**Expense Tracker Application - Phase 4: Business Process Automation**

**Objective:** To implement the business logic and automation required for a functional expense management process. This phase focuses on ensuring data integrity with validation rules, automating the manager approval process, and creating a guided user experience with flows.

**Step 1: Implement Validation Rules**

First, we will create rules to ensure high-quality data is entered *before* a record is saved.

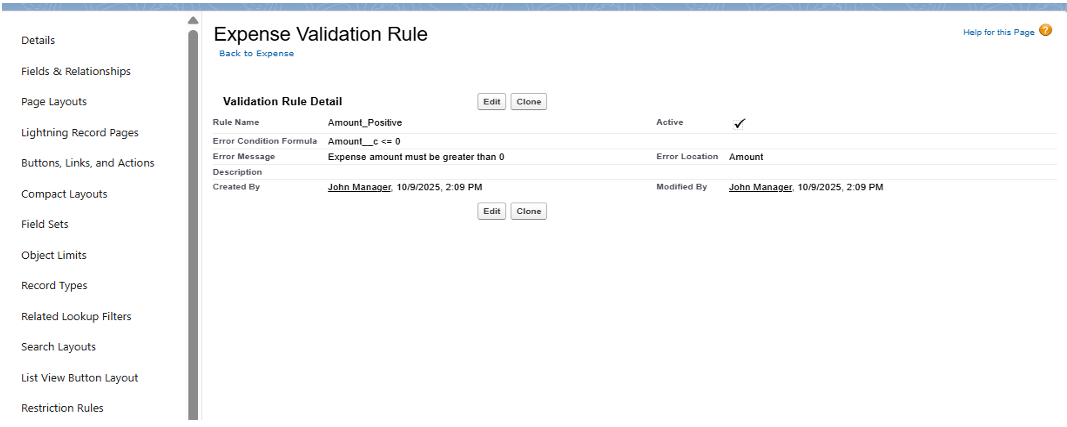
* **Purpose:** To enforce data quality standards, such as ensuring all expense amounts are positive and dates are not set in the future.
* **Navigation:** Go to **Setup** → **Object Manager** → **Expense** → **Validation Rules** → **New**.

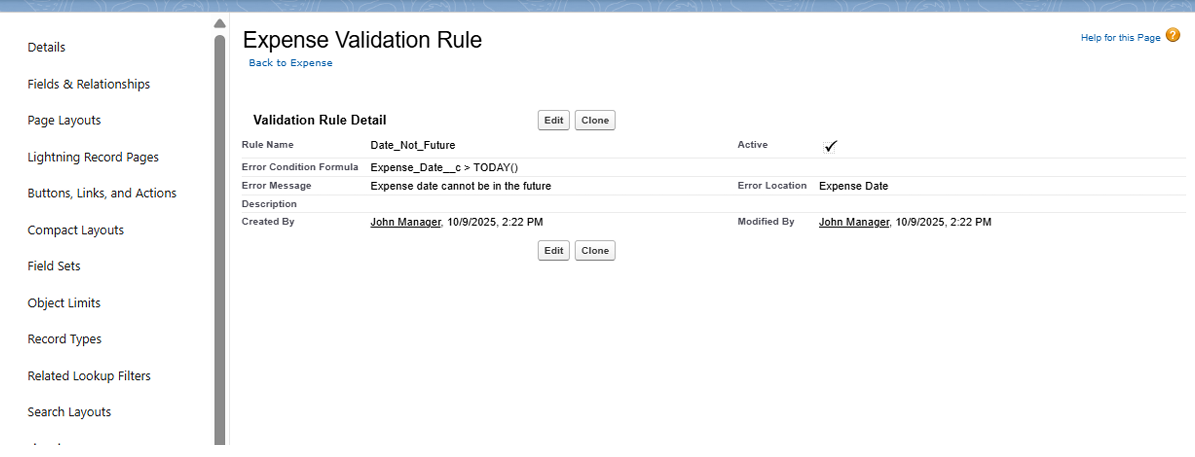
**Rule A: Ensure Positive Amount**

* **Rule Name:** Amount\_Positive
* **Error Condition Formula:** Amount\_\_c <= 0
* **Error Message:** "Expense amount must be greater than 0"
* **Error Location:** Select the Amount field.

**Rule B: Prevent Future Expense Dates**

* **Rule Name:** Date\_Not\_Future
* **Error Condition Formula:** Expense\_Date\_\_c > TODAY()
* **Error Message:** "Expense date cannot be in the future"
* **Error Location:** Select the Expense Date field.



****

**Step 2: Create a Guided Submission with a Screen Flow**

We will create a user-friendly wizard to guide employees through submitting a new expense.

* **Purpose:** To provide a simple, guided form for employees, ensuring all required information is captured in a sequential flow.
* **Navigation:** Go to **Setup** → **Flow** → **New Flow** → **Screen Flow**.

**Flow Structure:**

1. **Screen Element:** Create a screen titled "Submit Expense Form".
   * Add input components for Amount (Currency, Required) , Expense Date (Date, Required) , Category (Picklist) , and Description (Text Area).
2. **Create Records Element:** Add this element after the screen.
   * **Label:** "Create Expense Record".
   * **How to Set Record Fields:** Select "Use separate resources and literal values."
   * **Map Fields:** Map the screen components to the fields on your Expense\_\_c object.
   * **Set Employee:** Critically, map the Employee\_\_c field to the global variable \$User.Id. This automatically stamps the record with the current user's ID.
3. **Connections:** Connect the **Start** element to the **Screen**, and the **Screen** to the **Create Records** element.
4. **Save and Activate** the flow.

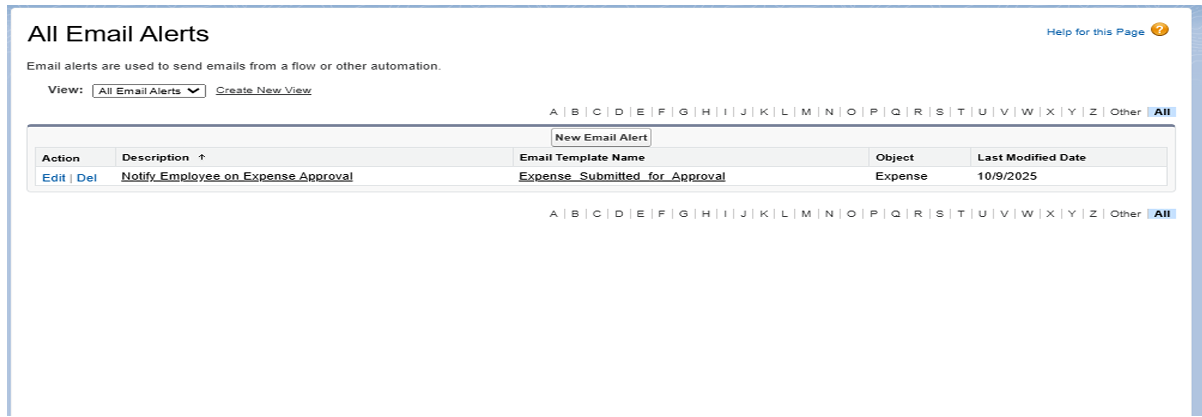
**Step 3: Build the Manager Approval Process**

This process will automatically route expenses for approval based on the amount.

* **Purpose:** To ensure that expenses exceeding a certain monetary threshold receive manager approval before being processed.
* **Navigation:** Go to **Setup** → **Approval Processes** → Select the **Expense** object → **Create New Approval Process** → **Use Standard Setup Wizard**.

**Configuration:**

* **Name:** Expense Manager Approval.
* **Entry Criteria:** Set the criteria to trigger this process, for example: Amount\_\_c > 5000.
* **Approver:** Choose "Manager" as the approver. The system will automatically use the Manager field on the submitter's User record.
* **Approval Actions:** Add a **Field Update** action to change the Approval\_Status\_\_c field to 'Approved'.
* **Rejection Actions:** Add a **Field Update** action to change the Approval\_Status\_\_c field to 'Rejected'.
* **Email Alerts:** (Optional) You can configure email alerts here to notify the manager of a new request and the submitter of the outcome.
* **Activate:** Once all steps are configured, **Activate** the approval process.



**Step 4: Automate Status Updates with a Record-Triggered Flow**

This flow will run in the background to update a record *after* it has been approved.

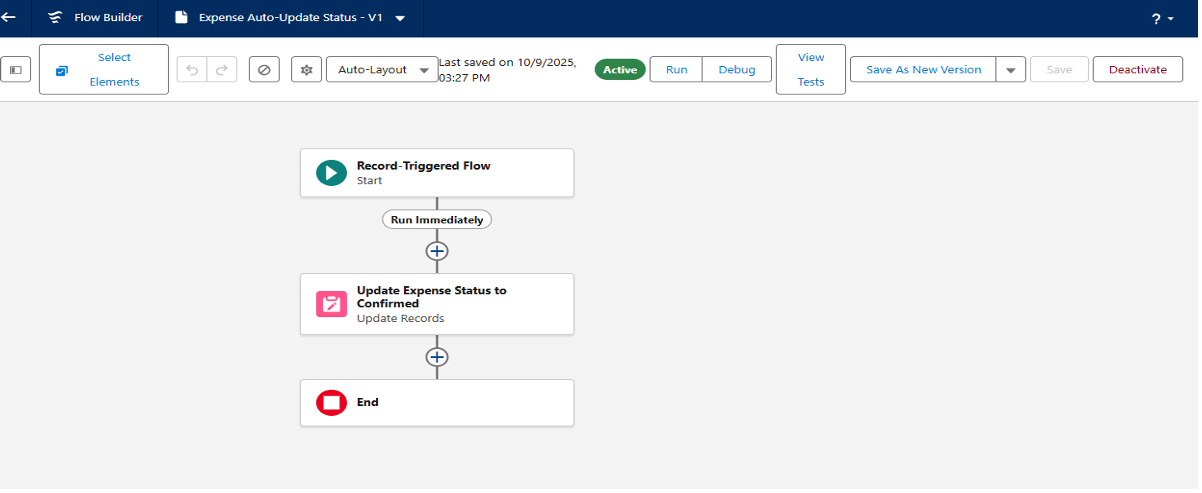
* **Purpose:** To automatically perform a final action, such as updating a "final" status to 'Confirmed' once the approval process is complete.
* **Navigation:** Go to **Setup** → **Flow** → **New Flow** → **Record-Triggered Flow**.

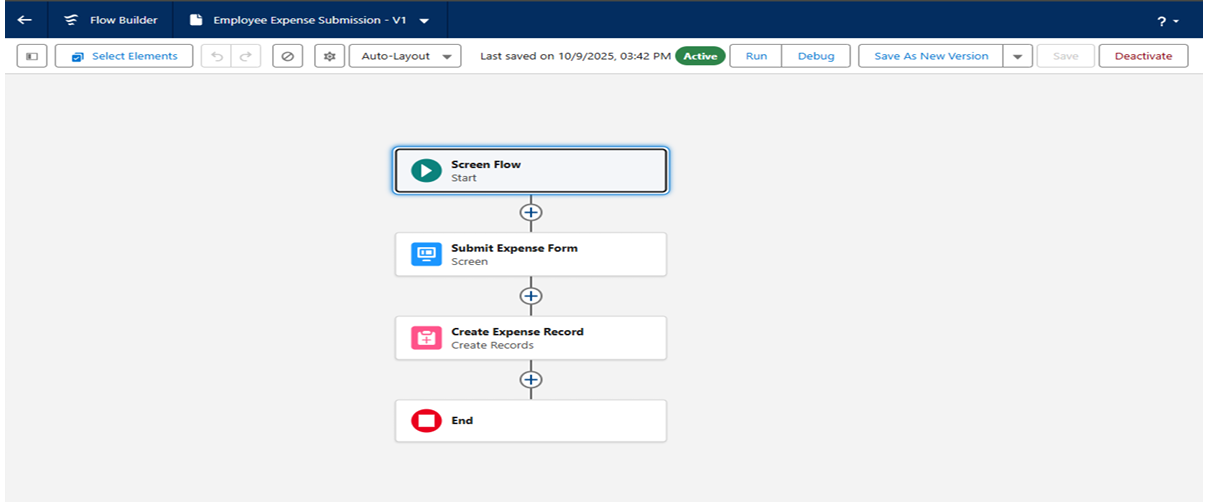
**Trigger Configuration:**

* **Object:** Expense.
* **Trigger:** Select "A record is created or updated".
* **Condition Requirements:** Set the condition for the flow to run: Approval\_Status\_\_c equals 'Approved'.
* **Optimize for:** Select "Actions and Related Records."

**Flow Element:**

1. Add an **Update Records** element to the flow.
2. **Label:** "Update Expense Status to Confirmed".
3. **How to Find Records:** Select "Use the expense record that triggered the flow".
4. **Set Field Values:** Set a field (e.g., a new field named Expense\_Status\_\_c) to 'Confirmed'.
5. **Save and Activate** the flow.

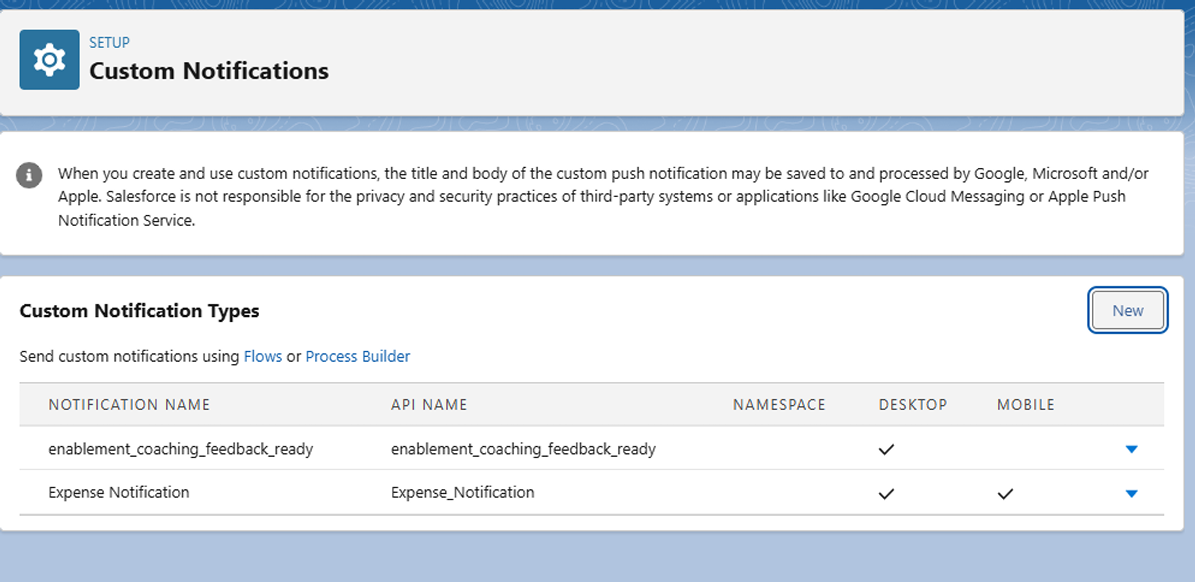




**Step 5: Configure In-App Notifications**

Finally, we will set up in-app (bell icon) notifications to alert users instantly.

* **Purpose:** To notify employees or managers directly within the Salesforce app when an action occurs, such as an approval.
* **Navigation (Part 1):** Go to **Setup** → **Notification Builder** → **Custom Notifications** → **New**.
  + **Name:** Expense Notification.
  + **Supported Channels:** Check **Desktop** and **Mobile**.
  + Save.
* **Navigation (Part 2):** Add this action to your **Record-Triggered Flow** (from Step 4).
  + Add an **Action** element to your flow.
  + **Action:** Search for "Send Custom Notification".
  + **Notification Type:** Select the Expense Notification you just created.
  + **Recipient:** Set the recipient (e.g., the record's owner \$Record.OwnerId or manager).
  + **Message:** Create a dynamic message, such as "Your expense {$Record.Name} has been approved".



**Phase 4 Complete**

With these automations, the Expense Tracker application is now intelligent. Employees are guided through a Screen Flow , data is validated, approvals are automatically routed, statuses are updated, and all relevant users are notified of the changes.