

Calculating Family Expenses using ServiceNow

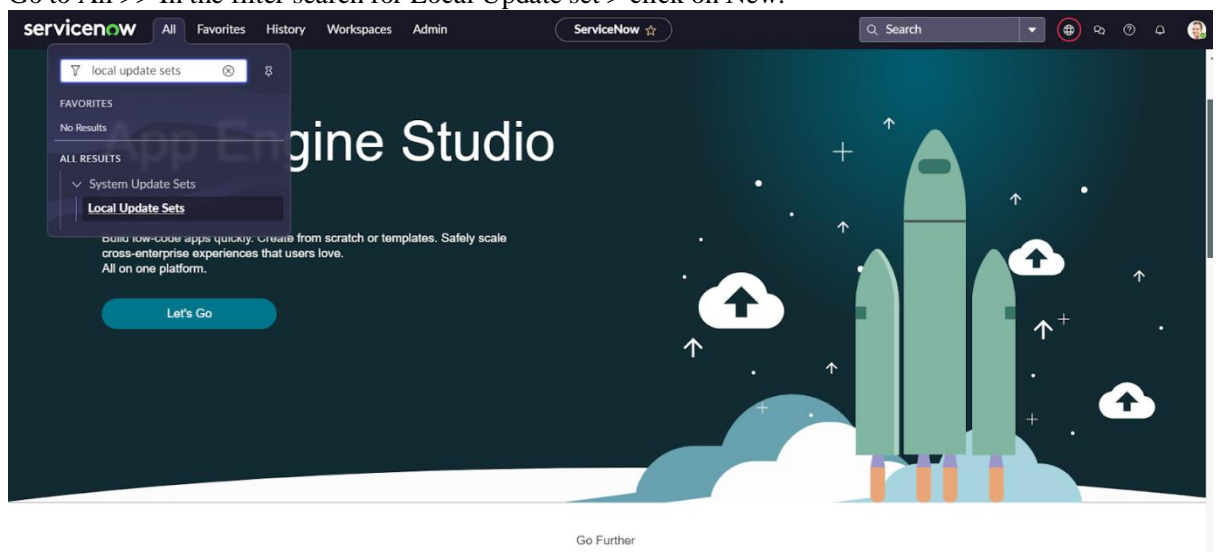
The project aims to develop a comprehensive expense calculation system using ServiceNow. **This system will enable users to track and manage family expenses efficiently. It will include** features such as expense categorization, budget setting, real-time tracking, and reporting capabilities. Utilizing ServiceNow's robust platform, the project will ensure seamless integration, user-friendly interface, and scalability to accommodate varying family sizes and financial complexities. The end goal is to empower users with the tools they need to make informed financial decisions and promote financial well-being within the family unit.

Setting up ServiceNow Instance

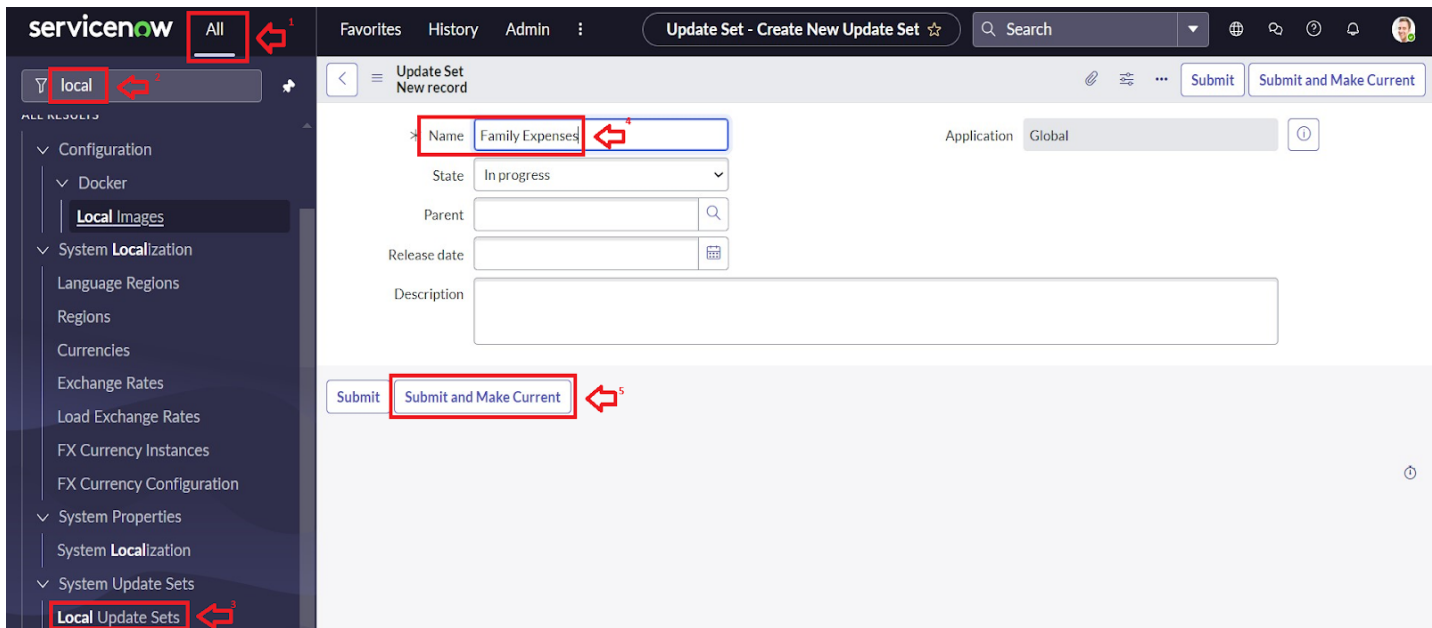
1. Sign up for a developer account on the ServiceNow Developer site “<https://developer.servicenow.com>”.
2. Once logged in, navigate to the "Personal Developer Instance" section.
3. Click on "Request Instance" to create a new ServiceNow instance.
4. Fill out the required information and submit the request.
5. You'll receive an email with the instance details once it's ready.
6. Log in to your ServiceNow instance using the provided credentials.
7. Now you will navigate to the ServiceNow.

Creation of New Update Set

1. Go to All >> In the filter search for Local Update set > click on New.



2. Enter the Details as:
Name : Family Expenses
3. Then click on Submit and Make current.



Creation of Family Expenses Table

1. Go to All > In the filter search for Tables > click on New.
2. Enter the Details:
 Label : Family Expenses
 Name : Auto-Populated
 New menu name : Family Expenditure

The screenshot shows the 'Table Columns' configuration page. The 'Label' field is 'Family Expenses' and the 'Name' field is 'u_st_family_expenses'. The 'New menu name' field is 'Family Expenditure'. The 'Table Columns' section shows three columns: Number (String), Date (Date), and Amount (Integer).

Column label	Type	Reference	Max length	Default value	Display
Number	String				false
Date	Date				false
Amount	Integer				false

3. Go to the Header and right click there>> click on Save.

Creation of Columns(Fields)

1. Near Columns Double click near insert a new row.
2. Give the details as:
Column label : Number
Type : String
3. Double click on insert a new row again
4. Give the details as:
Column label : Date
Type : Date
5. Double click on insert a new row again
6. Give the details as:
Column label : Amount
Type : Integer
7. Double click on insert a new row again
8. Give the details as:
Column label : Expense Details
Type : String
Max length : 800

Column label	Type	Reference	Max length	Default value	Display
Number	String				false
Date	Date				false
Amount	Integer				false
Expense Details	String		800		false

Go

to the Header and right click there>> click on Save.

Making Number Field an Auto-Number

1. Double click on the Number Field/Column.
2. Go down and double click on Advanced view
3. In Default Value:
Use dynamic default : check the box
Dynamic default value : Get Next Padded Number
4. Click on Update.

- 5.
6. Go to All >> In the filter search for Number Maintenance >> select Number Maintenance
7. Click on New.
8. Enter the below Details:
Table : Family Expenses
Prefix : MFE

9. Click on Submit.

Configure the Form

1. Go to All >> In the filter search for Family Expenses >> Open Family Expenses
2. Click on New
3. Go to the Header and right click there>> click on Configure >> Select Form Design
4. Customize or Drag Drop the form as per your requirement.

5. Make Number Read-Only Field by clicking on the gear icon and checking Read-Only
6. Make Date, Amount Mandatory Field by clicking on the gear icon and checking Mandatory
7. Click on Save.

Creation of Daily Expenses Table

1. Go to All > In the filter search for Tables > click on New.
2. Enter the Details:
Label : Daily Expenses
Name : Auto-Populated
Add Module to menu : Family Expenditure

Label: Daily Expenses ↔ 1

Name: u_daily_expenses ↔ 2

Extends table:

Application: Global

Create module: ☒

Create mobile module: ☒

Add module to menu: Family Expenditure ↔ 3

Application Menu

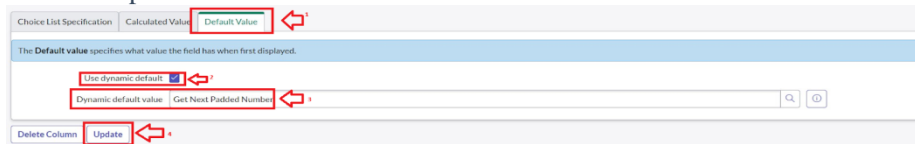
3. Go to the Header and right click there>> click on Save.

Creation of Columns(Fields)


1. Near Columns Double click near insert a new row.
2. Give the details as:
Column label : Number
Type : String
3. Double click on insert a new row again
4. Give the details as:
Column label : Date
Type : Date
5. Double click on insert a new row again
6. Give the details as:
Column label : Expense
Type : Integer
7. Double click on insert a new row again
8. Give the details as:
Column label : Family Member Name
Type : Reference
Max length : 800
9. Double click on insert a new row again
10. Give the details as:
Column label : Comments
Type : String
Max length : 800
11. Go to the Header and right click there>> click on Save.

Making Number Field an Auto-Number

1. Double click on the Number Field/Column.
2. Go down and double click on Advanced view
3. In Default Value:
Use dynamic default : check the box
Dynamic default value : Get Next Padded Number
4. Click on Update.



- 5.
6. Go to All >> In the filter search for Number Maintenance >> select Number Maintenance
7. Click on New.
8. Enter the below Details:
Table : Family Expenses
Prefix : MFE



9. Click on Submit.

Configure the Form

1. Go to All >> In the filter search for Daily Expenses >> Open Daily Expenses
2. Click on New
3. Go to the Header and right click there>> click on Configure >> Select Form Design
4. Customize or Drag Drop the form as per your requirement.



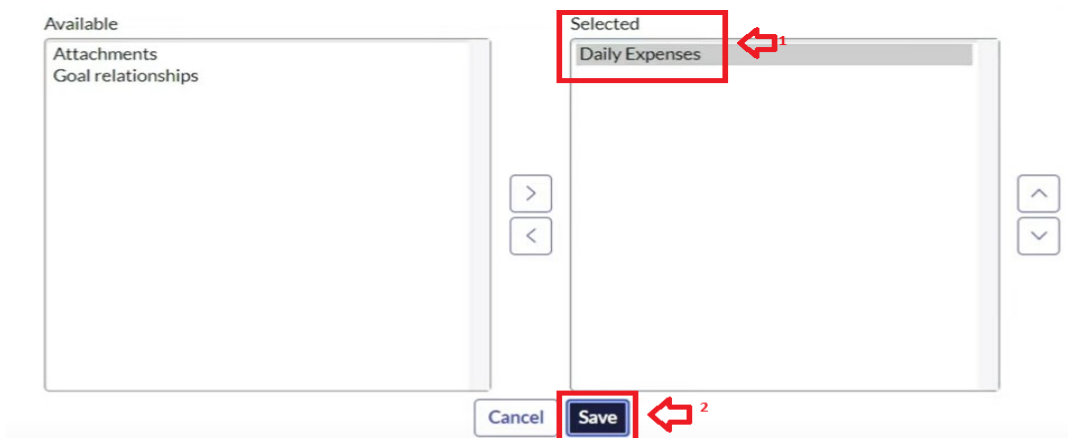
5. Make Number Read-Only Field by clicking on the gear icon and checking Read-Only

6. Make Date, Family Member Name Mandatory Field by clicking on the gear icon and checking Mandatory
7. Click on Save.

Creation of Relationship

Creation of Relationship between Family Expenses and Daily Expenses tables

1. Go to All >> In the filter search for Relationships >> Open Relationships
2. Click on New.
3. Enter the details:
Name : Daily Expenses
Applies to table : Select Family Expenses
Daily Expenses : Select Daily Expenses
4. Click Save.



Configuring Related List on Family Expenses

1. Go to All >> In the filter search for Family Expenses >> Open Family Expenses
2. Click on New
3. Go to the Header and right click there>> click on Configure >> Select Related Lists
4. Add Daily Expenses to the Selected Area.
5. Click on Save

Creation of Business Rules

1. Go to All >> In the filter search for Business Rules.
2. Under System Definition Select Business Rules then click on New.
3. Enter the Details:
Name : Family Expenses BR
Table : Select Daily Expenses

Check Advanced

Business Rule
New record

ss rule is a server-side script that runs when a record is displayed, inserted, deleted, or when a table is queried. Use business rules to automatically change values in form fields when the specified conditions are met.





Name: Family Expenses BR 

Table: Daily Expenses (u_daily_expenses) 

Application: Global 


Active: ☒

Advanced: ☒ 

4. In when to run Check Insert and Update
5. In Advance (we write the code): Write the below code >>

```
(function executeRule(current, previous /*null when async*/) {  
    FamilyExpenses = new GlideRecord('u_family_expenses');
```

var

When to run  Advanced

Specify whether the business rule should run on Insert or Update. Use Filter Conditions to specify under which conditions.

When: before

Order: 100

Insert: ☒ 

Update: ☒

Delete: ☐

Query: ☐

Filter Conditions: [Add Filter Condition](#) [Add "OR" Clause](#)

-- choose field -- -- oper -- -- value --

Role conditions: 

```
FamilyExpenses.addQuery('u_date',current.u_date);  
FamilyExpenses.query();  
if(FamilyExpenses.next())  
{  
    FamilyExpenses.u_amount += current.u_expense;  
    FamilyExpenses.u_expense_details +=  
    ">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";  
    FamilyExpenses.update();  
}
```

```
else  
{  
    var NewFamilyExpenses = new GlideRecord('u_family_expenses');  
    NewFamilyExpenses.u_date = current.u_date;  
    NewFamilyExpenses.u_amount = current.u_expense;  
    NewFamilyExpenses.u_expense_details +=  
    ">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";  
    NewFamilyExpenses.insert();  
}
```

Configure the Relationship

1. Go to All >> In the filter search for Relationships >> Open Relationships.

2. In that, open Daily Expenses Relationship.
3. For Applies to table : Select Family Expenses.
4. In Query with : write the below Query.

```
(function refineQuery(current, parent) {
// Add your code here, such as current.addQuery(field, value);
current.addQuery('u_date',parent.u_date);
current.query();
})(current, parent);
```

5. Click on Update.

The screenshot displays the Salesforce interface for configuring a relationship between two tables. At the top, a script editor shows a JavaScript function `executeRule` that refines the query for the 'Family Expenses' table based on the 'Daily Expenses' table. The script includes logic to add a new record or update an existing one based on the 'u_date' field.

Below the script editor, the 'Relationship' configuration page is shown. The 'Name' field is set to 'Daily Expenses', and the 'Application' is set to 'Global'. The 'Applies to table' dropdown is set to 'Family Expenses [u_family_expenses]', and the 'Queries from table' dropdown is set to 'Daily Expenses [u_daily_expenses]'. A red box highlights the 'Applies to table' dropdown, with a red arrow pointing to it.

Below the configuration page, a 'Query with' section shows the same JavaScript function as the script editor. A red box highlights this section, with a red arrow pointing to it. At the bottom, the 'Update' button is highlighted with a red box and a red arrow, indicating the next step in the process.

Conclusion