEBUG	"userId": 1,
01:08:57:000	USER_DEBUG	"id": 1,
01:08:57:000	USER_DEBUG	"title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
01:08:57:000	USER_DEBUG	"body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas
01:08:57:000	USER_DEBUG	}

API LIMITS:-

The API limits are usually used so they can monitor the usage of the org over a given time period, which can be done manually-

By going to the company Information we can find the current usage in last 24 hrs and what is the max limit

For EG:
for the Expense on Page org:
The API Usage is : 244
The Max Api Usage is: 15,000

API Requests, Last 24 Hours	244 (15,000 max)

WEB SERVICES (REST/SOAP):-

These are the API that are used or built in the salesforce are called externally.
The Web Services are of two types-

- *REST API (Representational State Transfer Application Programming Interface)*: It is used when an external app wants to fetch or submit record to the salesforce
- *SOAP (Simple Object Access Protocol)*: It is rarely used now days as they are used only for integrating older enterprises system

In the project Expense on a Page we do not have any external APIs connected so will not be using it now but we will use it in future enhancement.

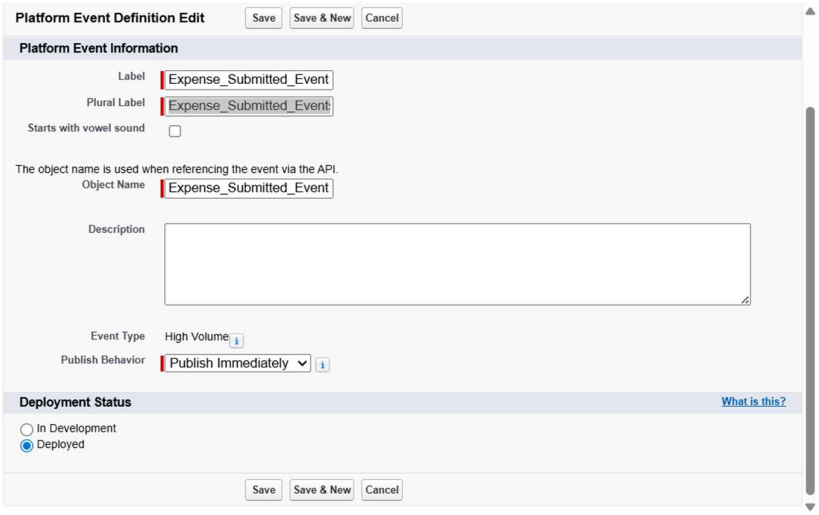
PLATFORM EVENTS:-

Platform Events are event driven architecture in salesforce. It is useful for real-time notification, Decoupled integration, and async processing

Using it in the project to notify the finance team when a record is submitted and to notify the employee if the Expense is approved.

Creating a new Platform Event:

Label: Expense_Submitted_Event
Plural Label: Expense_Submitted_Events
Object Name: Expense_Submitted_Event
Publish Behaviour: Publish Immediately
SAVE



Creating Custom Fields & Relationships for a Platform Event:

Field One-

Data Type: Text
Field Label: ExpenseId
Length: 18
Field Name: ExpenseId

Field Two-

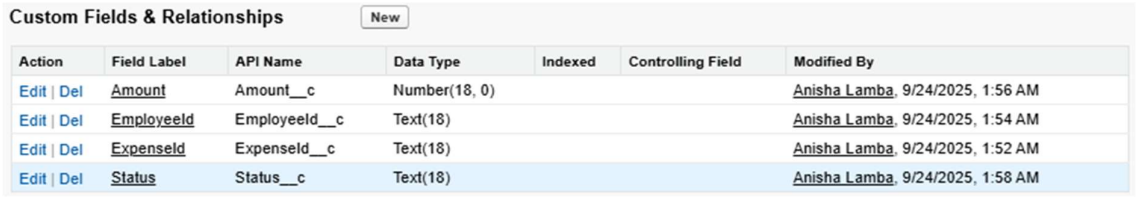
Data Type: Text
Field Label: EmployeeId
Length: 18
Field Name: EmployeeId

Field Three-

Data Type: Number
Field Label: Amount
Length: 18
Field Name: Amount

Field Four-

Data Type: Text
Field Label: Status
Length: 18
Field Name: Status



Custom Fields & Relationships | New

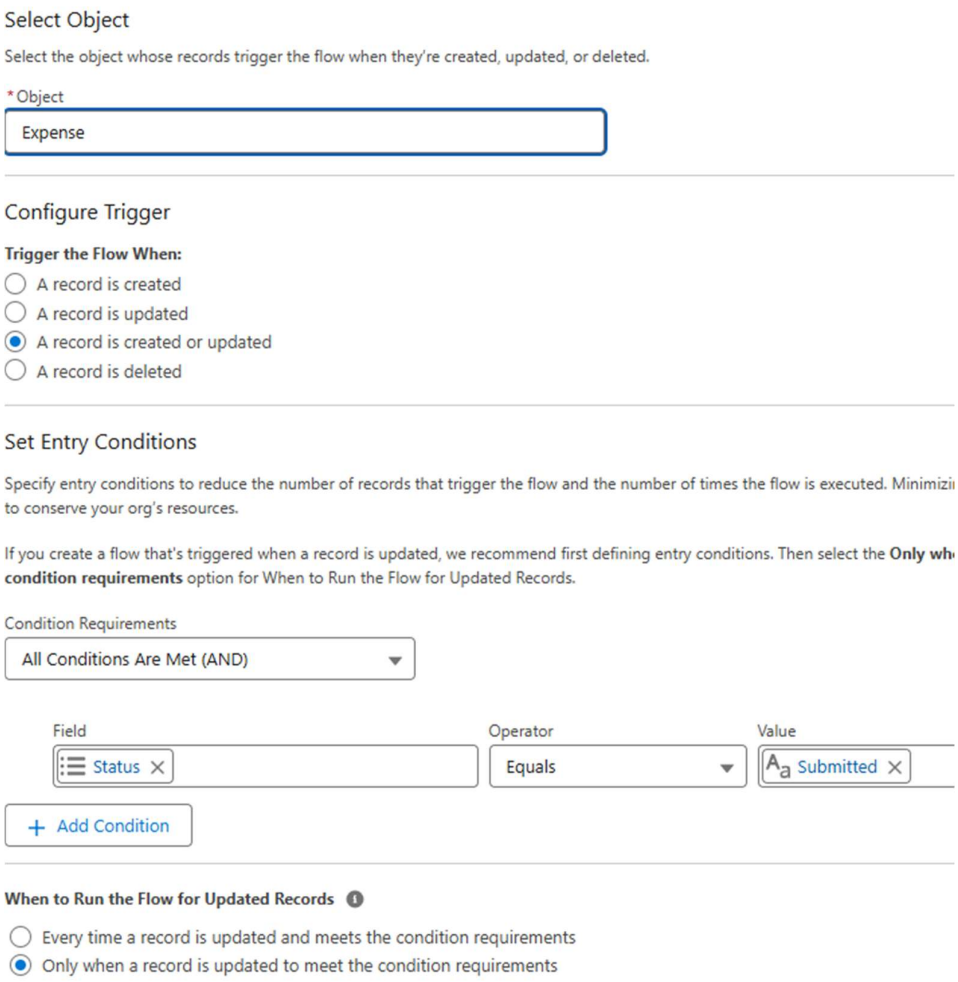
Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Edit | Del	Amount	Amount__c	Number(18, 0)			Anisha Lamba, 9/24/2025, 1:56 AM
Edit | Del	EmployeeId	EmployeeId__c	Text(18)			Anisha Lamba, 9/24/2025, 1:54 AM
Edit | Del	ExpenseId	ExpenseId__c	Text(18)			Anisha Lamba, 9/24/2025, 1:52 AM
Edit | Del	Status	Status__c	Text(18)			Anisha Lamba, 9/24/2025, 1:58 AM

Publishing the Event with the help of a flow:

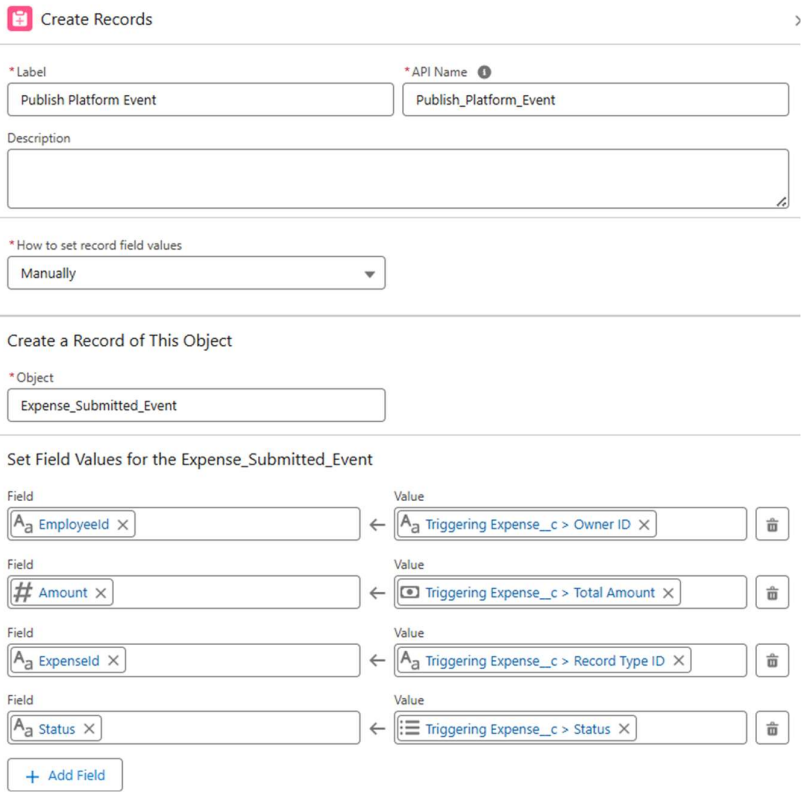
Creating a flow:

- *New Automation Type:* Record-Triggered flow
- *Configure Start-*
  Object: Expense
  Configure Trigger: A record is created or updated
  Optimize Flow: Actions and Related Records (After Save)

Set Entry Condition: Condition Requirement- All Conditions Are Met (AND)
Condition 1: Field: Status
Operator: Equals
Value: Submitted
When to Run the Flow for Updated Records: Only when a record is updated to meet the condition requirements.
Optimize the Flow: Action and Related Records

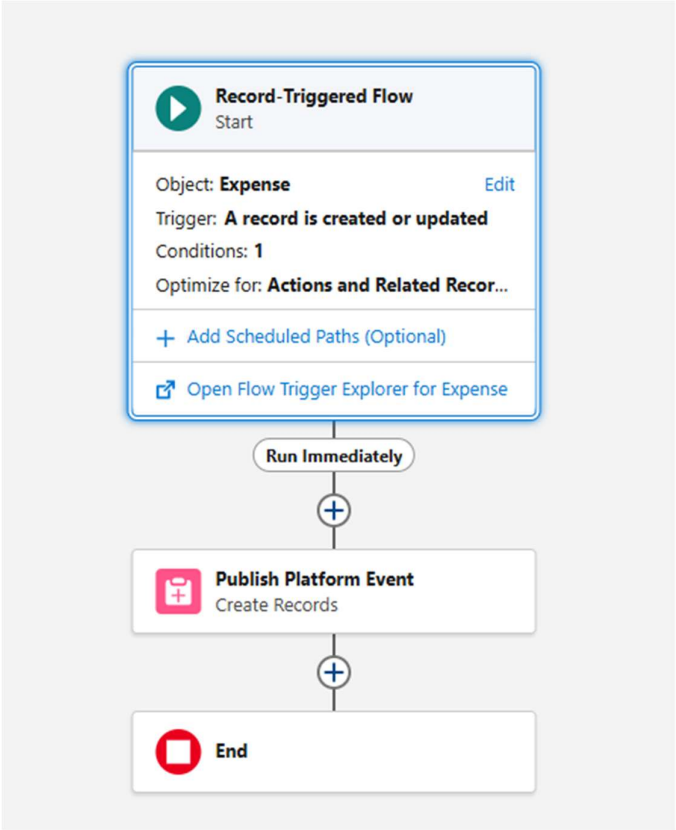


- *Create Record Element-*
  Label: Publish Platform Event
  API Name: Publish_Platform_Event
  How to set record field values: Manually
  Object: Exoense_Submitted_Event
  Set Field Values:
  Value 1- EmployeeId<- Triggering Expense_c > Owner Id
  Value 2- Amount<- Triggering Expense_c > Total Amount
  Value 3- ExpenseId<- Triggering Expense_c > Record Type ID
  Value 4- Status<- Triggering Expense_c > Status



Save the Flow:

Flow Label: publish platform event
*Flow API Name: p*ublish_platform_event
ACTIVATE IT



EXTERNAL SERVICE:-

We do not need External services in our project as our project is not connected to any External API.

CHANGED DATA CAPTURE:-

It is to send real time changes to subscriber which we do not have any in the project so we do not need it.

SALESFORCE CONNECT:-

We have no connection to any external data so do not need this in real life too.

OAUTH & AUTHENTICATION:-

We already have a sample login system using Named Credential.