wTools User Guide

wTools User Guide

Introduction

Sugar enables businesses to create extraordinary customer relationships with the most innovative, flexible and affordable CRM solution on the market. As an open-source, web-based CRM solution, Sugar is easy to customize and adapt to your organization's changing needs. wTools is a library of customizations for SugarCRM, which includes the following products:

- Double Click to Edit List View Row
- Gauge Chart Dashlet
- Drill Down Report Chart Dashlet
- Create Note Function in ListView
- Custom Notification Field Type
- Colored Fields
- Custom Buttons
- Custom Results per Module Dropdown
- Embedded Content Field
- Sparkline List Dashlet

This guide covers how users can engage with wTools after they been installed by an administrator. For more information on installing these plug-ins, please refer to the W-Systems Product Installation Guide.

Note: This guide pertains to the latest release of wTools. If you are not using the latest version of wTools, then you may not have access to some of the features listed on this page. Please refer to the W-Systems Supported Platforms page for more information on the latest package versions.

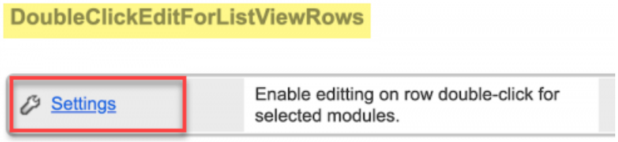
Double Click to Edit List View Rows

This simple module makes it a little quicker to edit a row in list view mode. Just double click on the row and it goes into edit mode. This replaces the need to press the row action menu and the edit choice. You can activate many rows for editing by repeatedly double-clicking the rows you want to edit.

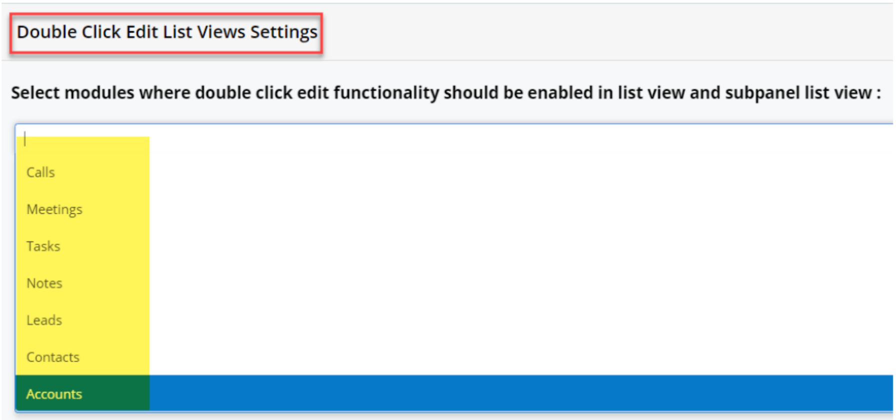
Configuring the Double Click to Edit List View Row Module

Before starting to use the **Double Click to Edit List View Row Module**, you have to configure it. In the **Administration** page, scroll down until you reach the **DoubleClickEditForListViewRows Configuration** panel. After clicking the

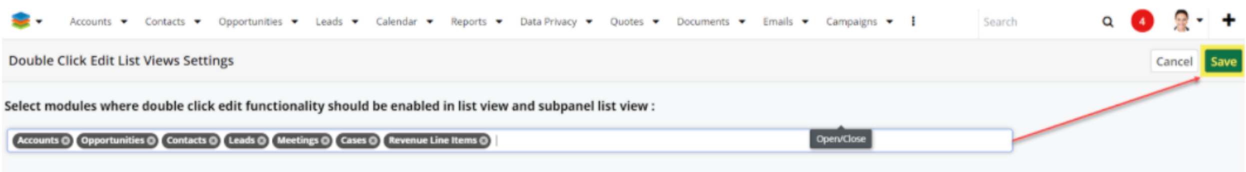
Settings link, a drop-down menu will open.



From the drop-down menu, select the Sugar Modules where you want the Double Click to Edit List View Row to activate.



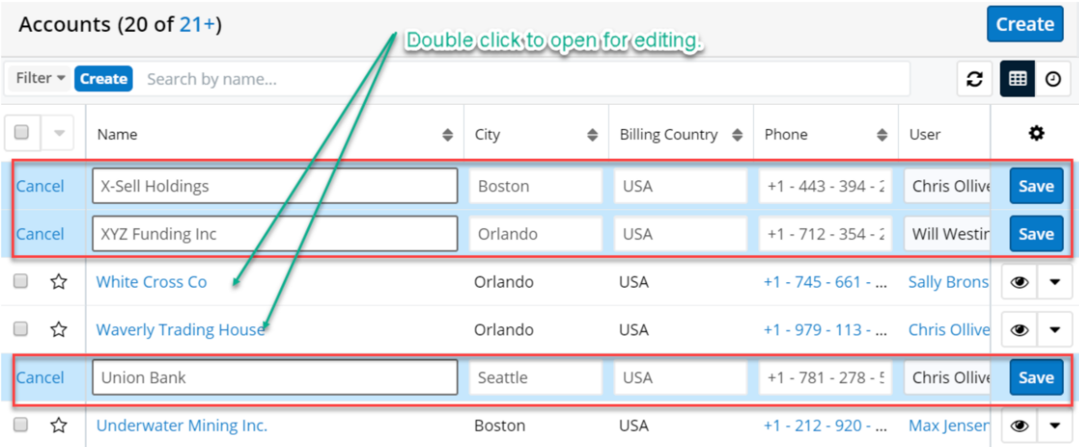
After selecting the Sugar modules, save your changes.



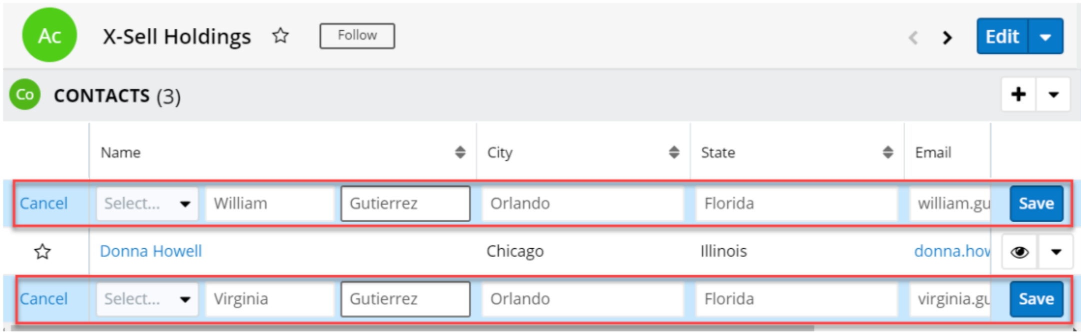
Now the Double Click to Edit List View Row Module is set up and ready to use.

Using the Double Click to Edit List View Row Module

Following is a screenshot showing three (3) rows opened for editing in the **Accounts** module (bordered in red). To open a row for editing, double-click within the row but not on a hyperlink (see green arrows). Once the row opens for editing, there will be a **Cancel** command on the left of the row, and a **Save** button on the right side of the row.



Sub-panels are supported too. The next image shows the **Contacts** sub-panel for the "X-Sell Holdings." account in the previous image. As you can see, double-clicking in a sub-panel also allows for editing within the row.



Gauge Chart Dashlet

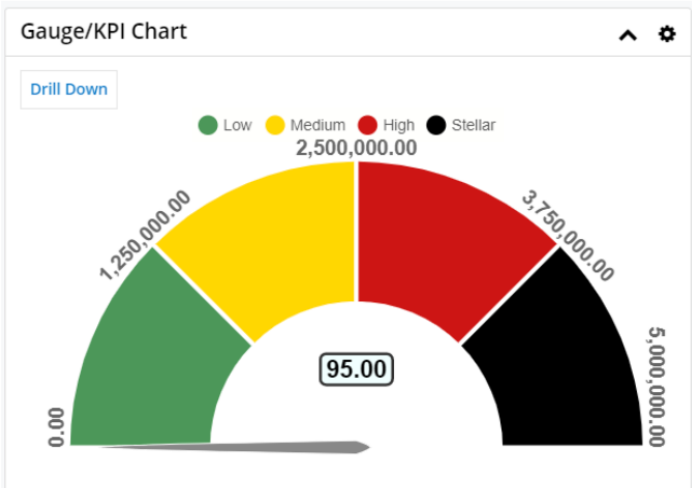
This new dashlet type can be configured to measure a numeric field aggregate value from any module and display the results in a gauge type chart. It is highly customizable.

Using the Gauge Chart Dashlet Module

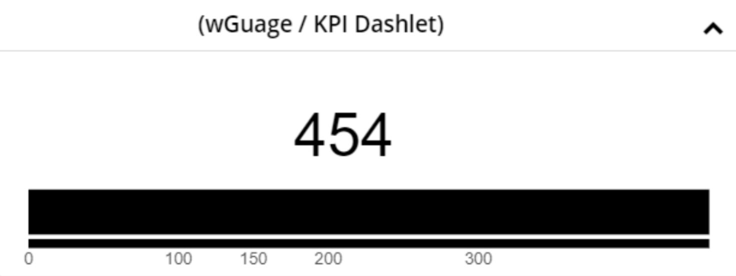
Gauge Dashlet Examples

This module provides dashlet functionality with two types of charts: Gauge and KPI (Key Performance Indicator). Both can be added to a dashboard. Here is an example of each type:

Gauge Chart:



KPI Chart:

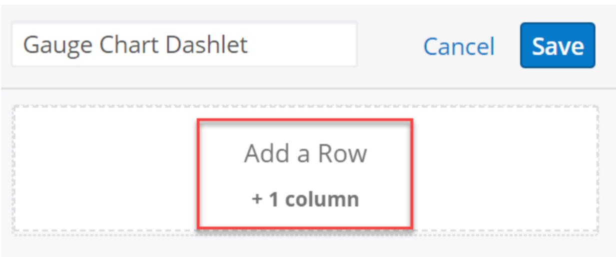


In the usage example below, we'll be adding a dashlet to the "Opportunities" module's dashboard to create the charts shown in the example above (charts can be used with other modules).

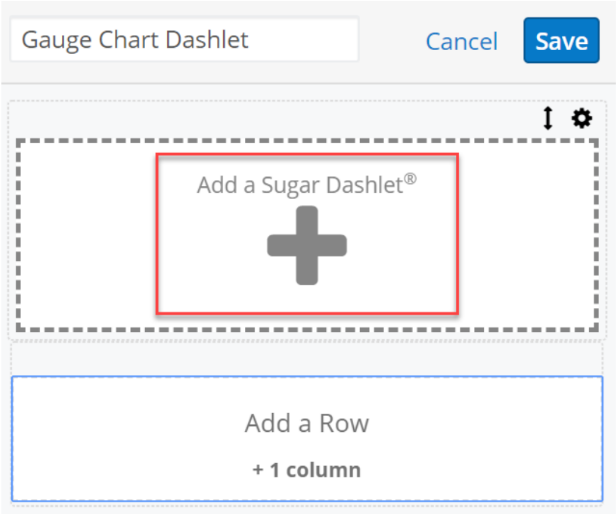
Setting Up a Gauge Dashlet

On the Account module, select a dashboard to add the dashlet to. Use the **Actions** menu drop-down to select **Edit**.

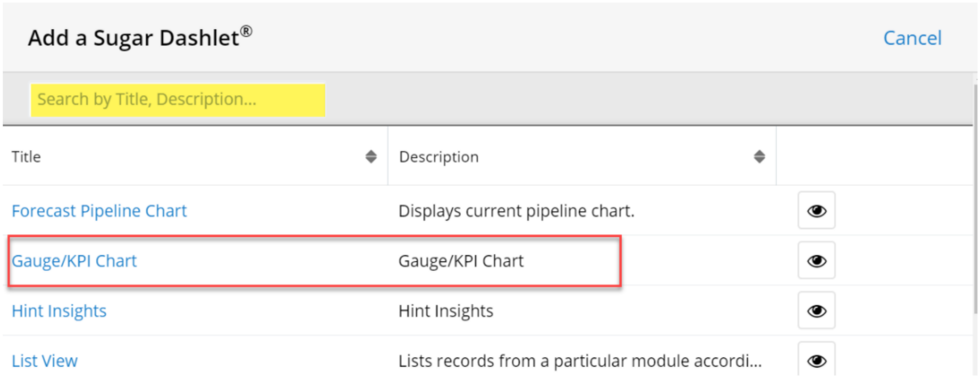
Scroll to the bottom of the dashboard and click "+ 1 column" in the **Add a Row** box.



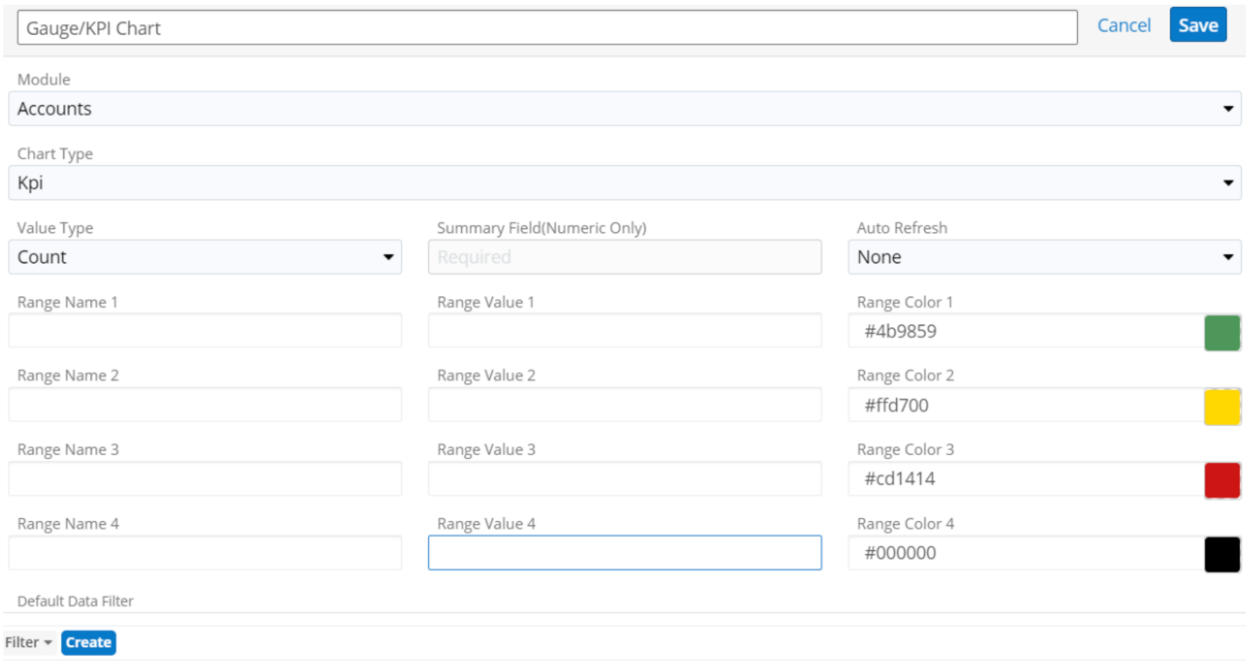
Click the **Add a Dashlet** box that appears:



When presented with the list of Sugar Dashlets, click the link for: **Gauge/KPI Chart**.

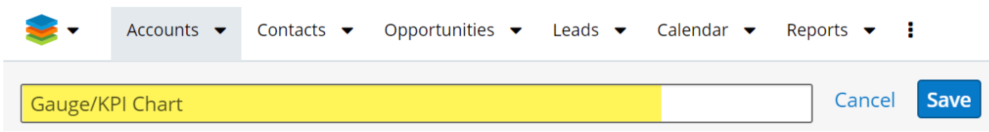


The configuration page for the dashlet will be displayed and a user can define the type of chart and the parameters:



The following parameters can be set:

Dashlet title: The dashlet title can be set by the user (the default setting is: "Gauge/KPI Chart").



Module (drop-down): The Module field is a drop-down containing all modules from Studio. The module selected defines which records are used by the chart.

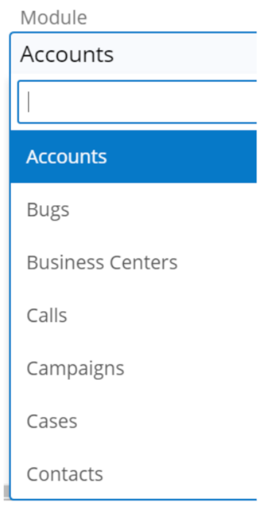
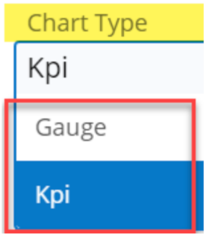
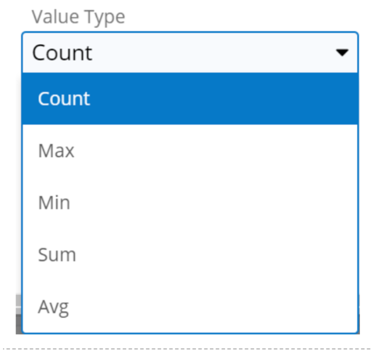


Chart Type (drop-down): This drop-down contains two options: **Gauge** and **KPI** . This selection defines the type of chart that will be created.



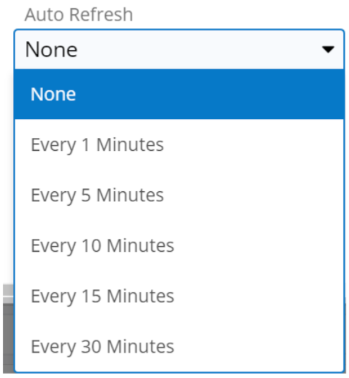
Value Type (drop-down): This drop-down contains the functions that can be used by the chart. The result (i.e., value) of the function selected in the **Value Type** field will be displayed inside the Gauge Chart and the pin will point to that value on the chart (please see the specific function descriptions). **Note**: On the Kpi Chart, the "value" will be displayed "over" the bar instead of as a pointer that is used by the Gauge Chart.



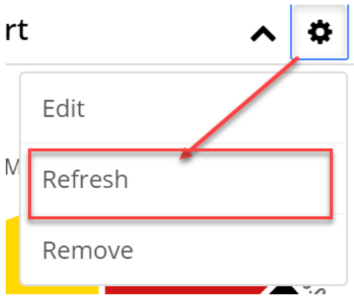
- **Count**: Counts the records from the module that meet the criteria. NOTE: When this function is selected, the "Summary Field" is not required.
- **Average**: Calculates the average value for the Summary Field.
- **Sum**: Calculates the sum of Summary Field from all records that meet the criteria.
- **Max**: Returns the maximum value of the Summary Field.
- **Min**: Returns the minimum value of the Summary Field.

Summary Field (**Numeric Only**) (drop-down): This drop-down contains all of the numeric fields from the selected module. This field is required for all functions selected from the Value Type drop-down except for the "Count" value. When "Count" is selected, this field becomes disabled as shown in this screenshot:

Auto Refresh (drop-down):

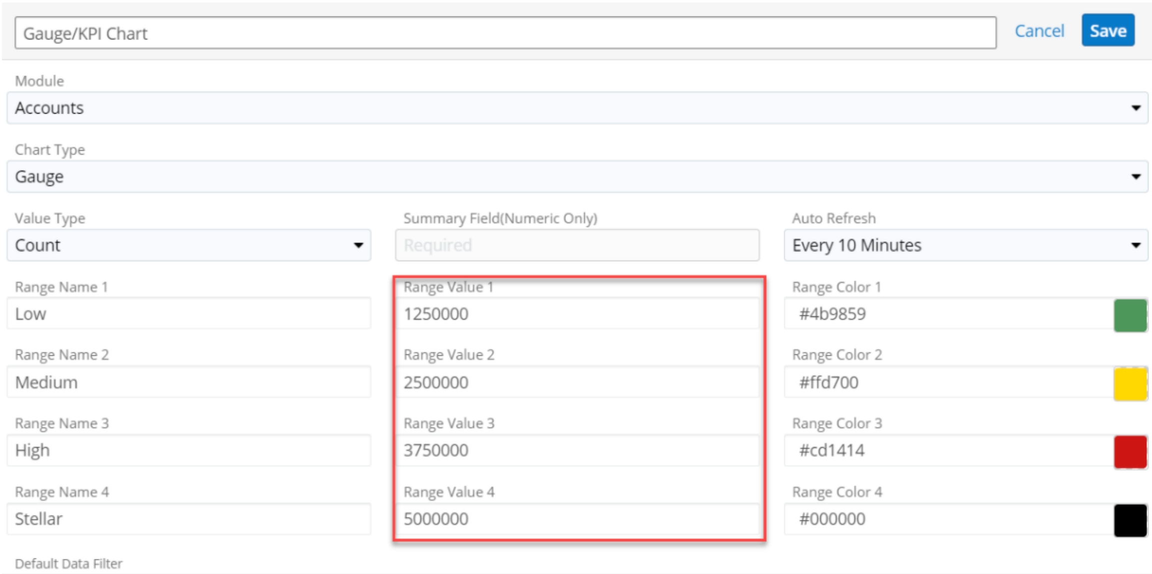


Please note that the dashlet can be refreshed at any time using the dashlet's **Configure** menu's **Refresh** command:

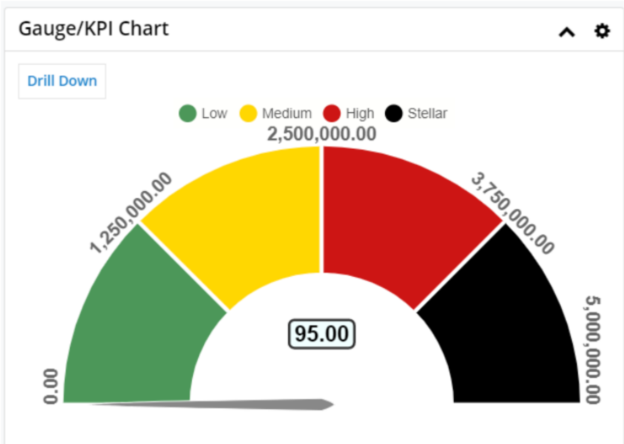


Range Values

The Range Value fields are four numeric fields that will define the sections of the Gauge chart and the X axis of KPI chart. The values must be in ascending order.

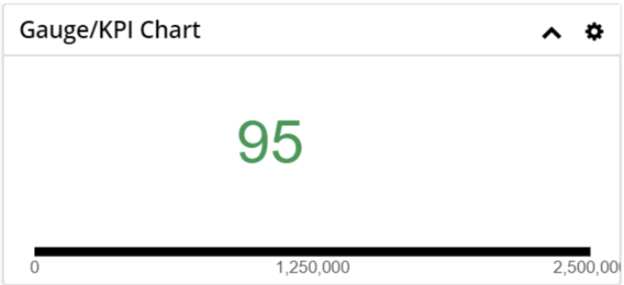


On Gauge Chart, the range values will be displayed above the chart at the end of each section:



If not all range values parameters are defined, the chart will be displayed only with the defined ranges and the missing range will be suppressed. Only the last range values can be undefined.

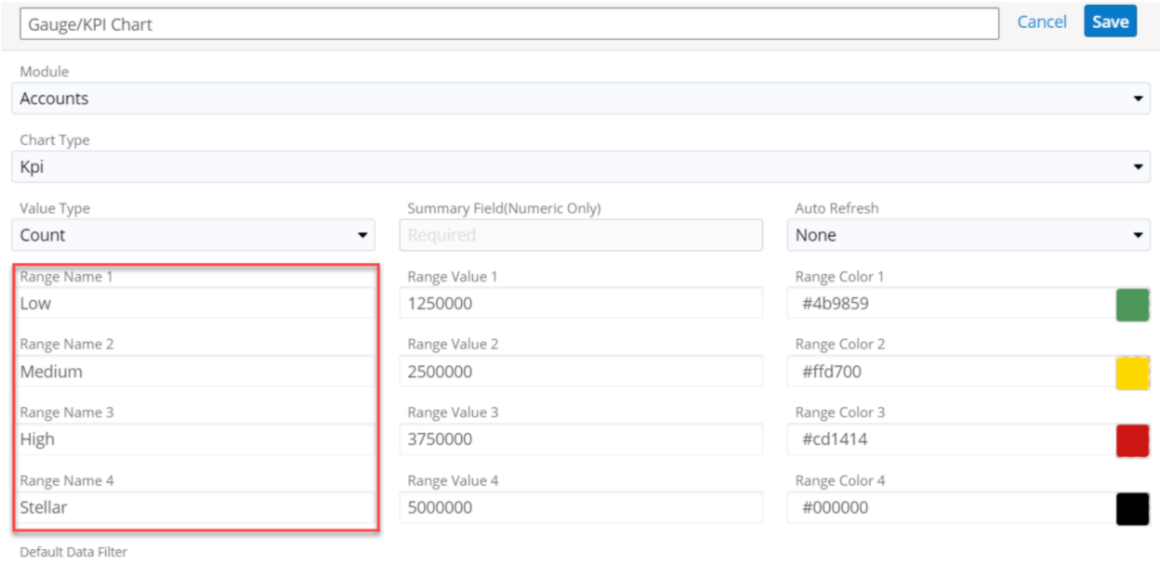
On the **KPI Chart** , the range values will be displayed below the X Axis:



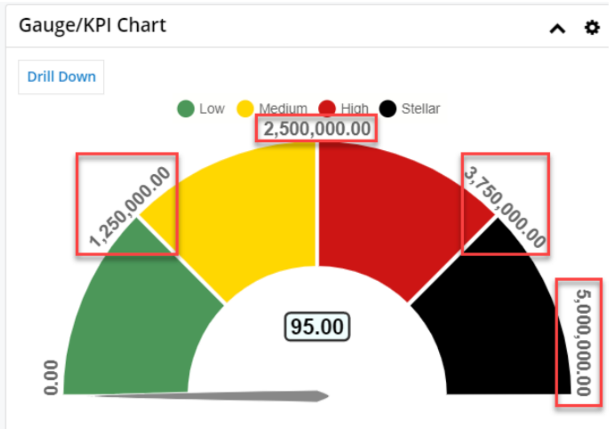
Above is an example of a Kpi Chart where the Range Value 3 and Range Value 4 are not defined.

Range Name

On both types of these charts, the Range Name must be completed in order for that Range Value to be represented on the chart.

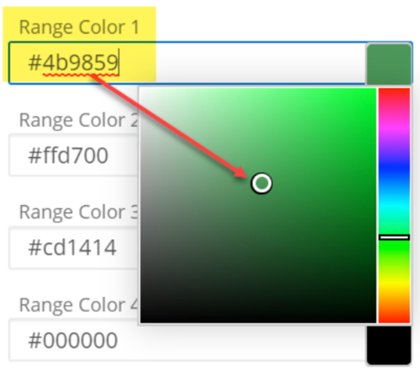


On Gauge Chart the Range Name are displayed above the chart as a legend.

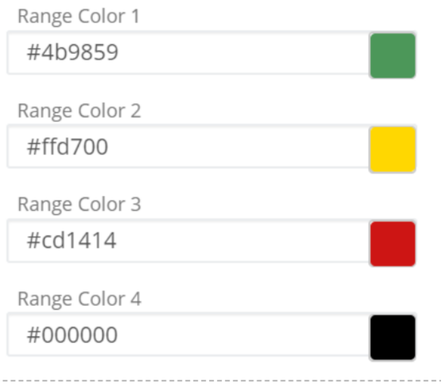


Range Color

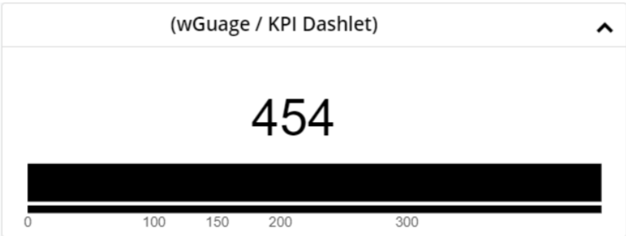
For each Range Value a different color can be selected using a color picker.



When a color is selected, the hex code for that color will be displayed in the field and a preview of that color will be placed on the right of the field.



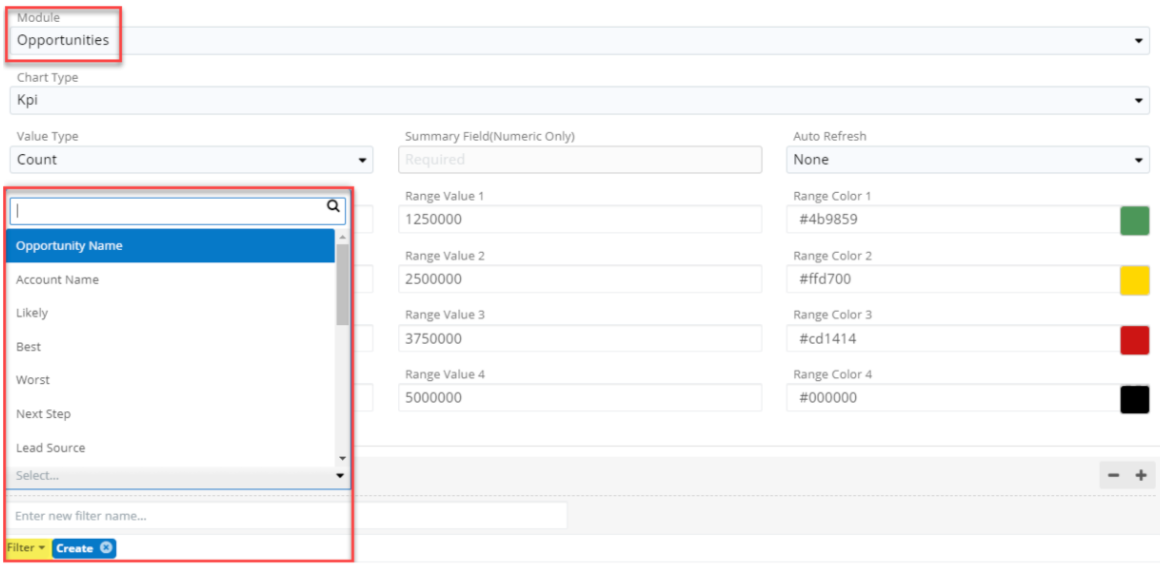
On the KPI chart, the displayed number and the horizontal bar chart will be colored.



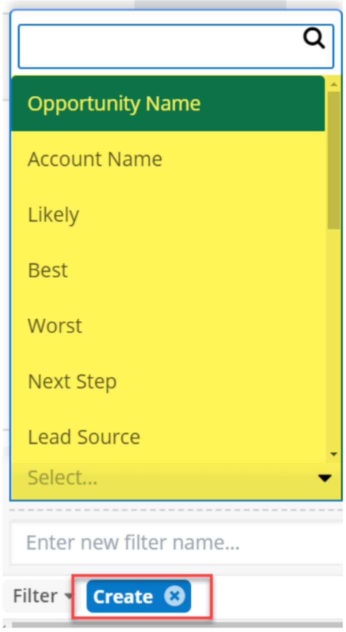
Filter

The records used by the charts can be filtered using an existing filter from the selected module, or creating a new filter on the Gauge/KPI Chart drawer.

The Filter list is populated with the existing filters from the selected module.



Users can create new Filters for that module by clicking on the **Create** button from the bottom of the drawer. Please refer to Sugar documentation for filter creation details.



Drill Down Report Chart

The wDrillDownReportsChart package contains the 'wChart Dashlet' which allows users to view onto a dashboard predefined report charts but also charts for newly created reports.

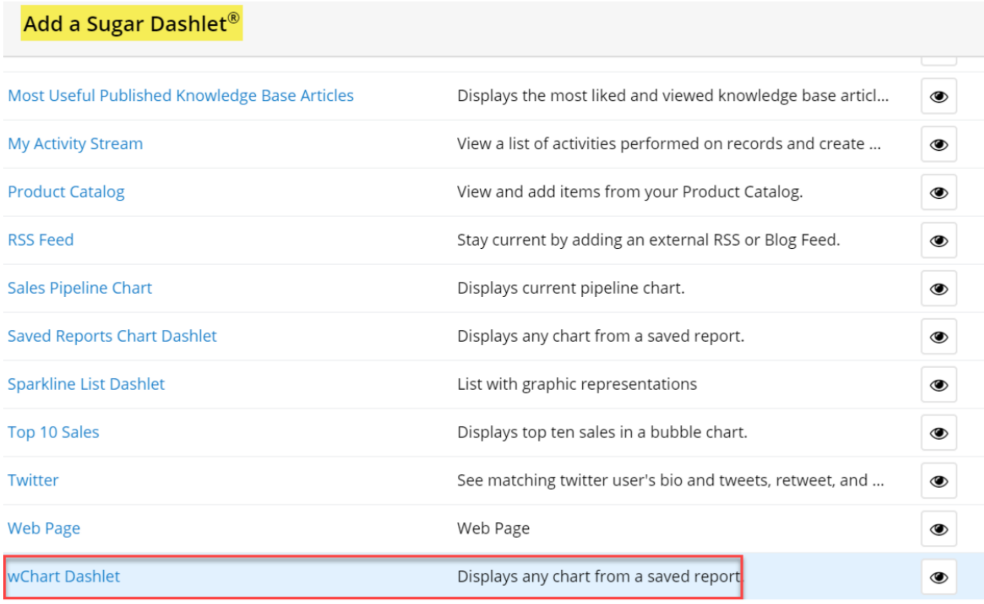
Since some of the most effective Sugar dashboard elements are charts based on reports, we have created a dashlet that allows you to click on the drill down link on any report based dashlet to open the dashlets underlying report in a new browser tab.

Configuring the Drill Down Report Chart Dashlet

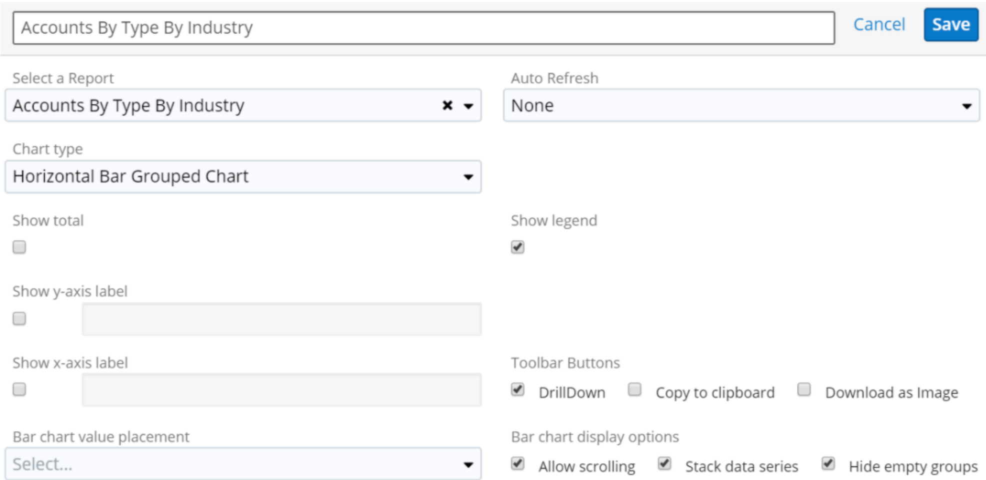
Using the Drill Down Report Chart Dashlet

Configurations

After the Drill Down Reports Chart package is successfully installed, the "**wChart Dashlet**" will be available in "**Add a Sugar Dashlet**" menu:



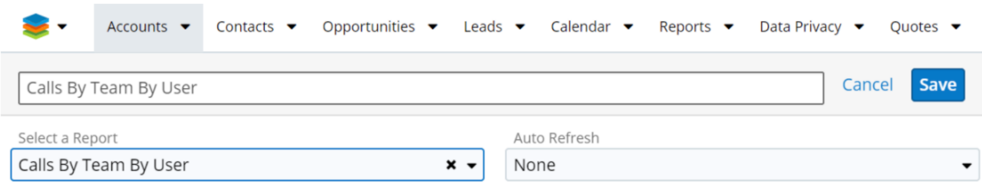
When selecting the "wChart Dashlet", the user can set the various configuration options available (i.e., Chart type, Bar chart value placement, Bar chart display options, etc.) to define how the report chart will be displayed in the dashlet. This allows users to customize each of their chart dashlets based on their own viewing preference:



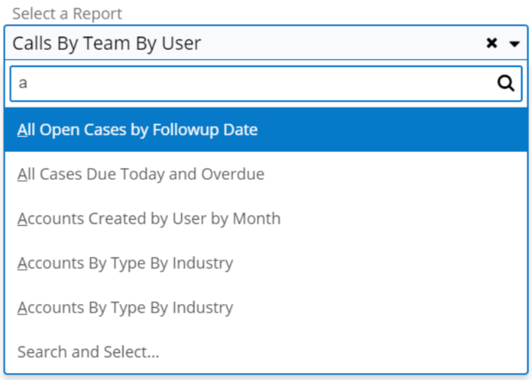
Configuration Options

The configuration options available may vary slightly between the report charts.

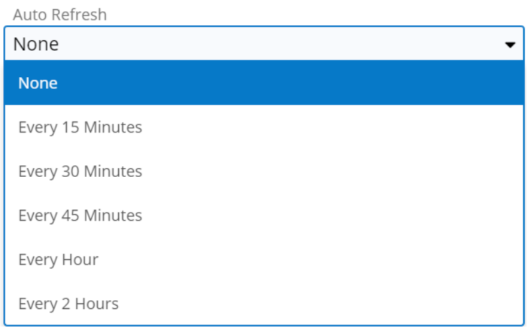
Dashlet name: This is auto populated with the name of the selected report from "Select a Report" drop down and can be changed by the user.



Select a Report: The "Saved Reports Chart Dashlet '' allows the user to view predefined report charts as well as charts of custom reports. All report types (i.e., Summation Report, Matrix Report, Summation with Details Report) except for "Rows and Columns" type reports, have the chart option available to view as dashlets.



Auto Refresh: This is a drop-down with time interval options (and a choice for "None"). The setting used determines the auto refresh rate of the dashlet.



Note: A "Saved Reports Chart Dashlet" can be refreshed in two ways. One way is using the previously mentioned "Auto Refresh" setting. The dashlet can also be manually refreshed anytime by using the dashlet's Configure button and selecting Refresh.

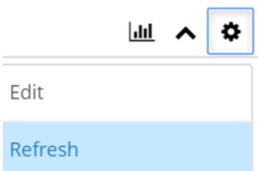
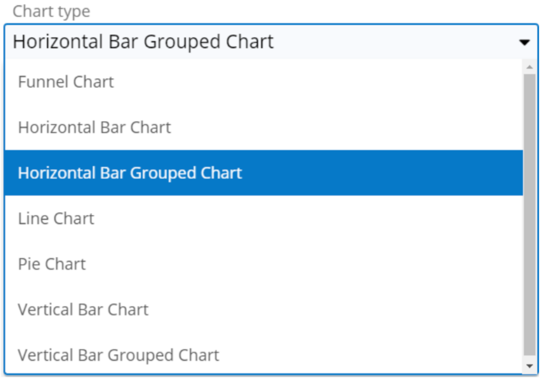
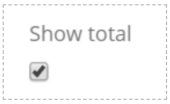
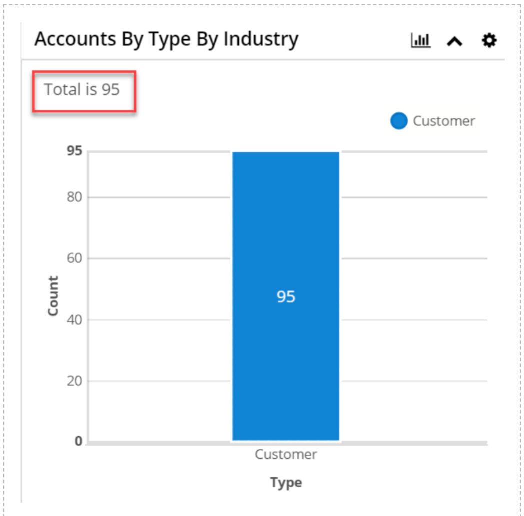


Chart type: This is a drop-down containing the types of charts used to define how the chart will be displayed. When a report chart is selected, the "Chart type" will default to the chart type defined in the originating report:

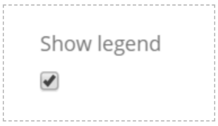


Show Total: This checkbox will determine if the chart's total number of records will be displayed on the top-left of the dashlet.





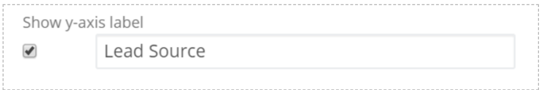
Show legend: This checkbox determines if the chart's full legend is displayed on the dashlet.

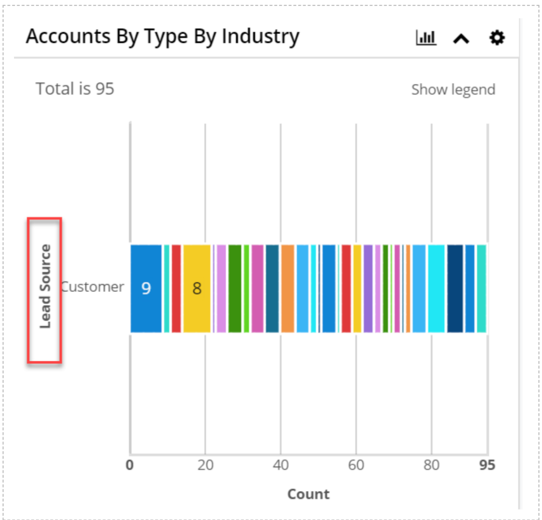


Show y-axis label: This checkbox determines if the chart's y-axis title is displayed on the dashlet.

A title from the originating report will populate automatically, but the user can specify a different title to be displayed in the dashlet by entering a new value in the open text box.

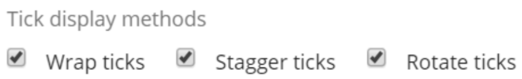
This option only applies to bar charts and line charts, for other types of chart the option will be hidden.



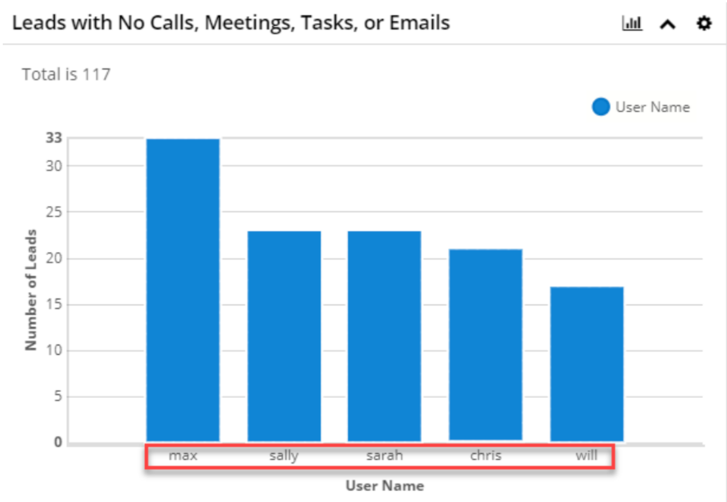


Show x-axis label: This checkbox determines if the chart's x-axis title is displayed on the dashlet. A title from the originating report will populate automatically, but the user can specify a different title to be displayed in the dashlet by entering a new value in the open text box. This option only applies to bar charts and line charts, for other types of chart the option will be hidden.

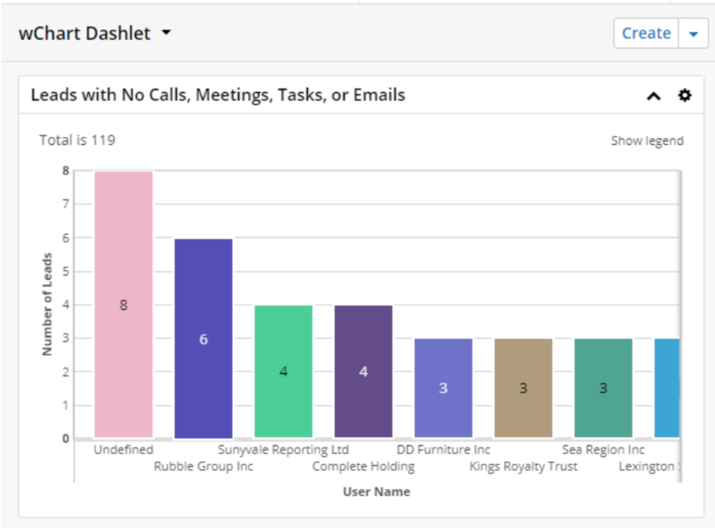
Tick display methods: Enable the appropriate options based on how the user wishes to display x-axis labels with long texts on the chart dashlet. When all three options are enabled, Sugar will apply the display methods in the order: "Wrap ticks", "Stagger ticks", and then "Rotate ticks", and display the x-axis labels based on the best option. This option only applies to bar charts and line charts, for other types of chart the option will be hidden.



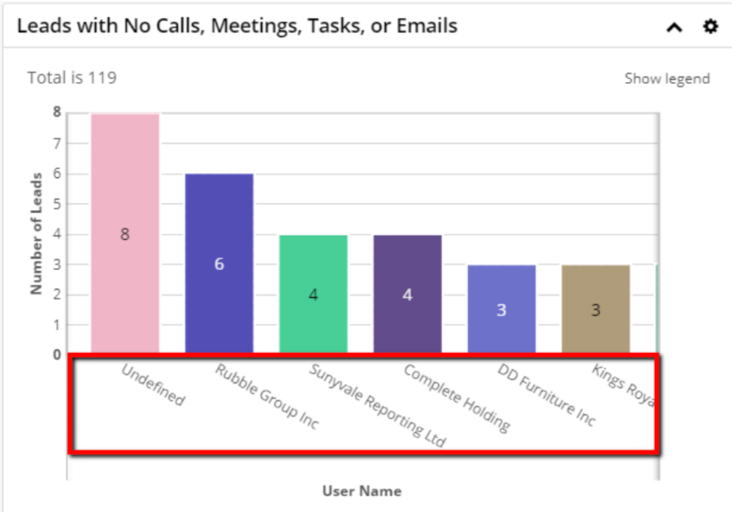
- **Wrap ticks**: Wraps x-axis labels to the next line



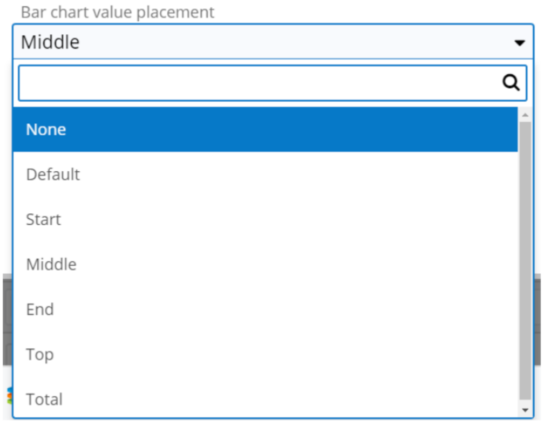
- **Stagger ticks**: Staggers x-axis labels in alternating positions:



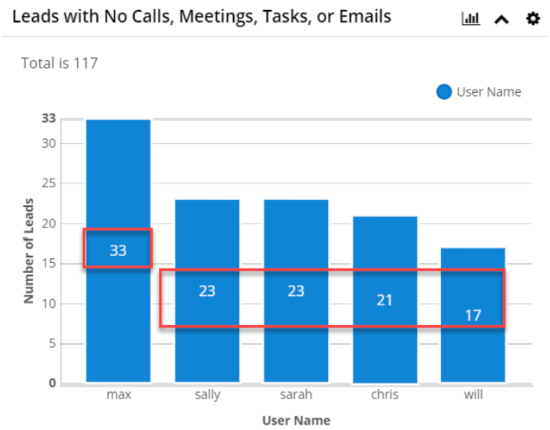
- **Rotate ticks**: Rotates x-axis labels diagonally instead of across:



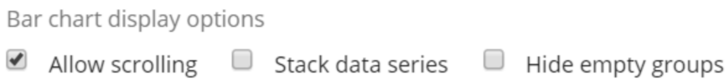
Bar chart value placement: This drop-down determines where the value will be positioned for bar charts values (e.g., None, End, Middle, etc.).



In the following example, the "Bar chart value placement" is set to "Middle" making the value appear in the "middle" of the bar:



Bar chart display options: Enable the appropriate options based on how the user wishes to view data on the chart using these checkboxes:



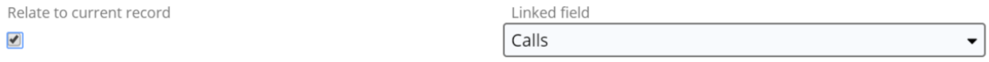
- **Allow scrolling**: Scroll the chart up-and-down or left-to-right on the dashlet, if applicable. Disabling this option re-sizes the chart to fit the dashlet's full width.
- Scrolling will work for charts where the data exceeds the size of the dashlet width.
- This option only applies to bar charts, for other types of chart the option will be hidden.
- **Stack data series**: Displays a stacked bar for each grouping in the report chart broken into sub-sections of data. Disabling this option displays separate bars for each sub-section in the same grouping.
- This option only applies to grouped bar charts (e.g. Horizontal Bar Grouped

Chart), for other types of chart the option will be hidden.

- **Hide empty groups**: Hides any empty data groups from the chart, if applicable.
- This option only applies to bar charts, for other types of chart the option will be hidden. **(This option is no longer available for Drill Down Reports Chart starting with version 8.x).**

Relate to current record: On a Record View, the "wChart Dashlet" has the option to display the selected report only with the records related to current record.

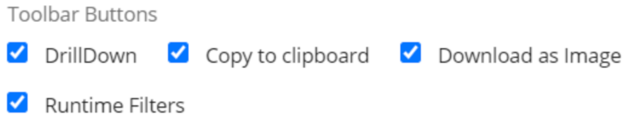
There are two options will make this possible:



- **Relate to current record**: Enable the records filtering related to the current record option. The checkbox is visible only on Record View.
- **Linked field**: Is a drop-down field with the related fields from the current module. This field is required and is visible only when "Relate to current record" is checked.

Note: Starting with **Sugar 10.0.x**, the **Relate to current** record option is no longer supported.

Toolbar Buttons: allows users to select which buttons will be displayed on the Drill Down Report Chart Dashlet:



Using the Drill Down Report Dashlet

1. Create wChart Dashlet

The 'wChart Dashlet' is available in 'Add a Sugar Dashlet' menu in the following screens:

- Home
- List View: Intelligence Pane
- Record View: Intelligence Pane

The only difference between the above screens is that on the Record View, the 'wChart Dashlet' can display the chart with the records linked to the current record if 'Relate to current record' is enabled and the Linked field is correctly selected.

The user can create a 'wChart Dashlet' on Contacts Record View that will display the 'Leads By Lead Source' report with the Leads linked to the current contact as follows:

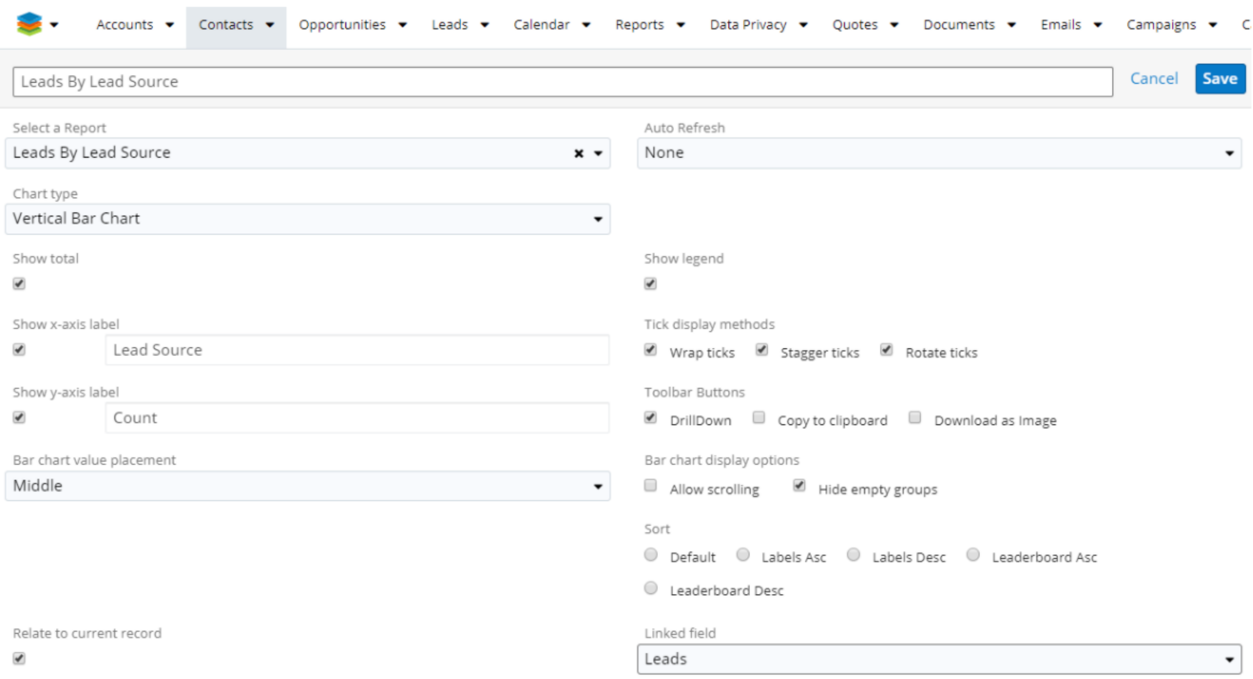
Step 1: Add the 'wChart Dashlet' on the Intelligence Pane.

Step 2: With the configuration page open, select 'Leads By Lead Source' report from 'Select a Report' drop-down.

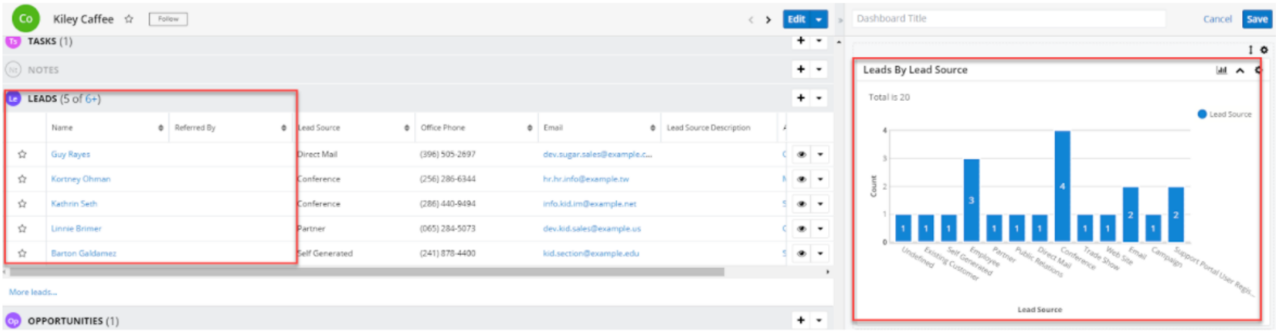
Step 3: Check 'Relate to current record' and select the corresponding field related to the Leads module.

Note: Starting with **Sugar 10.0.x**, the **Relate to current record** option is no longer supported.

Step 4: Change the other configurations as desired:

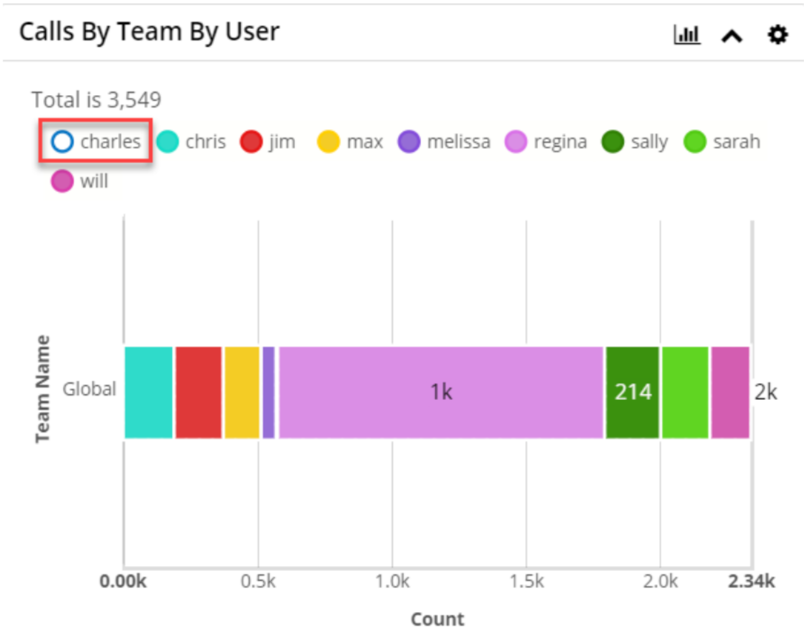


Step 5: Save the configuration and the chart will be displayed including only the Leads linked to the current contact:

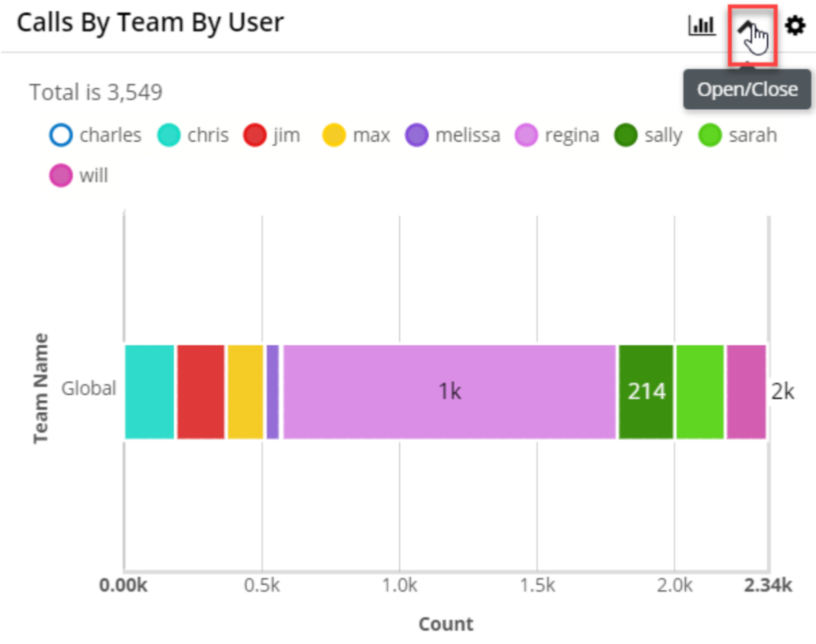


2. Edit "wChart Dashlet"

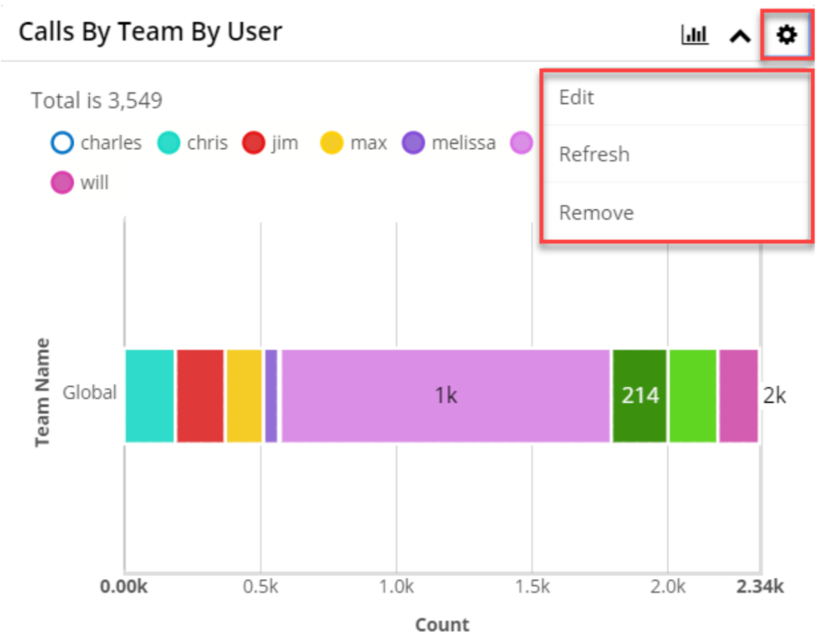
If the '**Show legend**' option was checked, the legend is displayed on the top right of the dashlet. Certain groups of data can be hidden from the chart by toggling (i.e., on/off) the legend keys by clicking on them in the chart. Hidden groups of data are represented on the legend by an empty circle. In this screenshot, the "Undefined" key was clicked and does not appear as a bar:



To collapse the dashlet, simply click the Up arrow to the left of the Gear icon (reverse this to re-display the dashlet by clicking the Down arrow):

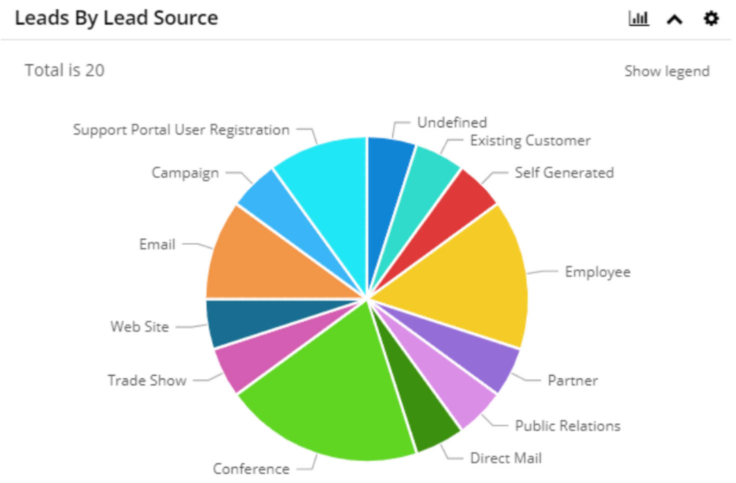


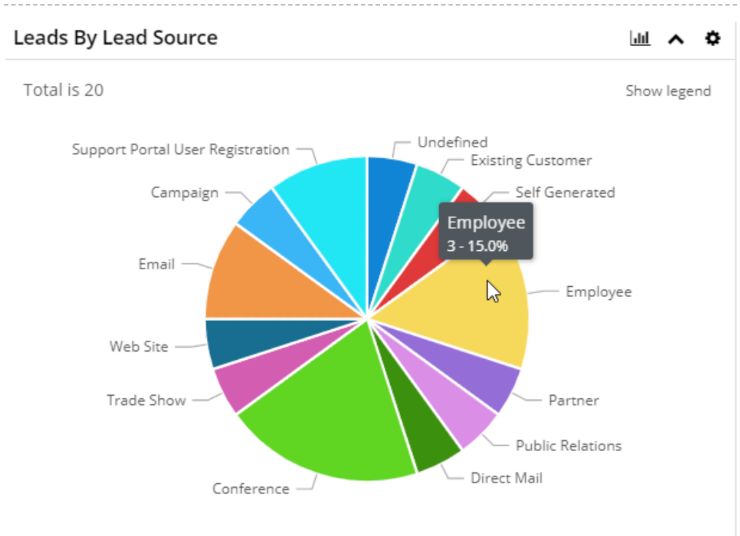
The user can make changes to the created dashlet by clicking the Gear icon (upper right of dashlet) which allows: editing the dashlet, refreshing the data, or removing the dashlet if desired. The Edit button will open the configuration drawer allowing the user to make changes to the dashlet. See the Create 'wChart Dashlet' topic for configuration information.



3. Clickable Elements

All chart types have clickable chart elements on a 'wChart Dashlet'. The clickable elements "lighten" when hovering over them:



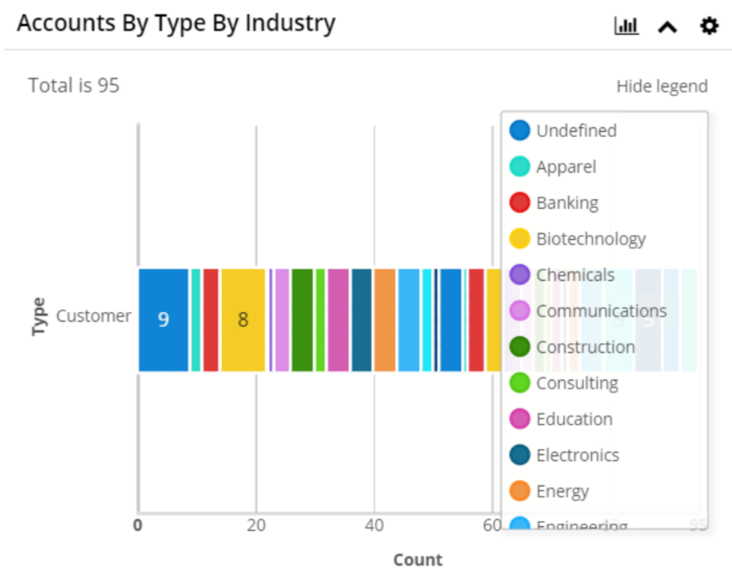


Based on the report type, the **"Report group drilldown**" drawer will display the following columns:

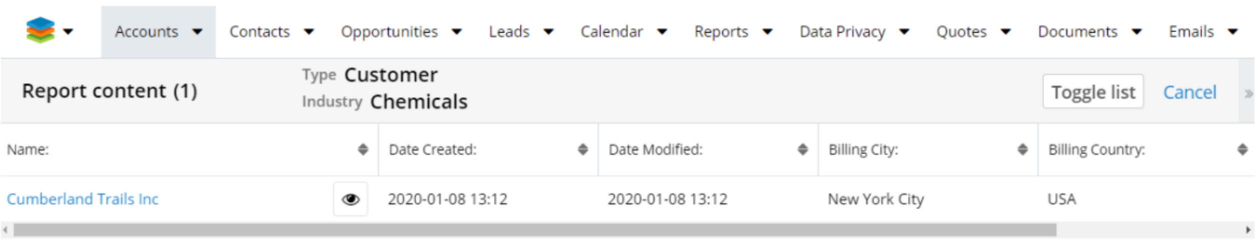
- For a "Summation Report with Details", the drawer will show the Display Columns of the report
- For other reports that don't have the Display Columns section, the drawer will show the List View (or part of List View if there are too many columns)

Examples of clicked elements from 'wChart Dashlets':

1. Home: wChart Dashlet' for the report "Accounts By Type By Industry"

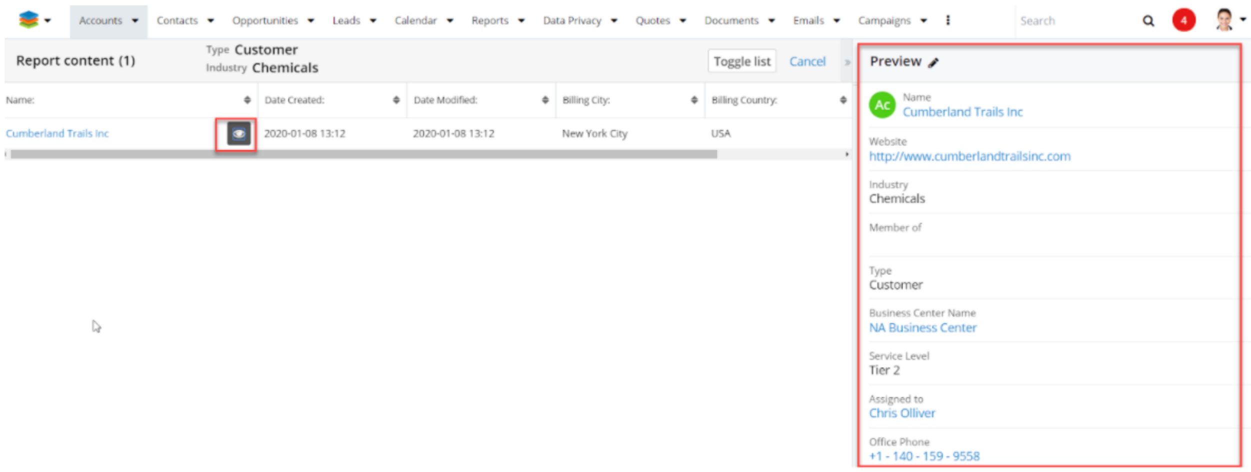


If the user clicks the section that corresponds to the "Chemicals" industry from the "Customer" type, 1 record will be displayed in the "Report group drilldown" drawer as follows:



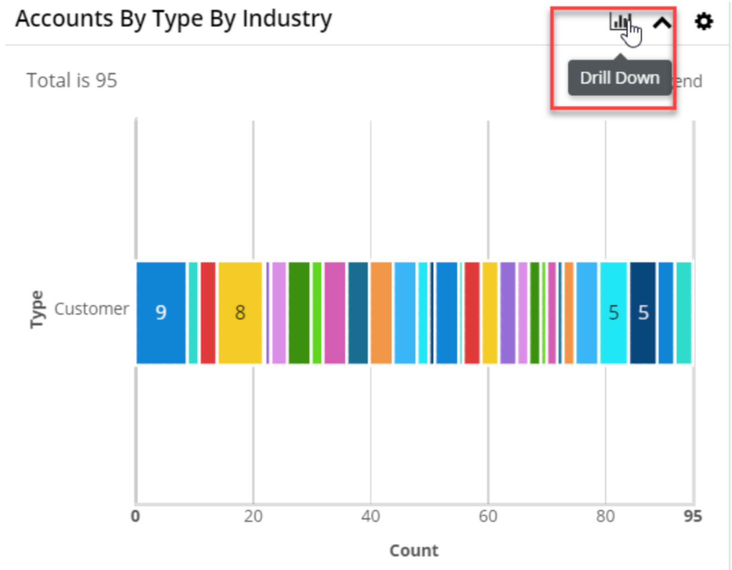
All the columns from "Report group drilldown" drawer are sortable.

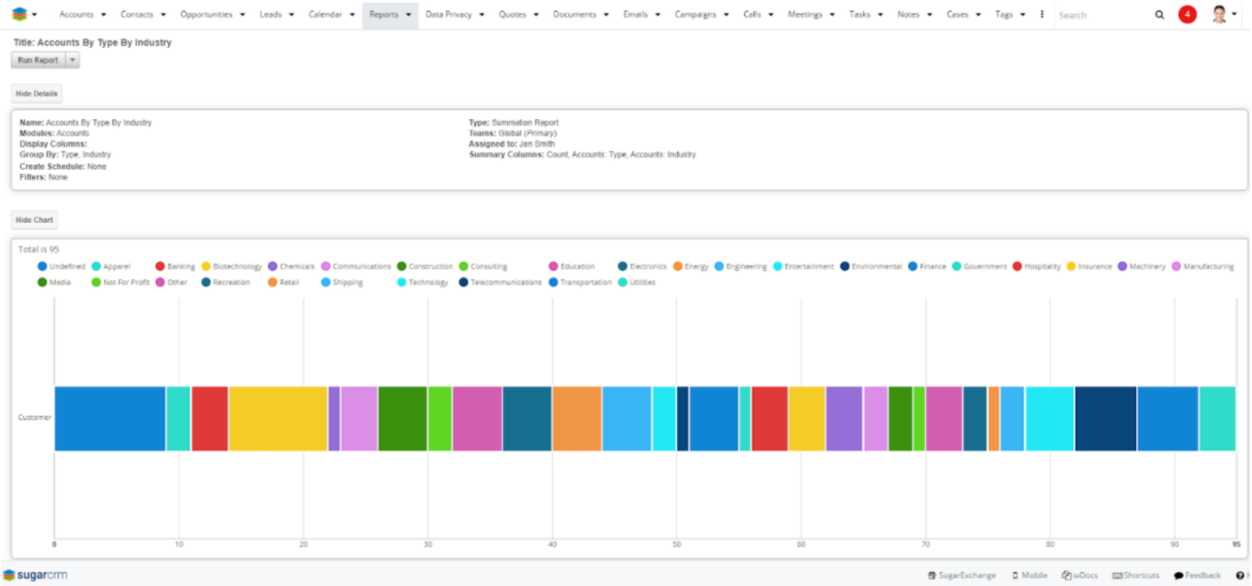
The name and related field types are links and the user is able to click on a link to open a record. Also, the user is able to click on the "Preview" button in order to see more details about the selected record (i.e., only for links).



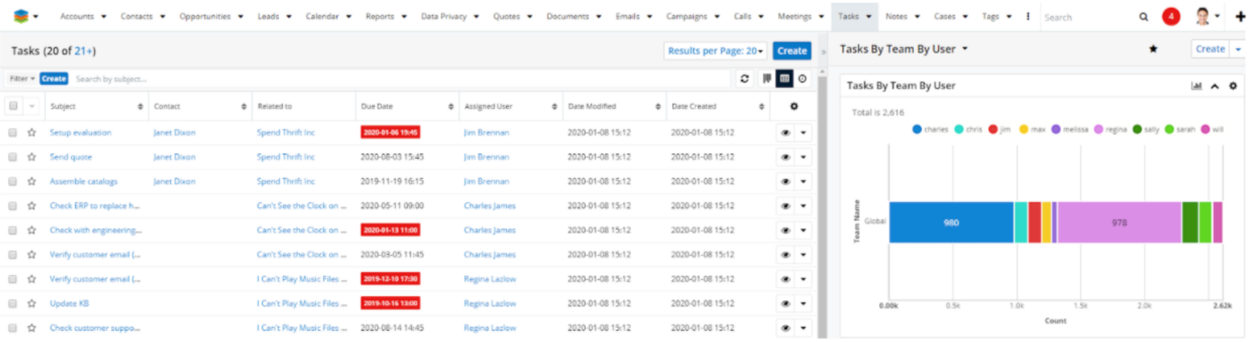
Drill Down button

The Drill Down button found in the upper right side of it is pointing to the current chart's Report record. The upper right side contains the text: ' **Click chart element to see group records**' to make users aware that the functionality exists.

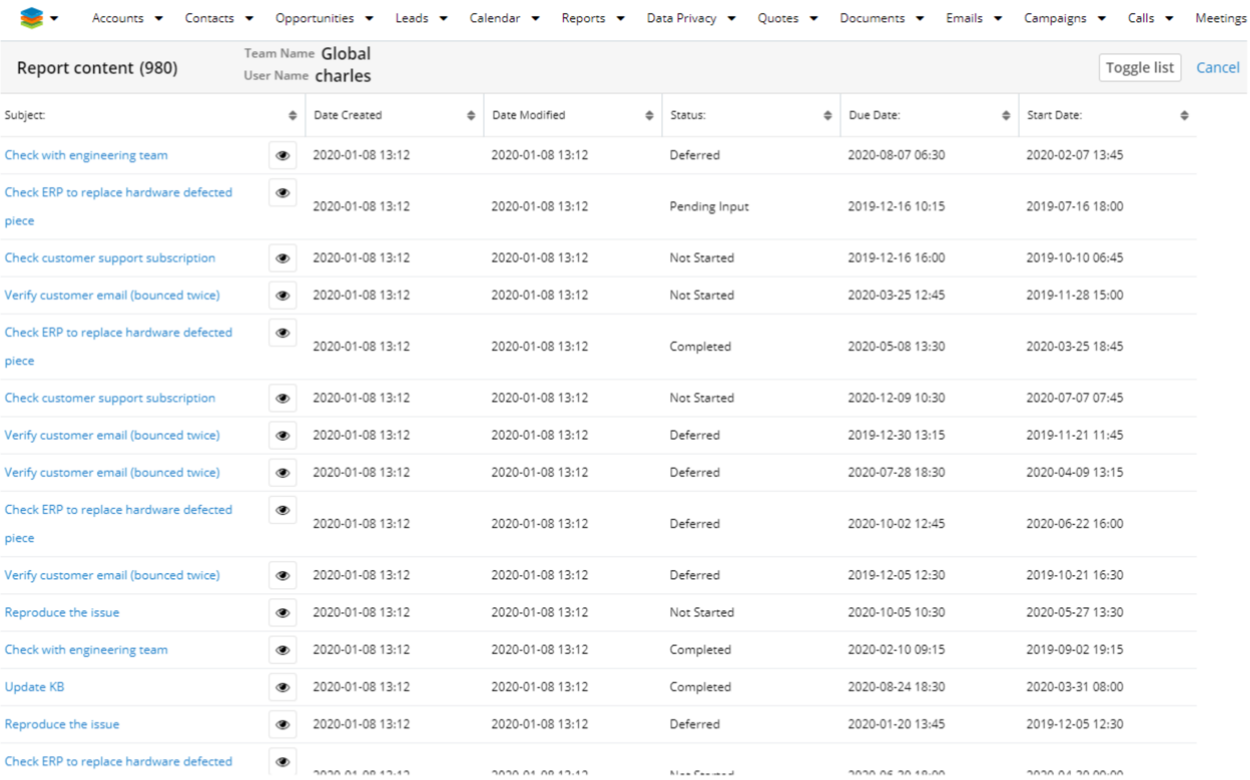


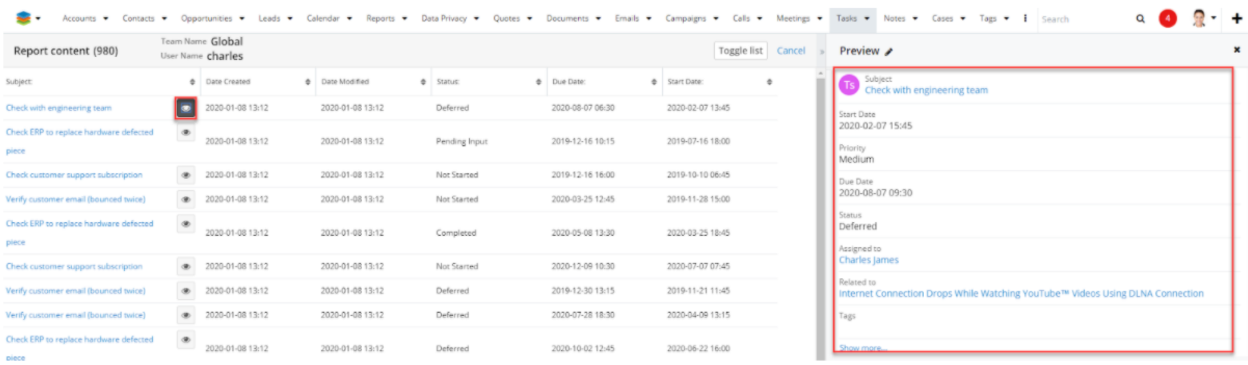


2. List View: 'wChart Dashlet' for the report: "Tasks By Team By User"



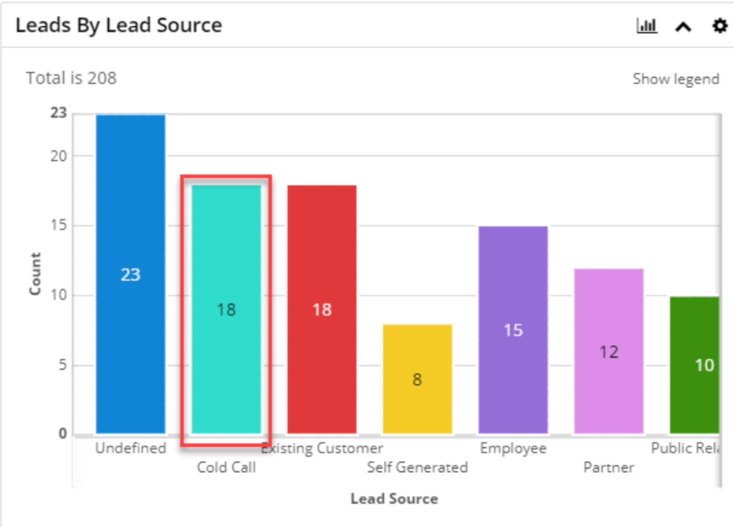
When the user clicks on the section that corresponds to the "Global" Team, the first 20 records will be displayed in the "Report group drilldown" drawer as follows:



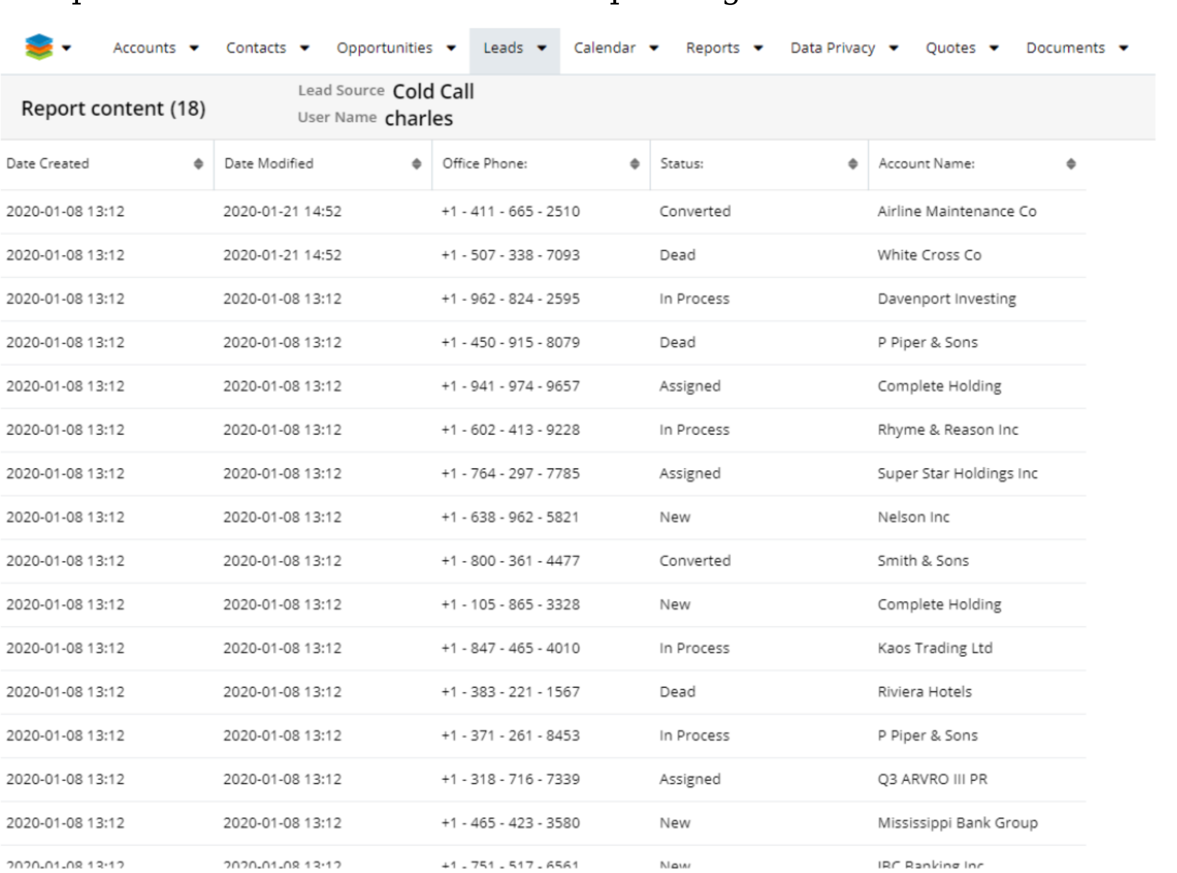


3. Record View: 'wChart Dashlet' for the report "Leads By Leads Source"

In the following example, the Cold Call vertical bar contains the 18 linked leads to the current record that have **Lead Source** equal to **Cold Call**.



If the user clicks on anywhere on the 'Cold Call' bar, the "Report group drilldown" will open with details about the corresponding 18 leads:

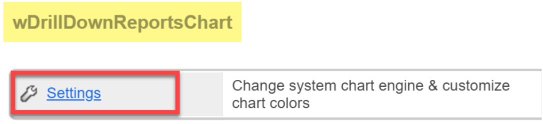


Date Created	Date Modified	Office Phone:	Status:	Account Name:
2020-01-08 13:12	2020-01-21 14:52	+1 - 411 - 665 - 2510	Converted	Airline Maintenance Co
2020-01-08 13:12	2020-01-21 14:52	+1 - 507 - 338 - 7093	Dead	White Cross Co
2020-01-08 13:12	2020-01-08 13:12	+1 - 962 - 824 - 2595	In Process	Davenport Investing
2020-01-08 13:12	2020-01-08 13:12	+1 - 450 - 915 - 8079	Dead	P Piper & Sons
2020-01-08 13:12	2020-01-08 13:12	+1 - 941 - 974 - 9657	Assigned	Complete Holding
2020-01-08 13:12	2020-01-08 13:12	+1 - 602 - 413 - 9228	In Process	Rhyme & Reason Inc
2020-01-08 13:12	2020-01-08 13:12	+1 - 764 - 297 - 7785	Assigned	Super Star Holdings Inc
2020-01-08 13:12	2020-01-08 13:12	+1 - 638 - 962 - 5821	New	Nelson Inc
2020-01-08 13:12	2020-01-08 13:12	+1 - 800 - 361 - 4477	Converted	Smith & Sons
2020-01-08 13:12	2020-01-08 13:12	+1 - 105 - 865 - 3328	New	Complete Holding
2020-01-08 13:12	2020-01-08 13:12	+1 - 847 - 465 - 4010	In Process	Kaos Trading Ltd
2020-01-08 13:12	2020-01-08 13:12	+1 - 383 - 221 - 1567	Dead	Riviera Hotels
2020-01-08 13:12	2020-01-08 13:12	+1 - 371 - 261 - 8453	In Process	P Piper & Sons
2020-01-08 13:12	2020-01-08 13:12	+1 - 318 - 716 - 7339	Assigned	Q3 ARVRO III PR
2020-01-08 13:12	2020-01-08 13:12	+1 - 465 - 423 - 3580	New	Mississippi Bank Group

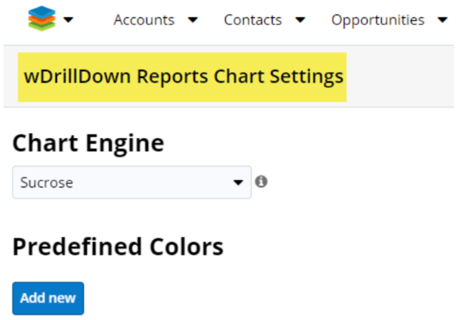
4. Predefined Colors Settings

Note: Starting with Drill Down Report Chart **v 8.x**, the **Predefined Color Settings** option is no longer available.

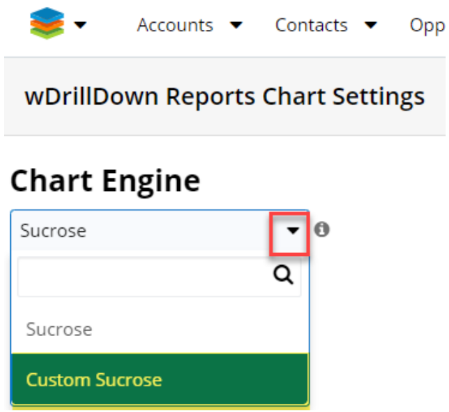
To configure the **Color Settings** for the wChart Dashlet, go to the **Administration** page, and find the **DrillDown Settings** at the bottom of the page. When the package is installed on your Sugar instance, the panel is automatically added to the Administration page.



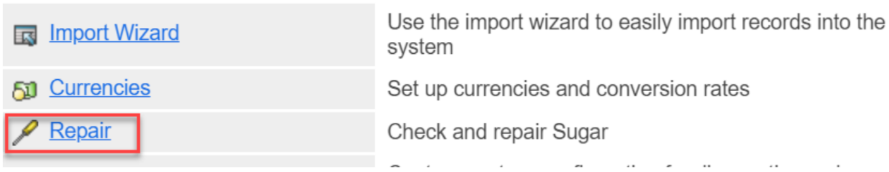
Click the **Settings** link and a new settings page will open.



By default, the color settings for the wDrillDown reports' **Chart Engine** are set to **Sucrose**. Click on the drop down menu and select the **Custom Sucrose** option to change your Color customizations.

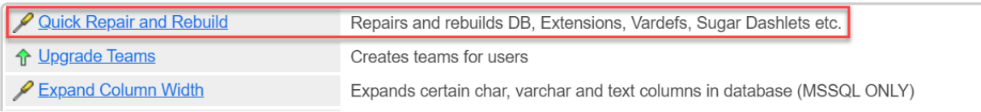


Once you change the **Chart Engine** color settings to **Custom sucrose**, a **Quick Repair and Rebuild** is necessary. Go back to the **Administration** page, to the **System** panel. Find the **Repair** link on the panel and click on it.



Next, perform a **Quick Repair and Rebuild** to make sure the Chart Engine color settings are properly saved on your instance.

Repair

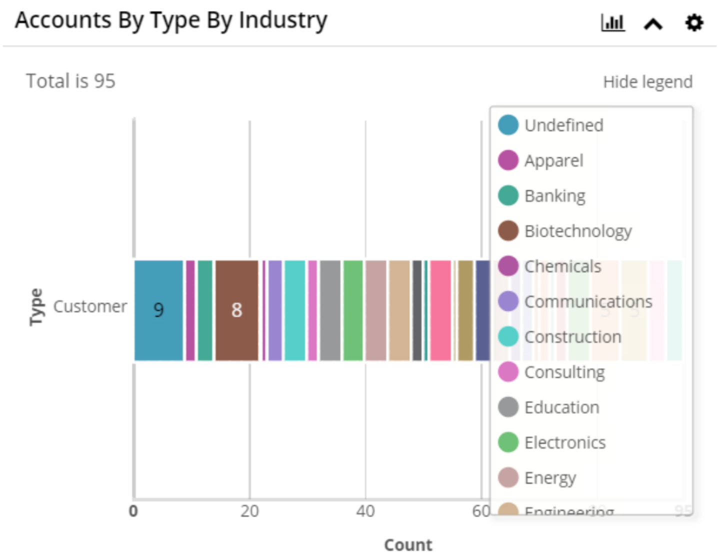


Quick Repair and Rebuild	Repairs and rebuilds DB, Extensions, Vardefs, Sugar Dashlets etc.
Upgrade Teams	Creates teams for users
Expand Column Width	Expands certain char, varchar and text columns in database (MSSQL ONLY)

After the **Quick Repair and Rebuild** action is complete, clear your browser's cache and then perform a browser refresh (see note).

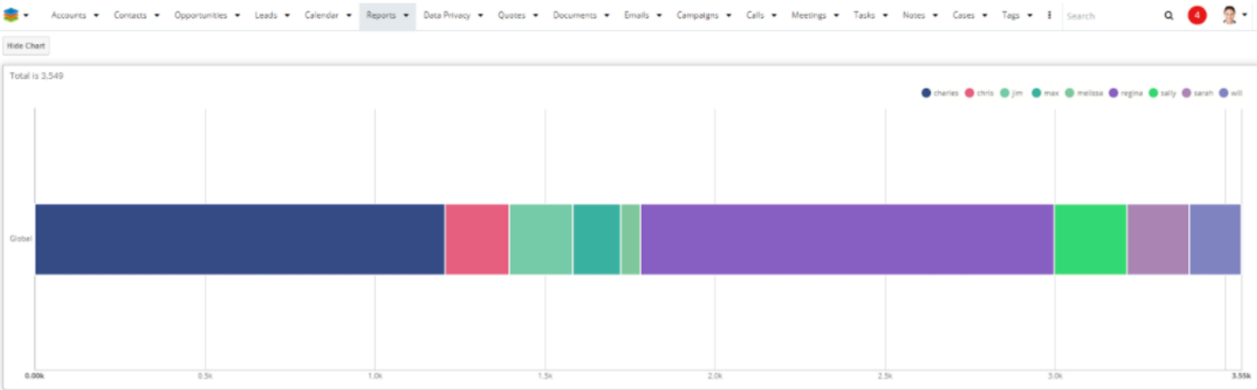
Note: **Clearing the Browser Cache and performing a hard browser refresh**. Please refer to the [Clearing Browser Cache] article for more information.

Once all these steps are complete, Chart colors will change automatically. For example, on "Accounts By Type By Industry" report, the default colors are as follows:

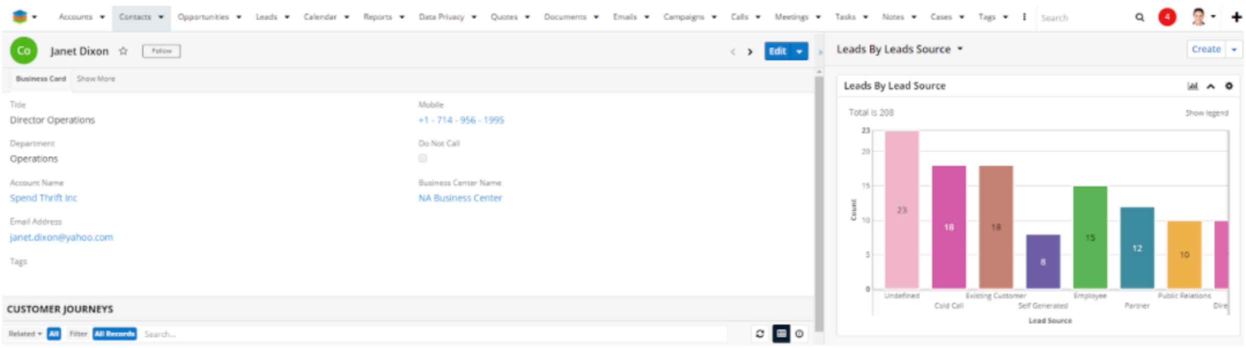


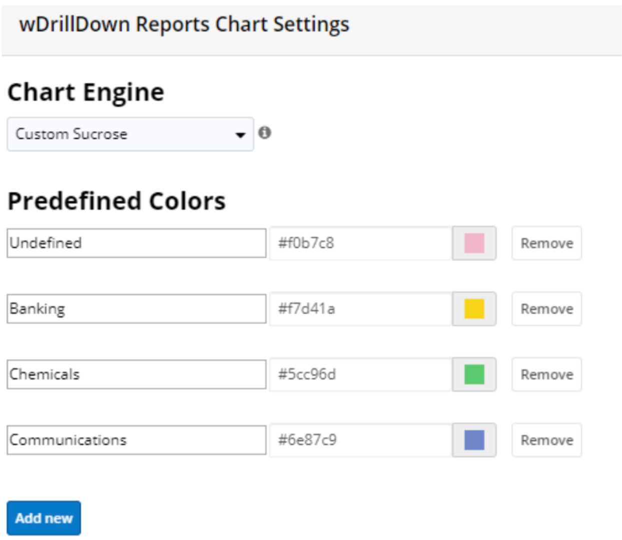
This feature is available for all criteria from reports. Default colors for "Calls By Team By User"

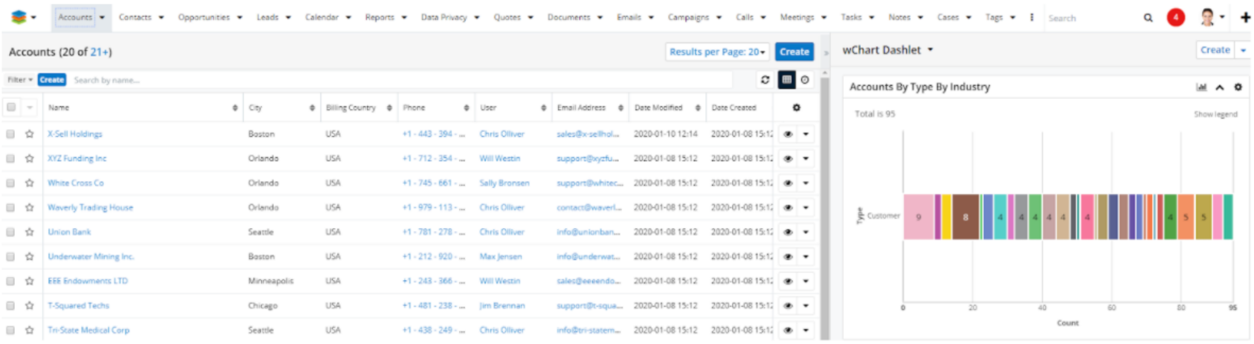
Defined colors for "Calls By Team By User":



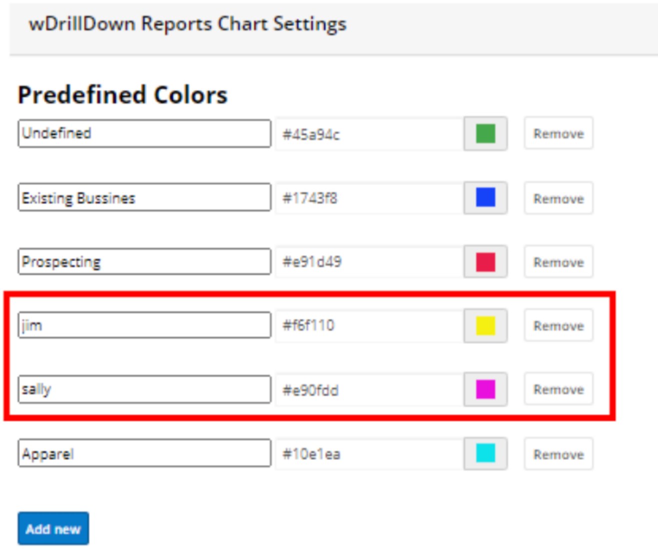
This functionality is the same on the List view and on the Record View:

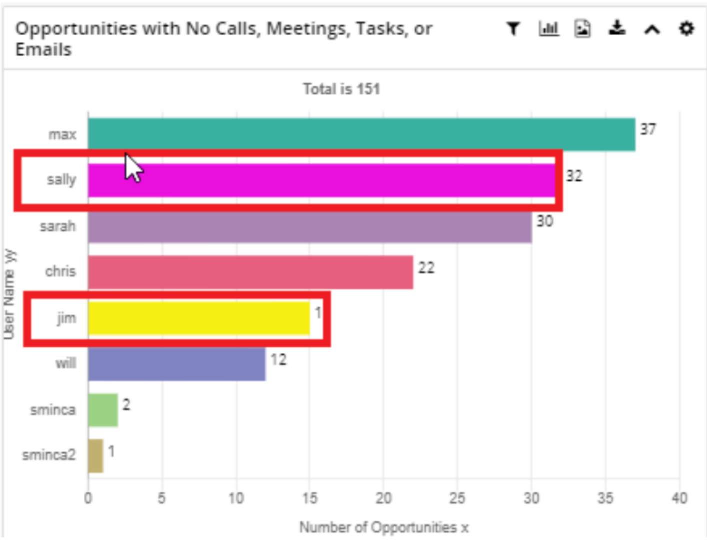






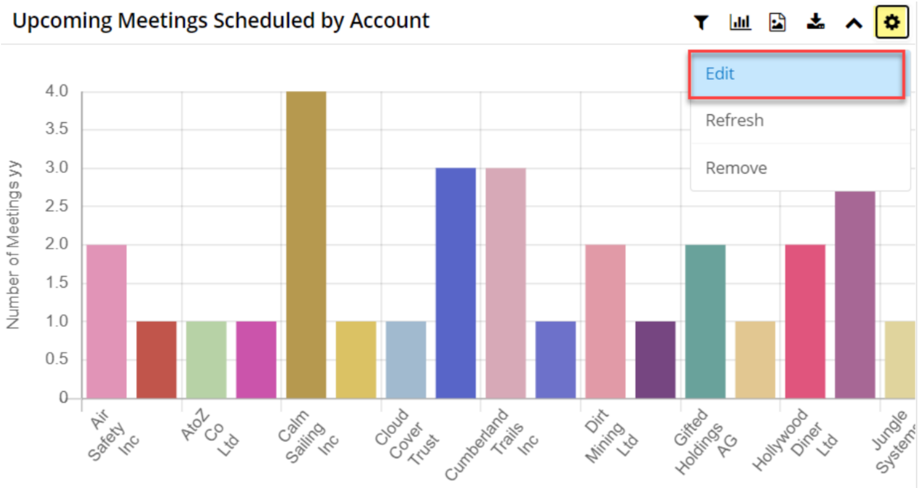
Note: Color names must match chart labels. After adding custom colors, a Quick Rebuild and Repair is needed for the changes to take effect



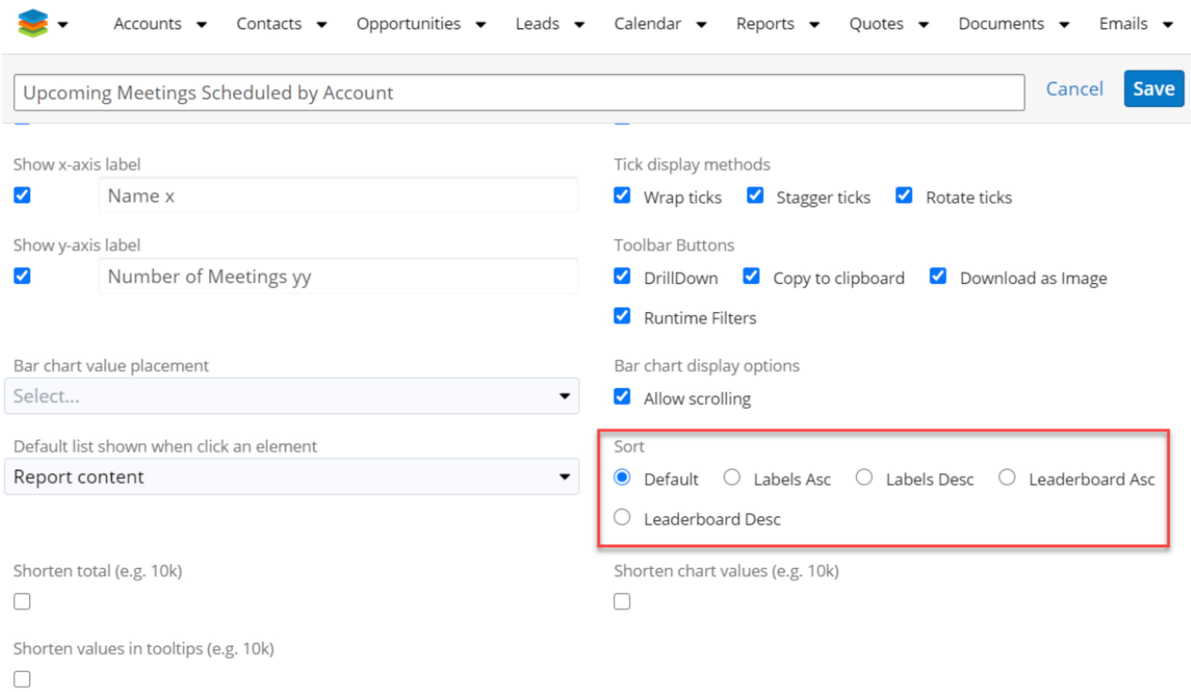


5. Sorting Drill Down Report Chart Data

Chart data and legend can be sorted.



1. Press the **Configure** button at the top of your dashlet.
2. Select **Edit**. A new view with customization options will open.



At the bottom-right side of your screen the **Sort** option will be available. The default sorting option is set to **Default**. This means that the dashlet will display data as it is stored in your report.

Other available sorting options:

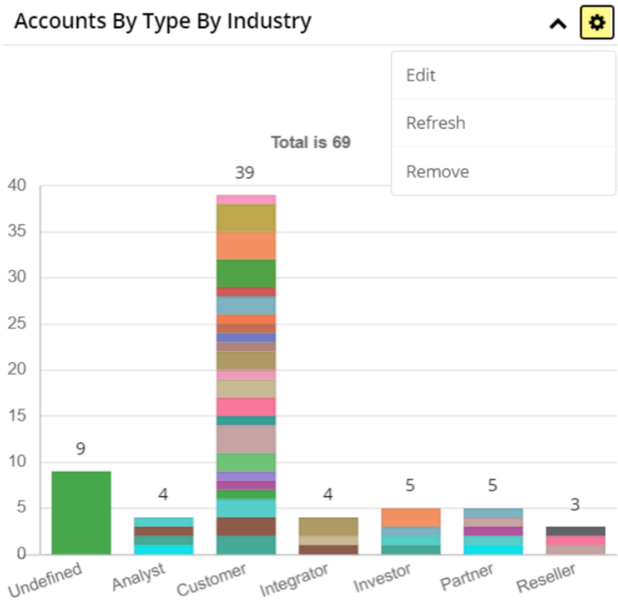
- **Labels Ascending** - Sorts records in ascending alphabetical order (A-Z);

- **Labels Descending** - Sorts records in descending alphabetical order (Z-A);
- **Leaderboard Ascending** - Sorts records depending on value, from the smallest value to the biggest;
- **Leaderboard Descending** - Sorts records depending on value, from the biggest, to the smallest.

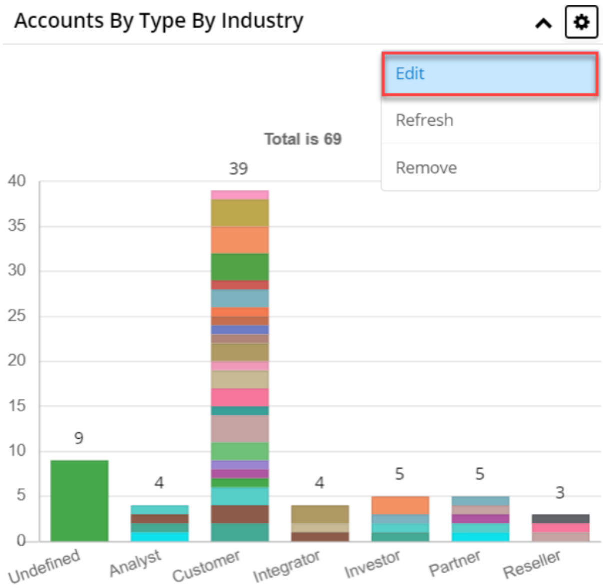
1. Select the desired legend sorting option from the available ones and **Save** your new configuration.

Three chart types support **Legend** sorting as well: **the Line Chart Dashlet, the Horizontal Bar Grouped Chart Dashlet, the Vertical Bar Grouped Chart**.

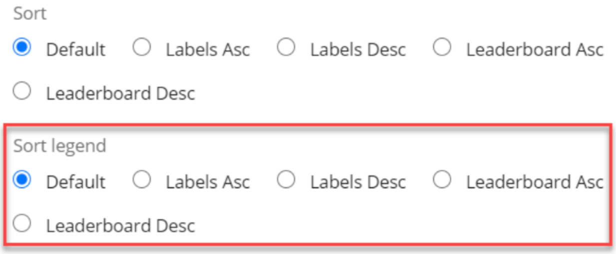
1. To sort chart legend, navigate to any of the chart types above and press the **Configure** button.



2. Select **Edit**.



3. A new view with customization options will open. At the bottom-right side of your new view the **Sort Legend** option is available.



4. Select the desired legend sorting option from the available ones and **Save** your new configuration.

Create Note Function in ListView

In order to quickly add notes to records shown in a List View, this module adds a **Create Note** command to the row's **Action** menu in a Sugar List View. This is useful because it allows the quick creation of a note without having to navigate into the Record View.

Configuring Create Note Function in List View

There is no special configuration required for this package.

For usage instructions, please visit the topic: Using the Create Note Function in ListView

Using Create Note Function in List View

This module adds a **Create Note** command to the row's **Action** menu in a Sugar List View of these modules:

- Accounts
- Contacts
- Opportunities
- Leads
- Calls
- Meetings

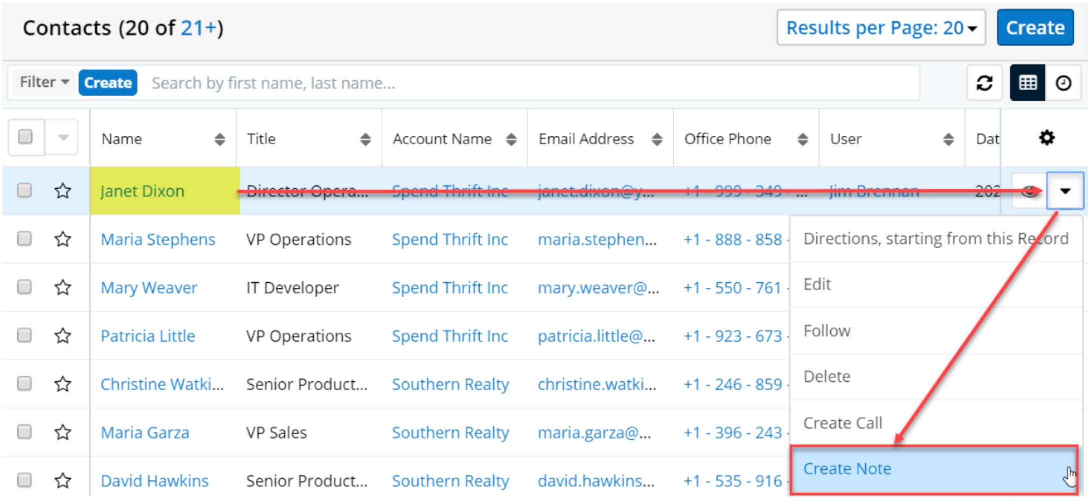
- Tasks
- Cases

- etc.

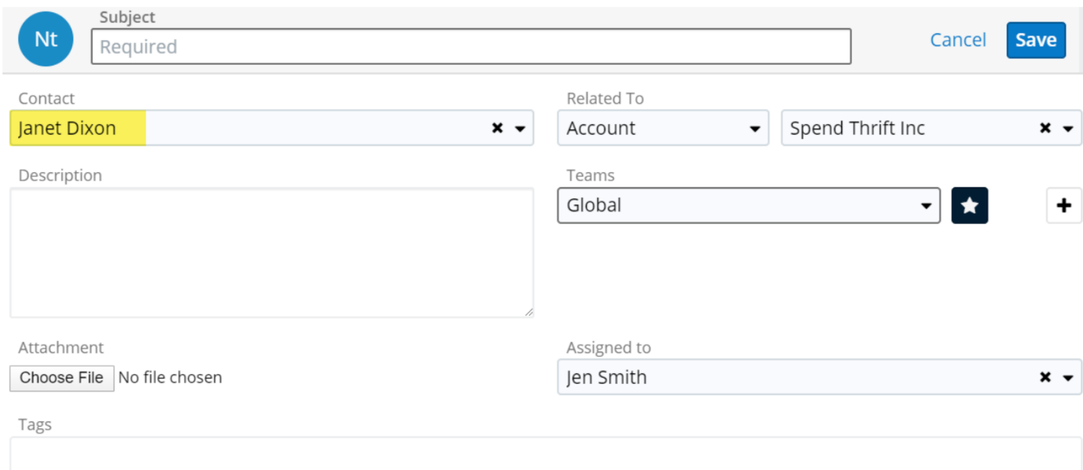
When the **Create Note** command is clicked (from the Actions menu), **Note** creation view will open with the relationships set to the current module and record if a relationship exists between the current List View module and the Notes module. If no relationship exists, then the note will be created without a default relationship.

Example #1: Create Note from Contacts module List View

Here is an example of creating a note from the Contacts List View for a contact named: Janet Dixon. The user clicks the Actions menu drop-down and then clicks Create Note:

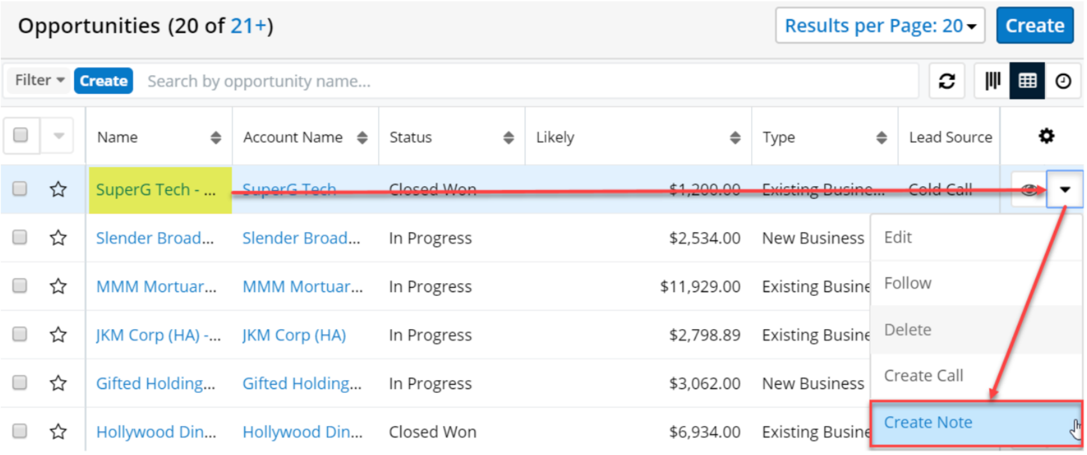


Sugar will display a Note screen with the corresponding name auto populated from the contact record:

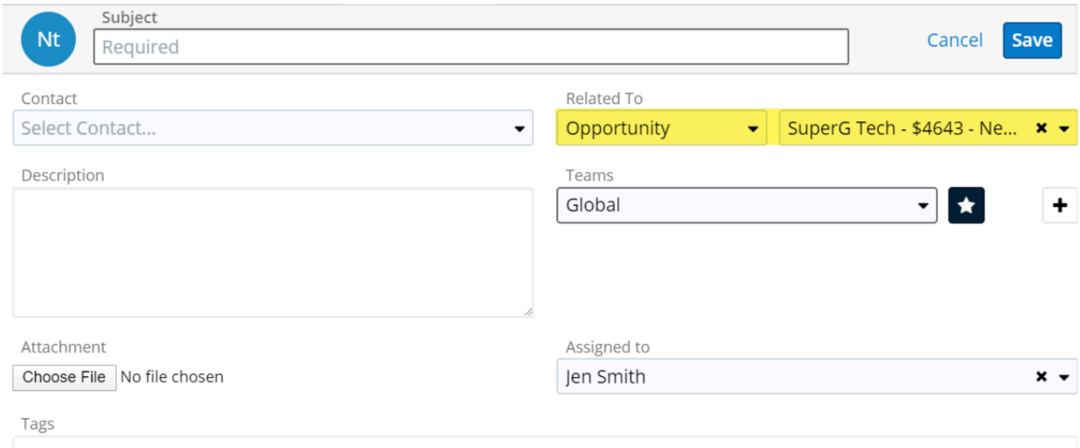


Example #2: Create Note from a module that has a relationship with the Notes module (other than Contacts module)

When a user clicks the **Create Note** command from the List View on modules like Account, Opportunity, Task, Lead, Call, etc., the **Relate To** field from the Note will be auto-populated with the corresponding module name and the field next to it will be auto-populated with the corresponding record name. In this example, the Opportunities module was used:

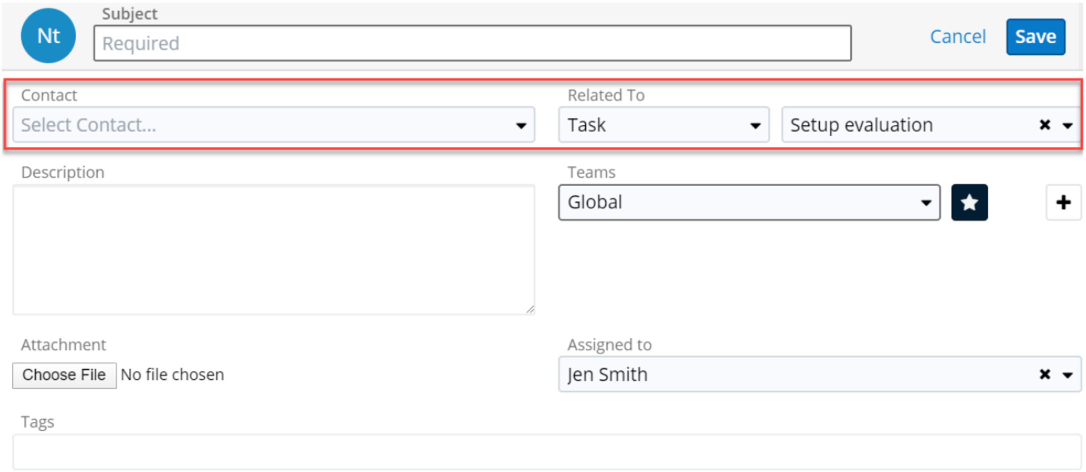


Sugar will display a Note screen with the corresponding **Related To** and **Name** value auto-populated from the opportunity record:



Example #3: Create Note from a module List View that has no relationship with the Notes module

When a user clicks the **Create Note** command from the List View on module that does not have a relationship with the Notes module, the **Note** creation page will open with no relationship set:



Custom Notification Field Type

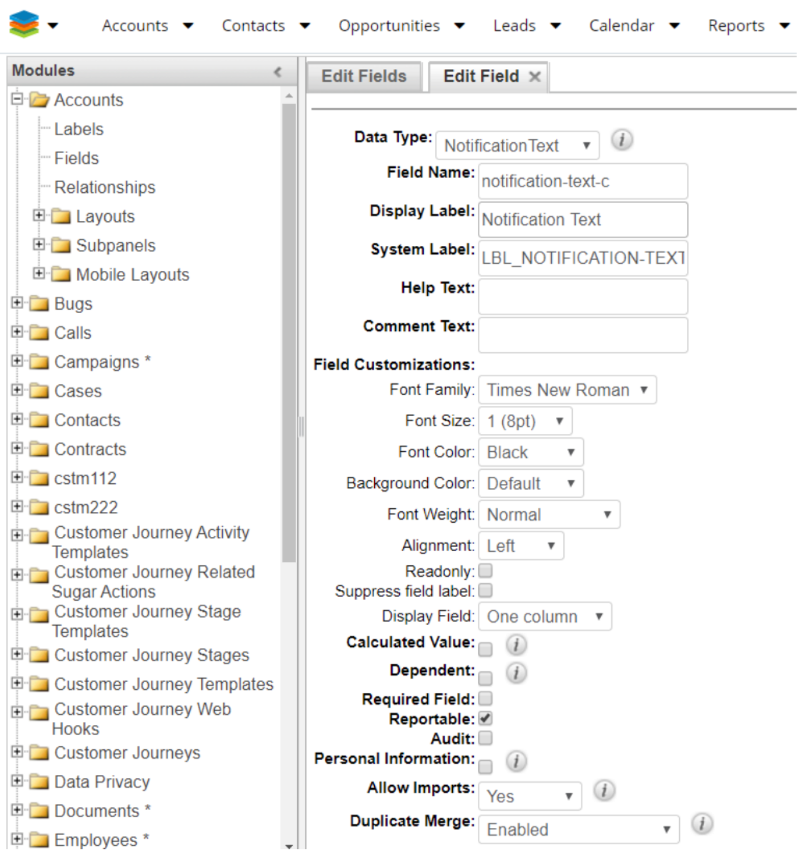
This loadable module creates a new field type that allows you to format specific text on a record view to make it more noticeable to the reader. For instance if you want to bring special instructions or alerts to the record reader you may want to increase the text font size, change the font color or font background.

When you load this module and afterward use the Sugar Studio tool to create a new field, you will notice a new " **Notification Text**" field type. This field type behaves like a **Text Field**, but adds formatting options for **Font Family, Font Size, Font Color, Background Color, Font Weight and Alignment**. In addition it allows the field to be a read only field, to suppress any field label and also to optionally stretch the field over a single or multiple columns in the layout.

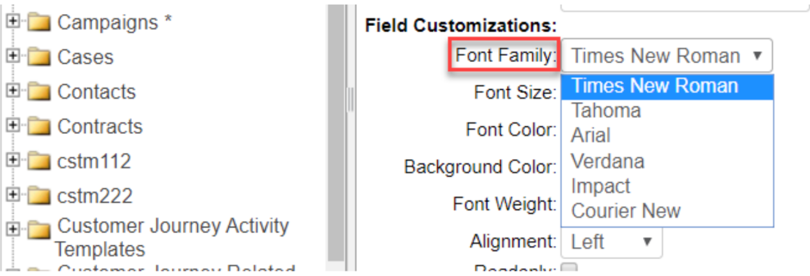
Using Custom Notification Field Type

The **Custom Notification Field Type** package creates a new type field that allows users to bring special instructions or alerts to the record reader, to increase the text font size, change the font color, or change the font background.

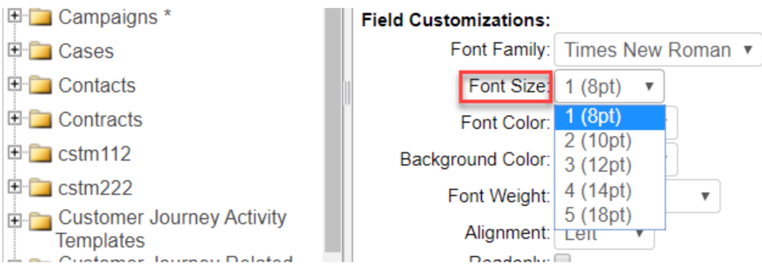
Screenshot of a "**NotificationText**" field type being added/edited:



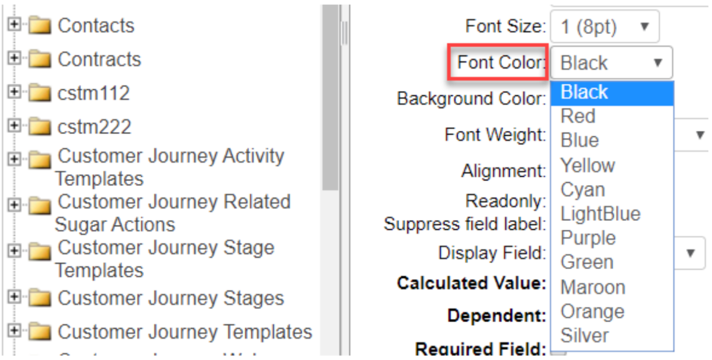
- **Font Family**: This drop-down provides the following values: Times New Roman, Tahoma, Arial, Verdana, Impact, and Courier New. This option allows users to choose which type of font will be used by the field:



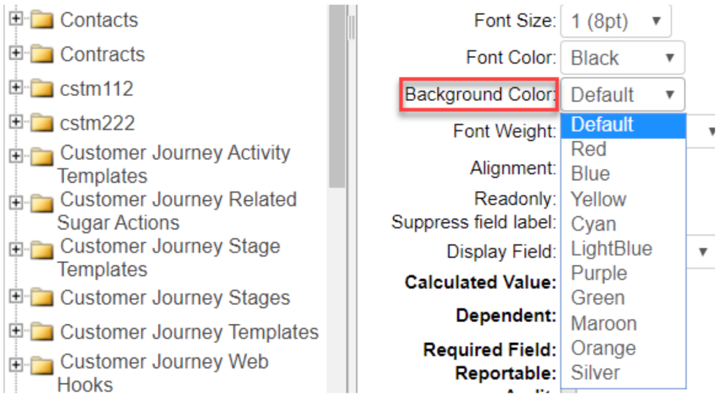
- **Font Size**: This drop-down provides the following values for the size of the font the field will display: 1 (8pt), 2 (10pt), 3 (12pt), 4 (14pt), 5 (18pt), 6 (24pt), 7 (36pt).



- **Font Color**: This drop-down provides the following values for selecting the color of the displayed font: default, red, blue, yellow, cyan, light blue, purple, green, maroon, orange, and silver.



- **Background Color**: This drop-down provides the following values: default, red, blue, yellow, cyan, light blue, purple, green, maroon, orange, silver. This option allows users to choose which color of background they want to use.



Colored Fields

The **Colored Fields** module offers the possibility to create new fields that where you can set the same background color from another field. Users can also select a new color individually for each record in Edit View.

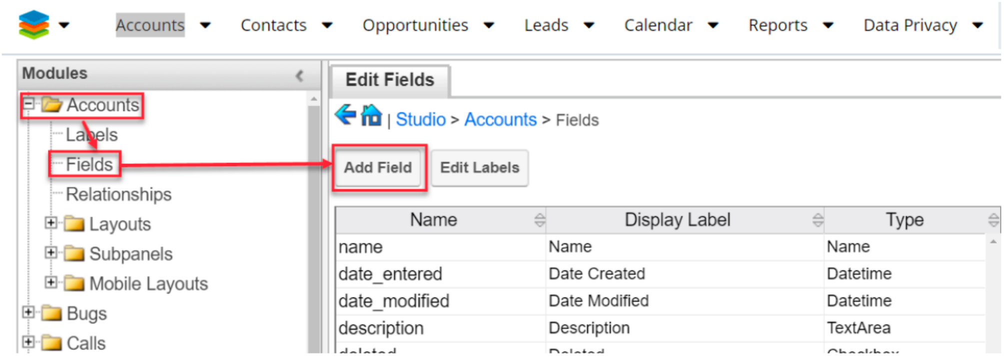
Using the Colored Fields Module

The **Colored Fields** module offers the possibility to create new fields that where you can set the same background color from another field. Users can also select a new color individually for each record in Edit View.

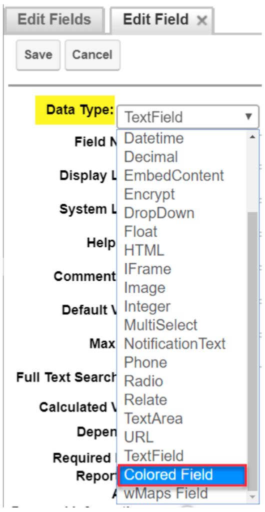
After installing the package and performing the installation steps, the admin user will be able to create fields of type "**Colored Field**" from Studio.

1. Get Background Color from Field

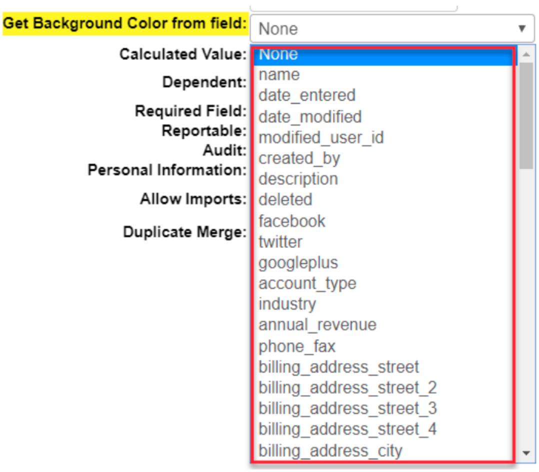
To set the background color from another field in, go to the **Administration** and click the **Studio** link found in the **Developer Tools** panel.



In this example, we will add a new field for the " **Get Background Color from Field**" functionality. Push the **Add Field** button and change the Data Type from **Text Field** to **Colored Field** from the dropdown menu.



Once you select the **Colored Field** Data Type and give your field a name (colored_field_c), you have to pick the field from where the background color is borrowed from. By default, the value is set to "**None**".



Pick your field of choice, but make sure it holds data type of any of the following.

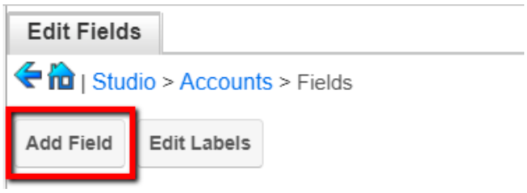
- TextField
- Dropdown
- Radio
- Text Area

Now **Save** your changes. The functionality is now ready to use.

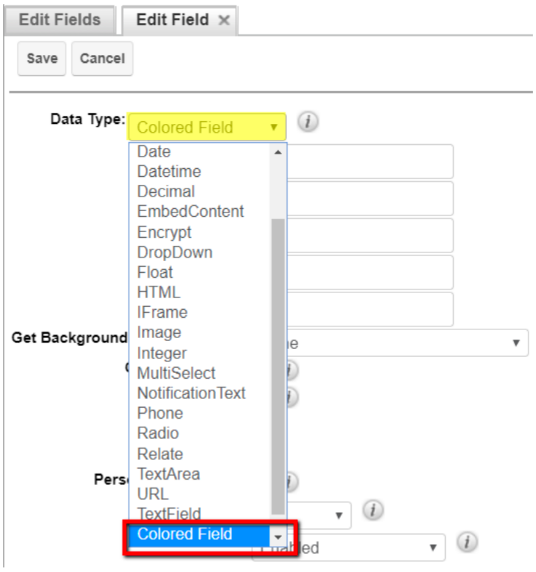
The **Get Background Color from Field** module can be set up to either allow you to use a color picker, or to set the background color from another field where the color you want can be found.

- **Color Field Color Picker**

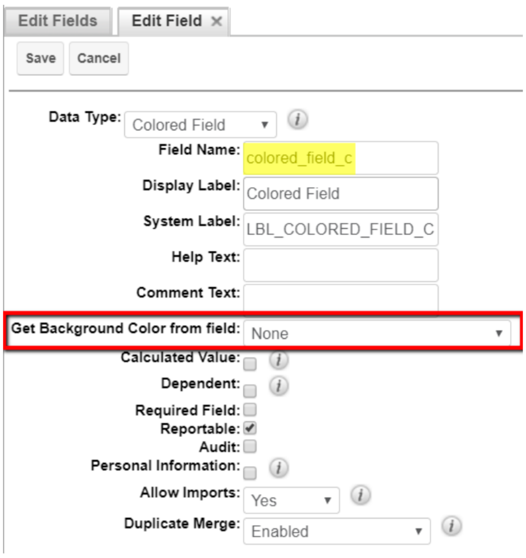
This functionality allows users to set the desired color in Edit View. To set it up, navigate to **Studio**, in the **Administration** panel. Go to the desired module (Account in our example) and select the **Fields** button. **Add a new Field**.



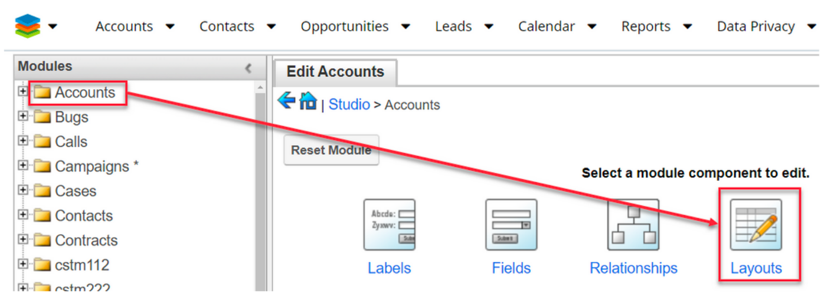
On the Edit Field screen, set the Data Type to Colored Field.



Fill the **Field Name f**ield with "colored_field_c" and set the Display Label to **Colored Field**.

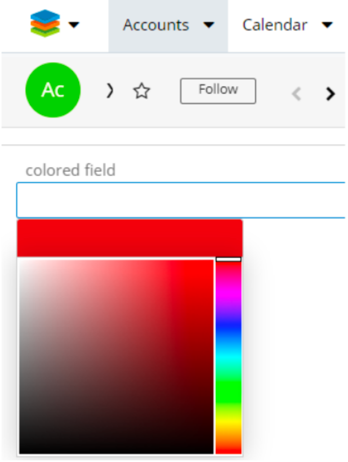


The **Get Background from Field** default field value should remain unchanged, set to **None**. In Studio, go back to your module of choice and make the field visible in Record View and List View from **Layouts**.

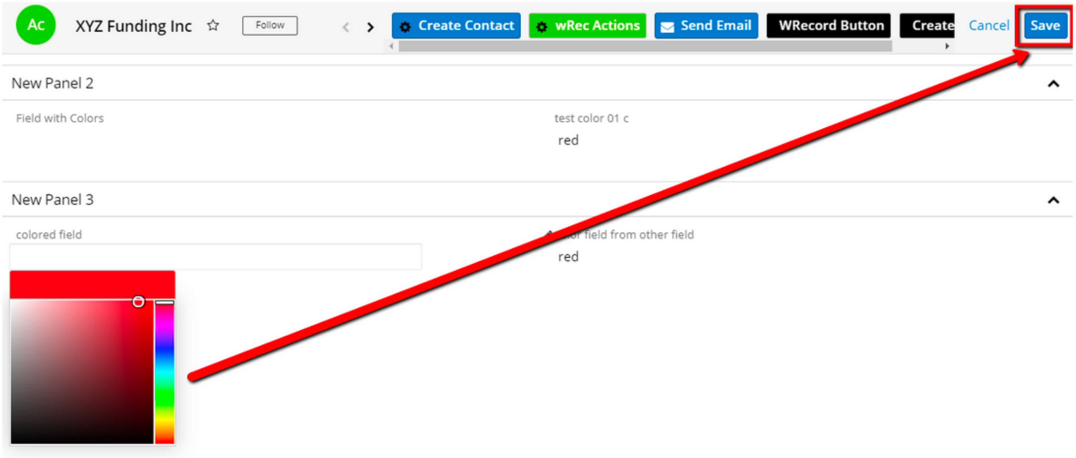


In **Layouts Record View**, create a new row and drag and drop the "**Colored Field**" filler from the left panel. **Save & Deploy** your changes. The changes made in **Record View** are automatically saved for **List View** as well. The colored field can now be used.

Go to the module you edited previously in **Edit View** and select a color from the color picker next to **Colored Field**.

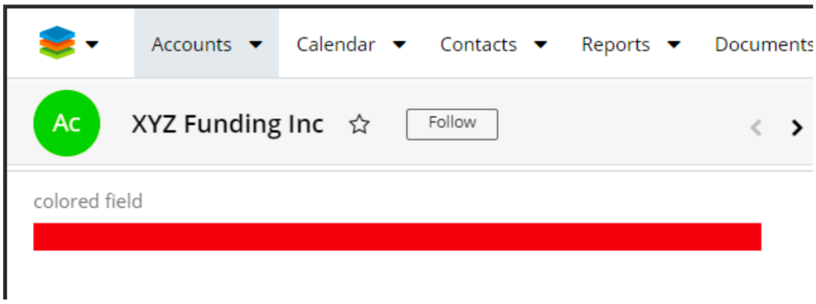


Save the changes you've made.

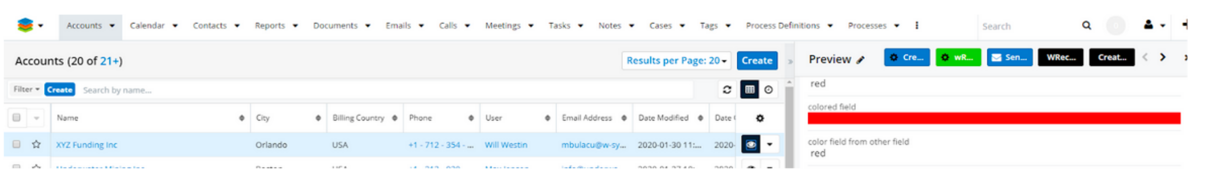


Now the background color picked will be visible in both **Record View** and **List View**.

- Record View:

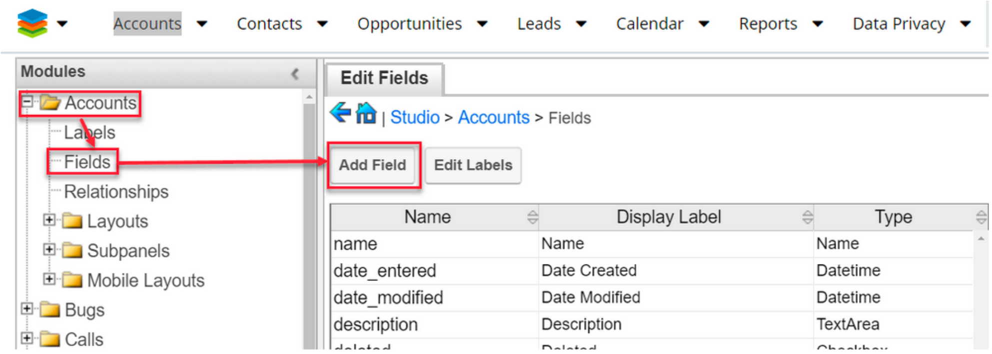


- Preview (in List View):

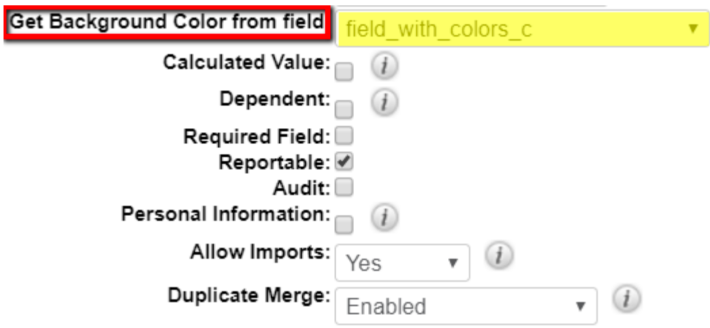


- **Colored Field with Color From Other Fields**

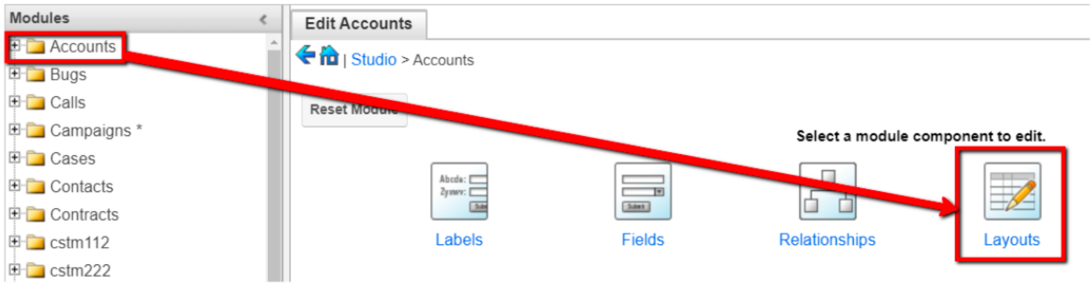
To enable the **Colored Field with Color from Other Fields** functionality, you have to go to **Studio** on the **Administration** page, and create a new custom field for the module of choice. In this example, we will work on the Accounts module.



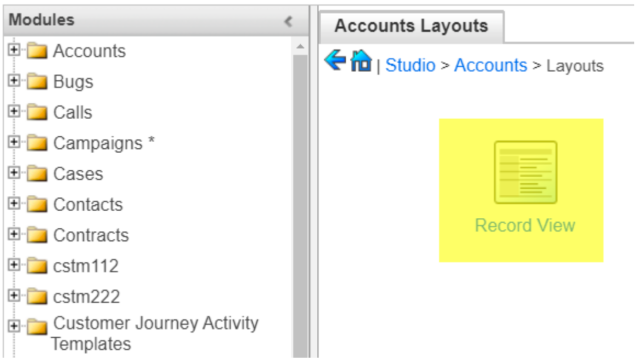
1. On the **Edit Field** panel, first create a new **Text Field** and name the new field "field_with_colors". set the label to Field with Colors and **Save** the new custom field.
2. Go Back to the **Fields** panel in **Studio** on your module of choice, and press the **Add Field** button. Create a new field with the Data Type set to "**Colored Field**". Name the field "colored_field_with_other_field" and set the label to **Colored Field From Other Field**. Now, set the color from the "**Get Background Color from field**" dropdown menu to "field_with_colors". **Save** your changes.



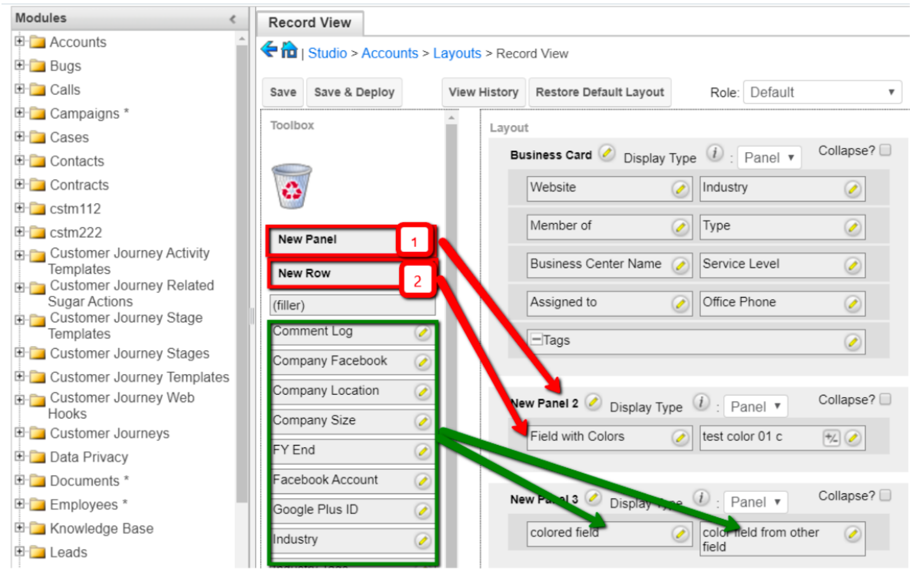
3. To make the changes visible in both **Record View** and **List View**, go to your module of choice in **Studio** and select **Layouts**.



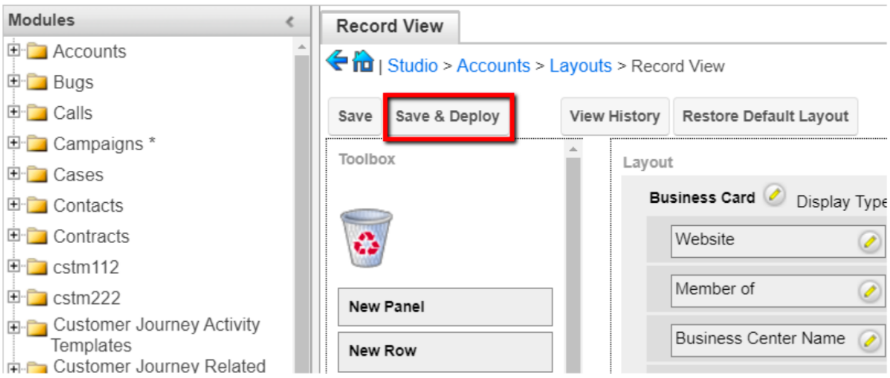
4. Now, select the **Record View** panel and make your changes visible.



5. Add a **New Panel** and rows (**New Row**) by dragging and dropping them from the left side of the menu. Drag and drop filler fields to make them visible in **Record View**. In this case, drag and drop the **Field With Colors** and the **Color Field from Other Field** fillers.

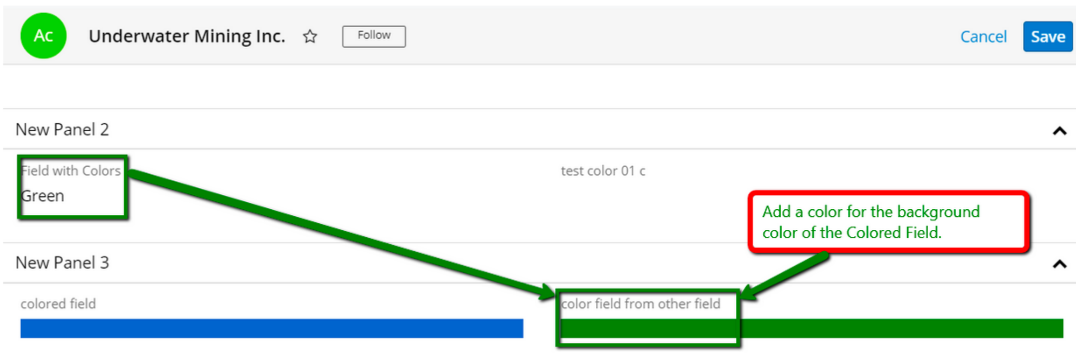


6. Click **Save & Deploy** to save your changes.

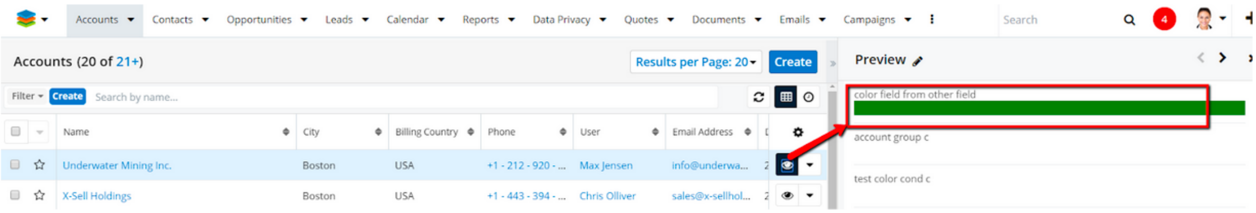


Usually, the layout changes made in **Record View** are autosaved to **List View** as well.

The changes you made are now applicable in your module of choice, in both **Record View** and **List View**. To change the background color, type a color name in the field that was selected from the "Get Background Color from Field" dropdown menu, in our case, the Field with Color field.



When you **Save** the changes, you will notice the field changing its color to the one you previously selected. The function is available in List View as well, by using the Preview button.



- **Colored Field Based on Condition**

To create colored fields based on conditions, you have to go in **Studio** and create three separate fields.

- account_group_c (Text);
- test_color_cond_c (Text);
- test_color_01_c (ColorField).

Each of the custom fields above will have a different **Data Type** and formula, as it follows:

1. account_grpund_c. Select the **Data Type** to **Text** and Save your changes (no formula for this field). **Save** the changes.
2. test_color_cond_c. Select the **Data Type** to **Text** and use the following formula: ifElse(equal($account_group_c,"A"),"red",

ifElse(equal($account_group_c,"B"),"green",

ifElse(equal($account_group_c,"C"),"yellow",

ifElse(equal($account_group_c,"D"),"blue",

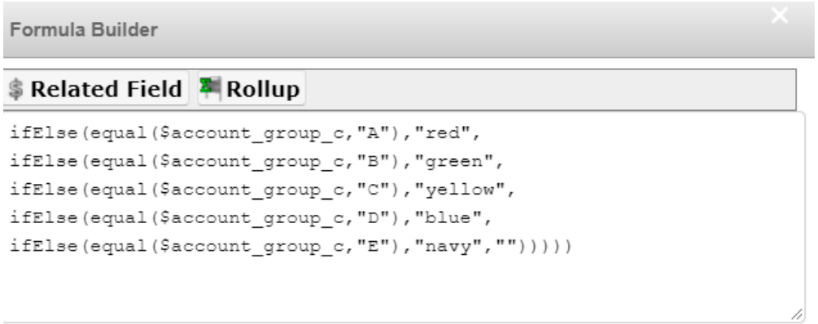
ifElse(equal($account_group_c,"E"),"navy","")))))

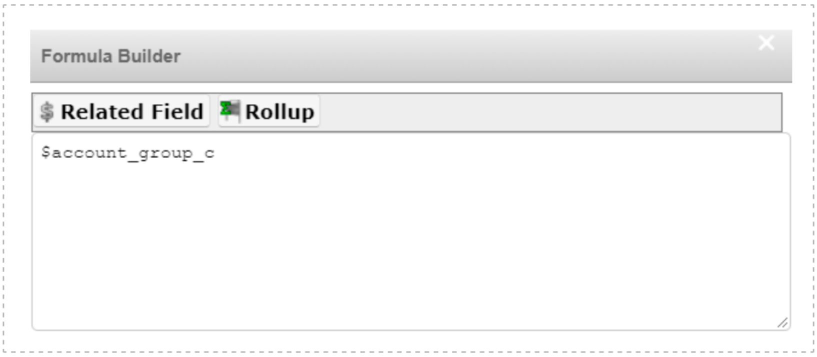
Save your formula and the newly created field.

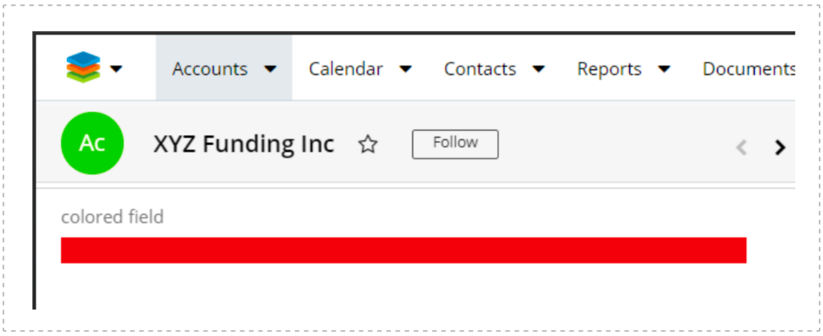
1. test_color_01_c. Set the **Data Type** to **Colored Field** and use the formula: $account_group_c. Choose the background color from the Get Background Color from the Field dropdown menu: test_color_cond_c.

Save your field.

Here is what the property values of the "test_color_cond_c" and "test_color_01_c" look like.







Custom Buttons

Custom Buttons allows users to add custom buttons to their Sugar interface for easy interactions and custom abilities. **Custom Buttons** is a module that will be added to Sugar modules from **Studio**.

Note: The Custom Buttons solution was integrated into Sugar as the Action Button field type starting with the 11.1 release. Please refer to the [Sugar Administration Guide] for more information.

Configuring the Custom Buttons Package

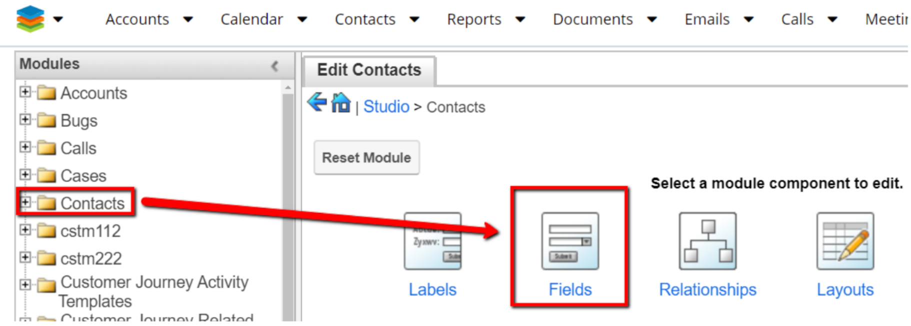
No configuration is necessary for the **Custom Buttons** module between installing and using it. See [Create Custom Buttons] for additional information.

Using the Custom Buttons Package

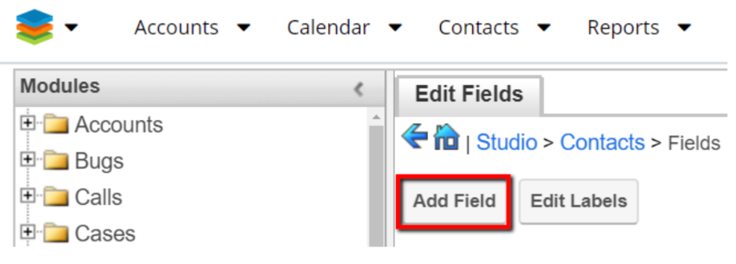
Custom Buttons can be used from the following views in Sugar: **Record View** (including header), **List View**, **Dashlet List View**, **Preview and Header Preview**. The package allows users to easily perform actions and commands without accessing additional menus.

Create Custom Buttons

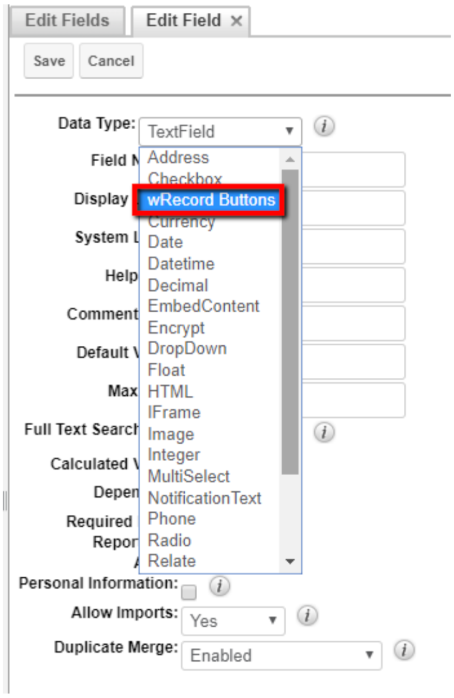
To create a **Custom Button** in Sugar, go to the **Developer Tools** section in **Administration a**nd click on **Studio**. In Studio, select the module where you want to create Custom Buttons, and click on Fields.



Push the **Add Field** button.

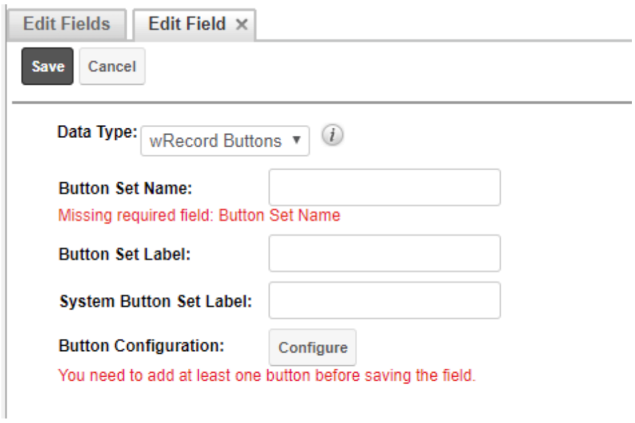


Set the **Data Type** field to **wRecord Buttons**.

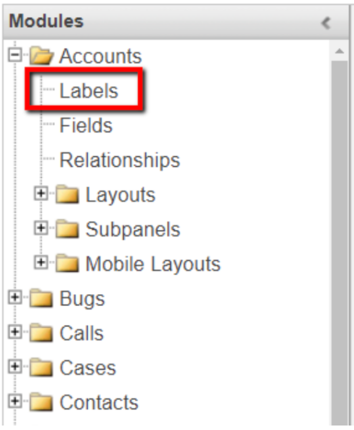


The new field has the following properties in **Studio**.

- **Button Set Name**. Users have to set a name for the new custom button. The field supports alphanumeric characters and underscore. Users must ensure their **Button Set Name** ends in _c.

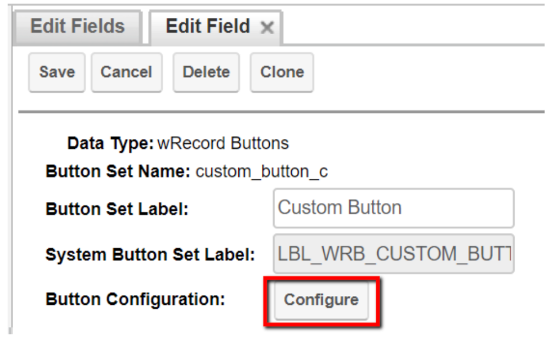


- **System Button Set Label**. Once given a name, the **Button Set Label** field, a label will be automatically attributed, preceded by "LBL_". Once created, the label cannot be changed.



Note: Users should not use the same System Button Set Label for multiple buttons to avoid similar values for different buttons in Layouts.

- **Button Configuration**. Before saving their newly created buttons, these have to be configured.

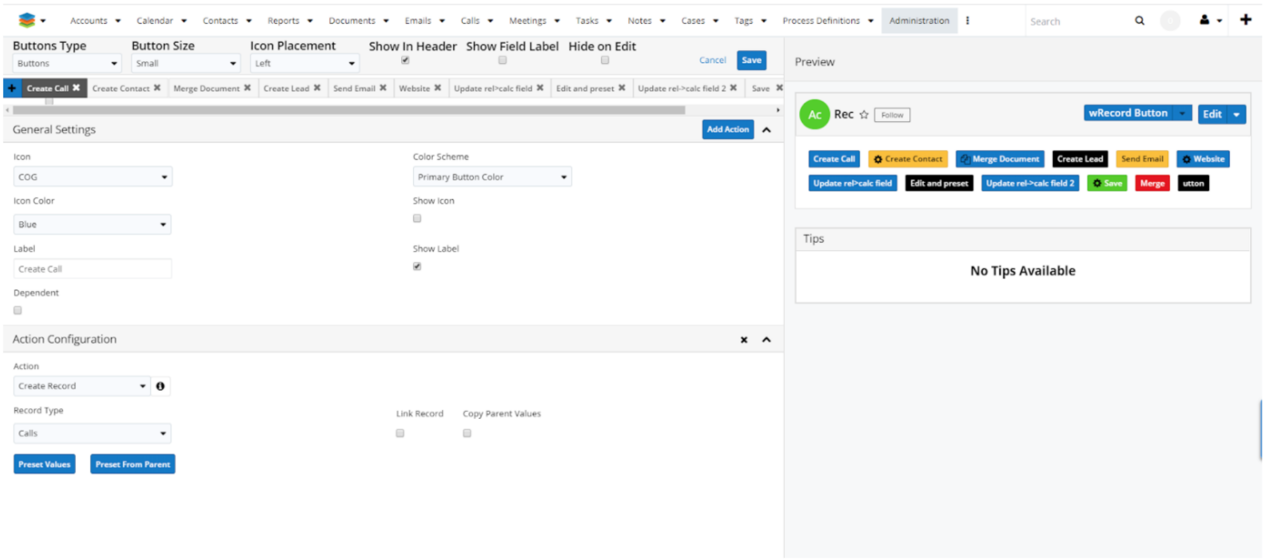


Note: Saving buttons without configuring them previously will not keep any of the changes.

Configuring Custom Buttons

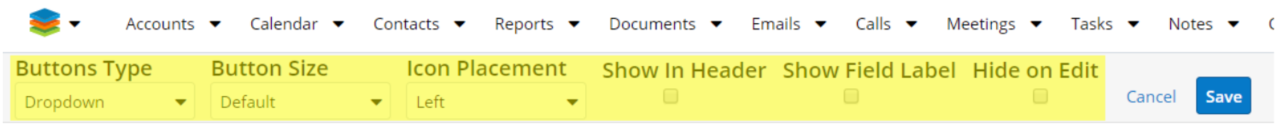
Custom Buttons comes with a series of personalization options.

Pressing the **Configure** button in **Edit Field** view will open a new window with customization options.



Custom Buttons Setup Options

Users have more configuration options in the new **Custom Buttons Setup** page header.



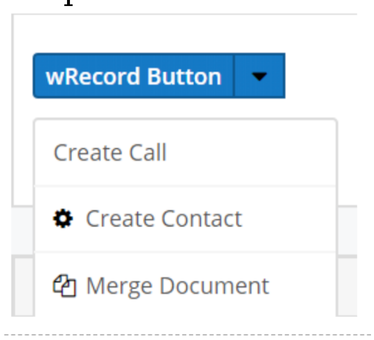
- **Button Type:**
  - Buttons:

  - Button Groups:

  - Dropdown:

- **Button Size:**

  - Default:

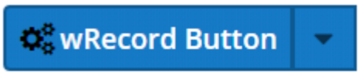
  - Small:

  - Large:

- **Icon Placement :**
  - Left:

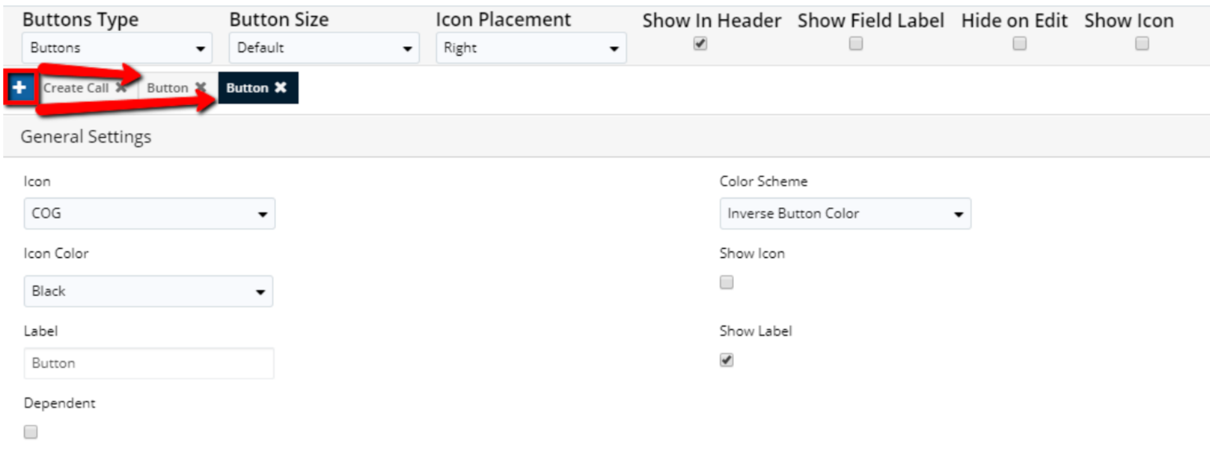
    

  - Right:

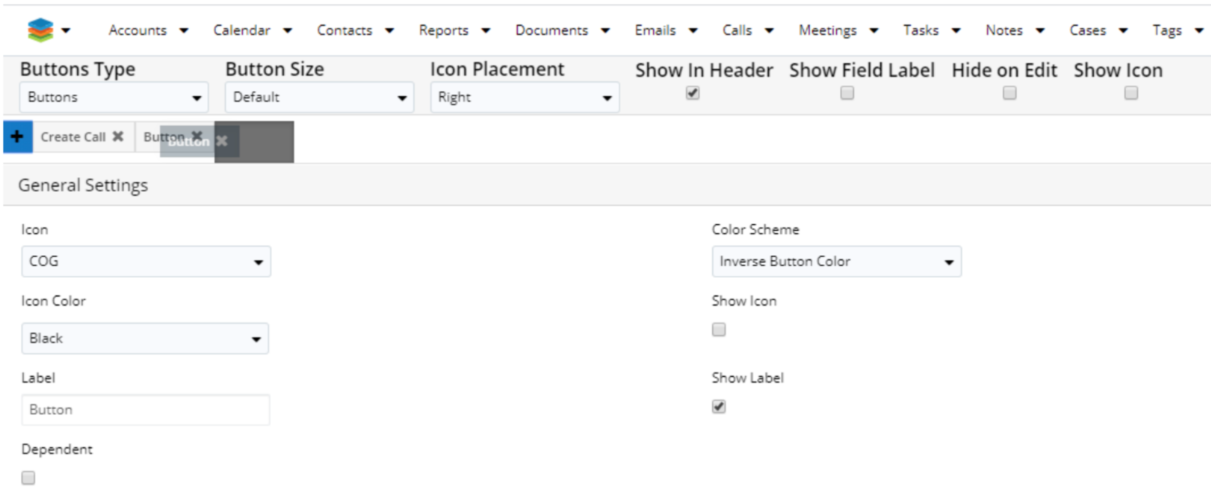
    

  - **Add Button**

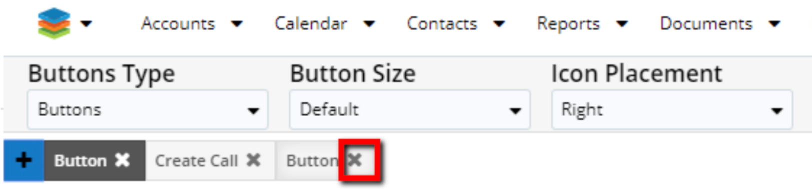
This will add a new button to Custom Buttons.



Users can also drag and drop button placement for further customization options.

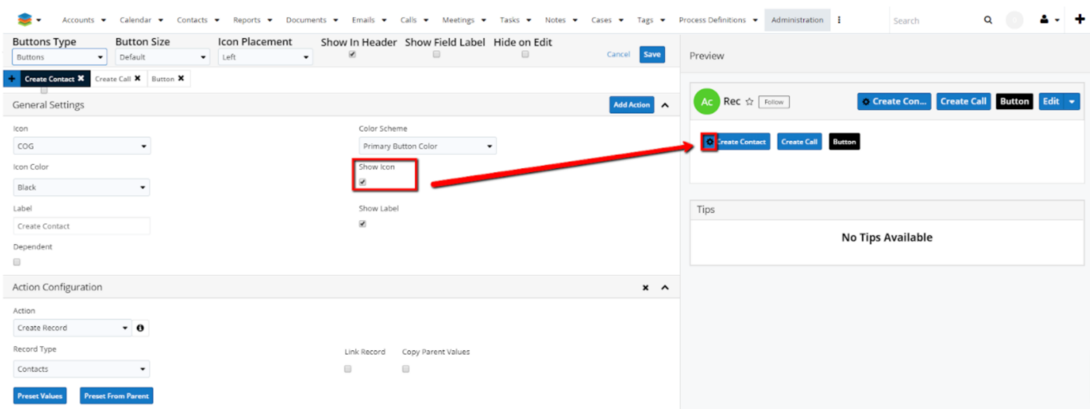


Users can also remove unwanted buttons by clicking the X icon on each button.

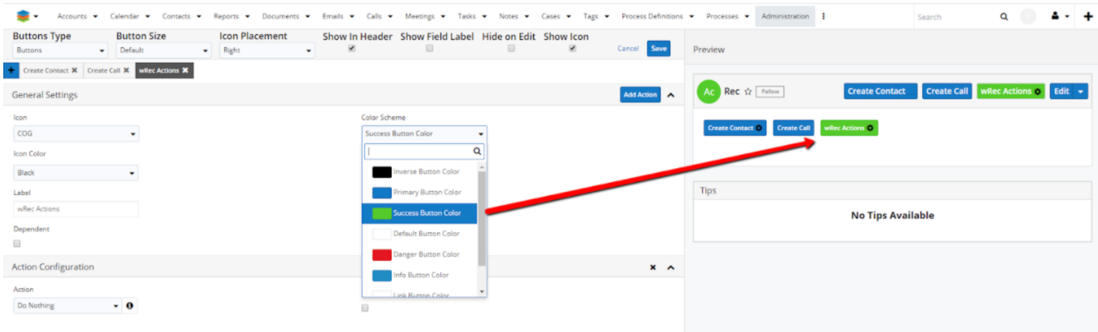


The **General Settings** panel in the **Configuring** window also supports different actions.

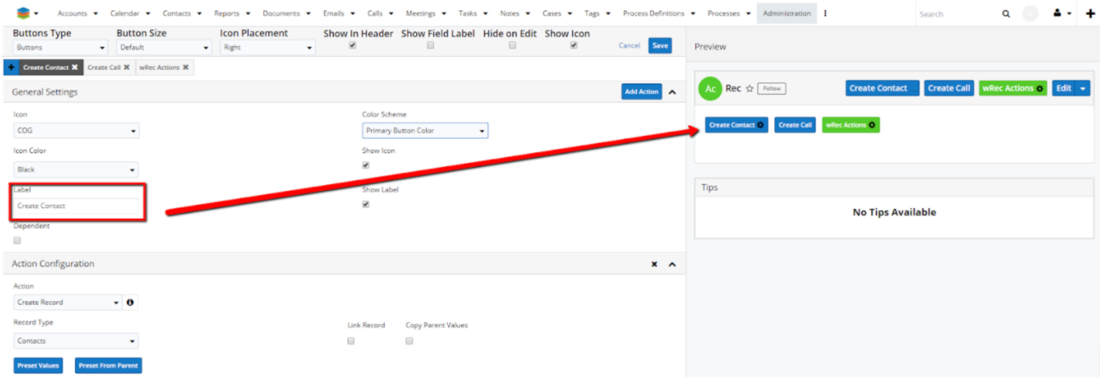
- **Show Icon.** When ticked, the **Show Icon** checkbox will make visible the button's selected icon.



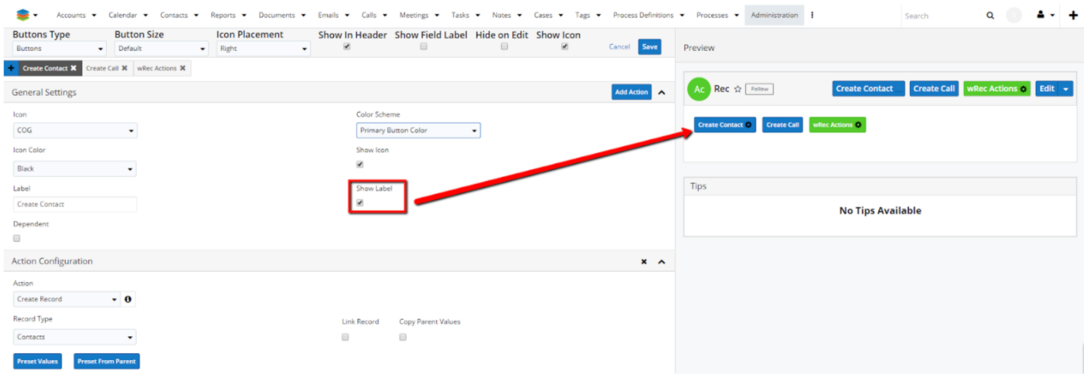
- **Color Scheme**. Each Custom Button can be set up to its particular color.



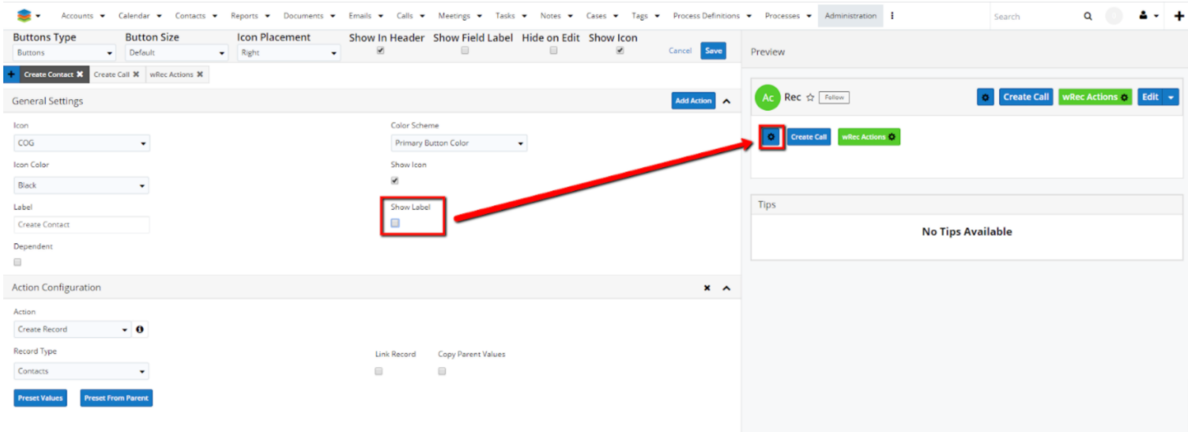
◦ **Label**. Users can set a different label for each button they create.

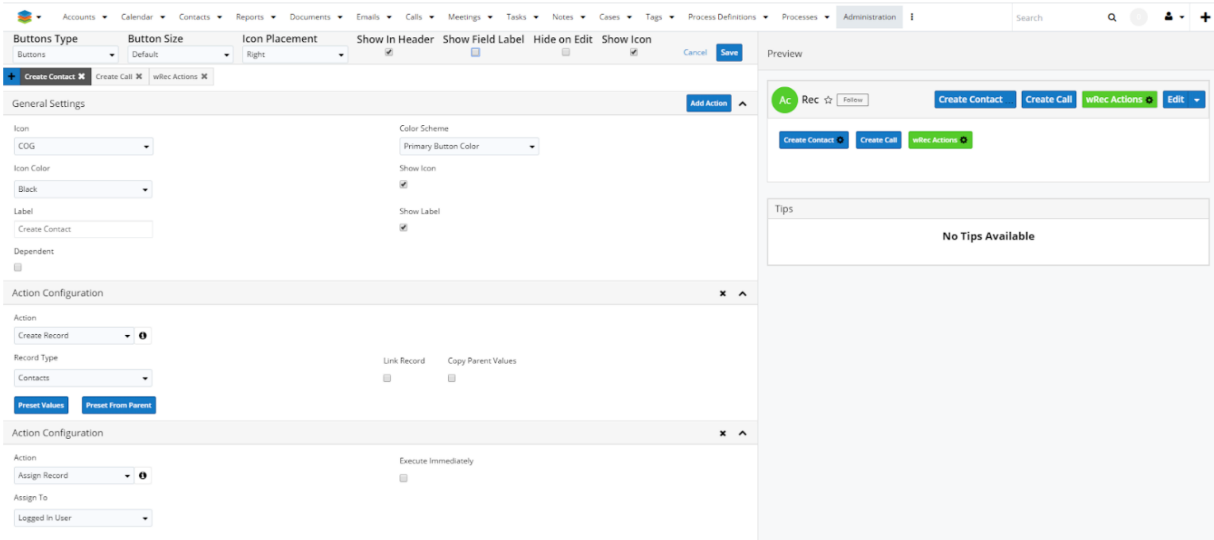


◦ **Show Label**. When ticked, the Show Label checkbox will show the attributed label to each button.



Note: When disabled, the button will only have attributed the icon, without a label, like in the following screenshot:





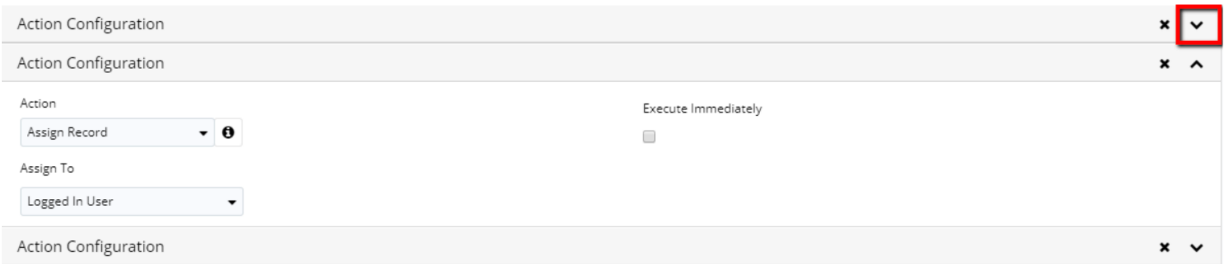
- **Dependent**. Ticking the **Dependent** checkbox in the **General Settings** panel will show or hide **Custom Buttons,** only if the conditions in the formula are met.
- **Actions**. Users can choose what happens when the Custom Button is pushed. The following Actions are supported:

1. Assign Record
2. Create Record
3. Run URL
4. Merge Document (only for wDocs users)
5. Convert Lead (only within the Leads module)
6. Update Field
7. Compose Email
8. Tabs/Panel Visibility Control
9. Run Job
10. Edit Record
11. Save Record

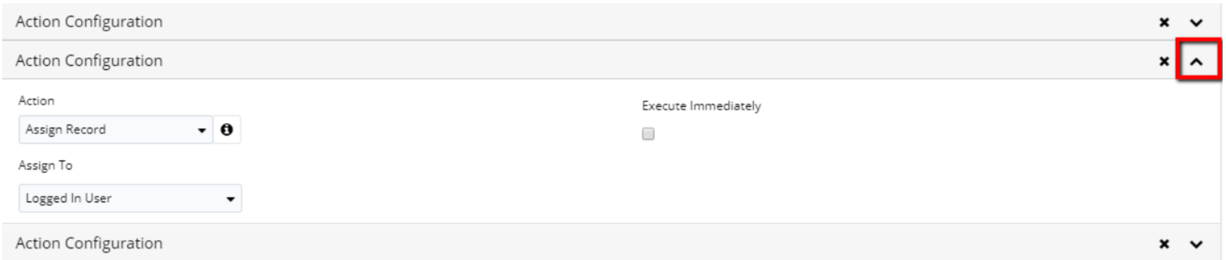
See the [Actions](#) section for usage and configuration information and tips.

Each action panel has the following options available.

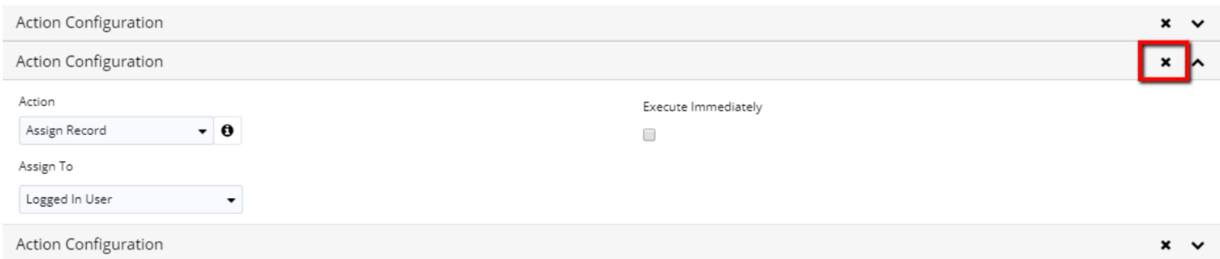
- Expand Panel



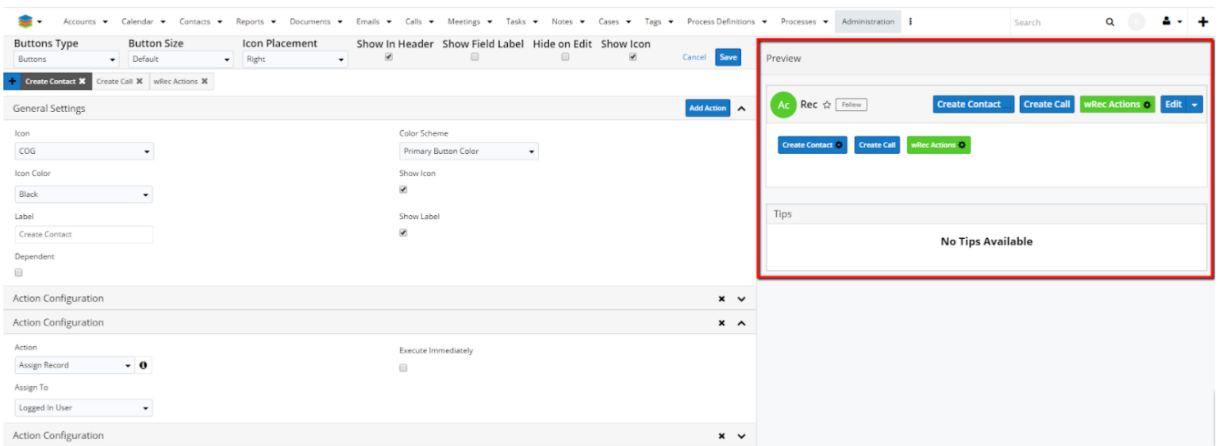
-
- Collapse Panel



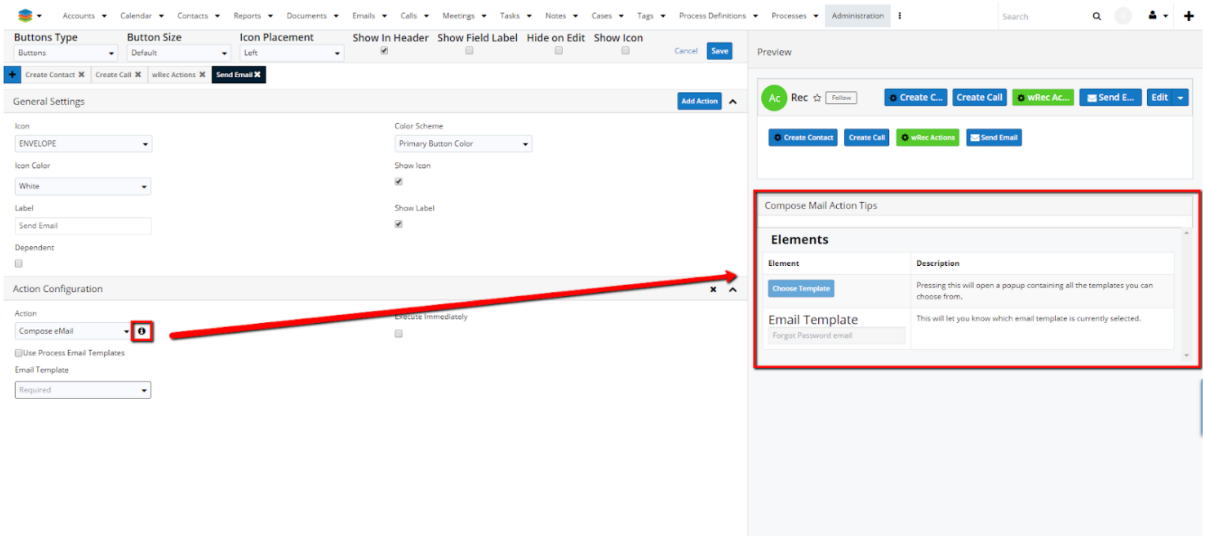
-
- Delete Unwanted Action



-
- **Preview**. The **Preview** function allows users to preview how the Custom Buttons they create look like, before saving.

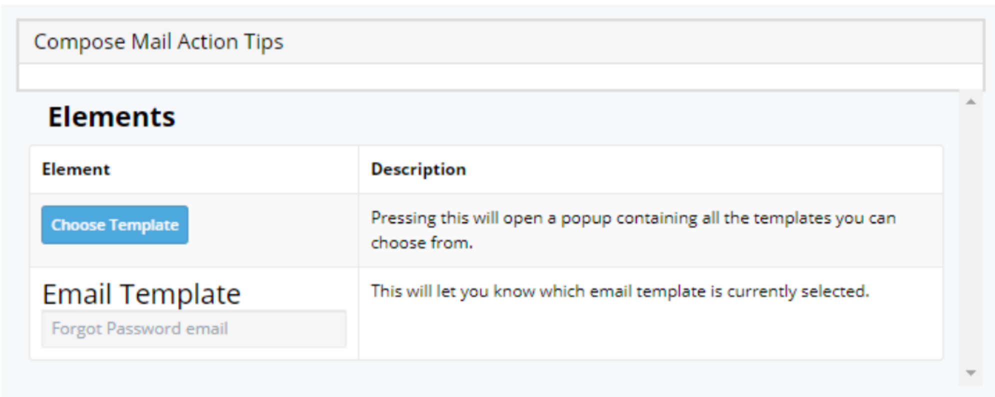


-
- **Tips/Tutorial System**. Some actions supported by Custom Buttons offer additional usage and configuration tips and tutorials. These are available by pressing on the icon.

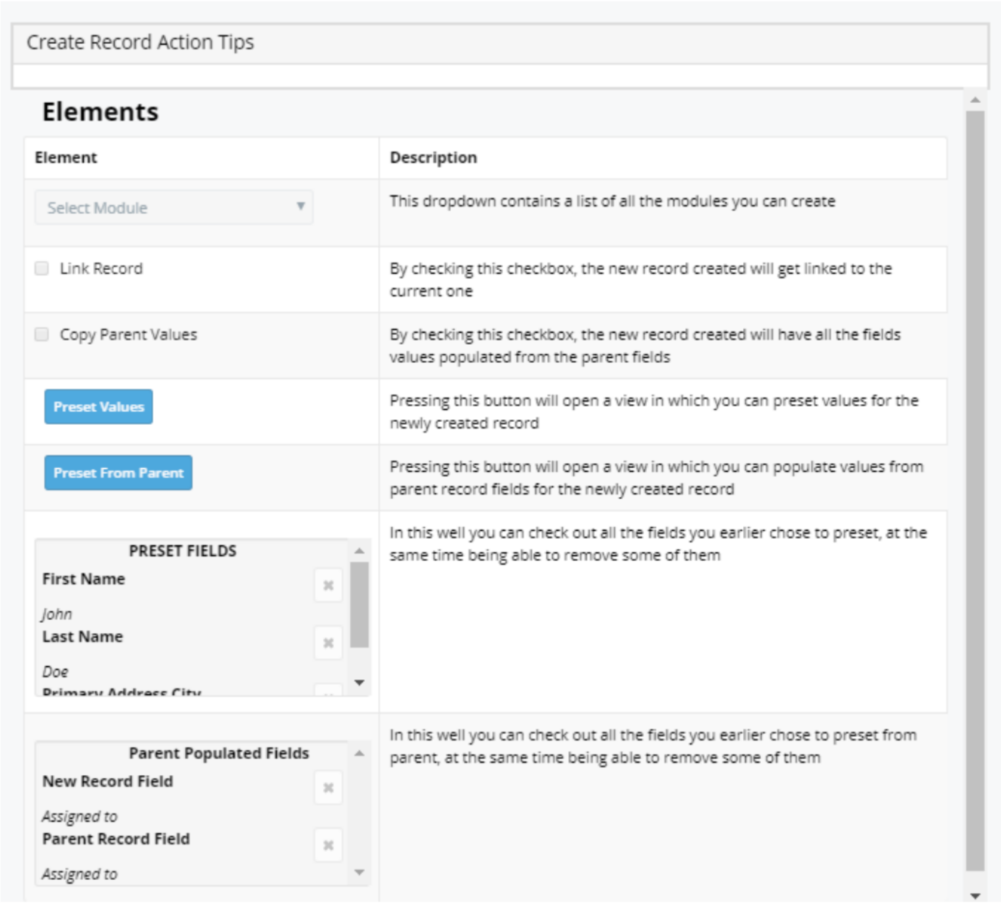


The **Tips and Tutorial System** are available for the following actions:

- Compose Email



- Create Record

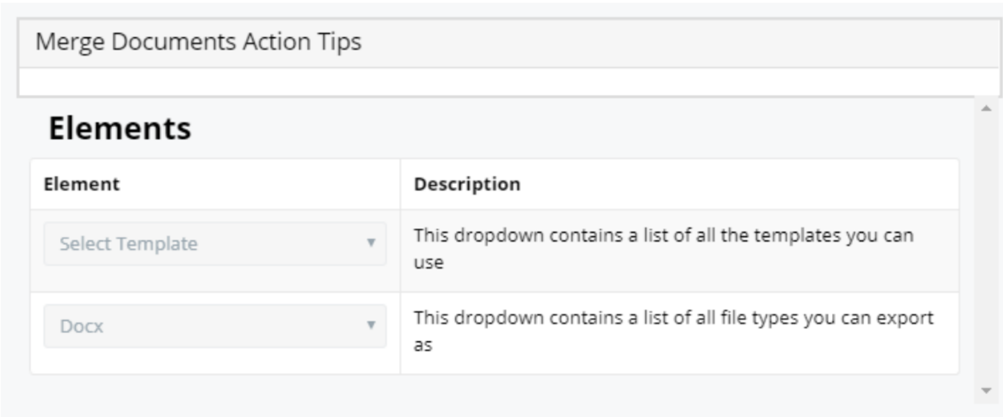


Create Record Action Tips

Elements

Element	Description
Select Module	This dropdown contains a list of all the modules you can create
Link Record	By checking this checkbox, the new record created will get linked to the current one
Copy Parent Values	By checking this checkbox, the new record created will have all the fields values populated from the parent fields
Preset Values	Pressing this button will open a view in which you can preset values for the newly created record
Preset From Parent	Pressing this button will open a view in which you can populate values from parent record fields for the newly created record
PRESET FIELDS First Name *John* Last Name *Doe*	In this well you can check out all the fields you earlier chose to preset, at the same time being able to remove some of them
Parent Populated Fields New Record Field *Assigned to* Parent Record Field *Assigned to*	In this well you can check out all the fields you earlier chose to preset from parent, at the same time being able to remove some of them

- Merge Document (only for wDocs users)

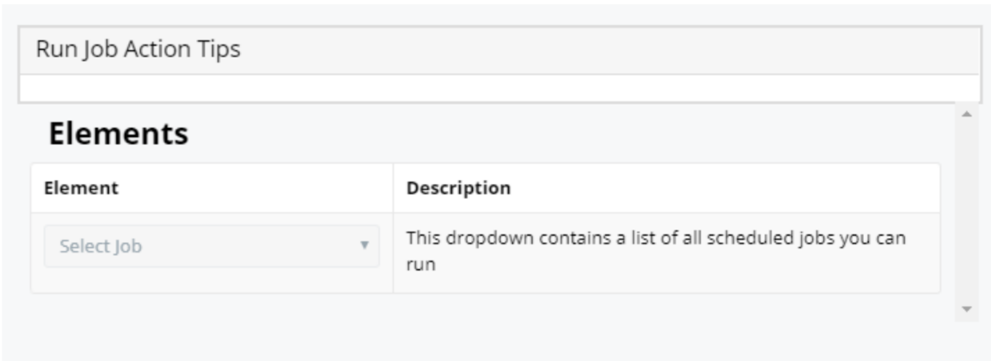


Merge Documents Action Tips

Elements

Element	Description
Select Template	This dropdown contains a list of all the templates you can use
Docx	This dropdown contains a list of all file types you can export as

- Run Job

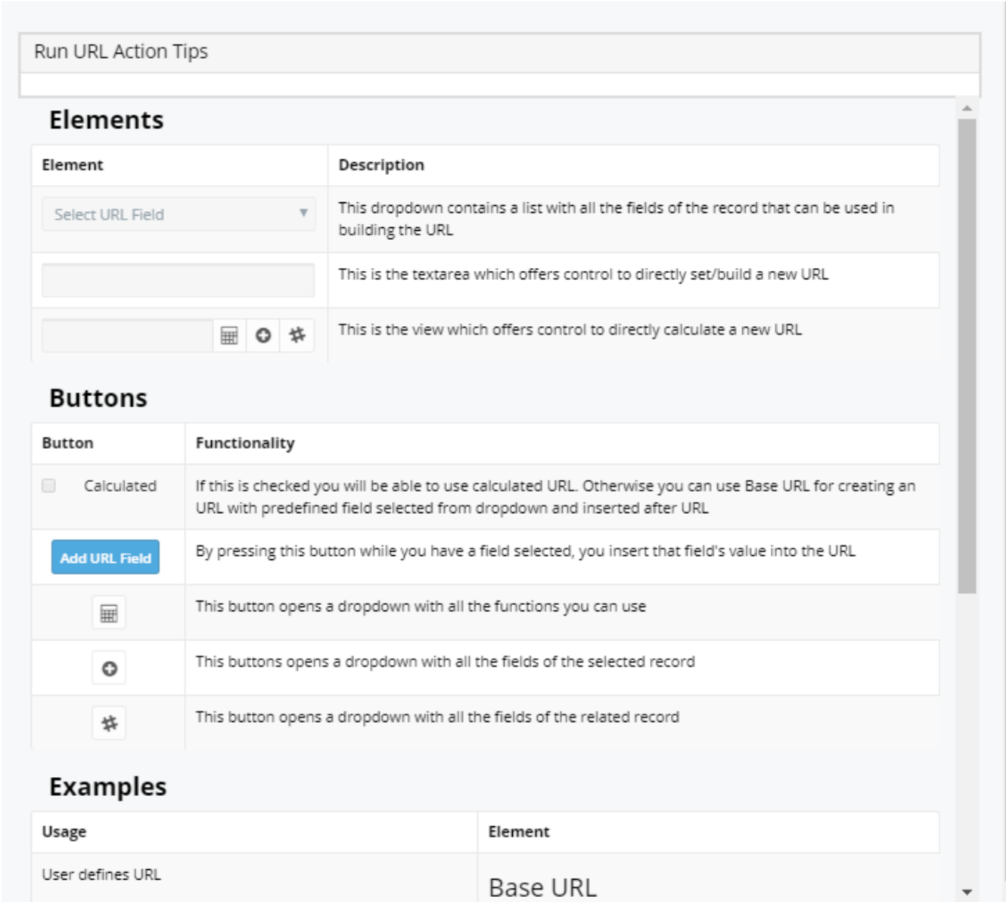


Run Job Action Tips

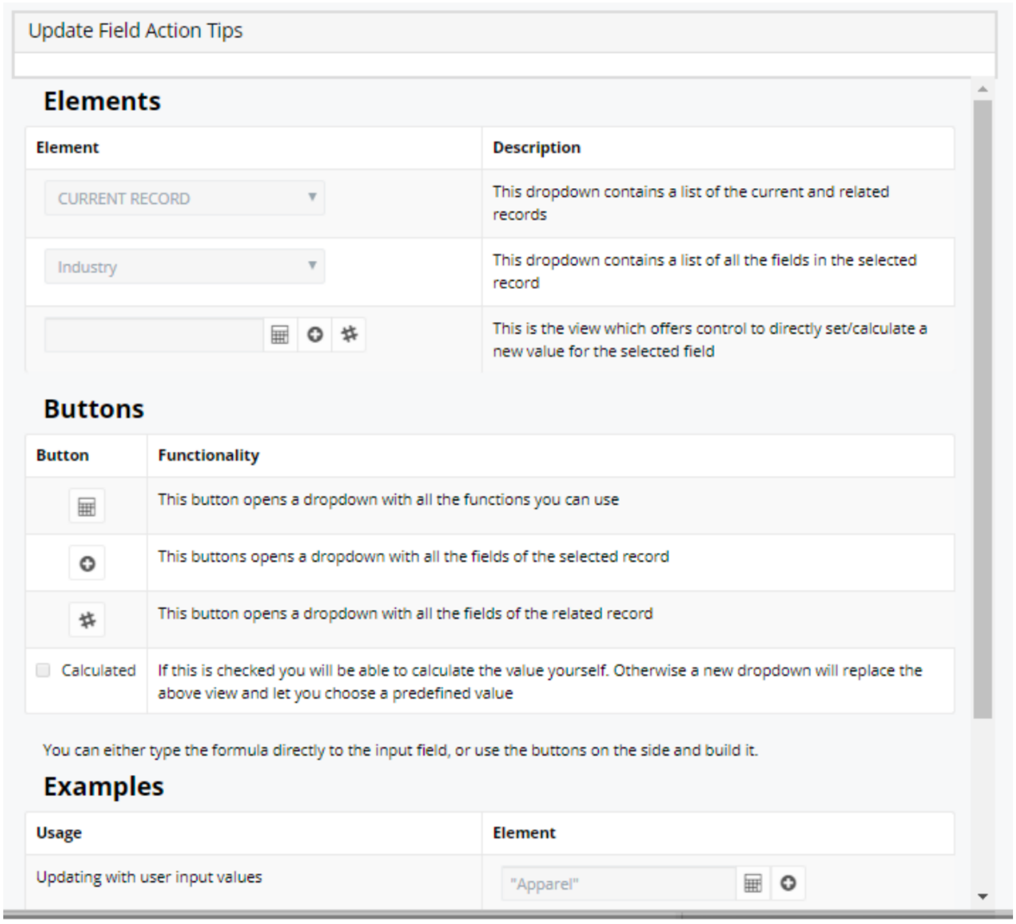
Elements

Element	Description
Select Job	This dropdown contains a list of all scheduled jobs you can run

- Run URL



- Update Field



Update Field Action Tips

Elements

Element	Description
CURRENT RECORD	This dropdown contains a list of the current and related records
Industry	This dropdown contains a list of all the fields in the selected record
	This is the view which offers control to directly set/calculate a new value for the selected field

Buttons

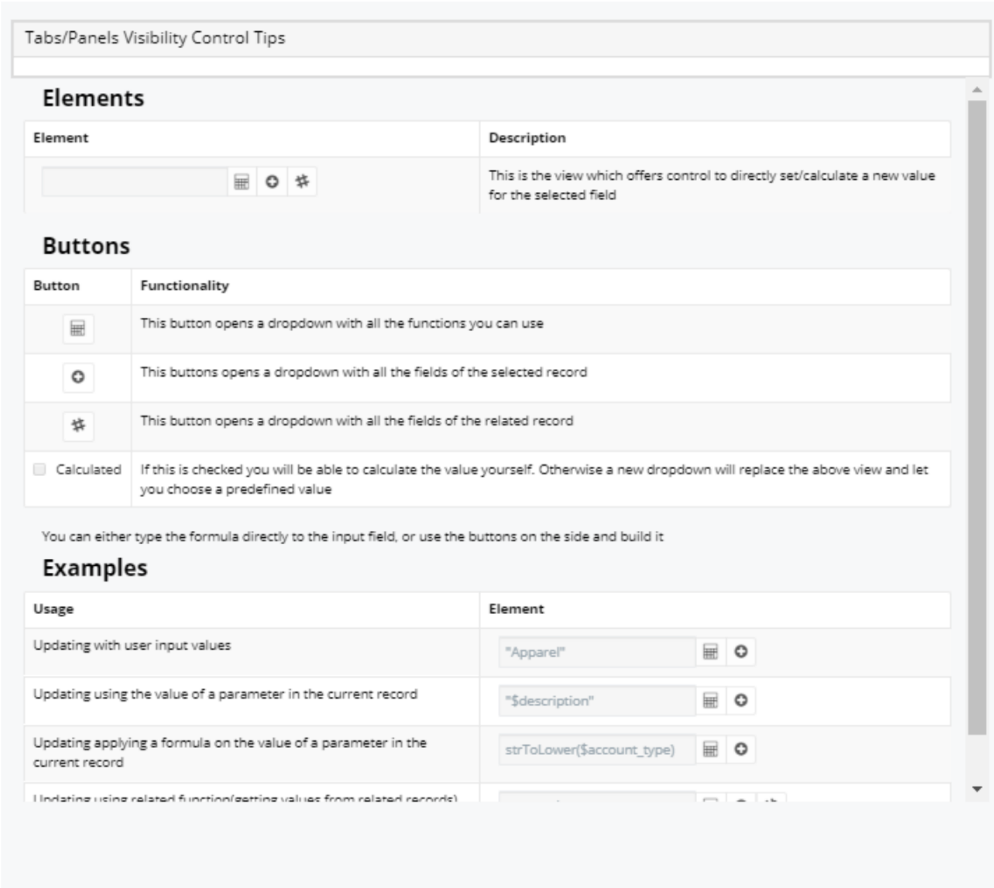
Button	Functionality
	This button opens a dropdown with all the functions you can use
	This buttons opens a dropdown with all the fields of the selected record
	This button opens a dropdown with all the fields of the related record
Calculated	If this is checked you will be able to calculate the value yourself. Otherwise a new dropdown will replace the above view and let you choose a predefined value

You can either type the formula directly to the input field, or use the buttons on the side and build it.

Examples

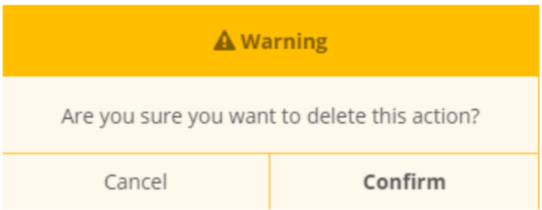
Usage	Element
Updating with user input values	"Apparel"

- Tabs/Panels Visibility Control



- **Unsaved Changes Validation**

- Users trying to leave the configuration window without saving their changes will be met with a warning pop up message.

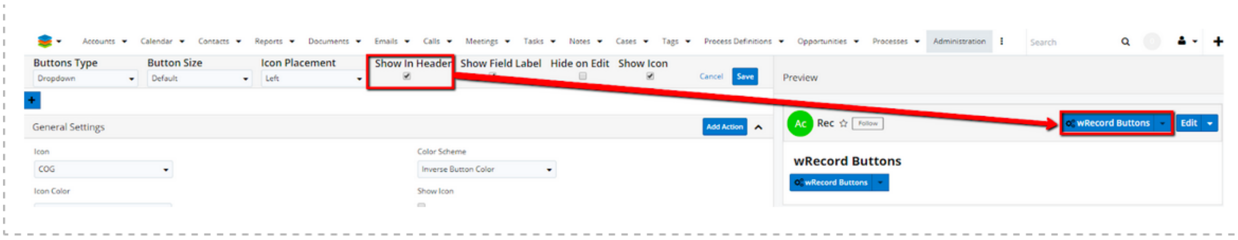


  - **Confirm** - if the Confirm button is pressed, all changes and configurations made to the Custom Buttons will be lost.
  - **Cancel** - pressing the Cancel button will keep the user on the page and allow them to save their configurations.

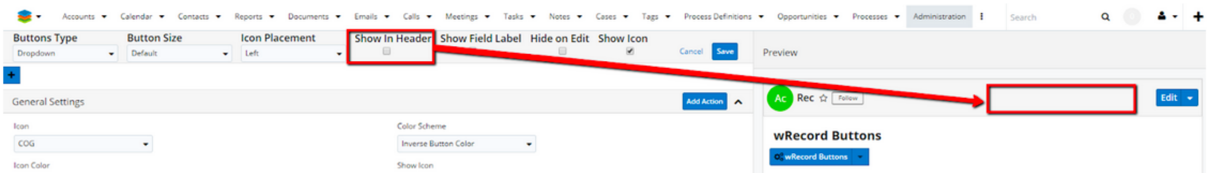
- **Show in Header**

- The Show on Header checkbox customization allows users to place Custom Buttons next to the header buttons of a record view. Ticking the checkbox will automatically place the buttons in the header.

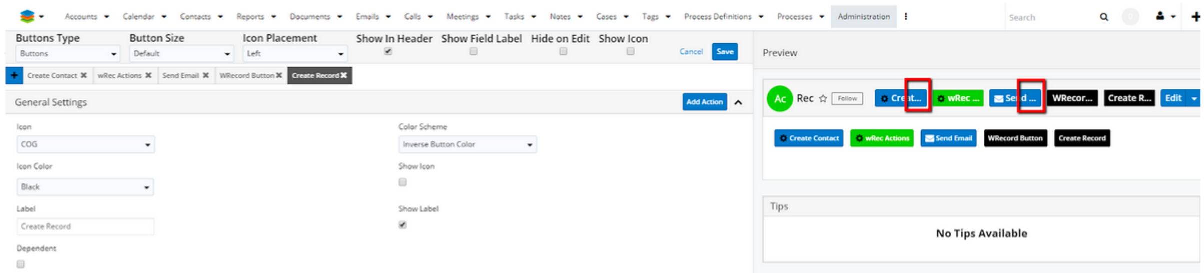
  Show in Header enabled:

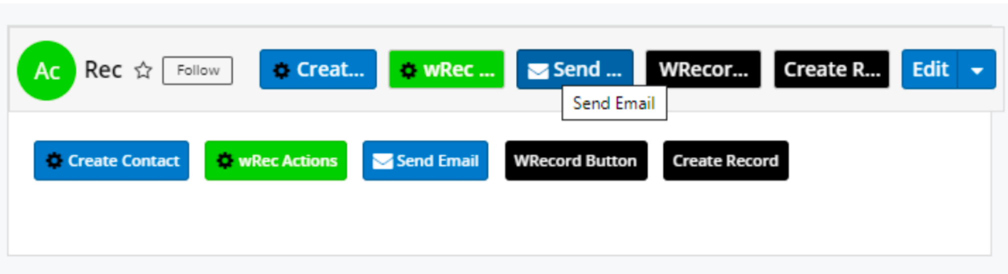


- Show in Header disabled:



  Note: Disabling this option will not remove the button from Record View.
  Note: Sometimes, on the Record View header, the custom button label might not be entirely displayed, due to space limitation. Hovering over the button allows users to see each button's entire label.

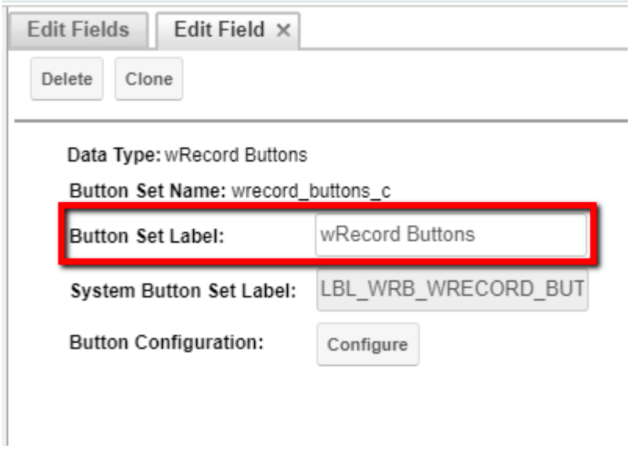


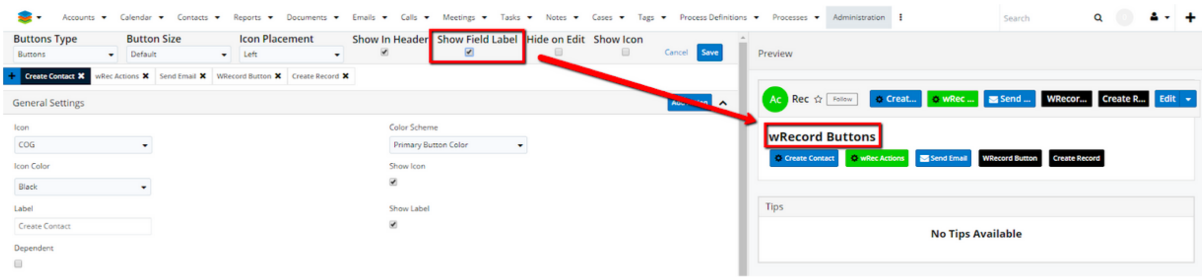


  - **Show Field Label**

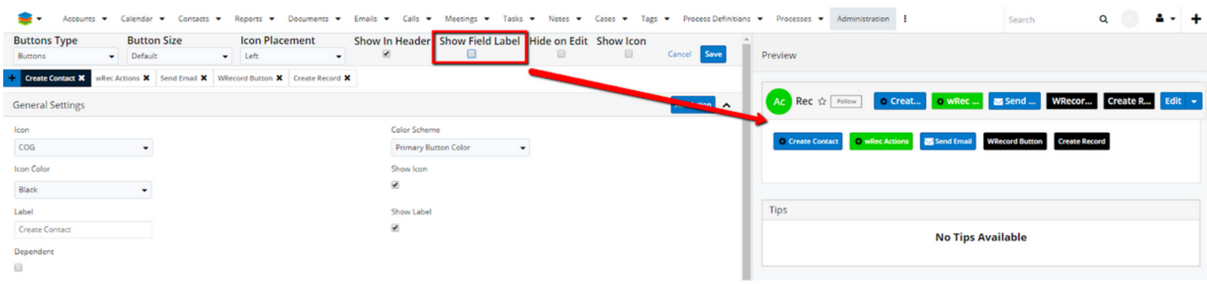
This customization allows users to choose if the label of the Custom Buttons they create is visible or not on views.

Show Field Label enabled:



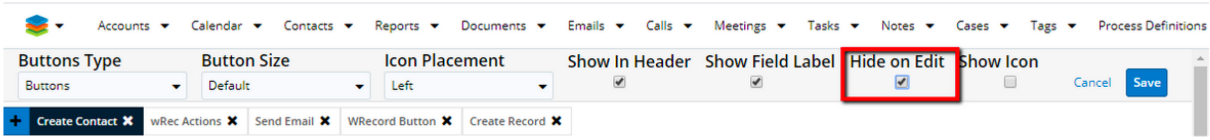


- Show Field Label disabled:

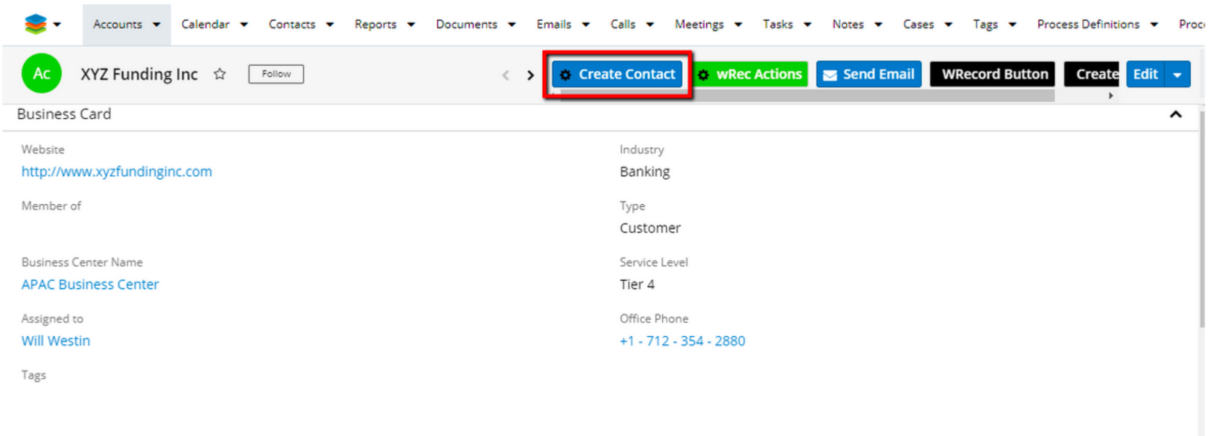


- **Hide on Edit**

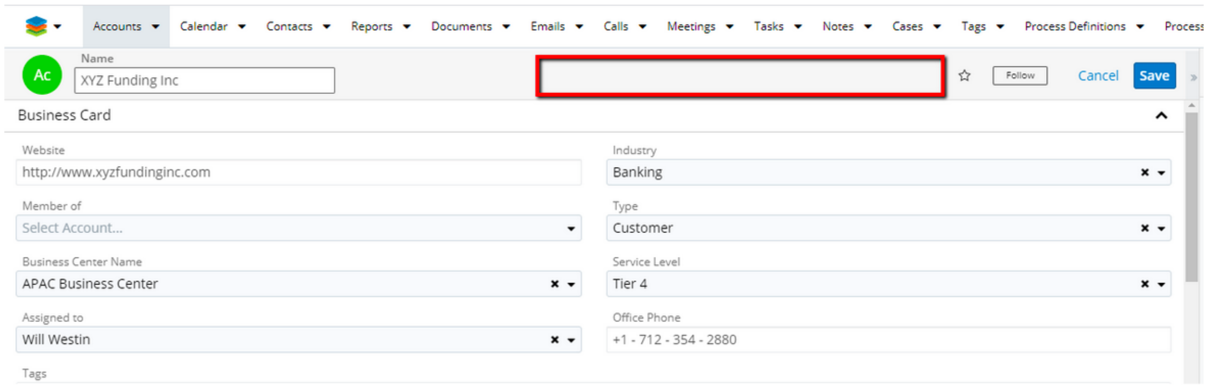
- This customization option allows users to hide buttons on edit mode. Ticking the checkbox will hide the button.



Hide on Edit disabled:



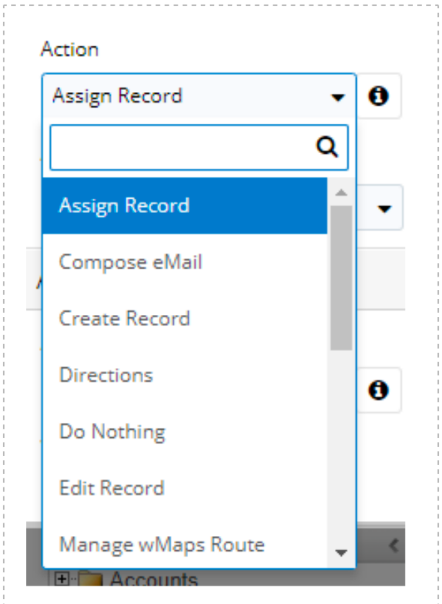
- Hide on Edit enabled:

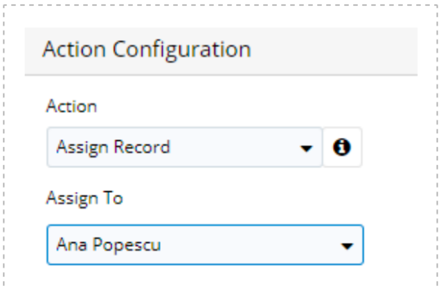


Actions

Assign Record

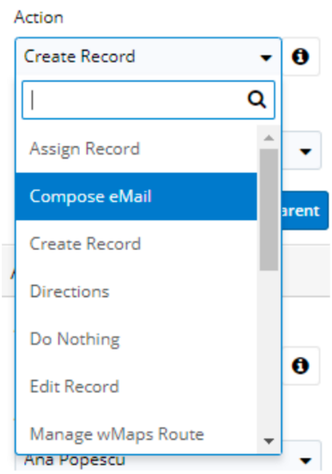
Clicking the custom button created with the **Assign Record** action enabled, will assign the actions attributed to the button to the selected user.



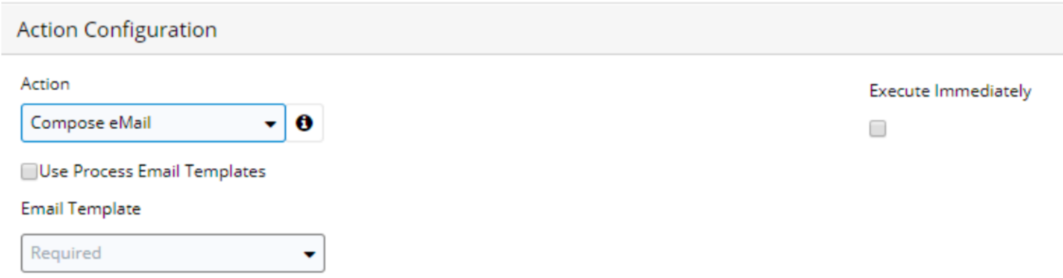


Custom Buttons support the following Actions:

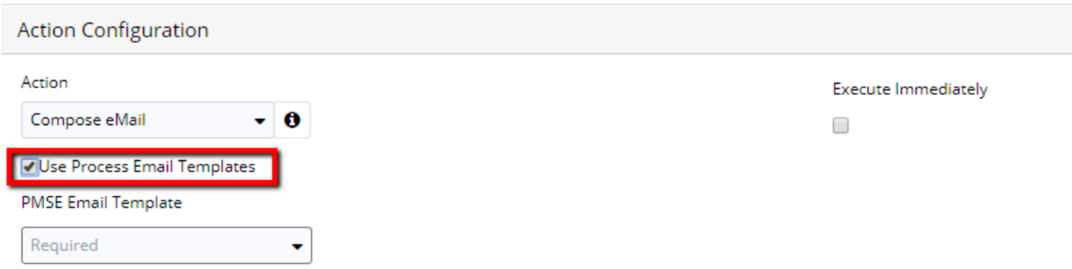
- **Compose Email**
  - The **Compose Email** action allows users to use email templates to quickly send emails.



- Clicking the **Email Template** function will open an email template drawer.

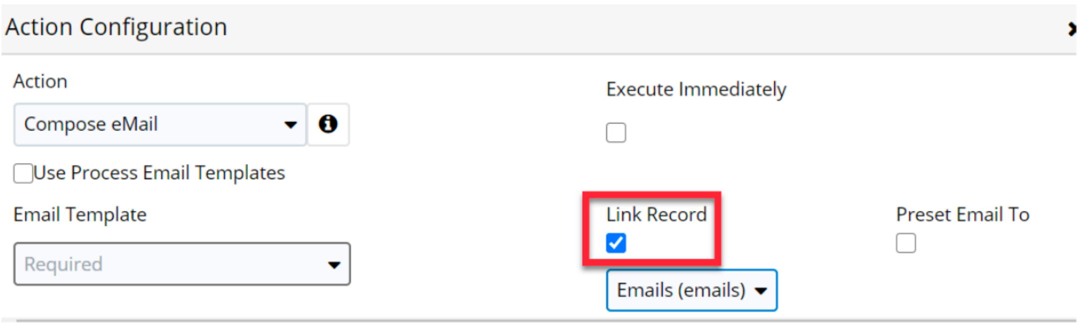


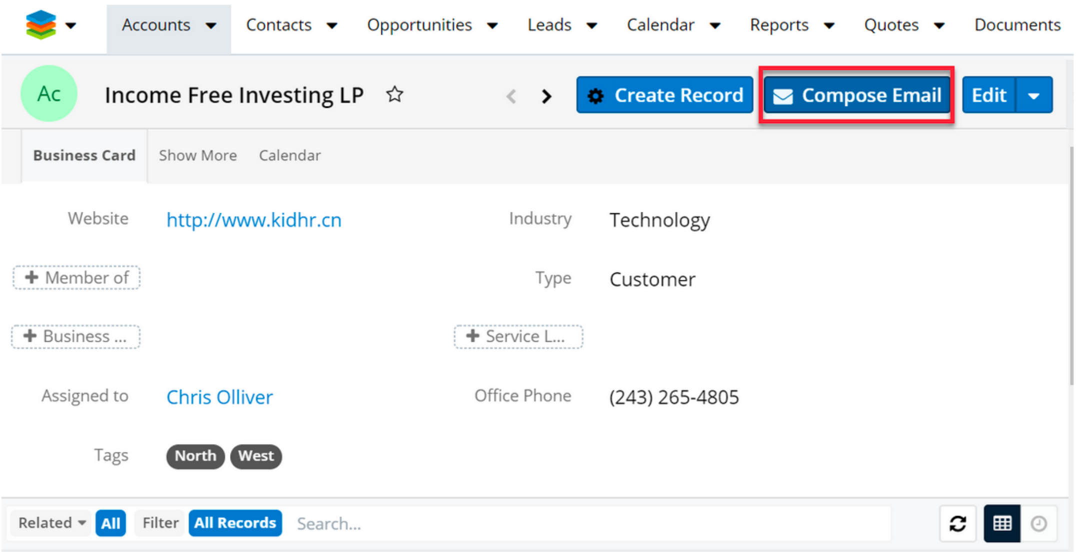
  - Ticking the **Use Process Email Templates** checkbox takes the user to a is similar to the Compose Email with Email Template, the only difference being that the Email Template Drawer contains only Process Email Templates.

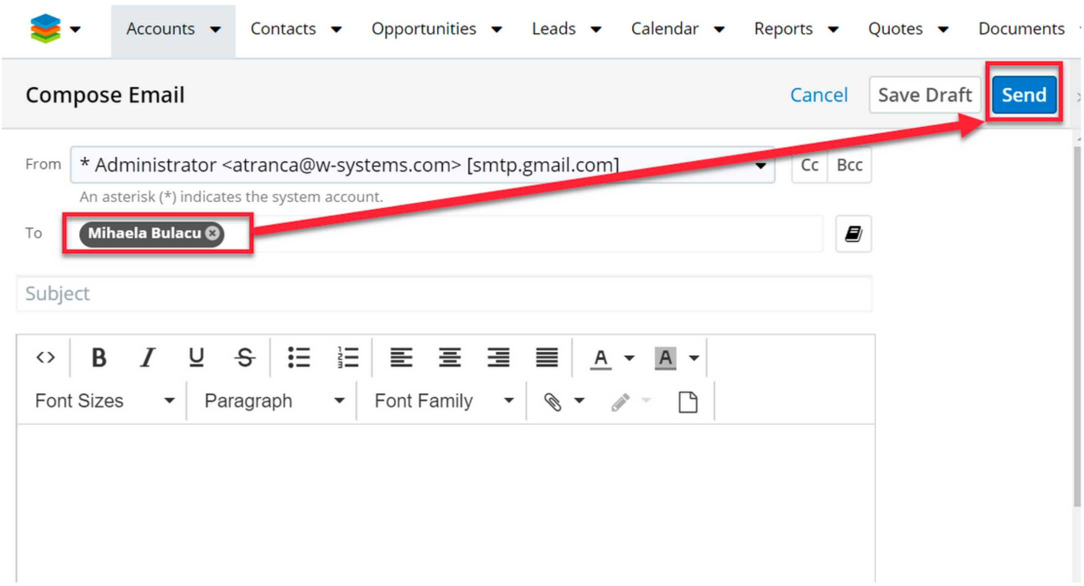


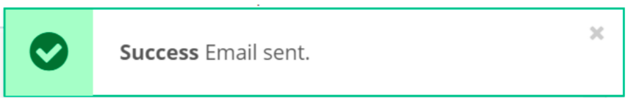
Note: Starting with the May 2020 package update, selecting an email template for this record button action type is no longer mandatory.

The Link Record feature on Compose Email. With this feature, users can automatically link an Email sent via the Send Email Record Button to the Emails Subpanel, by ticking the **Link Record** checkbox in **Studio Fields**, when configuring a custom button.

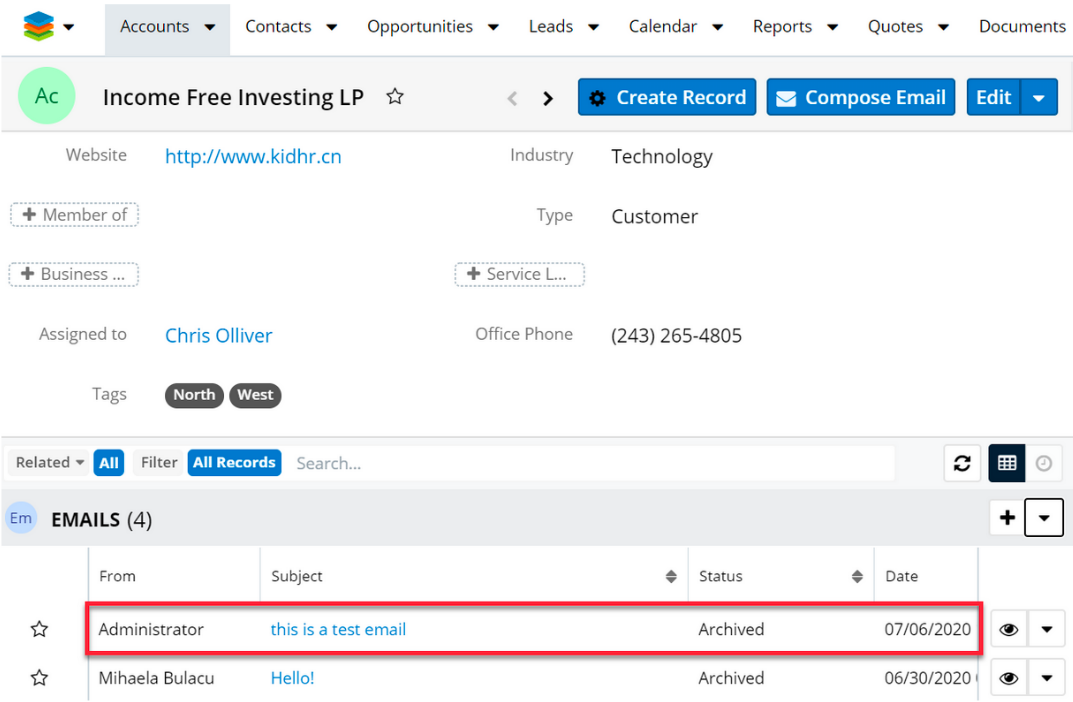






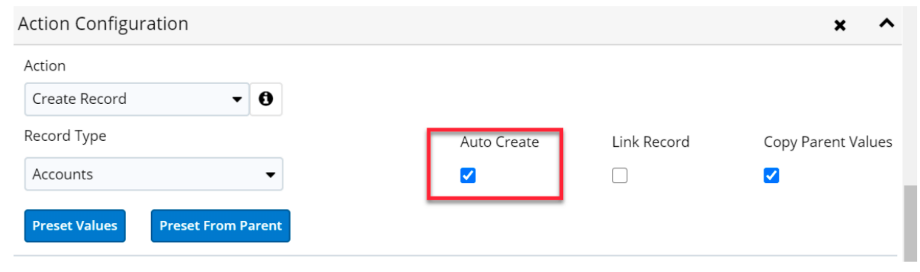


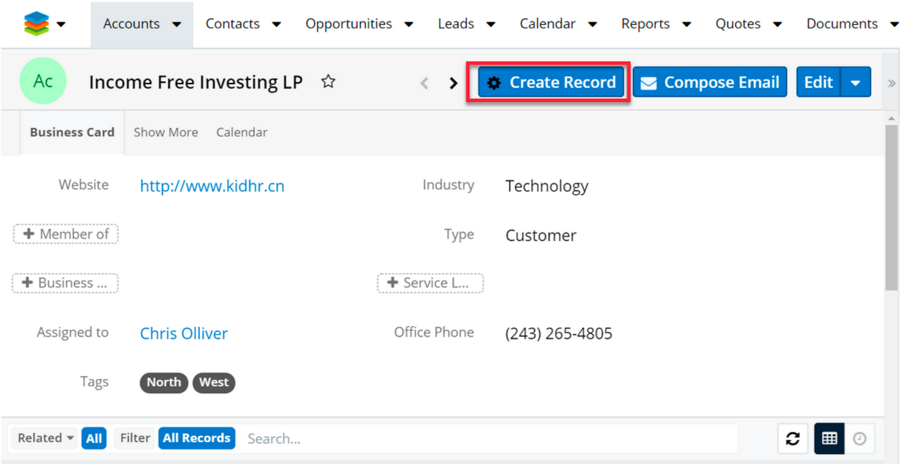
Once the email is successfully sent, it will appear attached in the **Emails** subpanel in **Record View**.

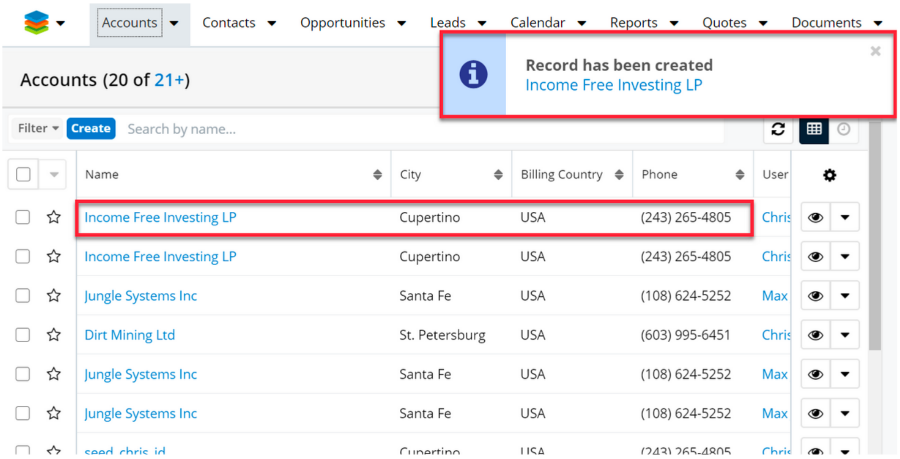


- **Auto-Create**

- **With the Auto-Create feature** the user can customize a custom button to automatically create a record on the desired module without the drawer opening by ticking the **Auto Create** checkbox in **Studio Fields,** when configuring a record button.

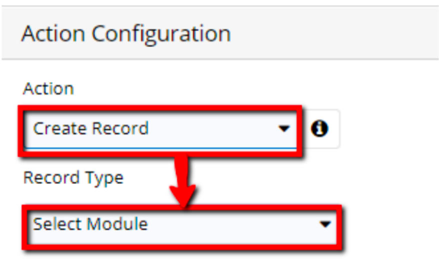




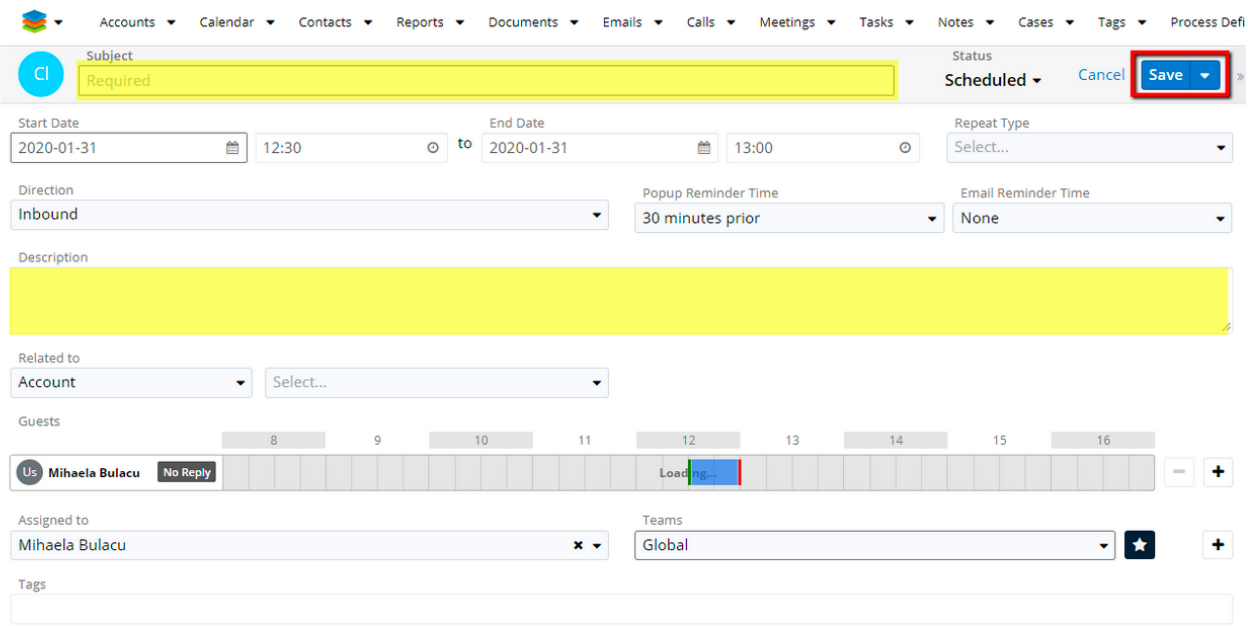


- **Create Record**

The **Create Record** action allows users to quickly create records in different modules.



- **Preset Fields**. The **Create Record** action allows users to preset values for certain fields. Once the user selects the record and record type and presses the "**Preset Fields**". Pressing the button will open a new drawer where the user can configure the button's functionality.



This functionality supports the following field types:

- TextField
- Name
- TextArea
- Currency
- Checkbox
- Date
- Datetime
- Encrypt
- Dropdown
- Float
- Integer
- Phone
- Radio
- Email
- Iframe
- URL
- Decimal
- Flex Relate
- Relate
- Color Status - custom field type

Note: The Preset Field functionality is not available for the Quotes module.

- **Do Nothing**

This Action is generally used for demos, as it behaves like a button without functionality.

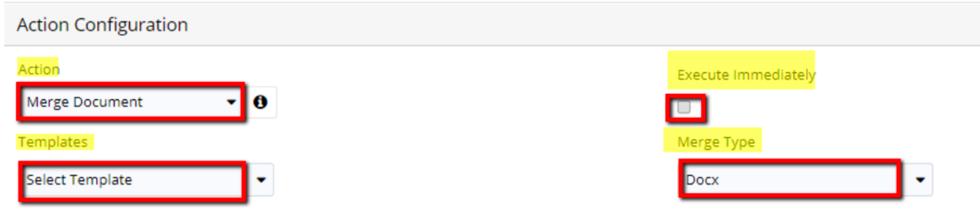
- **Edit Record**

The **Edit Record** button action allows users to create a custom button that opens **Edit View** on a record.

Note: This action button is supported on Record View, List View - inline, and Preview. When placed on Dashlet view, the button's functionality is not applicable.

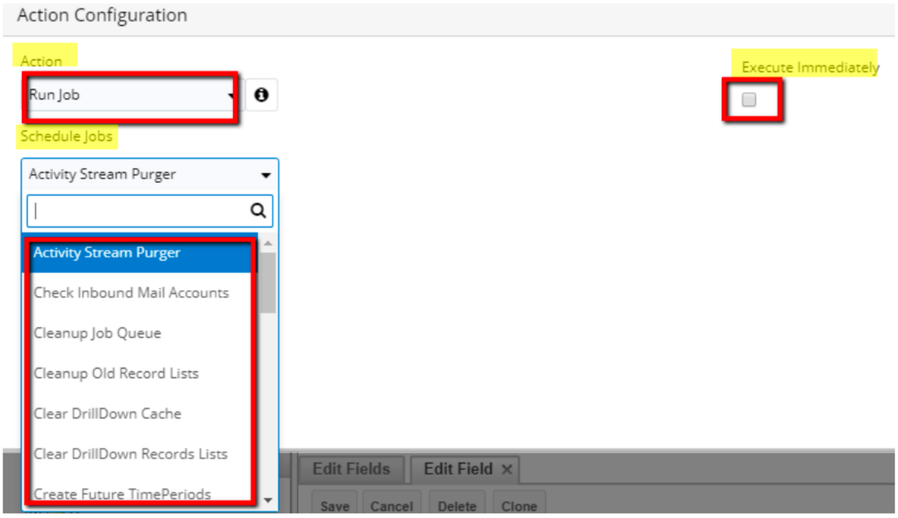
- **Merge Document**

The **Merge Document** action allows wDocs users to merge documents by using templates.



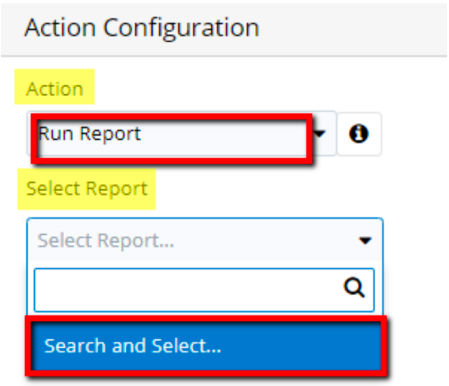
- **Run Job**

The **Run Job** action allows users to run jobs at the time they want, without waiting for the scheduled job run.

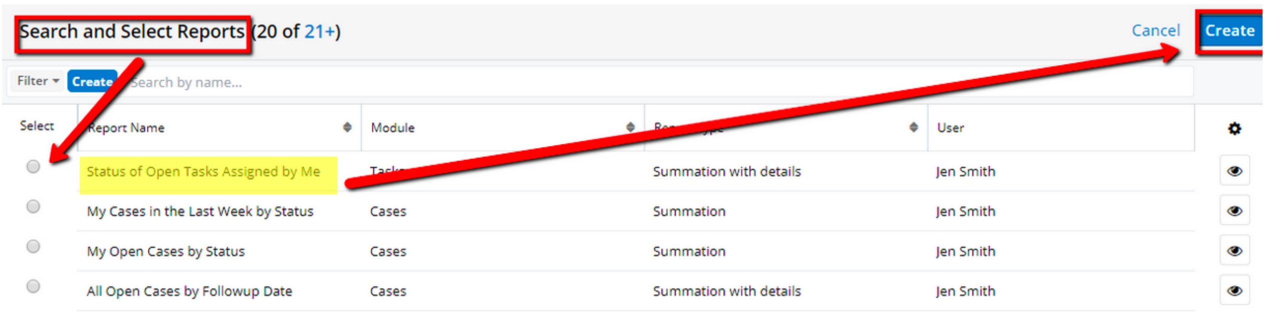


- **Run Report**

This activity allows users to automatically run a report type when pressing the custom button.



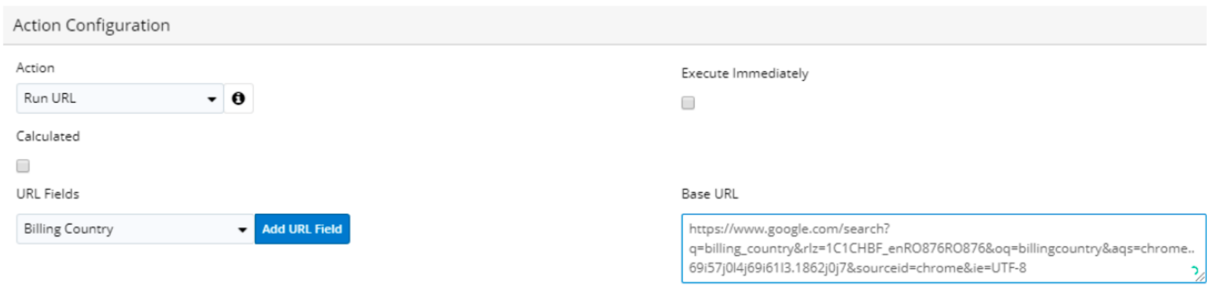
Pressing on the **Search and Select** field will open a report drawer from where the user can select what type of report they want to use for this button.



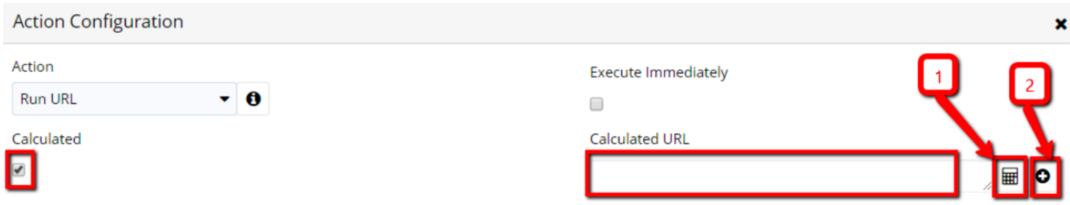
- **Run URL**

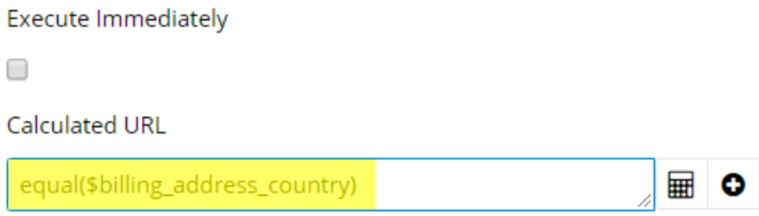
Users can add custom buttons that automatically open URLs in a new window in their browser. The **Run URL** activity type with Custom Buttons. This action type can be used in several ways.

- **Base URL**. By using this type of Run URL action users can simply type a valid URL address. Once pressed, the button will automatically open the URL in a new browser window.
- **Build URL.** Users can also build URLs using the available fields. For example, if you want to search on Google a certain value from a field, you can add the Google Search link, together with the chosen field from the dropdown list.



- **Calculated URL**. When the Calculated checkbox is ticked, the URL can be built by using formulas. You can use all formulas that you normally use in the studio, but the most relevant formulas in order build formulas are the ones that usually return a string (like : ValueAt ; Related ; GetDropdownValue ; Concat).





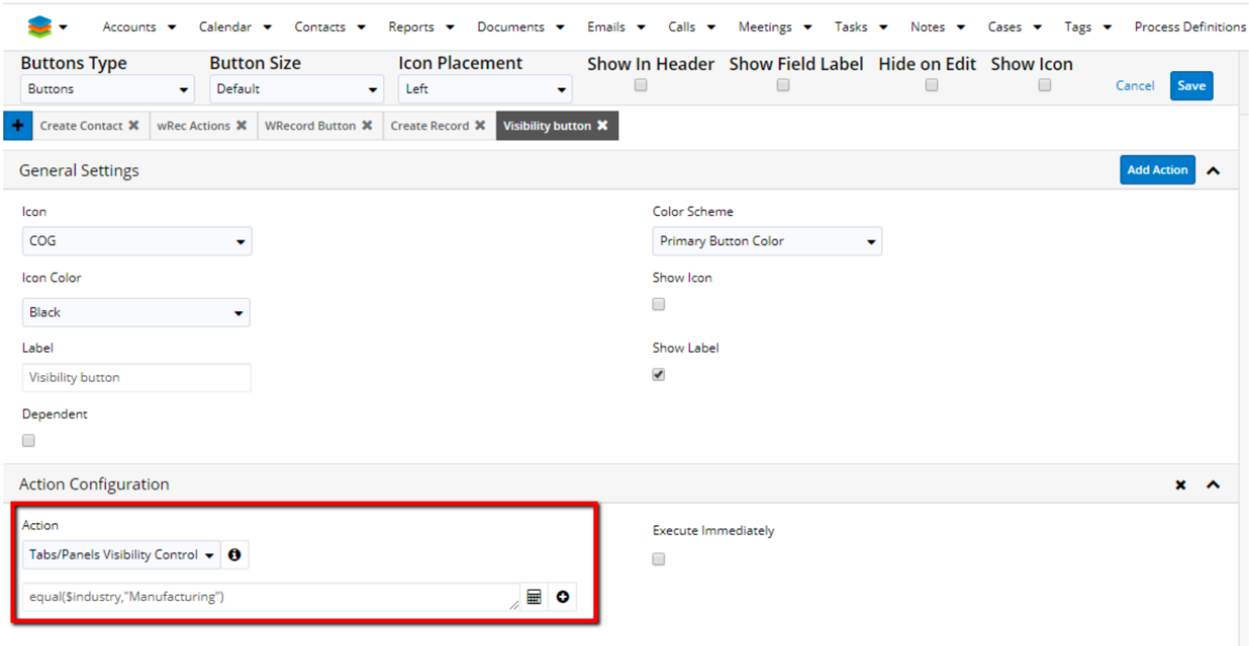
- **Save Record**

The **Save Record** action will simply save the button with all its configurations.

- **Tab and Panel Visibility Control**

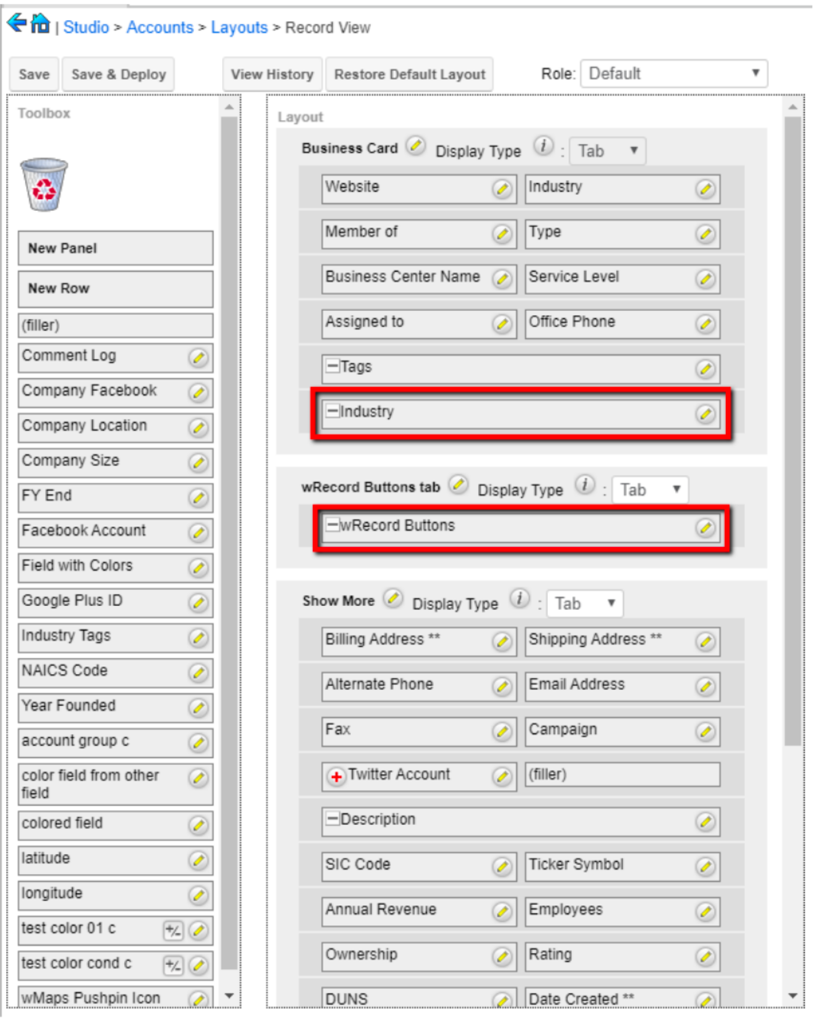
Users can customize if they want a certain Record Panel or Tab to be displayed in different fields. To configure this, users must follow the steps below.

1. Create a new Custom Button field on the record they want to configure.
2. Set the action of the Custom Button to "Tabs/Panels Visibility Control".
3. Insert the dependency formula they want to apply for a Panel or Tab.
   When created, these Custom Buttons will not be visible in Record View because it configures everything behind the scenes.

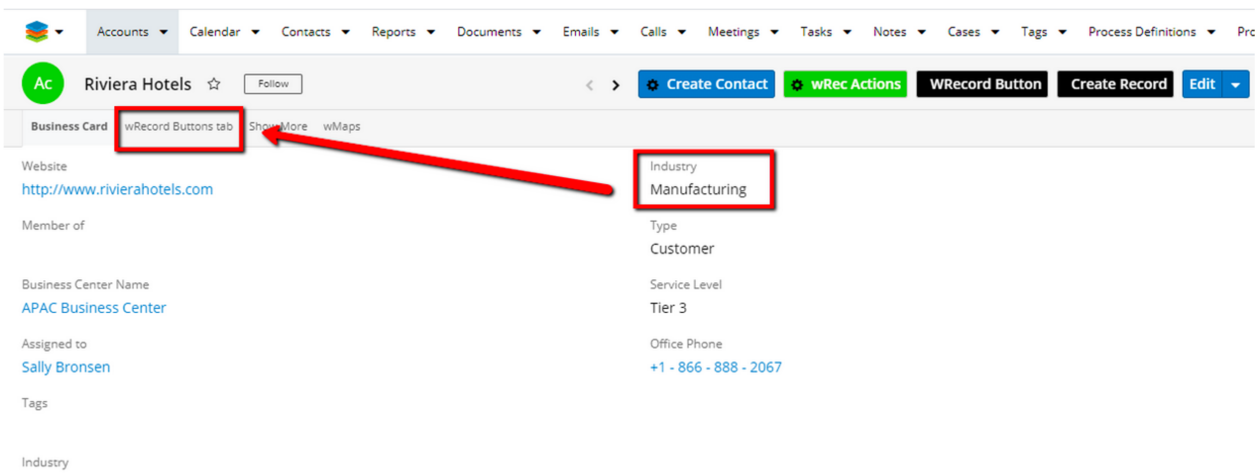


4. After saving the field, go to Record Layout in Studio and drag and drop the new Custom Button into the panel/Tab that you want to apply the action to, with the dependency formula that you previously inserted.

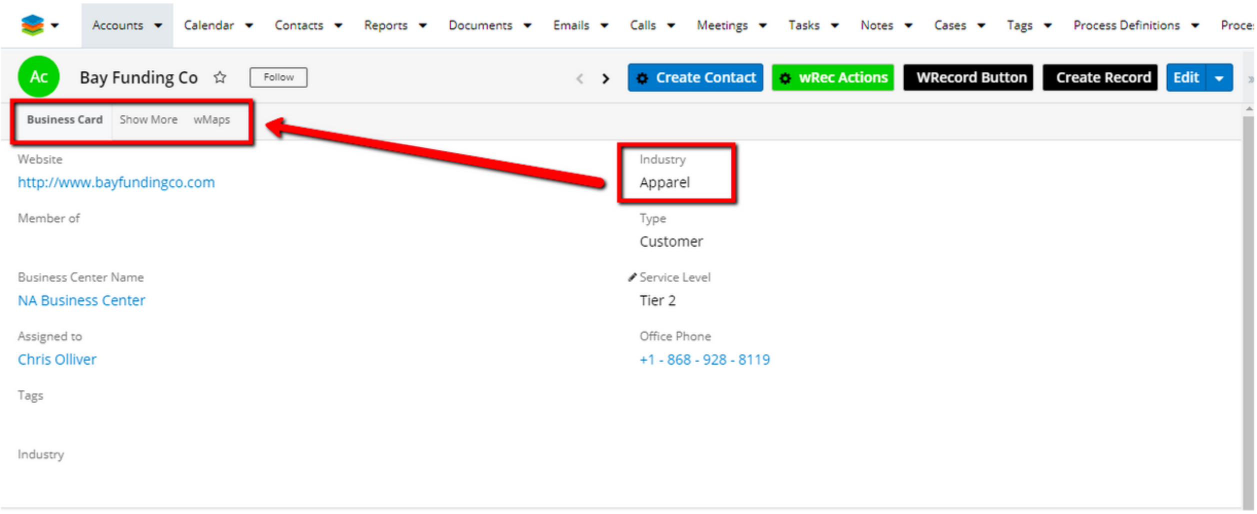
Note: Users must ensure that the fields they used inside the calculated formula will not be placed on a Panel/Tab layout with controlled visibility. Otherwise, they will not be able to change the values and the tab or panel will remain hidden.



This is the result of our example. The "Custom Button" tab will always be visible when the Account Type is "Analyst".



Otherwise, it will remain hidden.

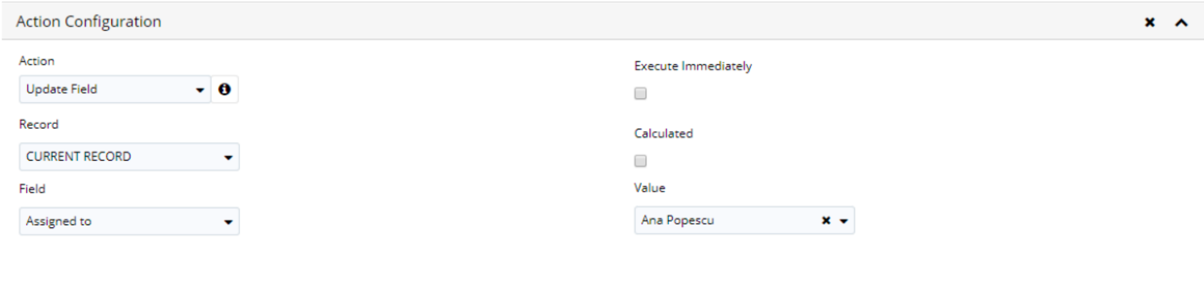


The advantage offered by this functionality is its flexibility. Users can leverage all formulas they would normally use in Studio. The functionality is also highly responsive to model changes.

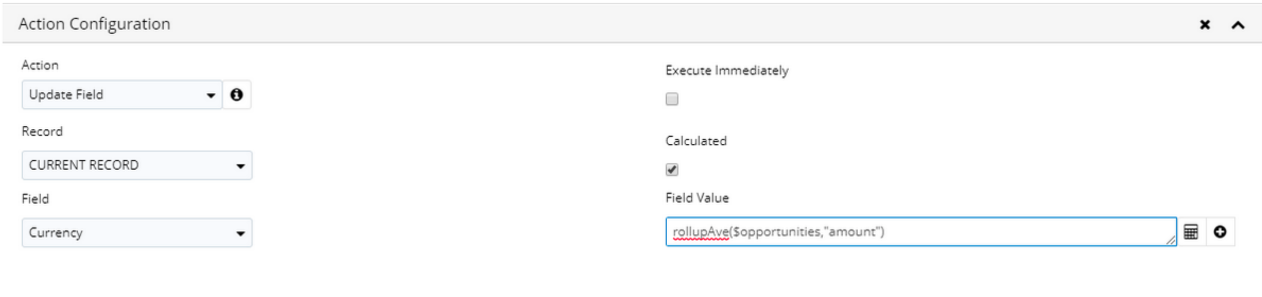
- **Update Field**

This action allows users to update fields based on a formula or based on a preset value.

1. Update Fields on **Current Record** without Calculated Formulas. This action supports updates on Current Record with the following Field Types.
    - Checkbox
    - Currency
    - Date
    - Datetime
    - Decimal
    - Dropdown
    - Encrypt
    - Float
    - Integer
    - Phone
    - Relate
    - TextArea
    - TextField
    - URL

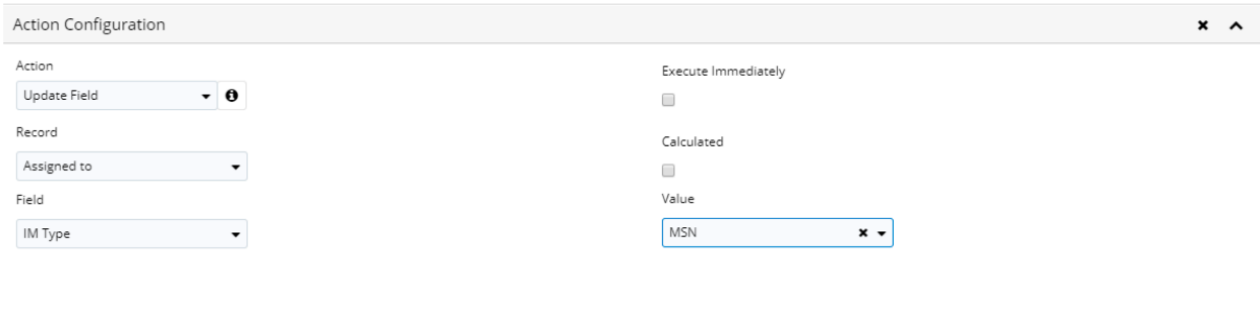


2. Update Fields on **Current Record** with Calculated Formulas.

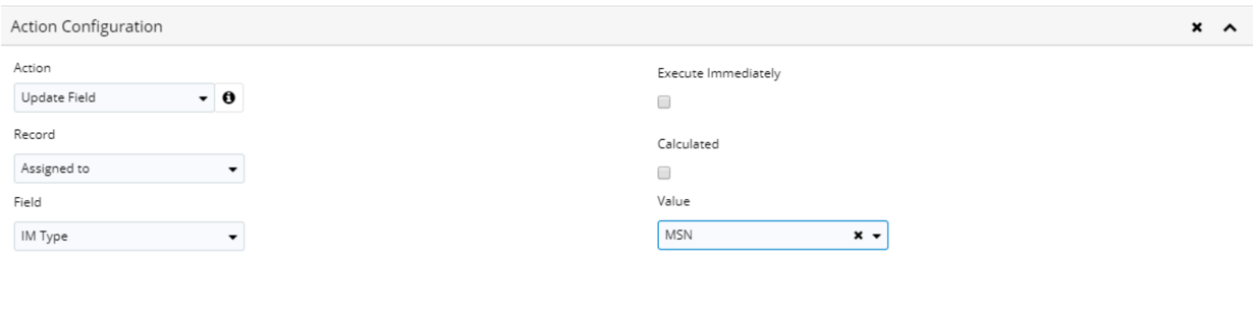


   Note: If users add empty spaces in formulas, these will be automatically removed when the user presses the cursor outside the Field Value area.

3. Update Fields on **Relate Record** without Calculated Value.



4. Update Fields on **Related Record** with Calculated Formulas.



- **Do not update fields unless all required fields are filled in**. When using Custom Buttons, users should ensure they fill in all fields. This feature is useful when records are imported with empty required fields. The feature is supported on **Record View**, **List View**, and **Preview**.
- **Supported Functions for Calculated Fields**. All functions available in Formula

Builder also work with Custom Buttons. Below is a table with all Sugar functions.

Function name	**Returned Type**	**Field Type that should be used according to the expression result.**	**Comments**
abs	Number	Integer Float Decimal	
add	Number	Integer Float Decimal	
addDays	Date		
and	Boolean	Checkbox	
average	Number	Integer Float Decimal	
ceil	Number	Integer Float Decimal	
concat	Text		
contains	Boolean		
count	Number	Integer Float Decimal	
createList	List		
date	Date		
dayofweek	Number	Integer Float	

Function name	Returned Type	Field Type that should be used according to the expression result.	Comments
		Decimal	
daysUntil	Number	Integer Float Decimal	
divide	Number	Integer Float Decimal	
equal	Boolean	Checkbox	
floor	Number		
getDropdownKeySet	List		
getDropdownValue	Text		
getDropdownValueSet	List		
greaterThan	Boolean	Checkbox	
hoursUntil	Number		
ifElse	Variable		
indexOf	Number	Integer Float Decimal	
isAfter	Boolean	Checkbox	
isBefore	Boolean	Checkbox	
isInList	Boolean	Checkbox	
isValidDate	Boolean	Checkbox	
isValidEmail	Boolean	Checkbox	
isWithinRange	Boolean	Checkbox	
ln	Number		
log	Number	Integer	

Function name	Returned Type	Field Type that should be used according to the expression result.	Comments
		Float Decimal	
max	Number	Integer Float Decimal	
maxRelatedDate	Date		
median	Number	Integer Float Decimal	
min	Number	Integer Float Decimal	
monthofyear	Number	Integer Float Decimal	
multiply	Number	Integer Float Decimal	
negate	Number	Integer Float Decimal	
not	Boolean	Checkbox	
now	Date		
number	Number	Integer	

Function name	Returned Type	Field Type that should be used according to the expression result.	Comments
		Float Decimal	
or	Boolean		
pow	Number	Integer Float Decimal	
related	Variable		
rollupAve	Number	Integer Float Decimal	
rollupMax	Number	Integer Float Decimal	
rollupMin	Number	Integer Float Decimal	
rollupSum	Number	Integer Float Decimal	
strlen	Number	Integer Float Decimal	
strToLower	Text		
strToUpper	Text		

Function name	Returned Type	Field Type that should be used according to the expression result.	Comments
subStr	Text		
subtract	Number	Integer Float Decimal	
timestamp	Date		
today	Date		
toString	Text		
translateLabel	Text		
valueAt			
Sugar Functions that should not be used with uRecord button:			
forecastincludedCommitstages	Not in Sugar guide but available in Formula Builder and also for uRecord button.		
forecastSalesStages	Not in Sugar guide but available in Formula Builder and also for uRecord button.		
getlistwhere	SUGAR ISSUE: Studio's formula builder displays several undesired functions (getListWhere, isForecastClosed, isForecastClosedWon, and isForecastClosedLost).		
isForecastClosed	SUGAR ISSUE: Studio's formula builder displays several undesired functions (getListWhere, isForecastClosed, isForecastClosedWon, and isForecastClosedLost).		
isForecastClosedLost			
IsForecastClosedWon			
round	Not in Sugar guide but available in Formula Builder and also for uRecord button.		

Function name	**Returned Type**	**Field Type that should be used according to the expression result.**	**Comments**
abs	Number	Integer Float Decimal	
add	Number	Integer Float Decimal	
addDays	Date		
and	Boolean	Checkbox	
average	Number	Integer Float Decimal	
ceil	Number	Integer Float Decimal	
concat	Text		
contains	Boolean		
count	Number	Integer Float Decimal	
createList	List		
date	Date		
dayofweek	Number	Integer Float Decimal	

Function name	Returned Type	Field Type that should be used according to the expression result.	Comments
daysUntil	Number	Integer Float Decimal	
divide	Number	Integer Float Decimal	
equal	Boolean	Checkbox	
floor	Number		
getDropdownKeySet	List		
getDropdownValue	Text		
getDropdownValueSet	List		
greaterThan	Boolean	Checkbox	
hoursUntil	Number		
ifElse	Variable		
indexOf	Number	Integer Float Decimal	
isAfter	Boolean	Checkbox	
isBefore	Boolean	Checkbox	
isInList	Boolean	Checkbox	
isValidDate	Boolean	Checkbox	
isValidEmail	Boolean	Checkbox	
isWithinRange	Boolean	Checkbox	
ln	Number		
log	Number	Integer Float	

Function name	Returned Type	Field Type that should be used according to the expression result.	Comments
		Decimal	
max	Number	Integer Float Decimal	
maxRelatedDate	Date		
median	Number	Integer Float Decimal	
min	Number	Integer Float Decimal	
monthofyear	Number	Integer Float Decimal	
multiply	Number	Integer Float Decimal	
negate	Number	Integer Float Decimal	
not	Boolean	Checkbox	
now	Date		
number	Number	Integer Float	

Function name	Returned Type	Field Type that should be used according to the expression result.	Comments
		Decimal	
or	Boolean		
pow	Number	Integer Float Decimal	
related	Variable		
rollupAve	Number	Integer Float Decimal	
rollupMax	Number	Integer Float Decimal	
rollupMin	Number	Integer Float Decimal	
rollupSum	Number	Integer Float Decimal	
strlen	Number	Integer Float Decimal	
strToLower	Text		
strToUpper	Text		
subStr	Text		
subtract	Number	Integer	

Function name	Returned Type	Field Type that should be used according to the expression result.	Comments
		Float Decimal	
timestamp	Date		
today	Date		
toString	Text		
translateLabel	Text		
valueAt			
Sugar Functions that should not be used with uRecord button:			
forecastincludedCommitstages	Not in Sugar guide but available in Formula Builder and also for uRecord button.		
forecastSalesStages	Not in Sugar guide but available in Formula Builder and also for uRecord button.		
getlistwhere	SUGAR ISSUE: Studio's formula builder displays several undesired functions (getListWhere, isForecastClosed, isForecastClosedWon, and isForecastClosedLost).		
isForecastClosed	SUGAR ISSUE: Studio's formula builder displays several undesired functions (getListWhere, isForecastClosed, isForecastClosedWon, and isForecastClosedLost).		
isForecastClosedLost			
IsForecastClosedWon			
round	Not in Sugar guide but available in Formula Builder and also for uRecord button.		

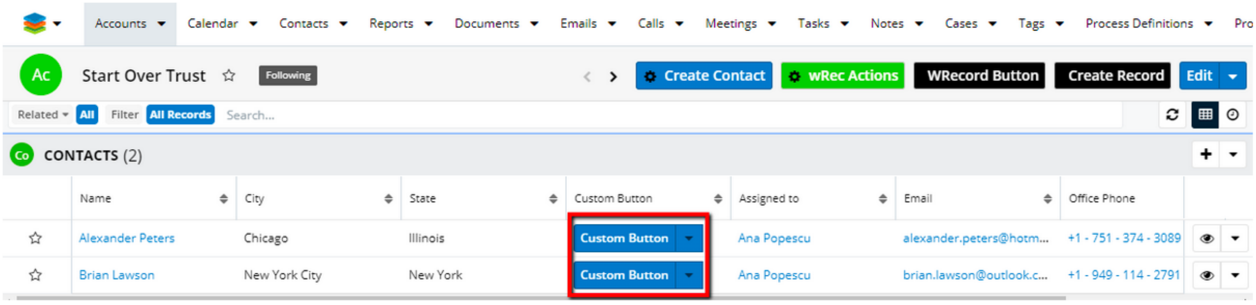
- **Custom Functions available with Custom Buttons.**

Function name	Returned Type	Field Type that should be used according to the expression result	Package name
Autoincrement	Number	Integer	wCalculatedFieldsFunctions
weekNumberReverse	Number	Integer	wCalculatedFieldsFunctions
getMultiSelectValues	Text	TextArea TextField	MultiSelectOnFormulaBuilder
weekEndDate	Date	Date Datetime	wCalculatedFieldsFunctions
dateSub	Date		wCalculatedFieldsFunctions
weekStartDate	Date		wCalculatedFieldsFunctions
Year	Number	Integer	wCalculatedFieldsFunctions
dateAdd	Date	Date	wCalculatedFieldsFunctions
weekNumber	Number	Integer	wCalculatedFieldsFunctions

- **Update Field with Current User, Using the getCurrentUSer() Function.** This function allows users to add an Update button to a text field or text area field with the name of the currently logged in user. This functionality is available with a new custom formula: **getCurrentUser()**. This update only applies to the username, not the linked user profile. Because of this reason, an update to the **Assigned To** field is not possible.

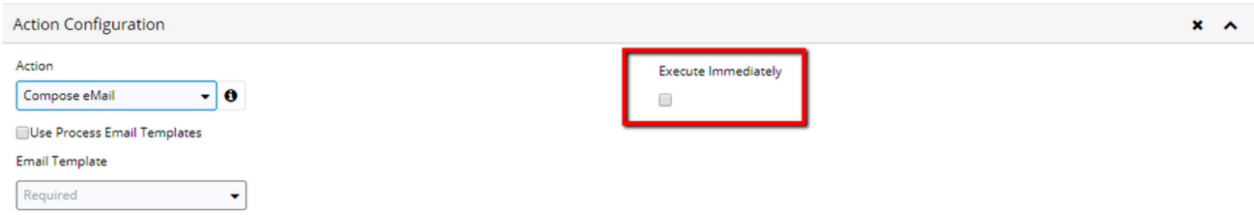
Custom Buttons in Subpanels

Custom Buttons can also be used from subpanels.



Execute Immediately

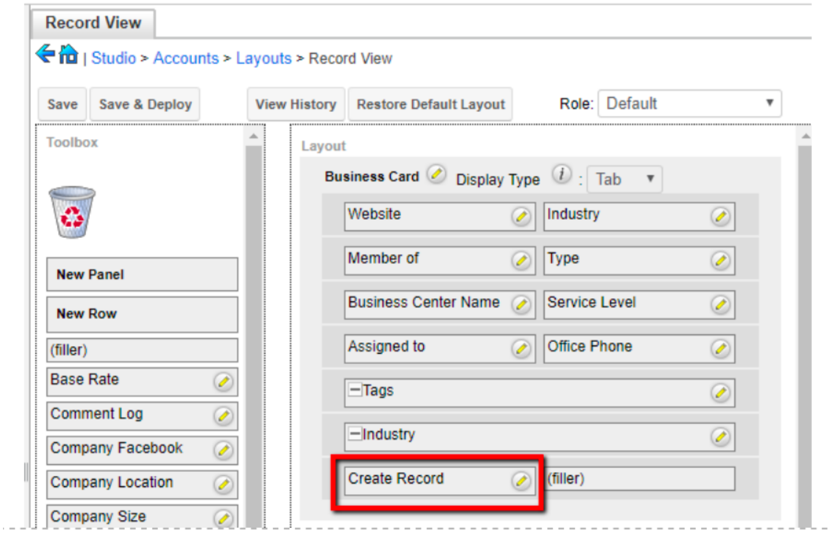
When ticked, the Execute Immediately checkbox automatically executes the actions attributed to the button when the button is pressed.



This new feature is especially useful when more than 1 actions are attributed to a single button. It allows users to execute the next action on the button without having to wait for the current one to end.

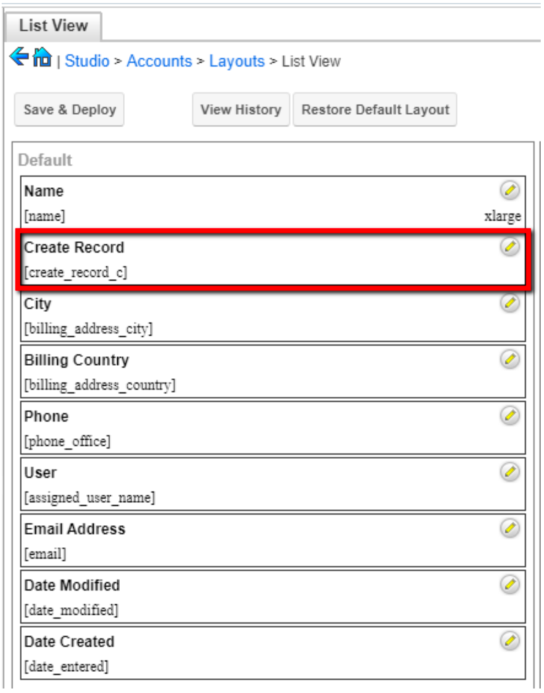
Show Buttons on Record View

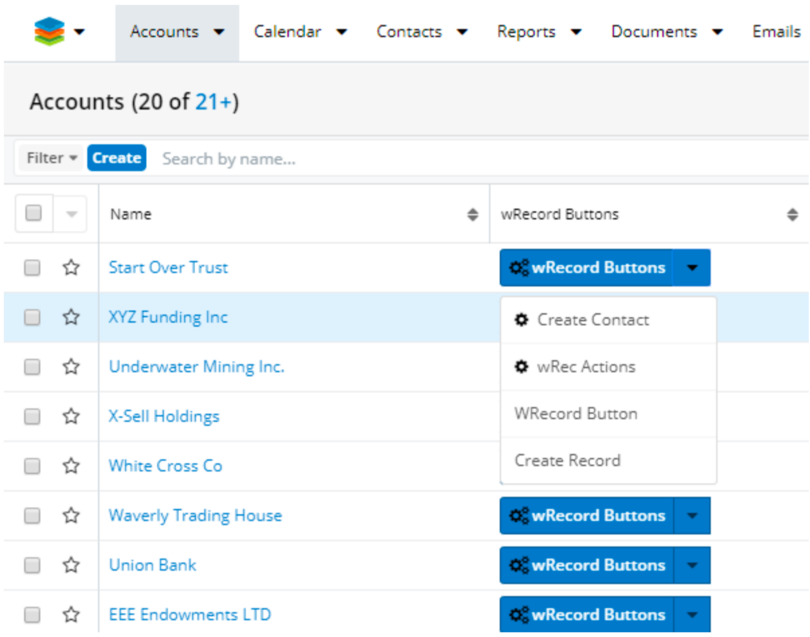
To see Custom Buttons on a module's Record View, users need to make the field visible form **Studio**, **Record View**. Drag and drop the field directly to the Record View layout. All button types are supported on Record View: Buttons, Button Groups, Dropdown.



Show Buttons on List View

To see Custom Buttons on a module's **List View**, users need to go to **Studio** and make these visible from **Layouts List View**. Drag and drop the field directly to the **Default Column** column.



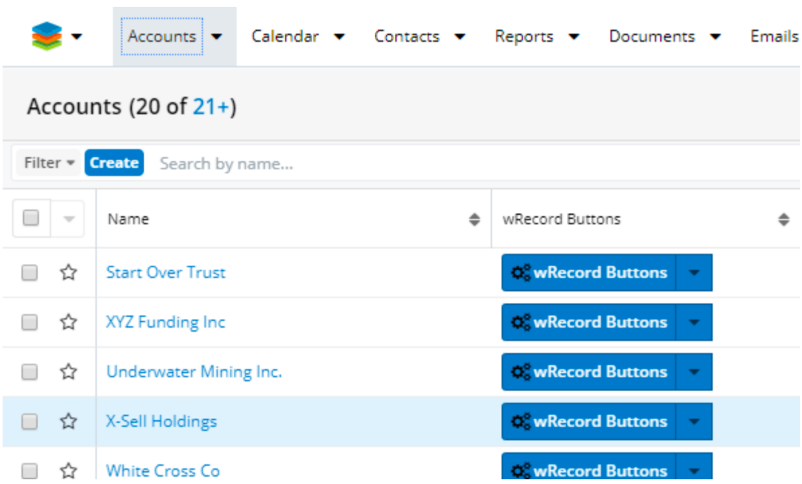


Note: If the Custom Button type is dropdown, when pressed the line in List View will be similar to the one in the screenshot above.

This function was similarly implemented to the Sugar dropdown field type; the record is in **Edit mode** when users try to pick a value.

Show Buttons on Dashlet List View

In List View dashlets, only the dropdown button type is supported, due to the fixed columns in this view mode.

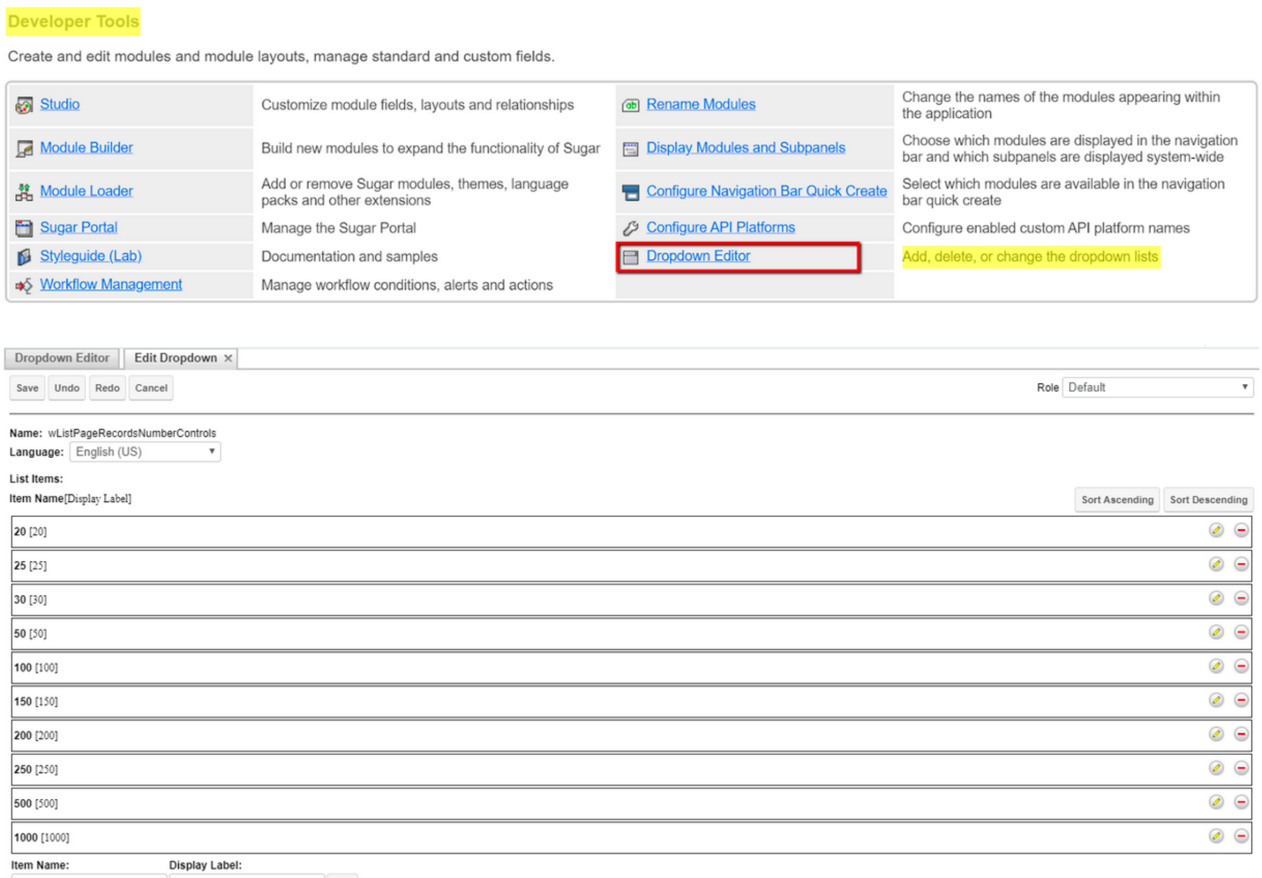


Custom Results per Module Dropdown

The **Custom Results per Module Dropdown** package allows users to control the number of records by using the **Custom Results per Module Dropdown** list strings. The package can be configured to meet each user's needs, from the **Dropdown Editor** subpanel on the **Administration** page.

Using Custom Results per Module Dropdown

The **Custom Results per Module Dropdown** package allows users to select how many entries are displayed in List View. The Results per Page numbers can be defined from the **Dropdown Editor** in the **Developer Tools** subpanel, on the **Administration** page.



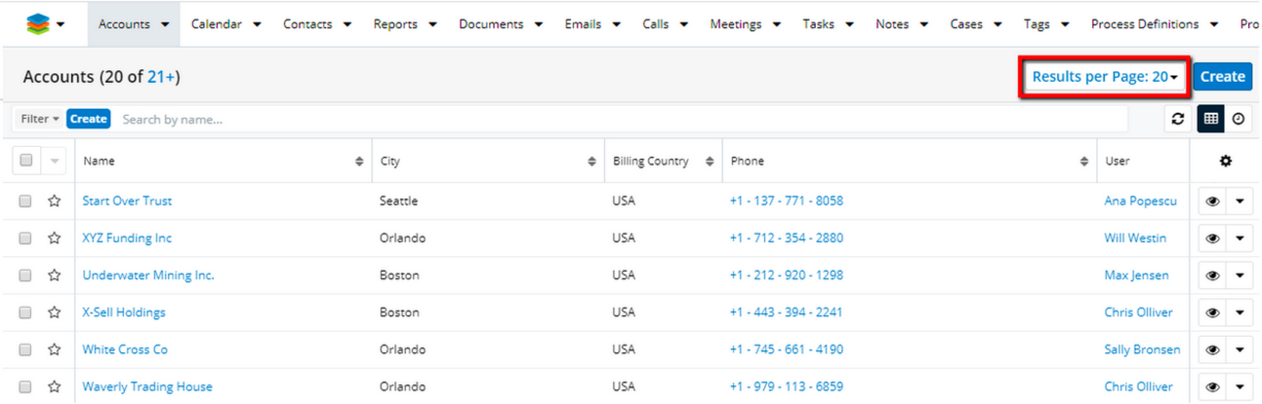
This module is available for all modules which have List View:

- Accounts
- Contacts
- Opportunities
- Leads

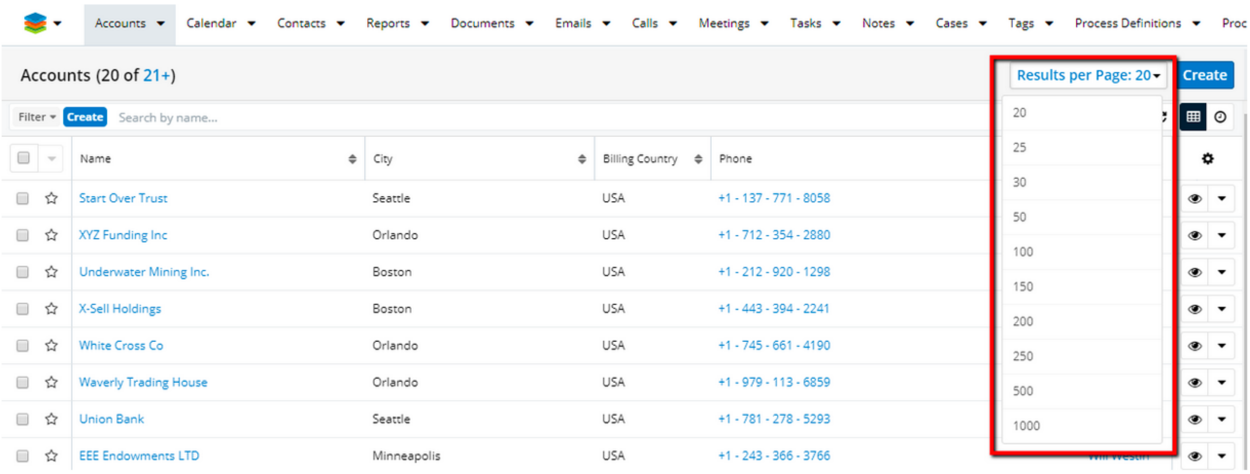
- Cases
- Calls
- Tasks
- Meetings
- Notes
- Targets
- TargetLists

List View

After installing the package, a new dropdown button will appear in List View, next to the **Create** button.



From the new dropdown field, users can select the number of entries they can view before scrolling down and clicking the " **More...**" link in **List View**.



Note: When selecting 500 or 1000 results per page, the load time for this action to finalize will significantly increase.

Embedded Content Field

The **Embedded Content** functionality allows users to import an element from internal or external sources to the current record in Sugar.

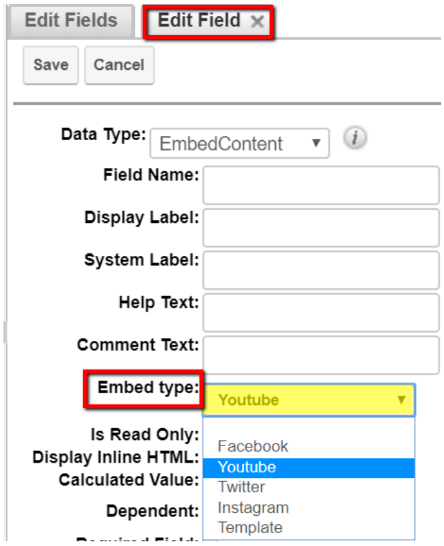
Configuring Embedded Content Field

Between installing and using the package, no additional steps are necessary.

Using the Embedded Content Field

The **Embedded Content Field** package allows users to create an **Embed Content** field type in Studio. This new field type allows users to import an element from internal or external sources to the current record in Sugar. It can be displayed on both Record View and List View.

Users have to specify the embedded type by selecting an option from the Embed type dropdown in Studio.

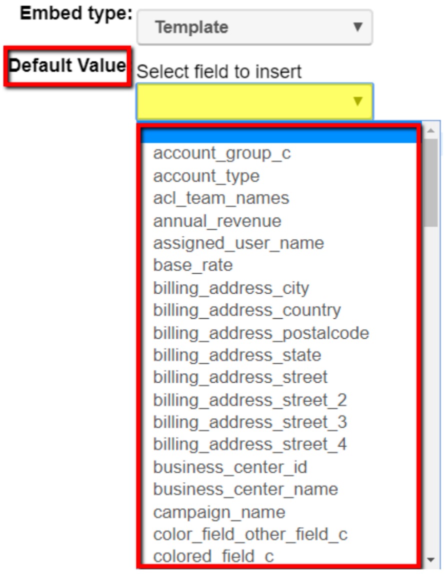


The Embedded Field Types available are the following:

- **Facebook**. The Facebook URL of a post will display, in this case, in Edit Mode. Hovering over the field will display a preview of the Facebook post.
- **YouTube**. The YouTube embedded field type will display in Edit Mode a YouTube URL. Hovering over the field will display a video preview.
- **Twitter**. The Twitter embedded field type will display in Edit Mode a link from a Twitter post. Hovering over the field will display a preview of the post.

- **Instagram**. The Instagram embedded Field type will display in Edit Mode a URL for an Instagram post. An image preview will appear when hovering over the field.
- **Template**. The embedded code can be configured only in Studio. Here the user can set up embedded Sugar variables to dynamically load embed code, merged with values from current contact record.

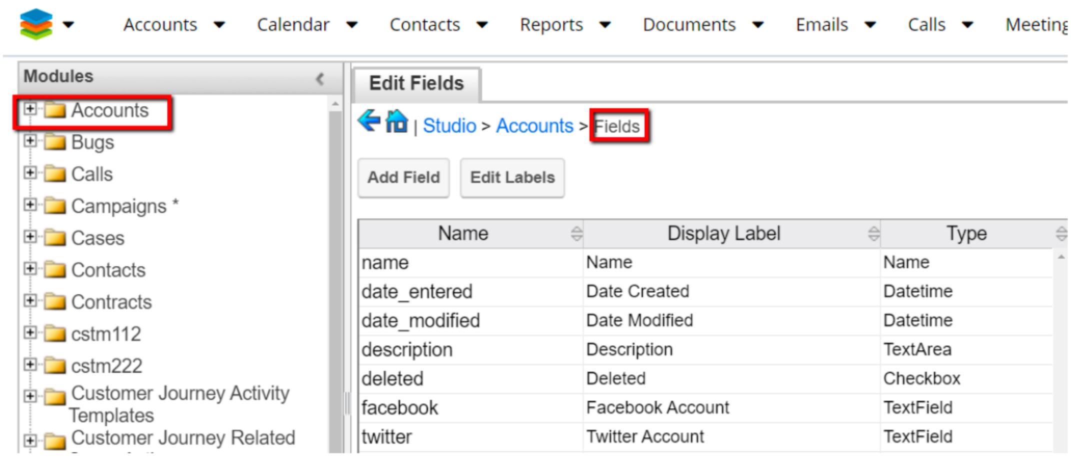
Setting the embedded field type to the **Template** option in Studio, the user can select fields to include in the embedded code entered. For this, users have to use the "Select Fields to Insert" dropdown menu.



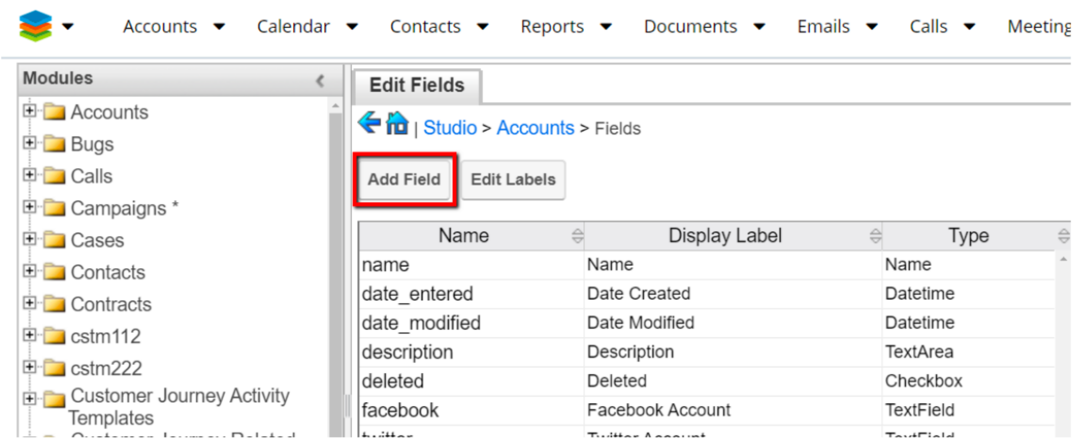
Create Embedded Content Field

Admin users can create new Embedded Content Fields by following the steps below.

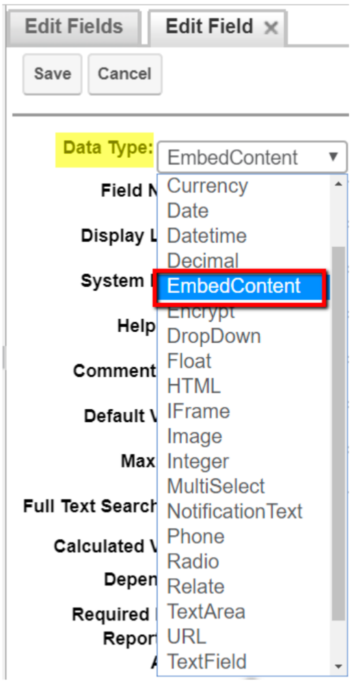
1. Navigate to **Studio** in the **Developer Tools** in **Administration**. Select the module where you want to create the **Embedded Content Field**, and go to the **Fields** section of that module.



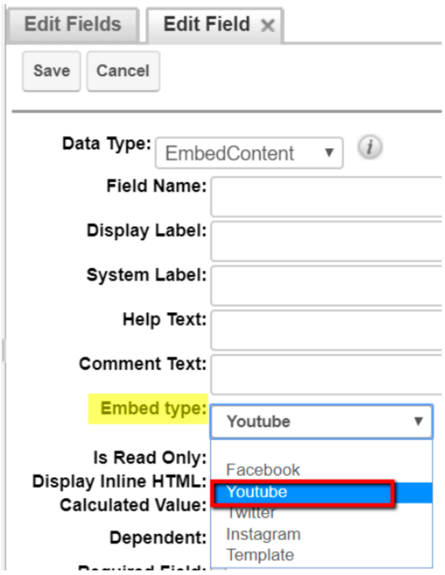
2. Press the **Add Field** button at the top of the panel.



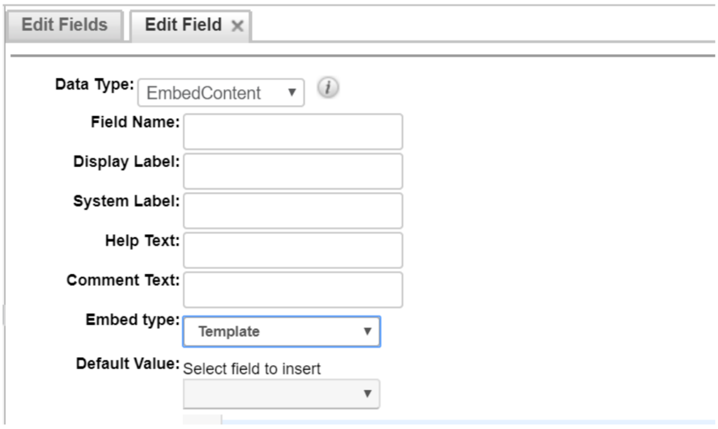
3. In **Edit Field** view, set the **Data Type** to **EmbedContent** from the dropdown menu available. Add a **Field Name** and **Display Label**.



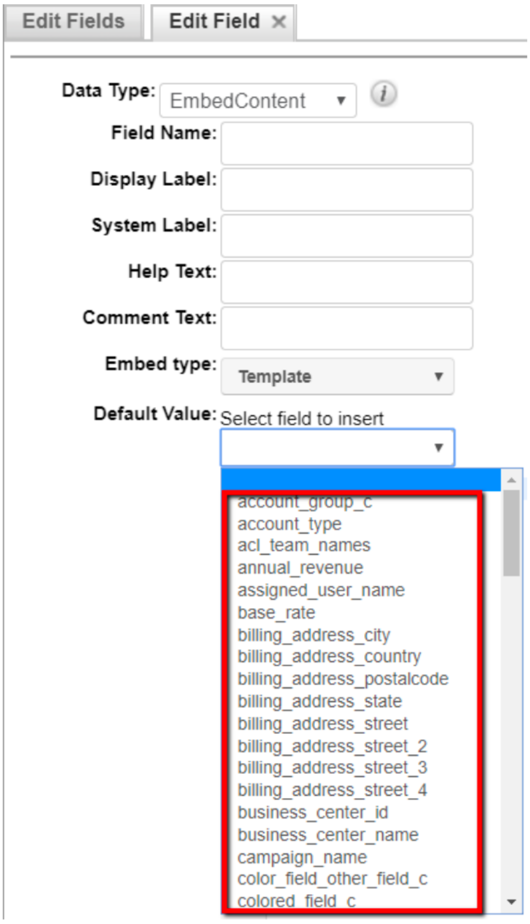
4. Select the desired option from the **Embed Type** dropdown menu.



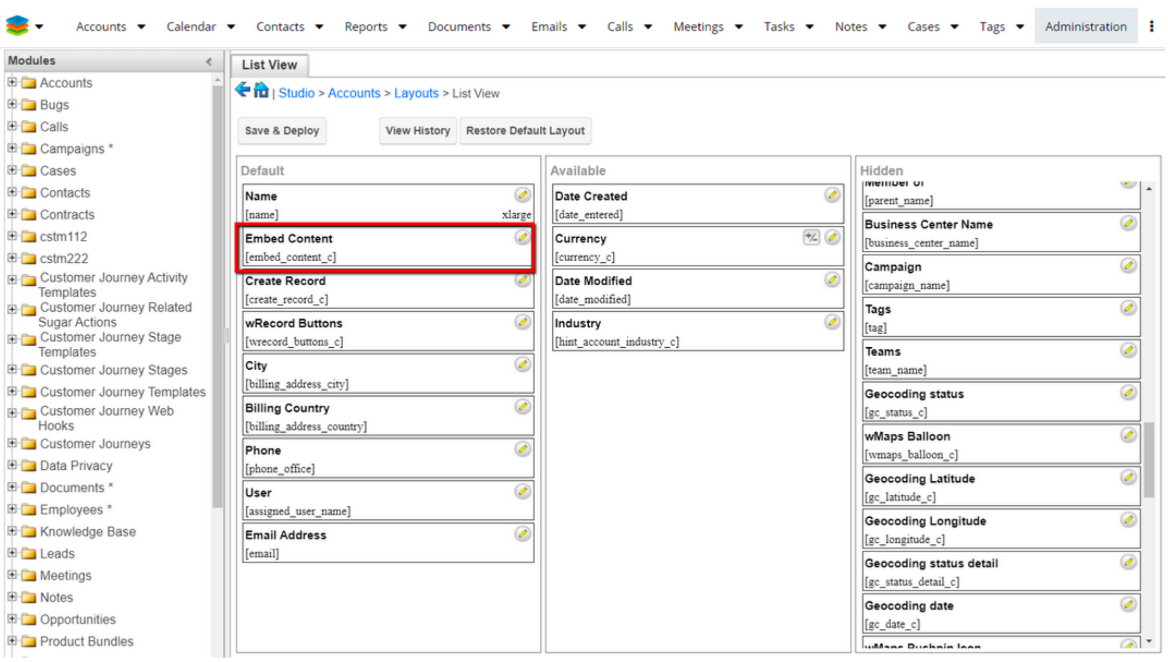
5. In case the user sets the **Embed Type** to **Template**, the **Default Value** field supports adding embedded code. This field will be displayed as default for each record created.

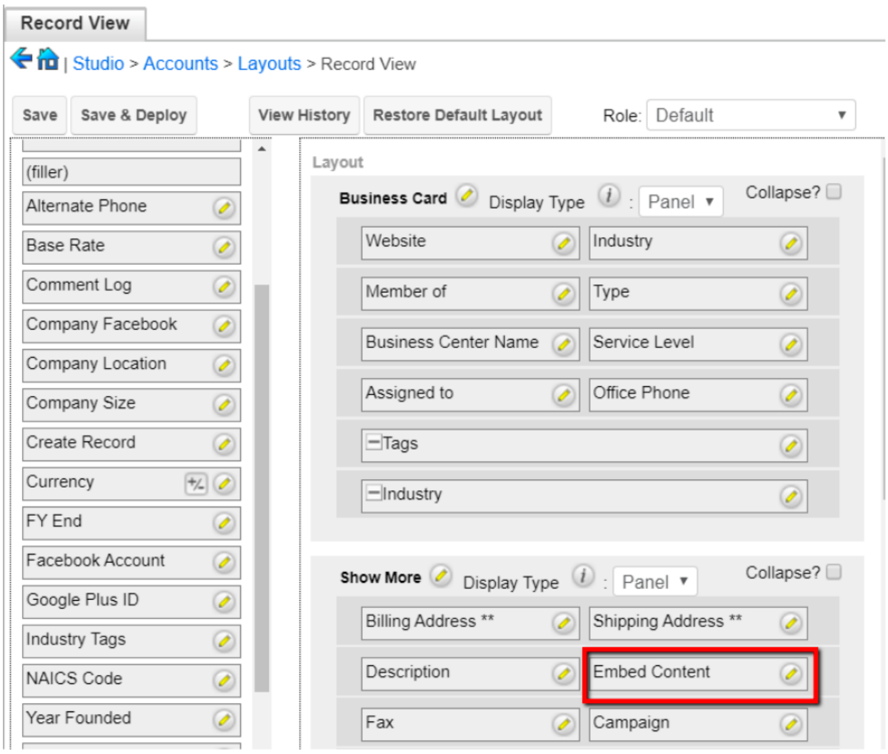


This option allows users to make fields visible for the current record. The Fields and be set up by using the " **Select Field to Insert**" dropdown menu. Here will be displayed all the fields available for the current module.



6. To save the created **Embedded Content Field**, press the **Save** button.
7. Add the **Embedded Content Field** created to both **List View** and **Record View** from the **Module Layouts**. This will make the module visible in Sugar.



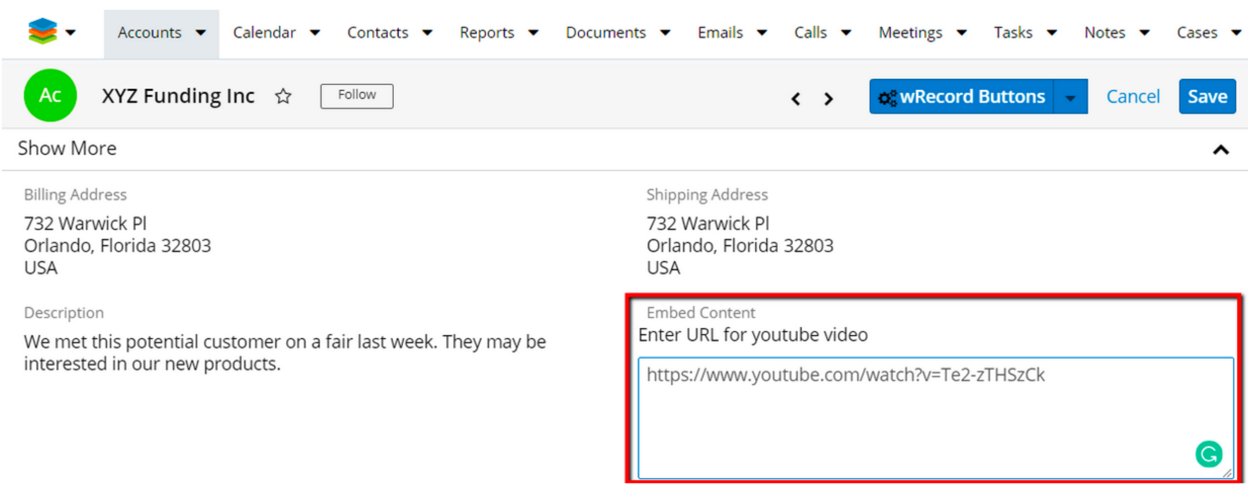


YouTube Embed Type

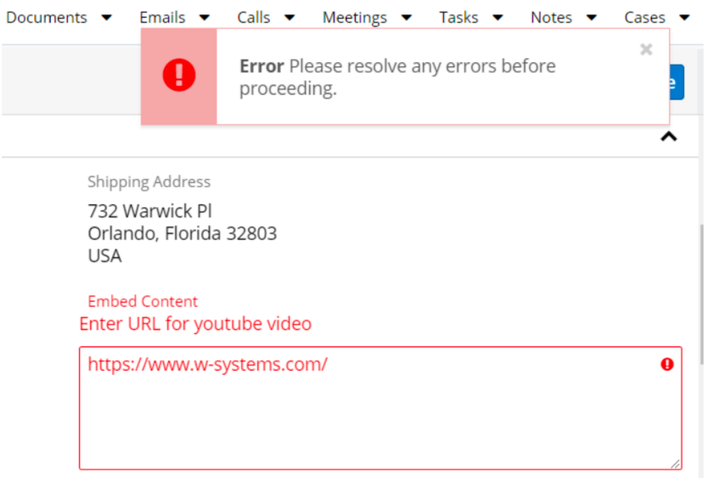
Before using the **YouTube Embed** field, users have to create a field as explained above, with the **Embed Type** set to **YouTube** (see screenshot above), and make the field visible in both **List View** and **Record View**. Using the YouTube embed field can be edited and viewed as follows.

- **Record View**.

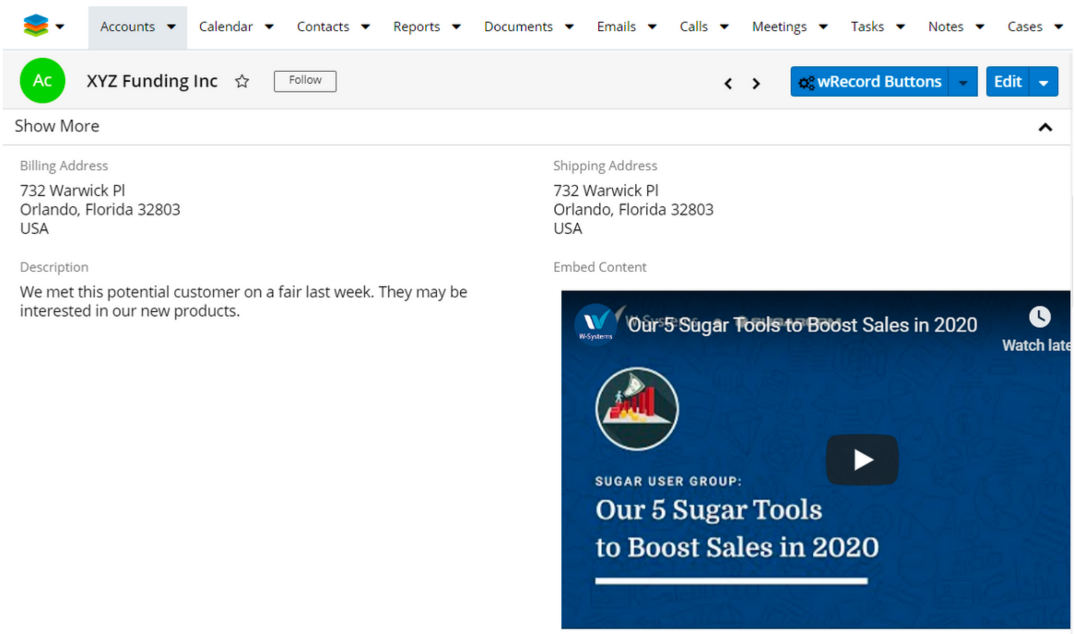
The YouTube embed field type can be quickly edited in Record View. When the field is in **Edit Mode**, users can enter and see an URL link of a YouTube video.



If the user enters a URL that is not directing to YouTube, an error message will be displayed and the changes can't be changed until the error is not corrected.

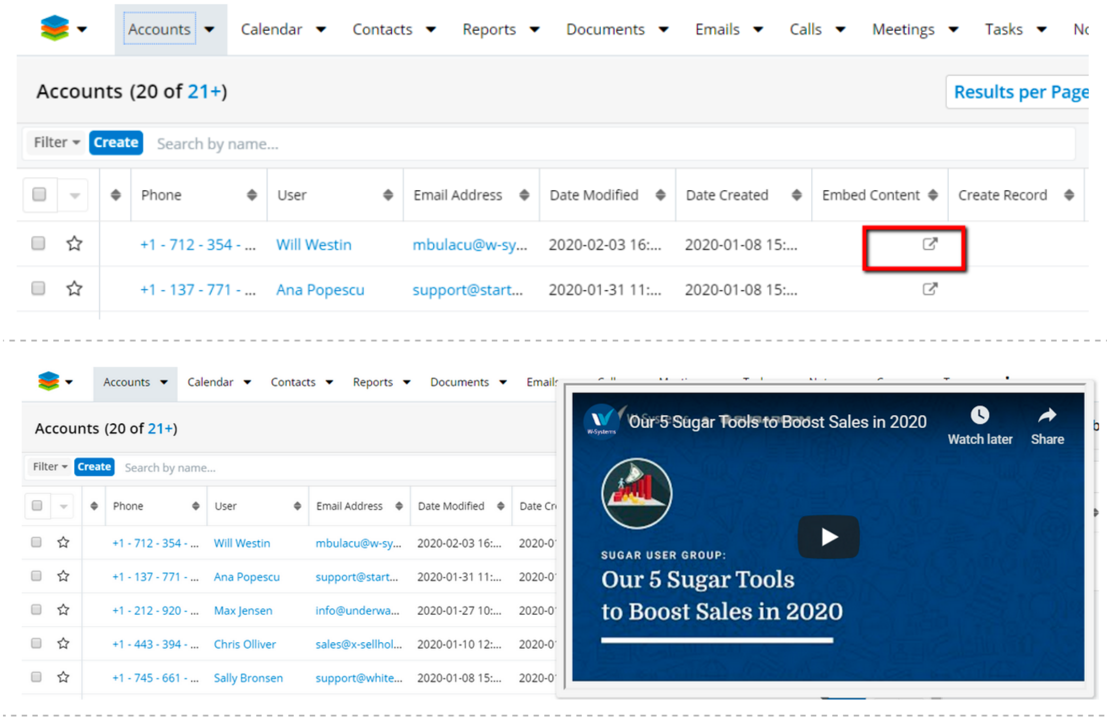


The new field will display the YouTube Video that can be played in Record View.



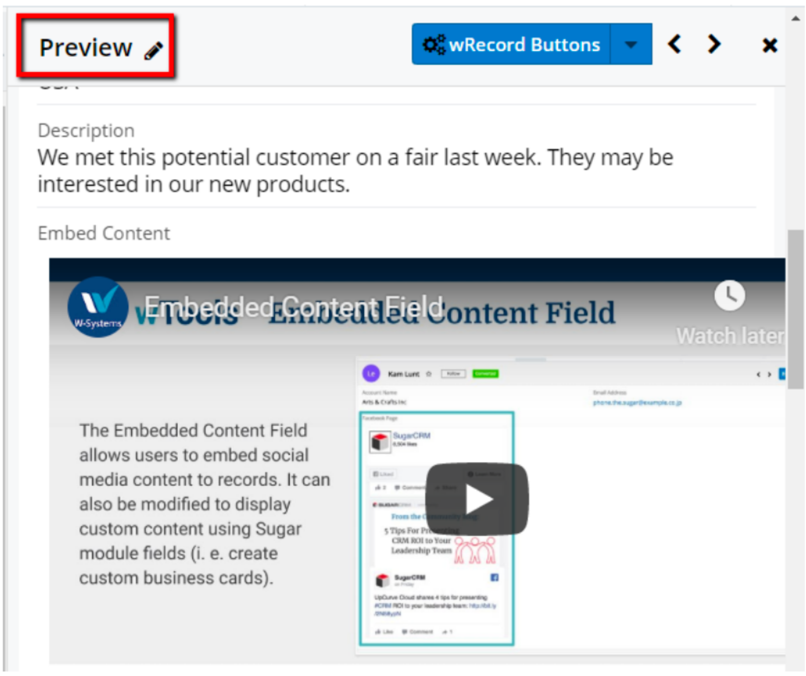
- **List View**

In **List View**, the **Embedded Content Field** has a button that opens a preview of the linked YouTube video.



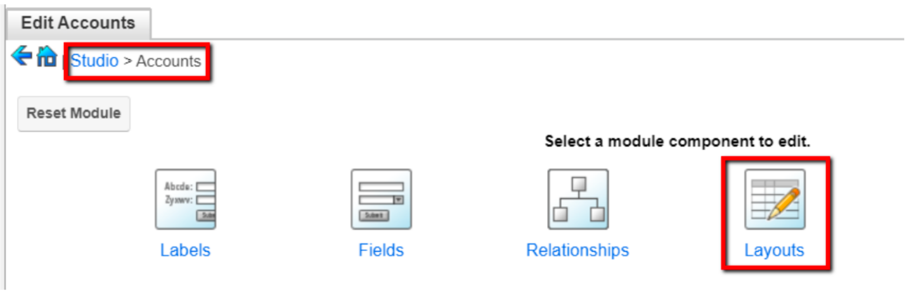
- **Preview**

In **Preview mode**, the YouTube video linked will be displayed and played in the Preview window.



Facebook Embed Type

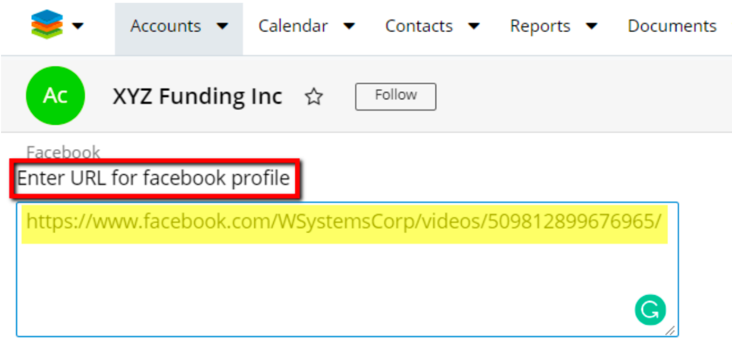
The **Facebook Embedded Content Field** has to be created with the **Embed Type: Facebook** and made visible in both **Record View** and **List View** from the **Layouts Panel** in **Studio**.



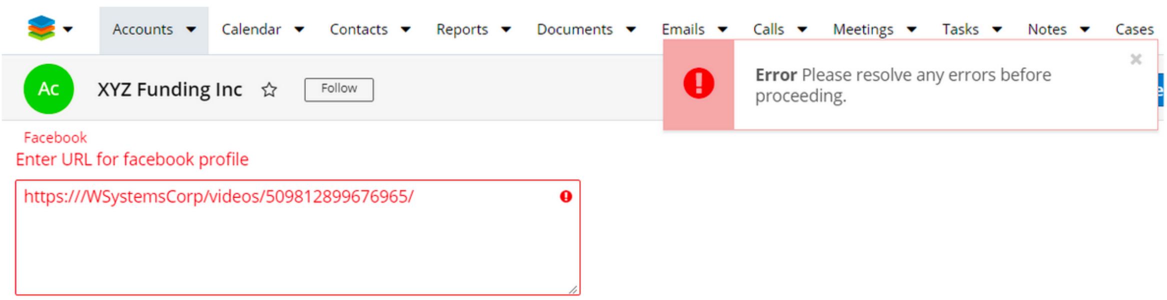
Once set up, the **Facebook Embed Type Field** can be used as follows.

- **Record View**

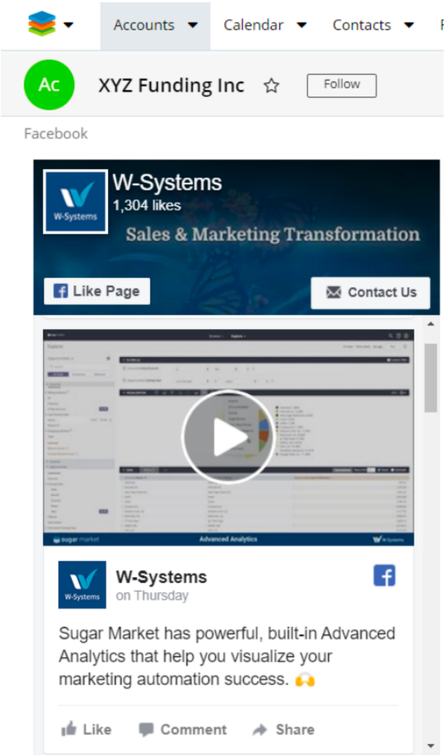
Facebook Embed Type fields can be edited in record view. When in **Edit Mode**, the users can enter and visualize Facebook URLs of Facebook posts.



If the URL entered is not a Facebook link, an error message will appear when trying to save the changes. This will prevent the user from saving their changes until the error is corrected.

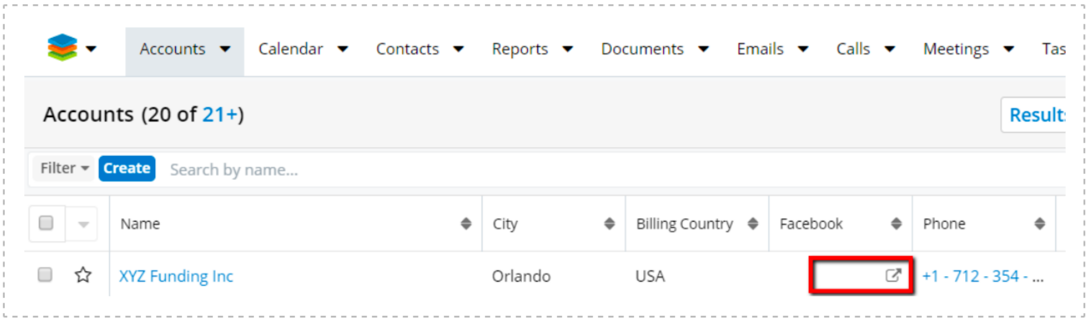


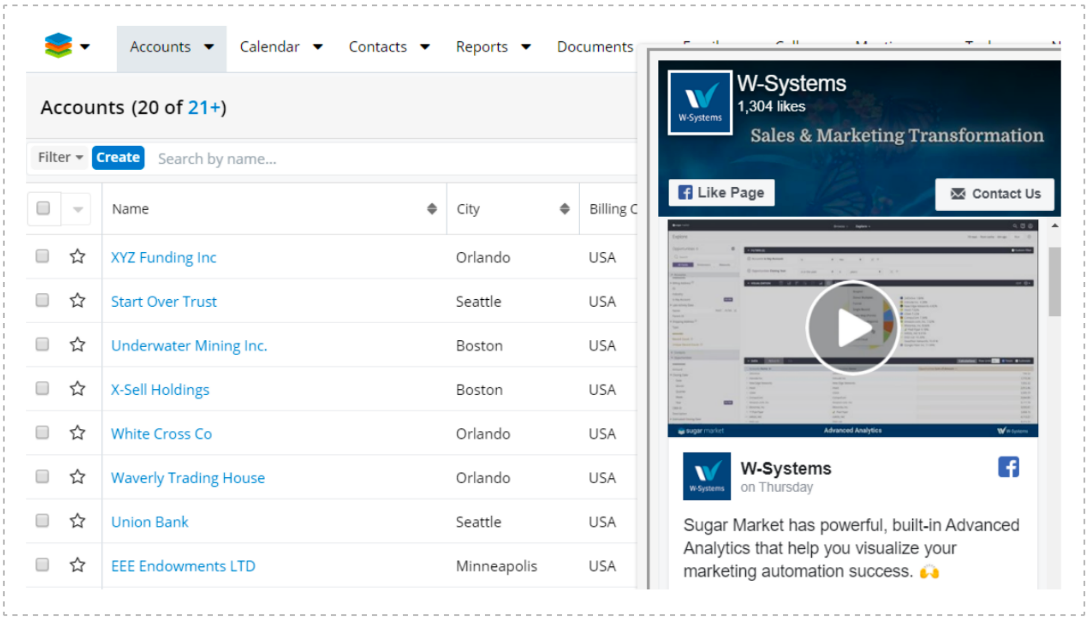
In **Record View**, the user can see a preview of the linked Facebook post.



- **List View**

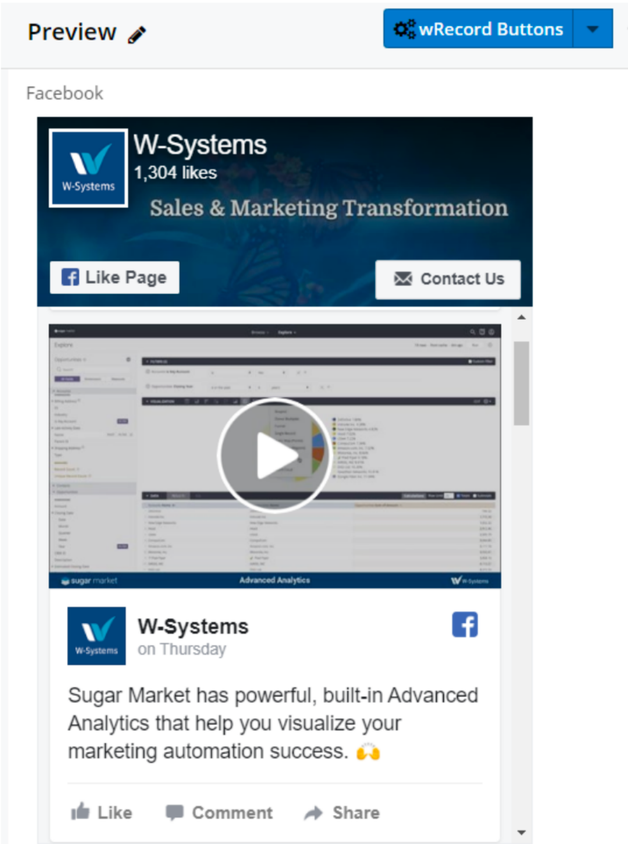
In **List View**, the **Facebook Embed Field Type** will display a button that opens a preview for the linked Facebook post.





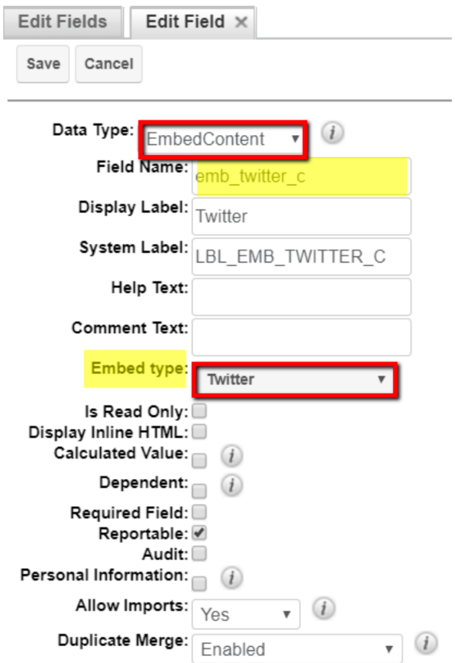
- **Preview**

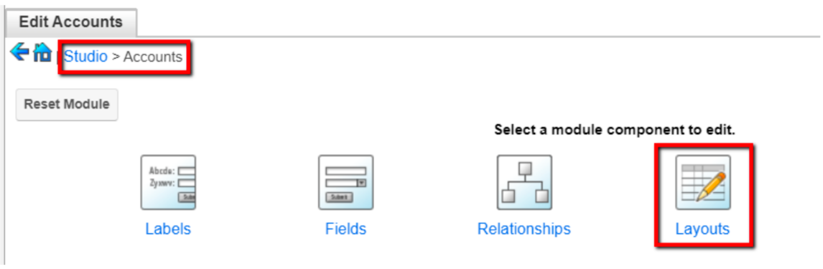
Users can also visualize the linked Facebook URL in **Preview** mode as well.



- **Twitter**

The **Twitter Embed Field Type** must be created with **Embed Type: Twitter** and made visible in both **Record View** and **List View**.

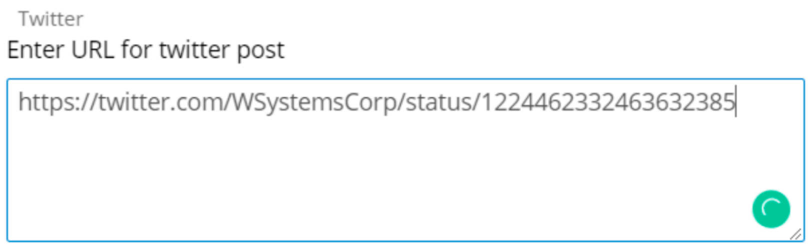




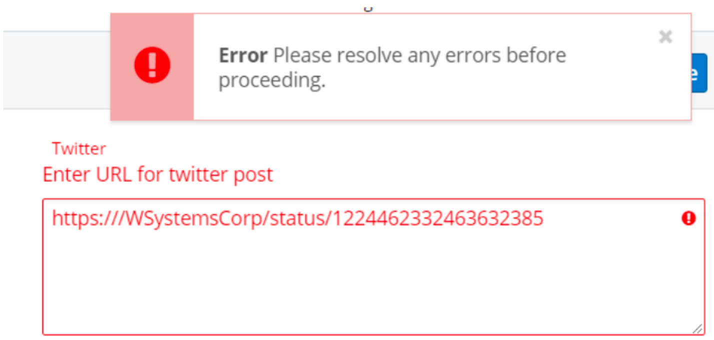
Users can edit and view the field as follows:

- **Record View**

This type of field can be edited and viewed in **Record View**. In **Edit Mode,** users can add Twitter URLs and view the posts.



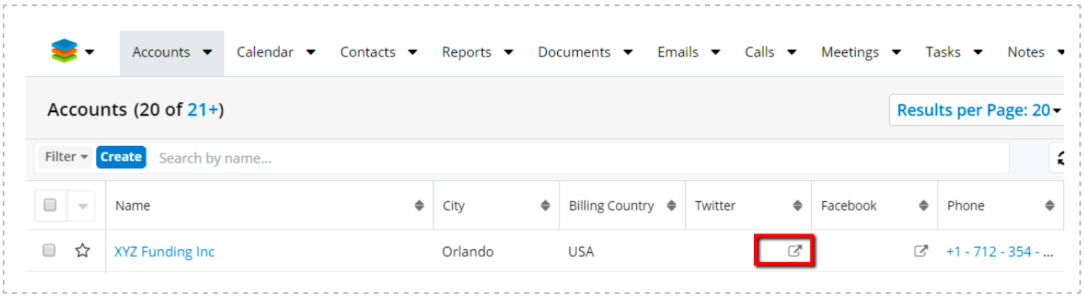
If the linked URL is not a Twitter link, the module doesn't allow saving the changes brought to the field until the error is fixed.

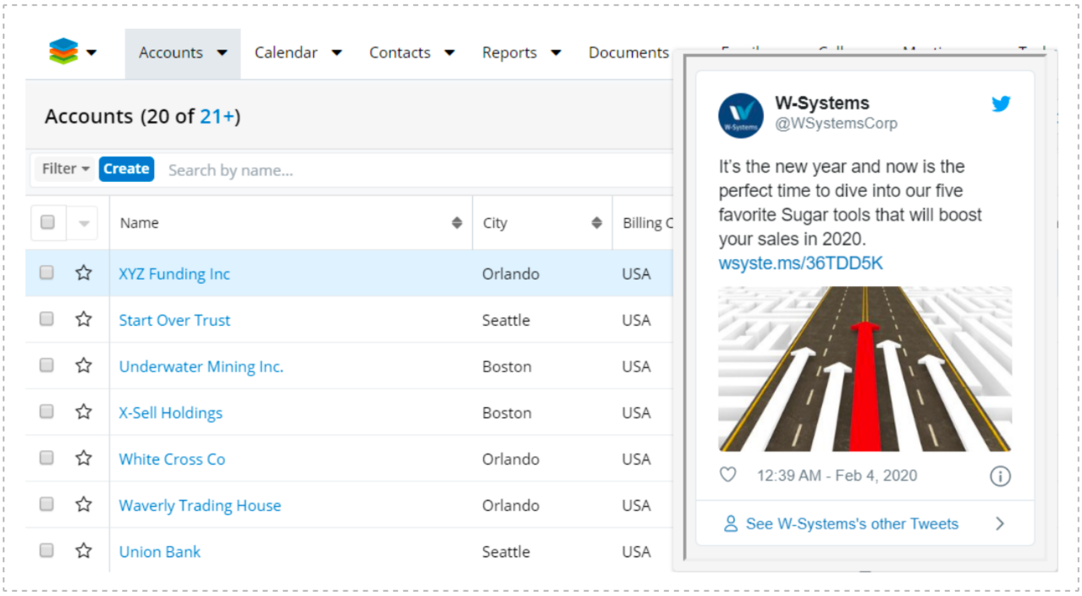


In **Record View**, users can see a preview for the Twitter post linked. In case this is a video, users can also play it in Record View.

- **List View**

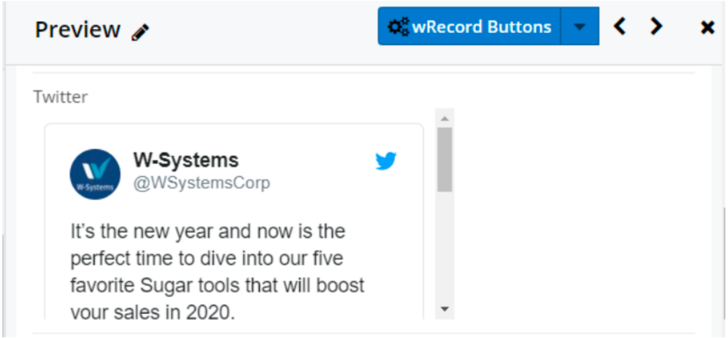
In **List View**, the **Twitter Embed Field Type** adds a new column that displays a button that opens in preview the liked post.





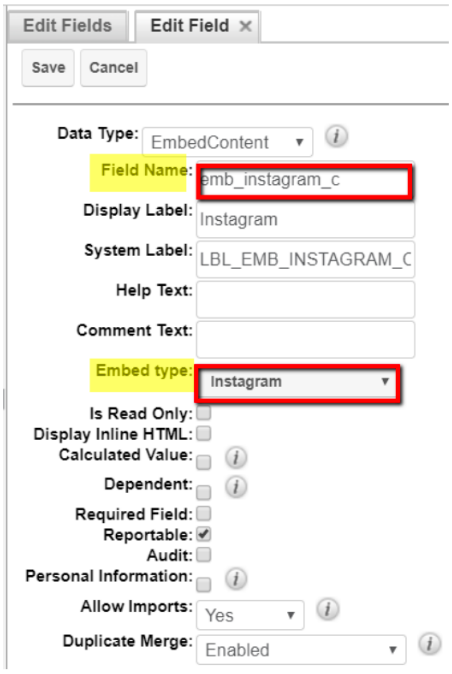
- **Preview**

Twitter posts will also be displayed in **Preview** Mode.

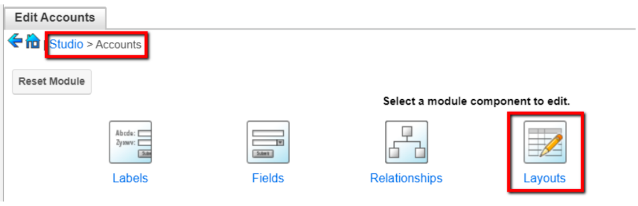


Instagram Embed Type

The **Instagram Embed Field Type** has to be created with **Embed Type: Instagram**.



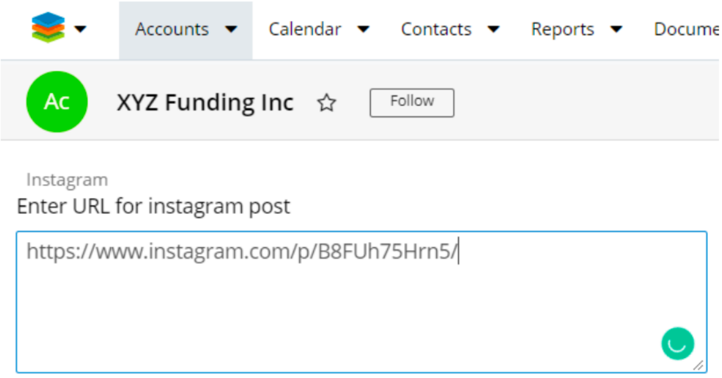
After its creation, the field must be made visible in both **Record View** and **List View**. Go to the **Layouts** panel in **Studio**, on your module of choice.



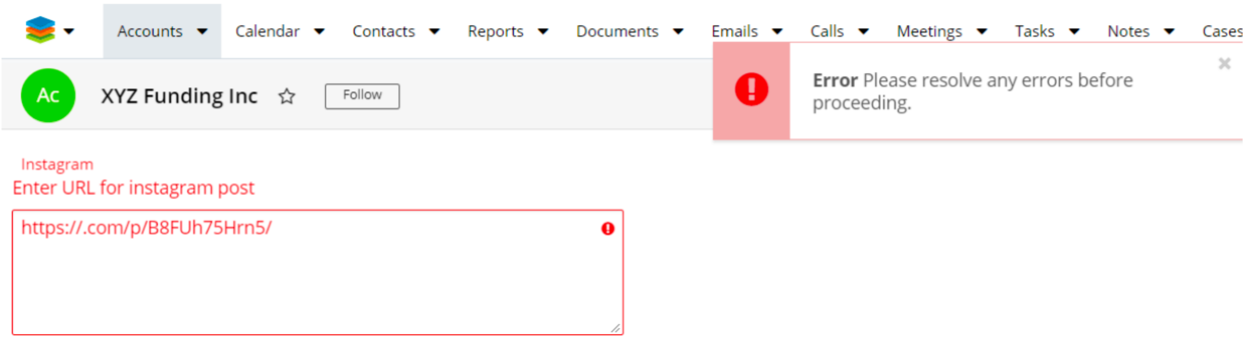
Once made visible, the field can be edited and viewed in three different ways.

- **Record View**

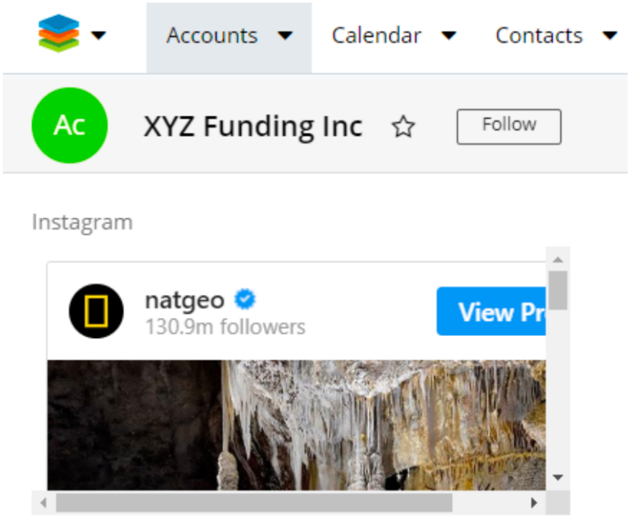
The **Instagram Embed Field Type** can be edited and viewed in **Record View**. In **Edit Mode**, users can add and see URL links of Instagram posts.



If users enter a URL that is not from an Instagram post, an error message will appear upon saving the changes. Until the error is not fixed and a valid URL is entered, the changes cannot be saved.

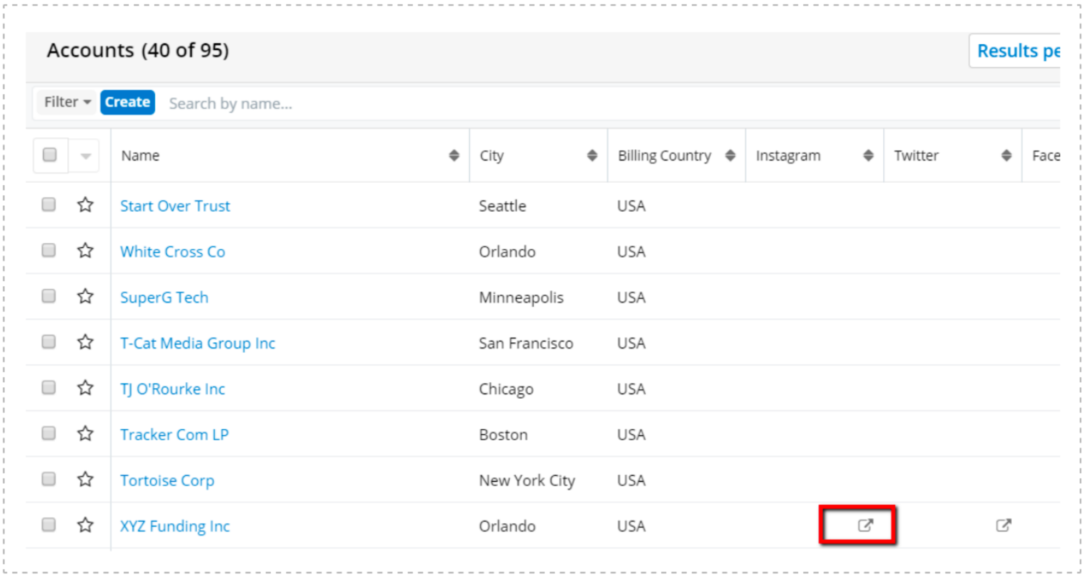


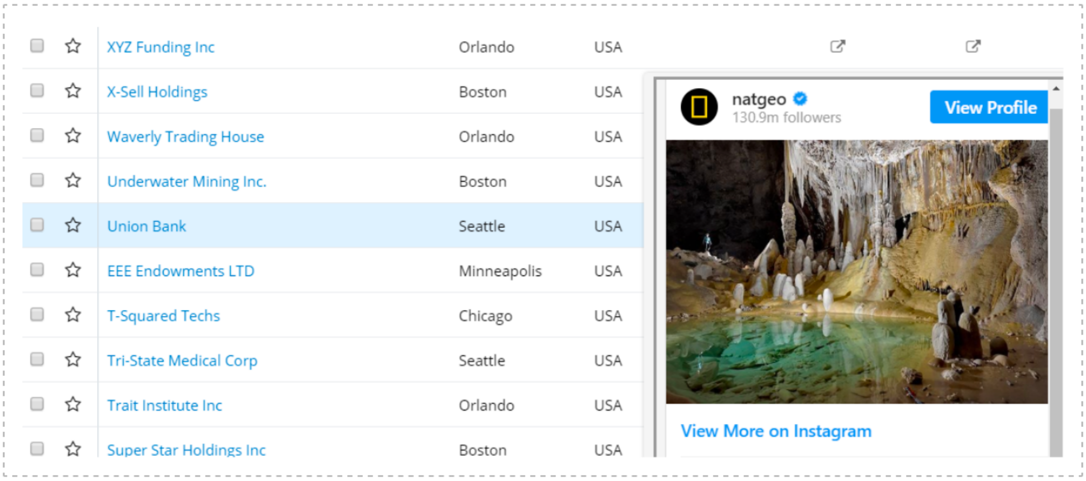
After a valid Instagram URL is entered and saved, users can see the linked post in **Record View**.



- **List View**

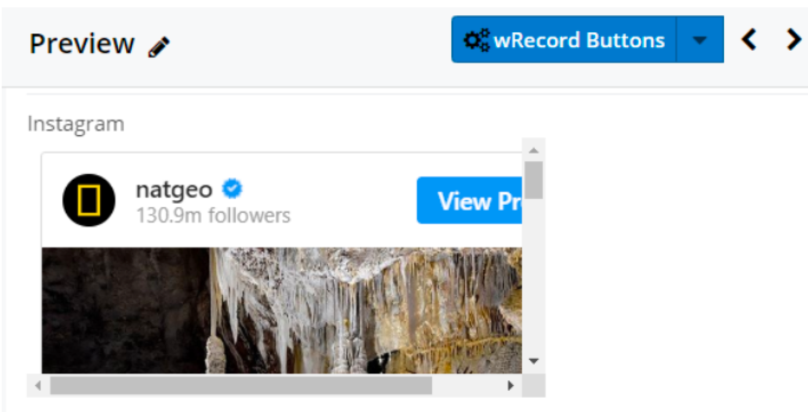
In **List View**, the field adds a new column, with a **Preview** button displayed. Clicking the **Preview button** will open a preview window for the linked Instagram URL.





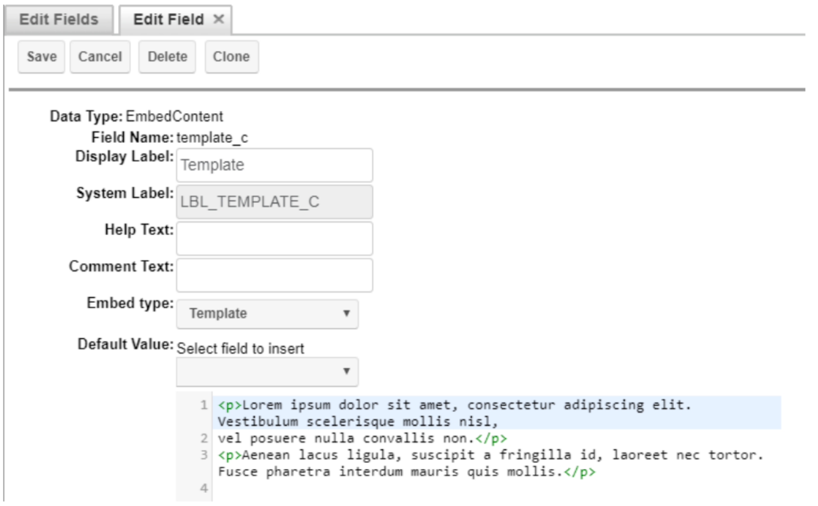
- **Preview**

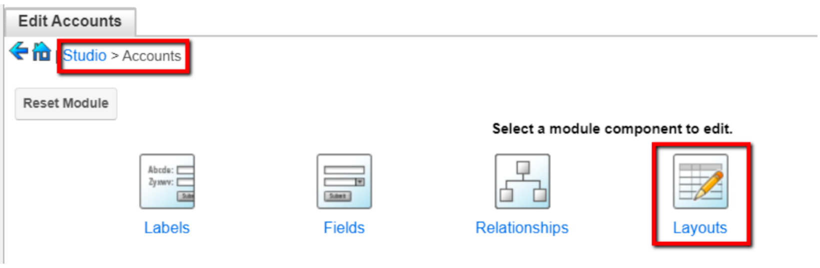
Instagram posts liked by using the **Instagram Embed Field Type** can also be viewed in **Preview mode**.



Template Embed Type

The **Template Embed Field Type** is also created in Studio. The Embed Field Type must be created with **Embed Type: Template** and made visible in both List View and Record View

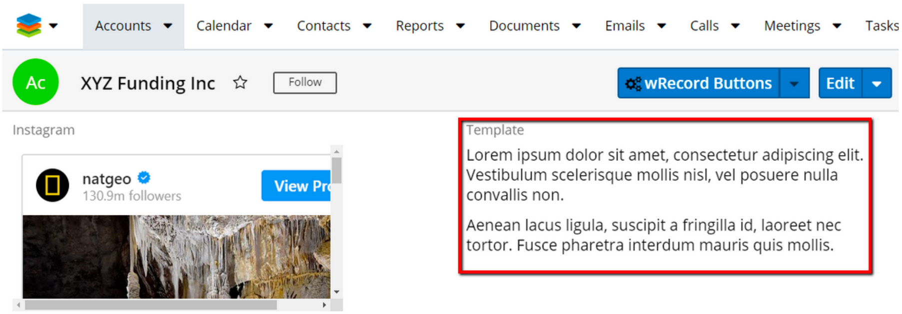




Once installed and made visible in both views, the field can be used in three different modes.

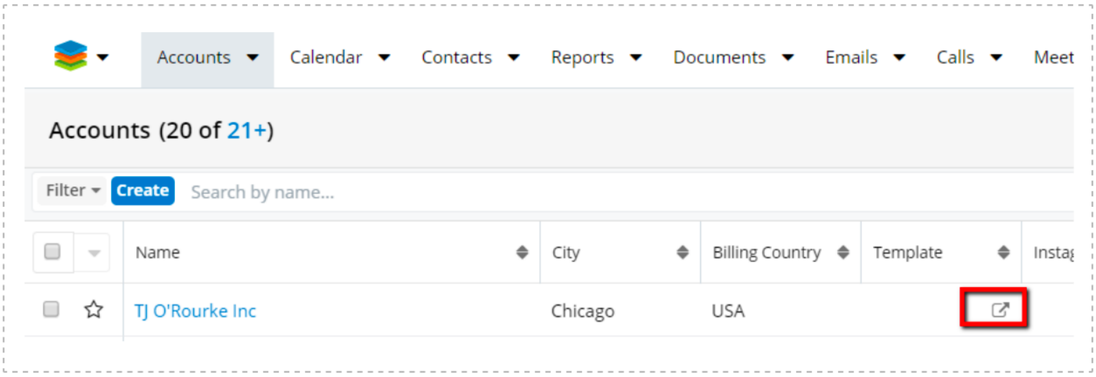
- **Record View**

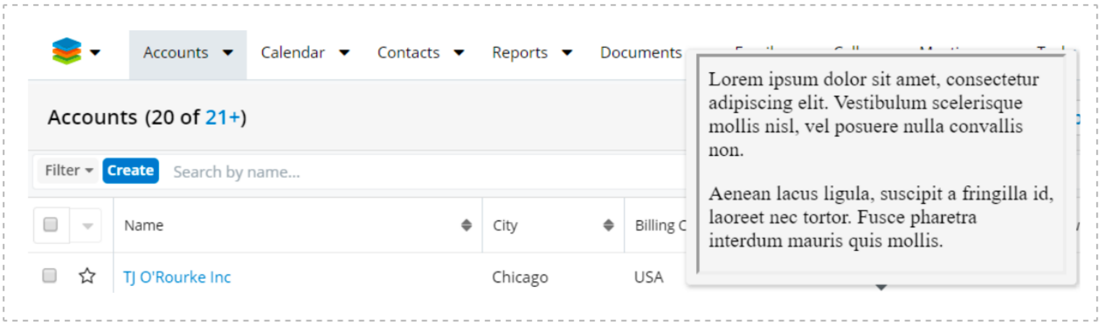
The **Template Embedded Field Type** allows users to see in **Record View** the field edited in **Studio.**



- **List View**

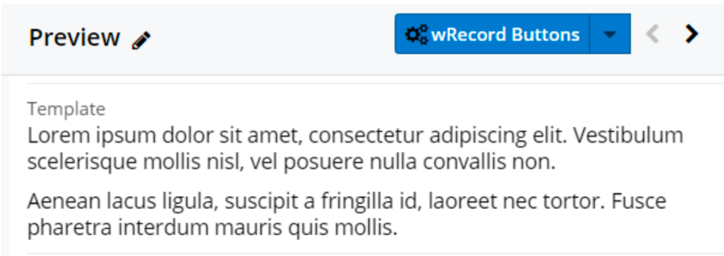
Once installed, the field will add a new column in List View with a preview button displayed for each record in List View. Users can also preview the content on the field here by **clicking the Preview** button.





- **Preview**

Preview Mode allows users to also see the content of the **Template Embed Field Type**.



SparkLine Dashlet

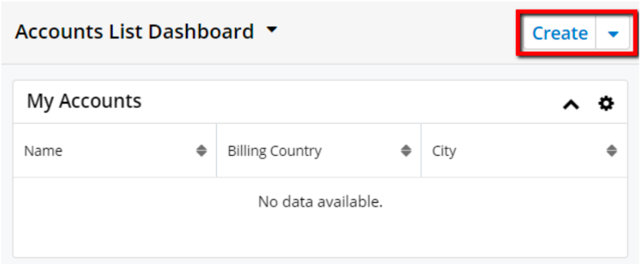
The **SparkLine Dashlet** package allows users to add charts at the top of each column in **List View**. The **SparkLine Dashlet** allows users to get a better visual representation of the data in each column.

Using the SparkLine Dashlet

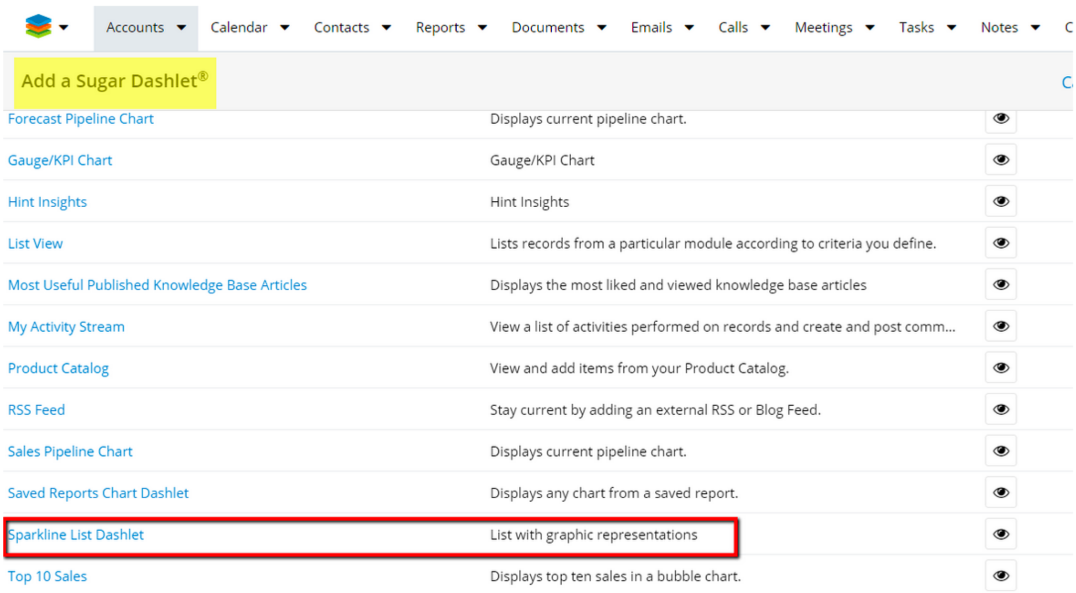
The **SparkLine Dashlet** allows users to view a list of records for a specific module in Sugar along with graphic representation at the top of each column. The supported graphic representation only supports number **Field Types: decimal, float, currency, or integer field type**.

Users have to create a new dashboard on any module in List View, once they create the dashboard in List view. To create the Dashlet, follow the steps.

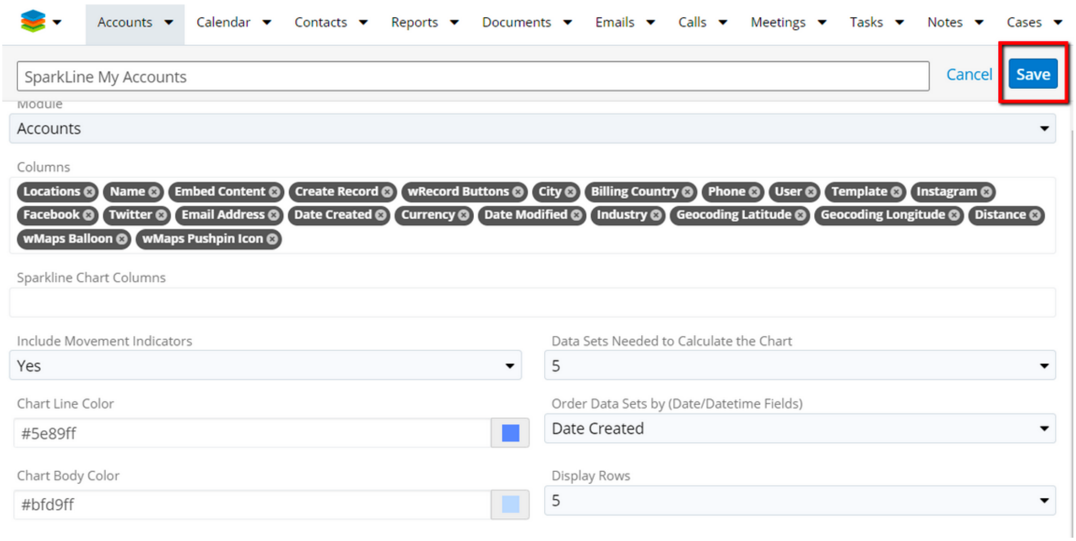
1. Go to your module of choice in **List View** and click the **Create** button on the **List Dashboard** panel.



2. Select the **wSparkLineList Dashlet** from the **Add a Sugar Dashlet** drawer that opens.



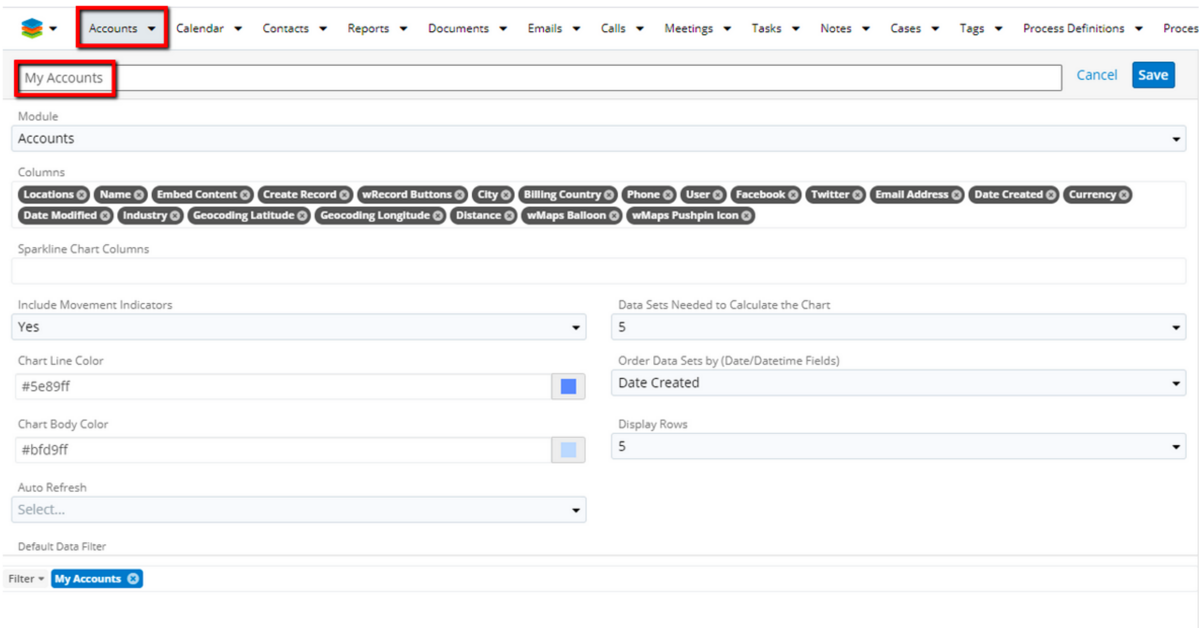
3. Once you select the **wSparkLineList Dashlet** from the drawer, a configuration panel opens.



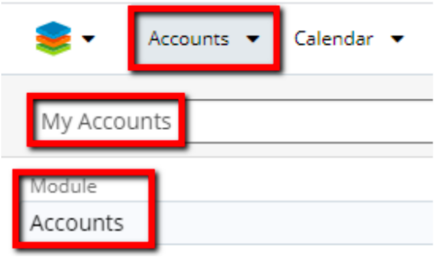
4. Configure your dashlet following the instructions in the **Configuration Options** section below.

Configuration Options

1. **Dashlet Name**
   The Dashlet Name will depend on the module where it is added. It will also be configured to match the module.



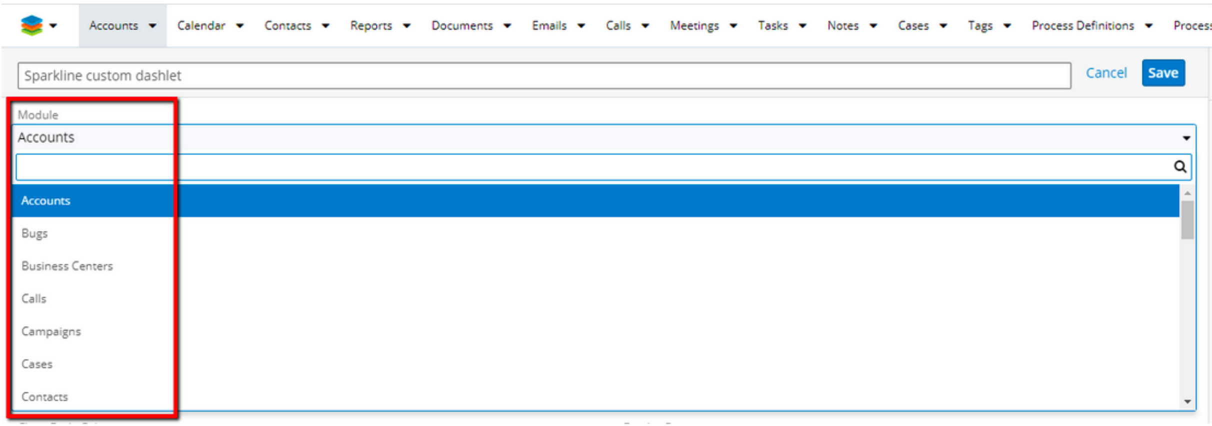
When the user selects a different module from the available dropdown, the dashlet name will also change.



After selecting the dashlet, the user can rename it.

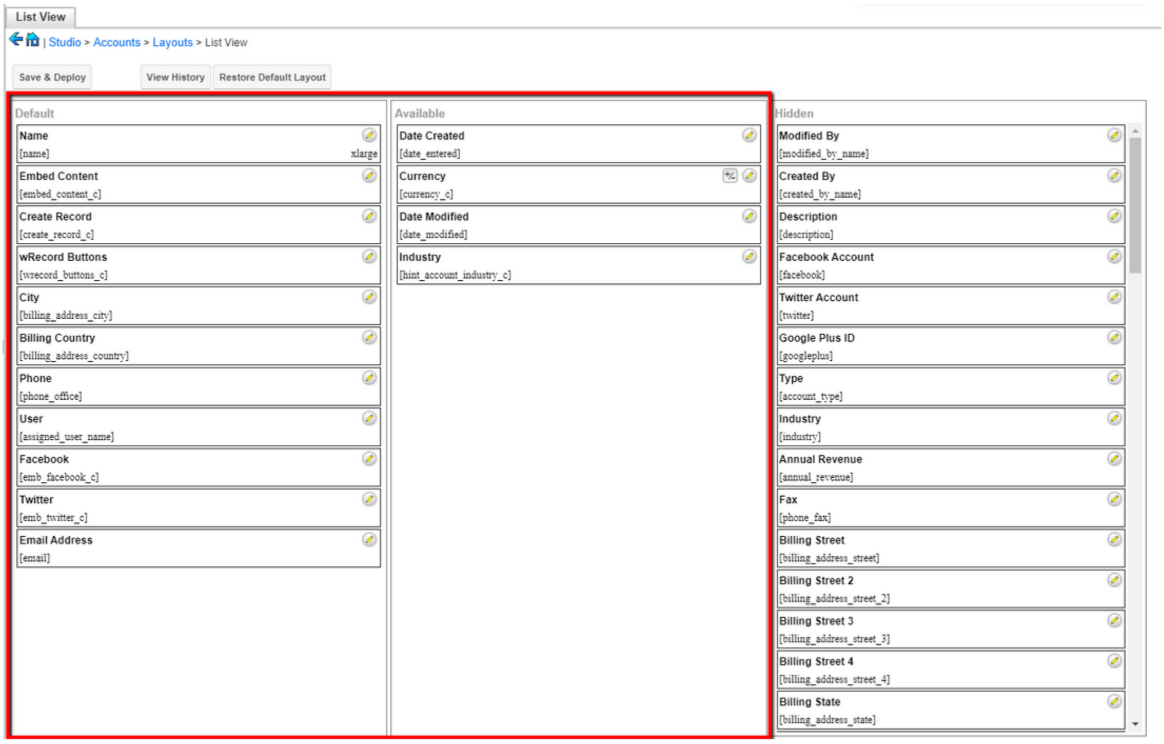
2. **Module**

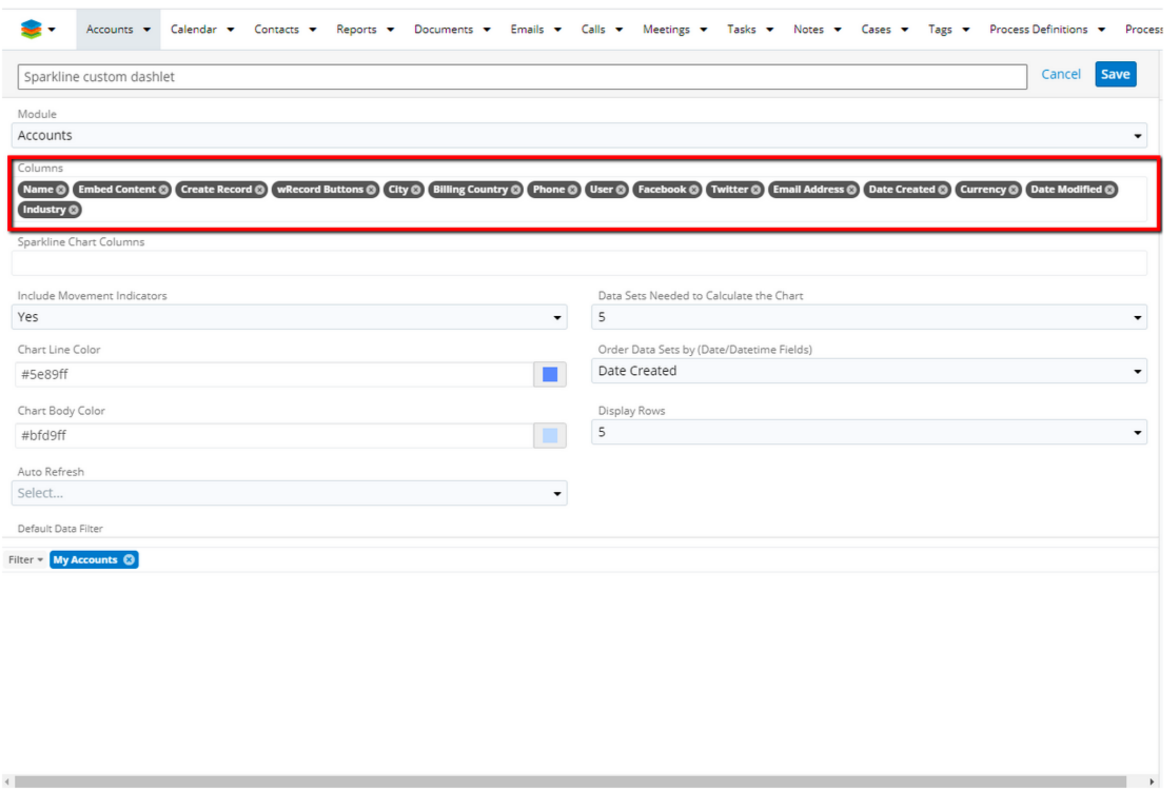
   Users are also able to pick the desired Module from the Module Dropdown.



3. **Columns**

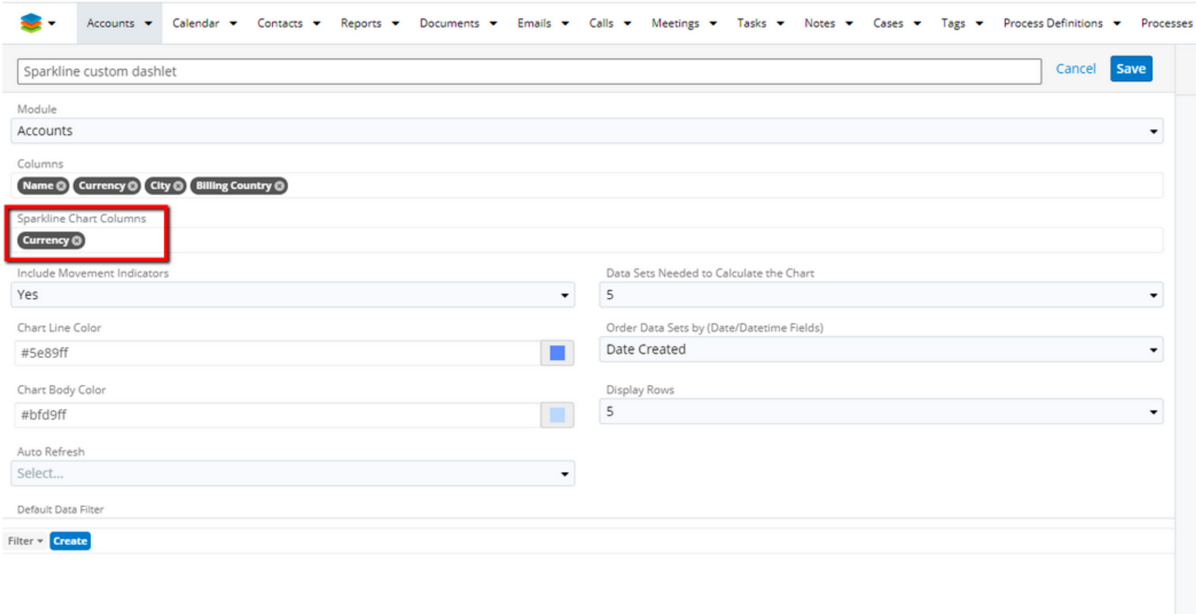
   After selecting the module, users can auto-populate with the entire list of **Column** fields available from the **List View** layout.



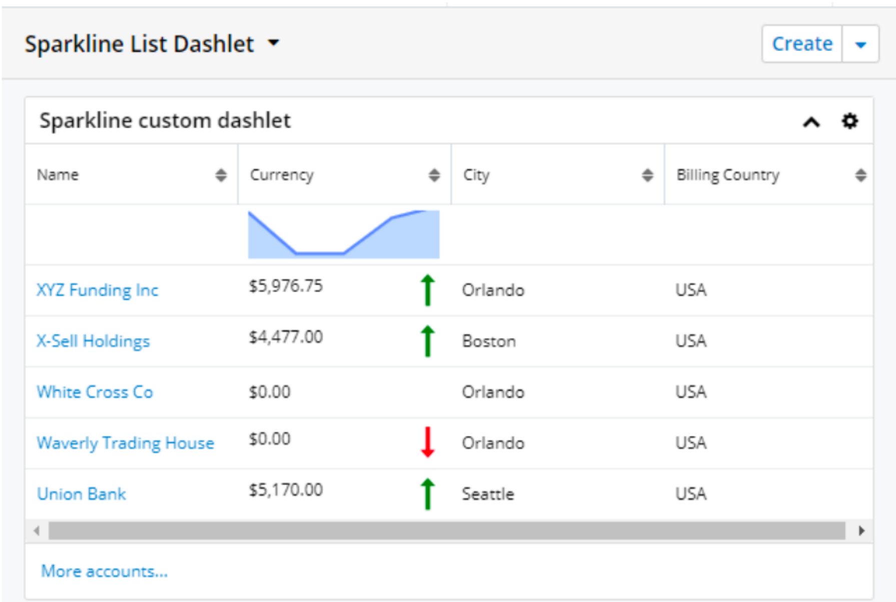


Note: The changes brought by Administrators to the List View Layout in Studio affects the available fields in the module's List View Dashlet. The changes saved will not display automatically in the corresponding module dashlet. The user will have to edit the dashlet to view the changes in the Columns field and add new fields to the layout afterwards.

4. **SparkLine Chart Columns**
   This dropdown field calculates and adds SparkLine charts, depending on the fields selected in Columns. Users can select which columns will be calculated in the Sparkline dashlet. **e.g.**: If a user wants to create a **Sparkline Dashlet** within the **Accounts** module and adds four new columns: **Name, Currency, City, and Billing Country**, he can choose the column on which the SparkLine chart will be created.

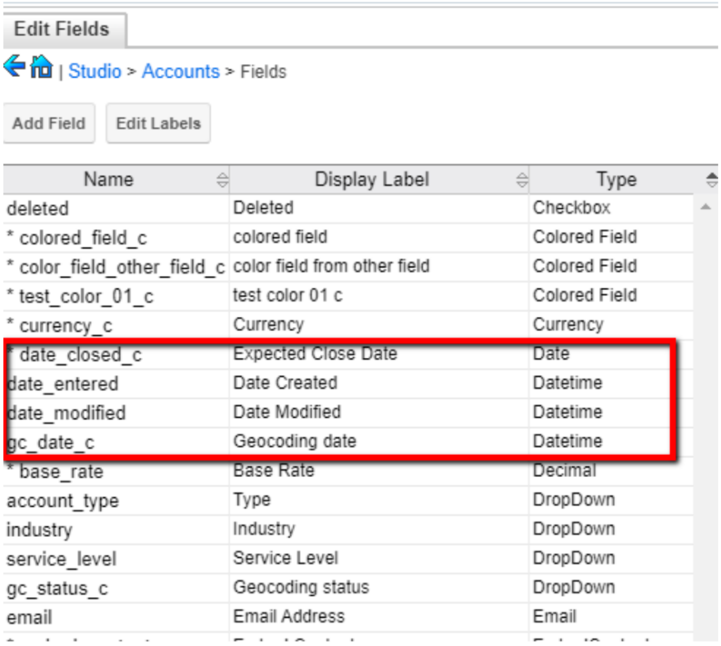


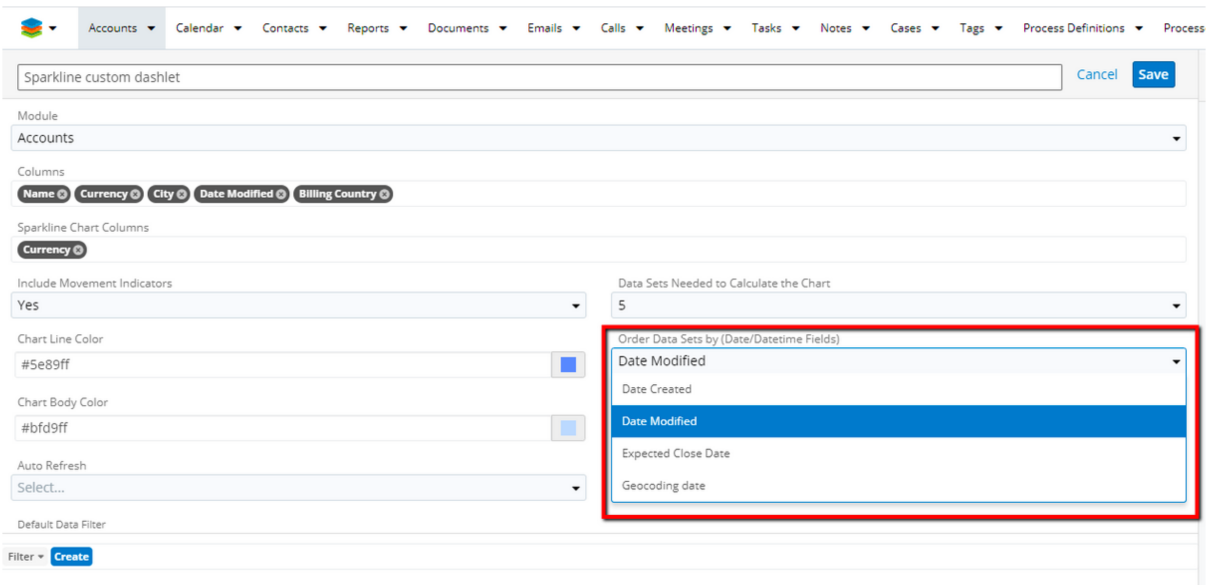
The columns selected are Name, Currency, City, and Billing Country. The chart will be attributed to the Currency column.



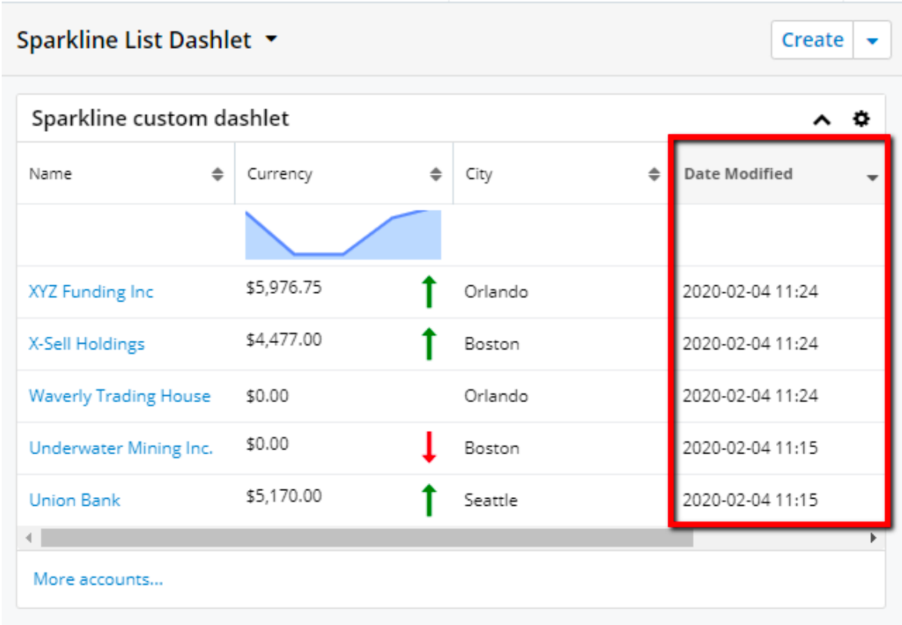
The dashlet created for the example above will look like the one in the previous screenshot.

5. **Order Data Sets by (Date/Datetime fields)**
   This configuration option is available only for the new custom **List View** dashlet created with graphic representation. The **Order Data Sets By (Date/Datetime fields)** is a dropdown field that shows all date and datetime fields available on the selected module.



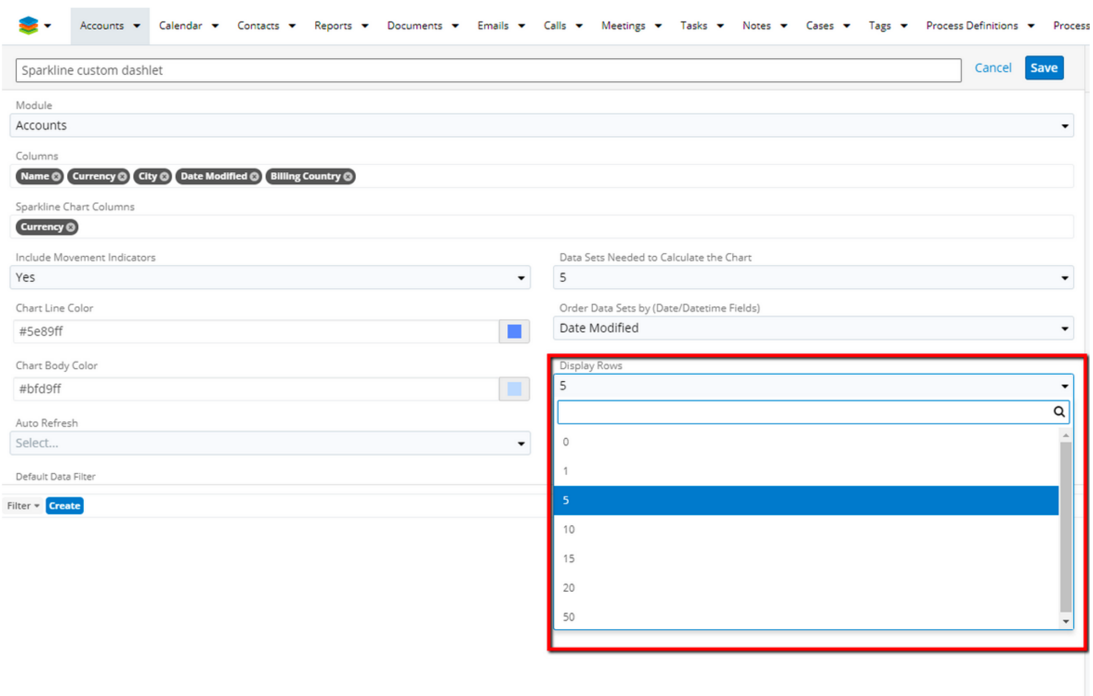


In **List View**, the chart will display the rows sorted by the selected date or datetime field in descending order.

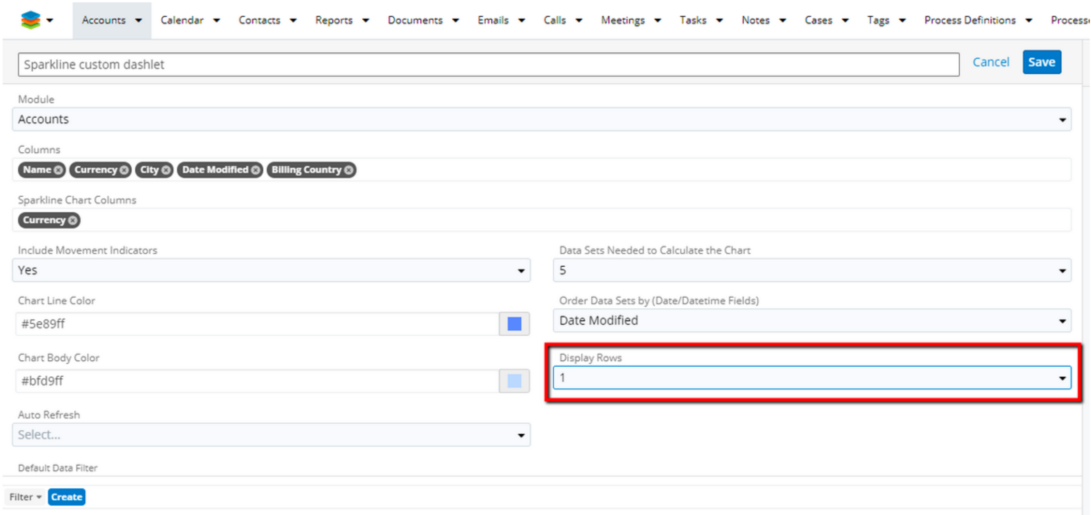


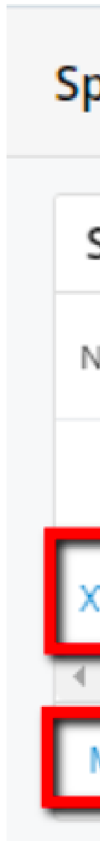
Note: When the user changes the sorting options to another column, the view will also change. Once the new sorting column is selected, the first 20 records will be sorted by that column and displayed, regardless of the Display Rows value selected in the configuration panel.

6. **Display Rows**
   By default, the "**Display Rows**" option is set in the **Configuration** page to 5 rows. This package was customized to have Display Rows options for:
   - 0 (zero) rows. When set to 0 Rows, the dashlet will only display the resulting chart, by taking the data from the **Data Sets Needed to Calculate the Chart** field.
   - 1 row. This option allows users to see the last record row in the dashlet, together with the SparkLine chart.

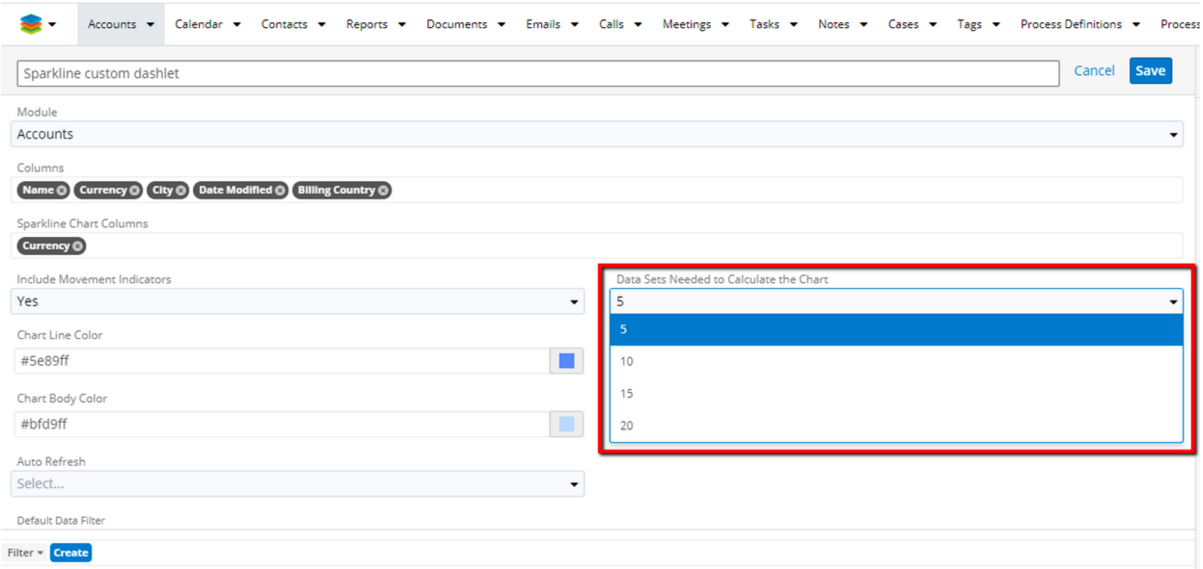


When the user presses the "**More records...** " option, in the SparkLine dashlet, another group of records will be displayed.

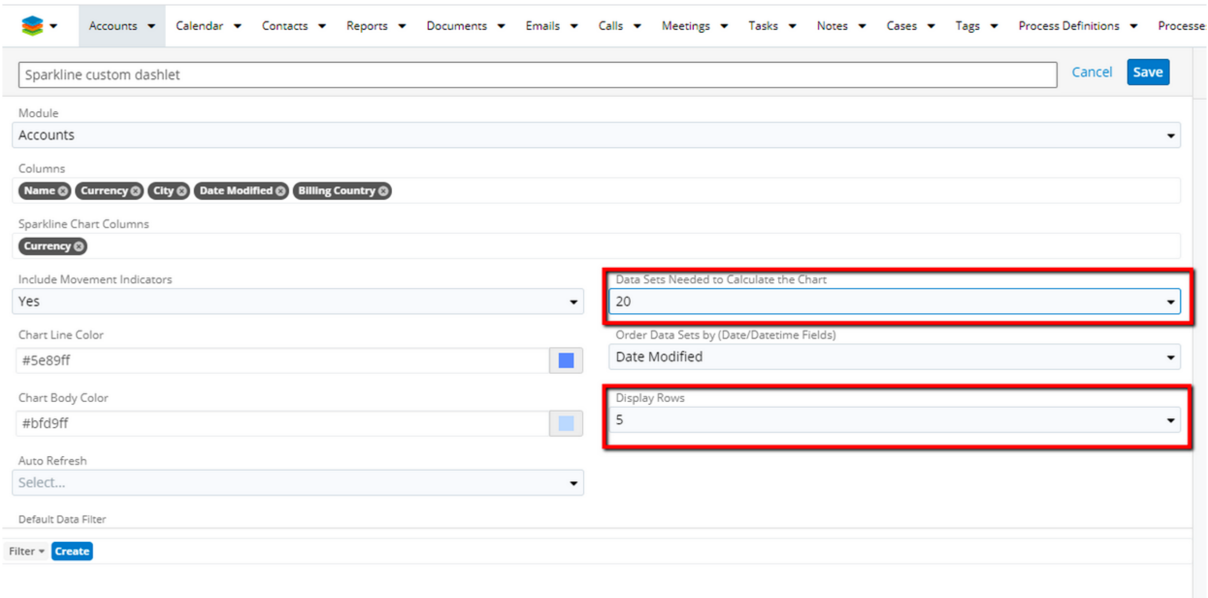




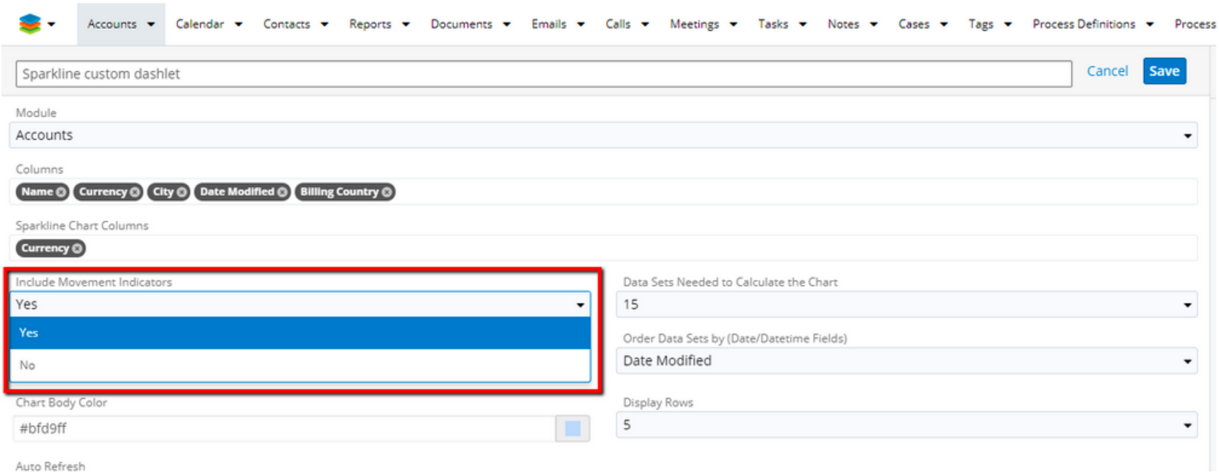
7. **Data Sets Needed to Calculate the Chart**
   This customization option works similarly to the Display Rows function. It allows the user to select the number of rows included in the data series for the SparkLine chart options: 5, 10, 15, 20.



For example, users can choose to display only 5 records, but to calculate the chart for the first 20 records.

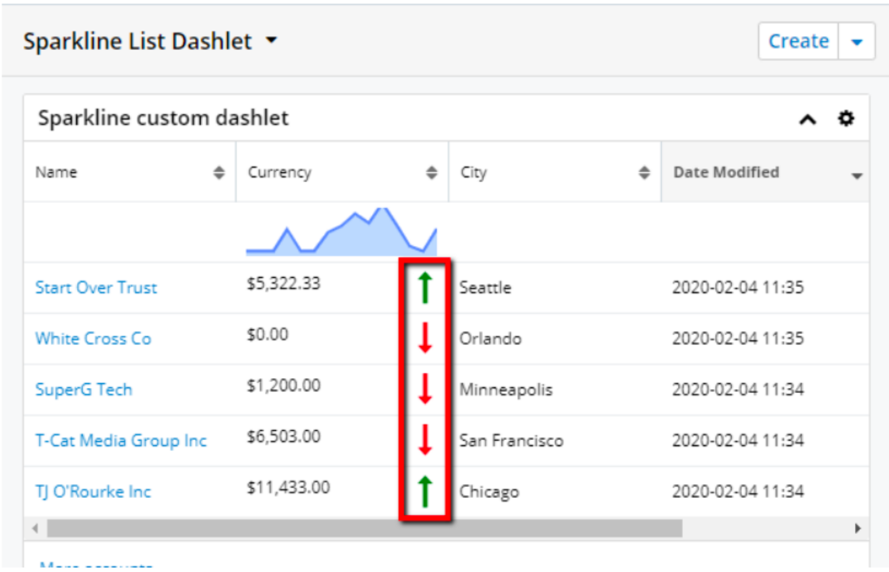


8. **Include Movement Indicators**
   This option allows users if they want to **Include Movement Indicators** in the chart. The field is a dropdown module from where users can check Yes/No. By default, the value for this field is set to Yes.

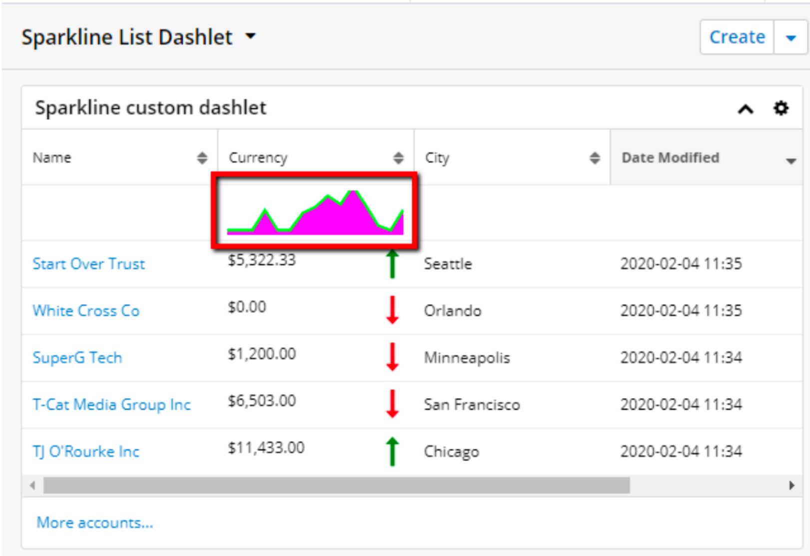


When used, this option will place:

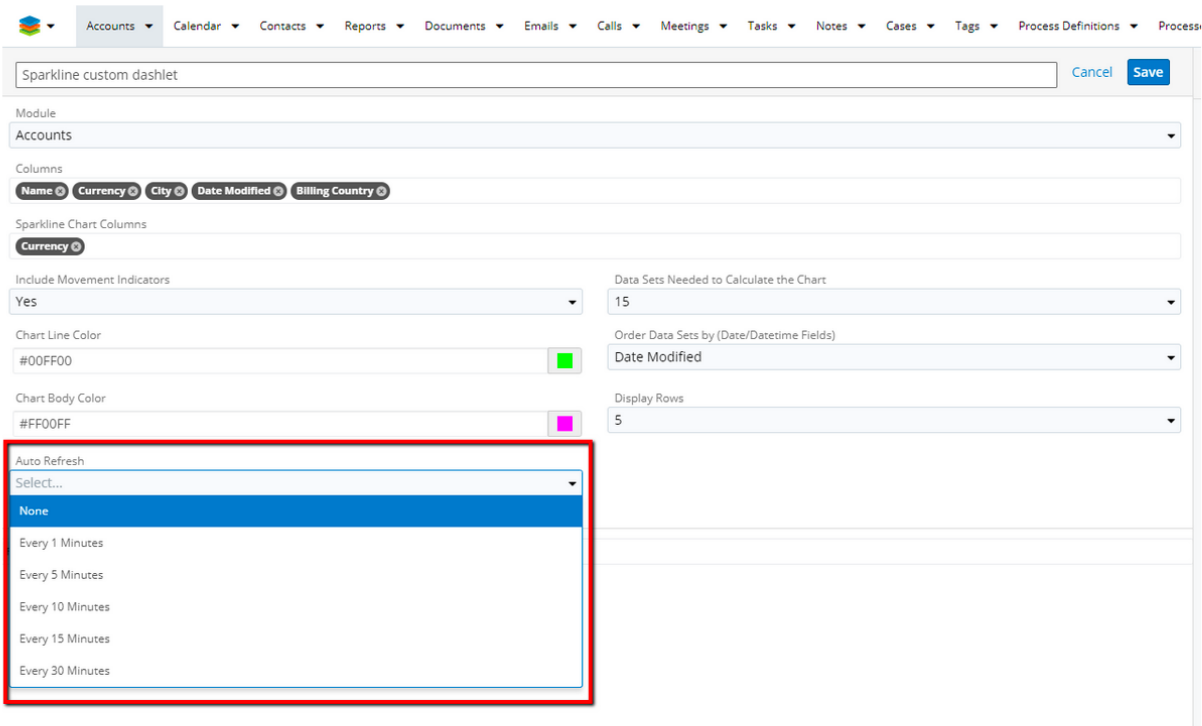
- A green upwards arrow next to the number.
- A red downwards red arrow next to the number.
  The arrow indicates trends for the numbers in the selected column.



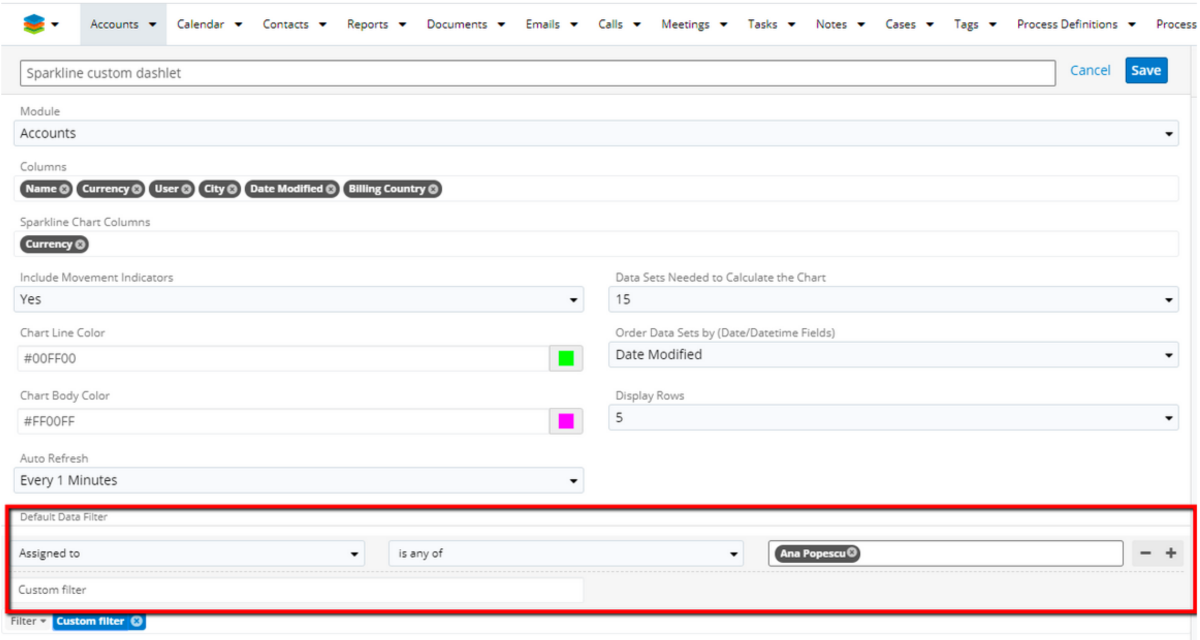
9. **Chart Line Color and Chart Line Shading Color**
   The **Chart Line Color** option allows users to set the color for the upper line of the chart. The **Chart Line Shading Color** option allows users to set the color for the filler of the chart.



10. **Auto Refresh**
    The Auto Refresh function allows users to set a time frame for the dashlet data to automatically refresh.



11. **Default Data Filter**
    This customization option allows users to select a pre-defined filter or to create a new one, exclusively used with the List View SparkLine Dashlet.



The charts will include data filtered by this option. In the example below, we have all the records assigned to a particular user.