NetSuite Viewer - User Guide

Salesforce Consultants

12 May, 2026

Table of Contents

1 Overview

This product enables real-time visibility of NetSuite data directly within Salesforce, eliminating the need to log into NetSuite. Designed for Salesforce admins and users, it provides a seamless experience for accessing key financial and operational data. Users can configure data columns, search specific records, and download invoice PDFs with ease. The solution streamlines workflows and improves efficiency by centralizing access to critical data. Its unique value lies in the real-time integration and user-driven customization, making it ideal for sales, finance, and operations teams who rely on up-to-date NetSuite insights within their Salesforce environment.

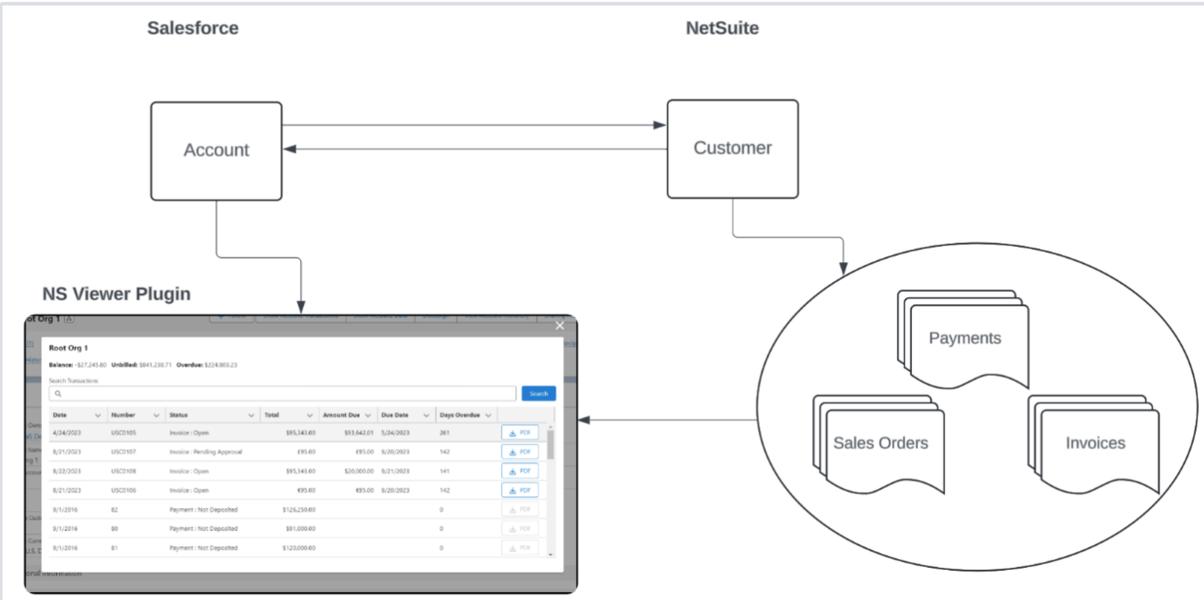
2 NetSuite Viewer - General Use

2.1 Features Overview

The NetSuite Viewer is a NetSuite and Salesforce app that enables organizations to showcase and engage with customer financial transactions sourced from NetSuite within Salesforce, encompassing Invoices, Payments, and Credits.

2.2 Architecture Overview

- Customers are linked in Salesforce by unique IDs matching NetSuite records.
- Financial data is created and updated in NetSuite.
- Data flows in real time from NetSuite to Salesforce, updating key fields like *Status* and *Amount*.
- Salesforce always shows the latest financial info linked to the right customer.



2.3 NetSuite Setup and Configuration

2.3.1 Pre-Installation Setup

Identify the user to install and run NetSuite Automation under - The following steps will need to be executed by a NetSuite Administrator. Additionally, it does require a full NetSuite user license in order to access the APIs. It could be the same user license used by other systems, integrations, etc...

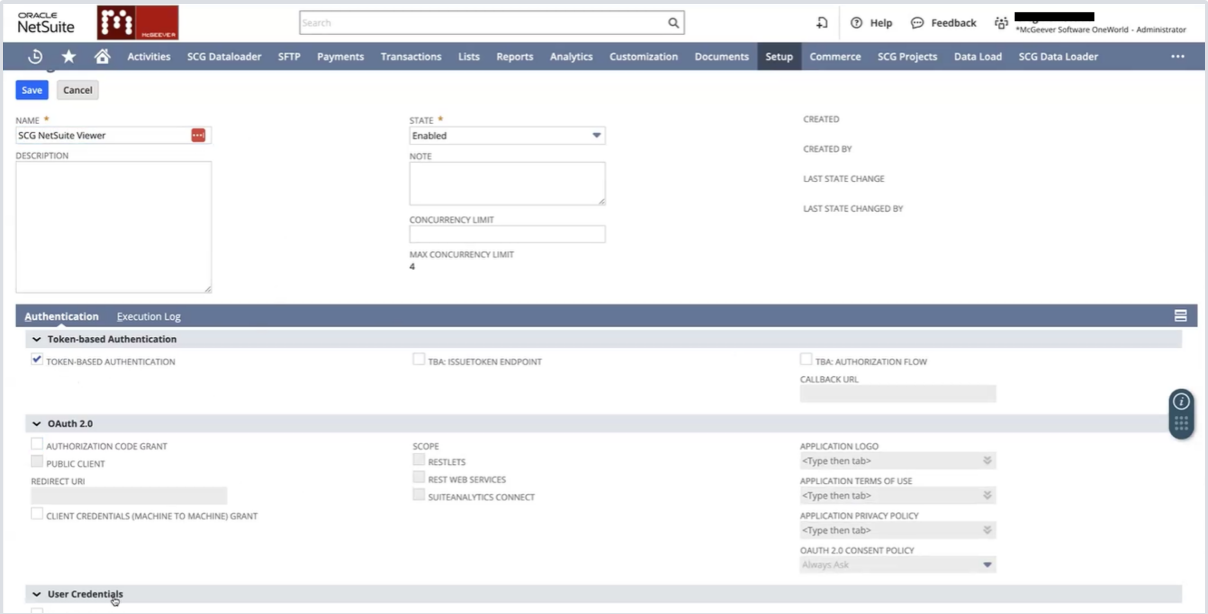
2.3.2 Post-Installation Setup

2.3.2.1 ***Create Integration Record in NetSuite***

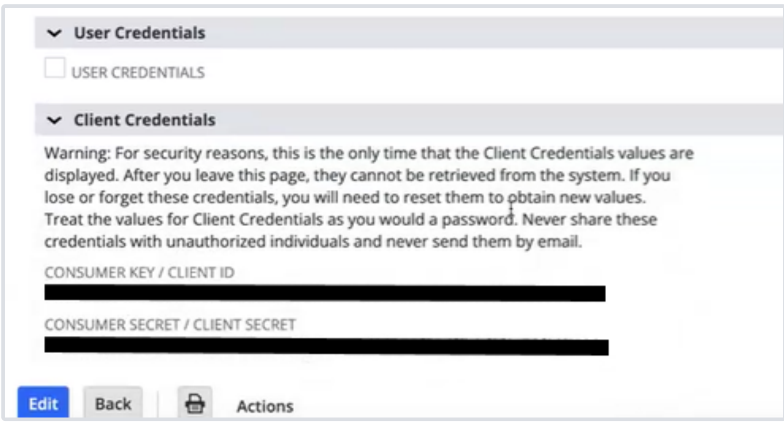
Create a new integration record (see screenshot below for an example)

- Navigate to Setup > Integration > Manage Integrations > New.
- On the Primary Information subtab:
  - Name: Enter a descriptive name. We recommend "SCG NetSuite Viewer".
  - Description: Add details about the integration's purpose (optional).
  - State: Select Enabled to activate it immediately (or Waiting for Approval for review).
- On the Authentication subtab:
  - For Token-Based Authentication:
    - Check Token-Based Authentication.
    - Uncheck TBA: Authorization Flow.
- For OAuth 2.0 (for modern REST integrations):
  - Uncheck Authorization Code Grant (unless interactive flows are needed).

Example of Integration Record:

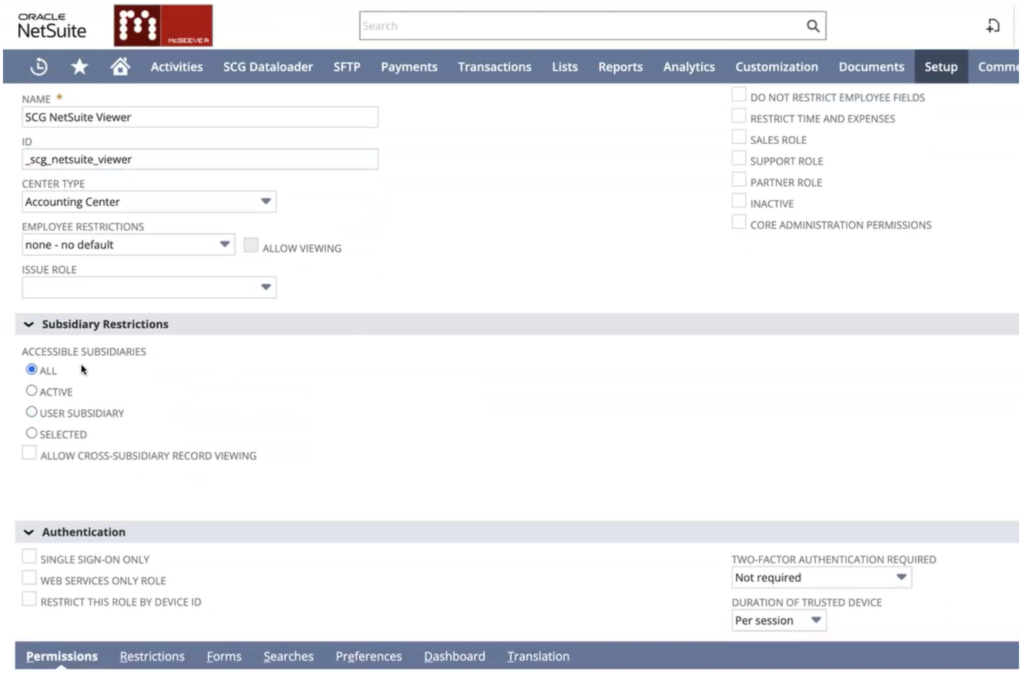


- Click Save. **(WARNING! After hitting save it is very important not to close this window as the Consumer Key and Secret are only displayed one time ever.)**
  - **Important:** Copy and securely store the Consumer Key and Consumer Secret (e.g., in a password manager). (These will be needed when setting up Salesforce, and cannot be found once you navigate away from this page.)

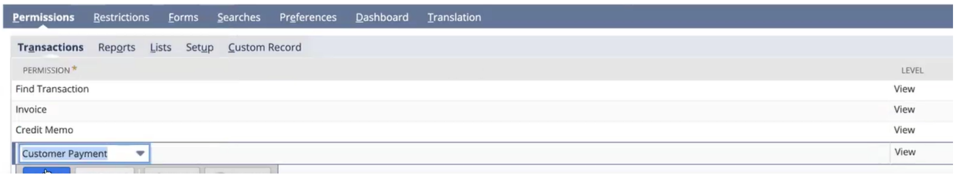


2.3.2.2 ***Create User Role***

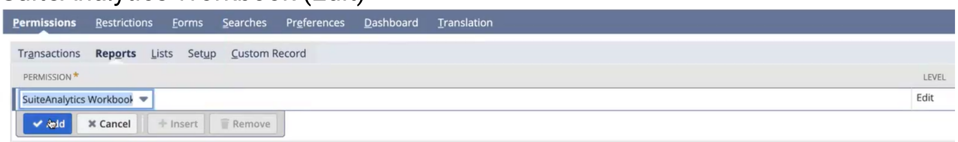
- Create a new user role
  - Navigate to Setup > Users/Roles > Manage Roles > New.
  - Enter a Name (e.g., "SCG NetSuite Viewer") and ID (e.g., "_scg_netsuite_viewer").
  - Under the Subsidiary Restrictions:
    - Select ‘ALL’



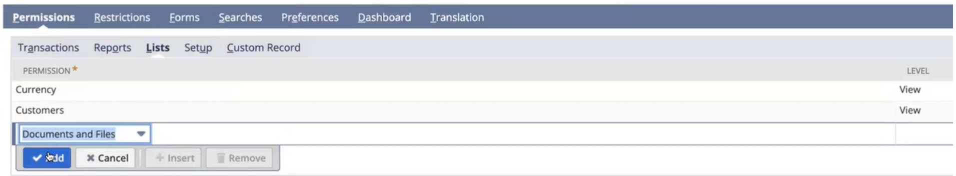
- Under the Permissions subtab:
  - Go to Transactions permissions and add the following permissions
    - Find Transaction (View)
    - Credit Memo (View)
    - Customer Payment (View)
    - Invoice (View)



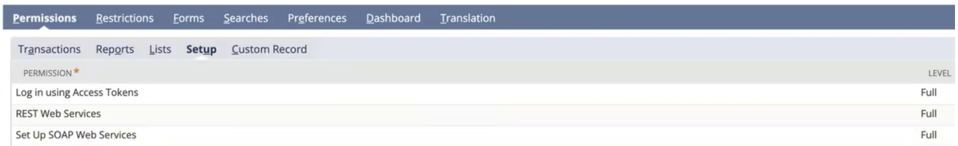
  - Go to Reports permissions and add the following:
    - SuiteAnalytics Workbook (Edit)



  - Go to Lists permissions and add the following:
    - Currency (View)
    - Customers (View)
    - Documents and Files (Full)

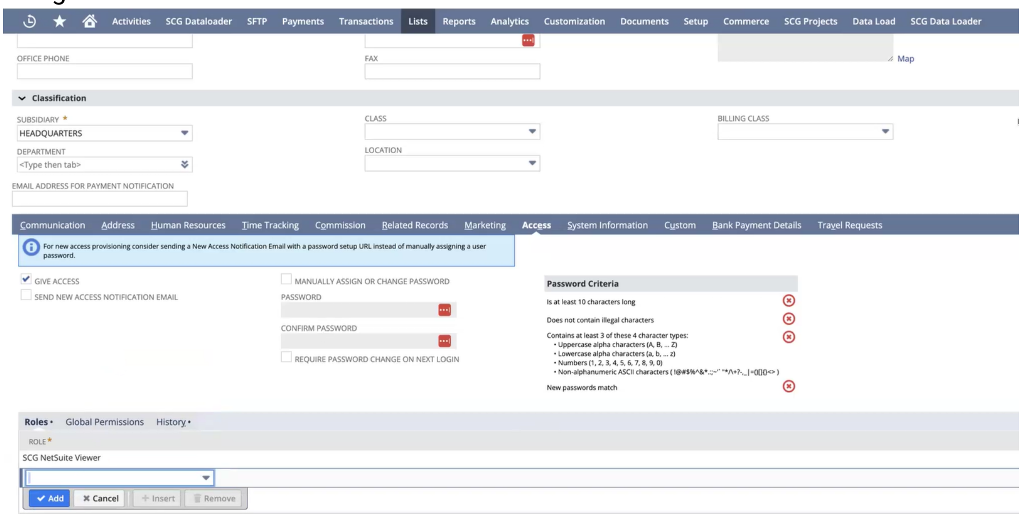


- Go to the Setup permissions and add the following:
  - Log in using Access Tokens (Full)
  - Rest Web Services (Full)
  - Set up SOAP Web Services (Full)



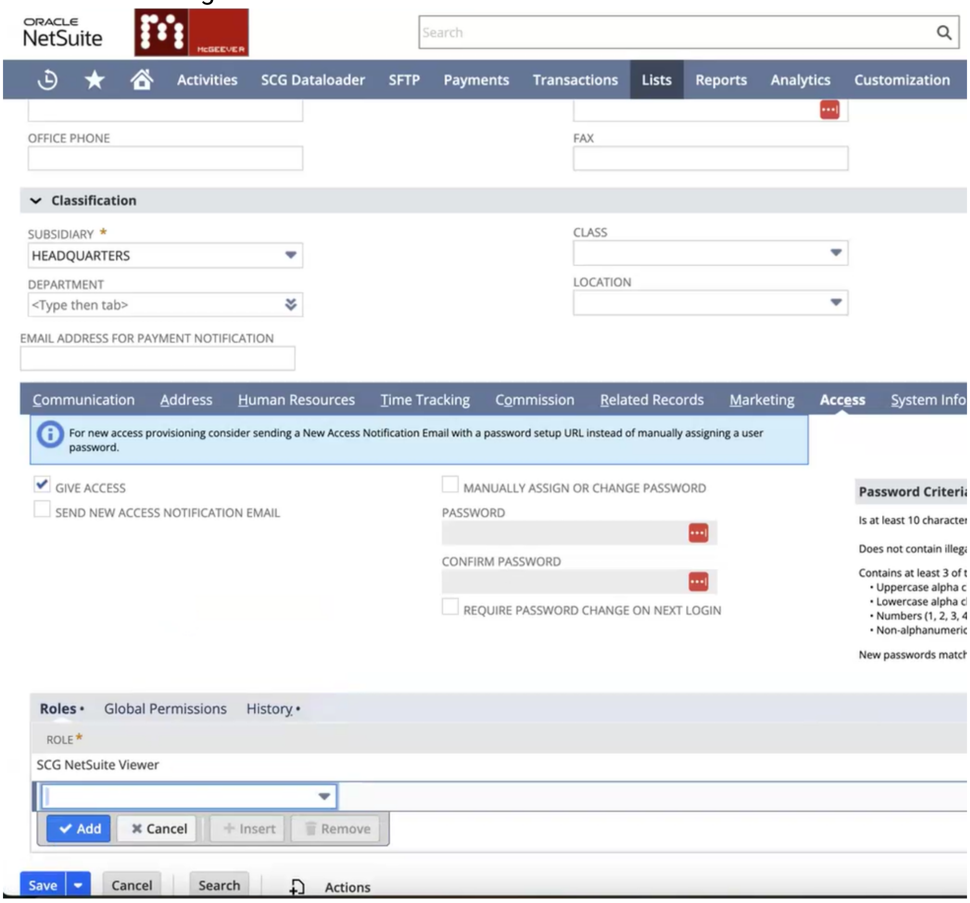
- Click Save.

- Add the role to the NetSuite employee record to be used for access
  - Navigate to the Employee Record:
    - Go to Lists > Employees > Employees.
    - Search for and select the user (employee) you want to assign the role to. If the user doesn't exist, create a new employee record by clicking New Employee.
  - Edit the Employee's Access Settings:
    - In the employee record, go to the Access subtab.
    - If the Give Access checkbox is not already checked, check it to enable role assignments for this user.



- Add the Custom Role:
  - In the Roles section of the Access subtab, click the dropdown under Role.
  - Select the custom role you created (e.g., "Integration User Role").
  - If the user needs access to specific subsidiaries (in NetSuite OneWorld), select the appropriate subsidiary in the Subsidiary column (if applicable).

- Click Add to assign the role.

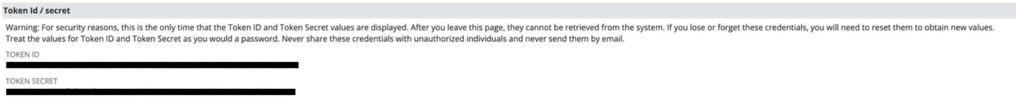


- Save the Changes:
  - Click Save on the employee record to apply the role assignment.

2.3.2.3 ***Create New Access Tokens***

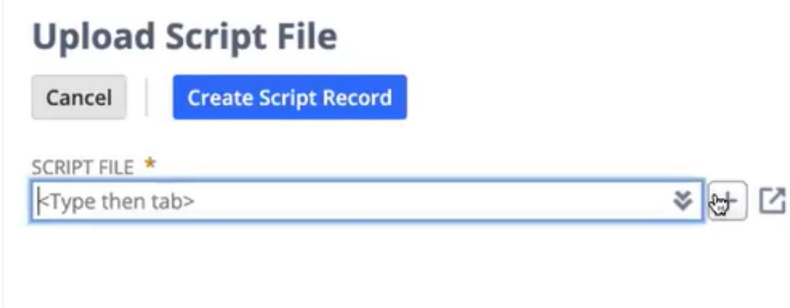
- Navigate to Access Tokens:
  - Go to Setup > Users/Roles > Access Tokens > New.
- Fill in Token Details:
  - Application Name: Select the Integration Record you created (e.g., "SCG NetSuite Viewer") from the dropdown.
  - User: Select the employee (user) assigned the custom integration role (e.g., the integration user).
  - Role: Select the custom role assigned to the user (e.g., "Integration User Role"). Only roles assigned to the selected user will appear.
  - Token Name: Enter a descriptive name (e.g., "MyApp Token 2025") to identify the token's purpose.
- Save the Token:

- Click Save.**(WARNING! After hitting save it is very important not to close this window as the Token ID and Token Secret are only displayed one time ever.)**
  - **Important:** Copy and securely store the Token ID and Token Secret (e.g., in a password manager). You will need these, along with the Consumer Key and Consumer Secret from the Integration Record, for API authentication.



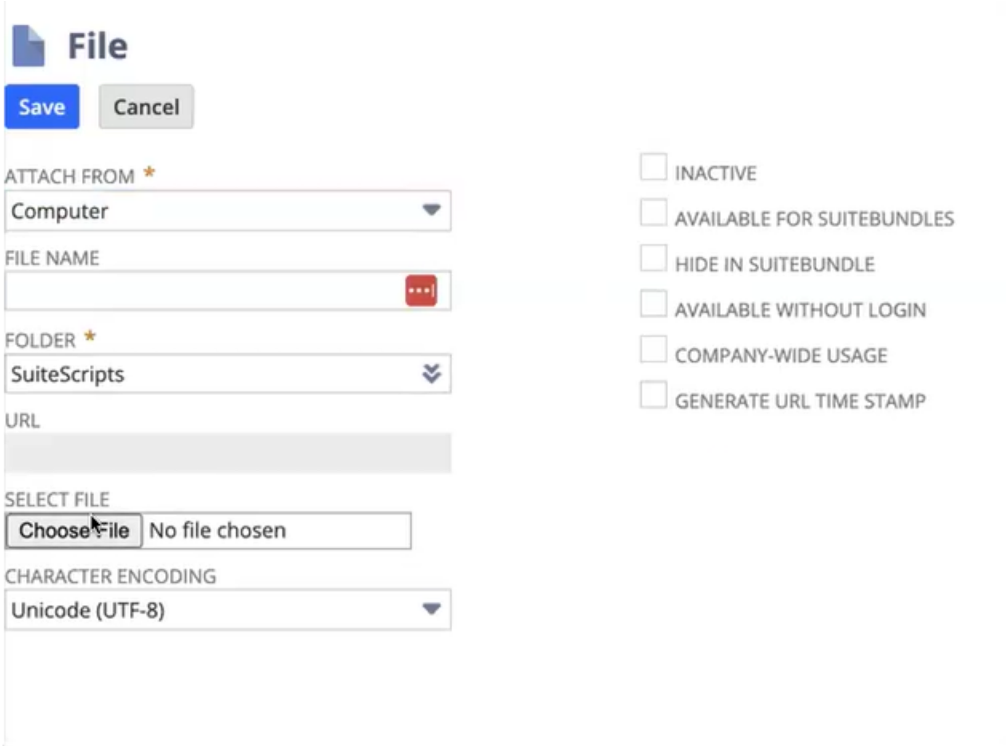
2.3.2.4 ***Create/Install Restlet Script***

- Download SCG RESTlet script from our website **here**[1]
  - Site Password: **SCG2025**
- Navigate to Script Creation:
  - Go to Customization > Scripting > Scripts > New.
  - Click the '+' under Script File

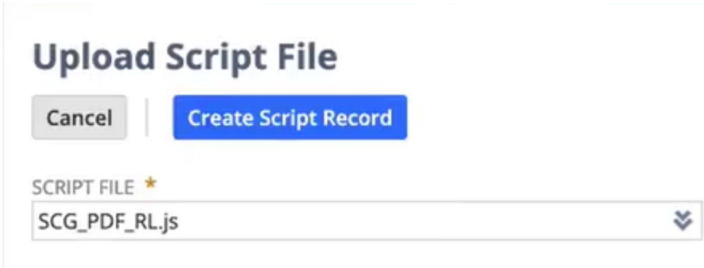


  - Choose the file that you just downloaded.

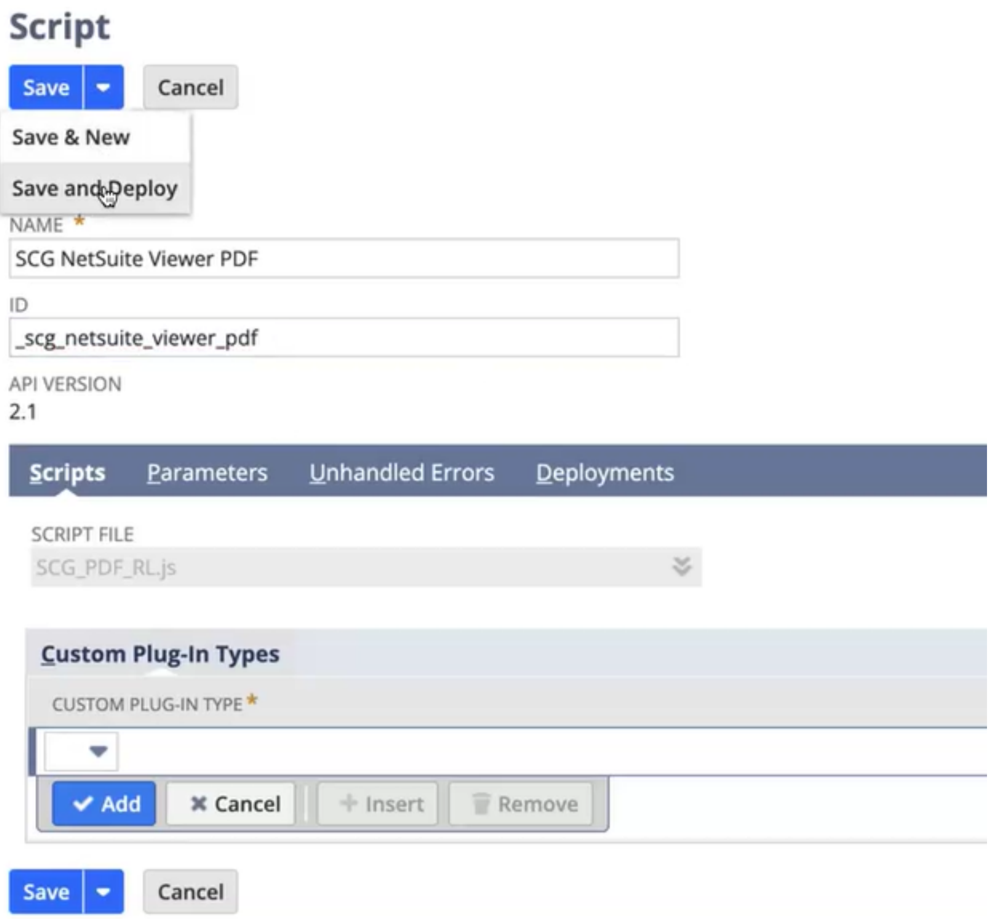
1. https://cdn.prod.website-files.com/68da50c8de0dc60b02cf6064/69cabda66da9ef3098112273_SCG_PDF_RL.zip



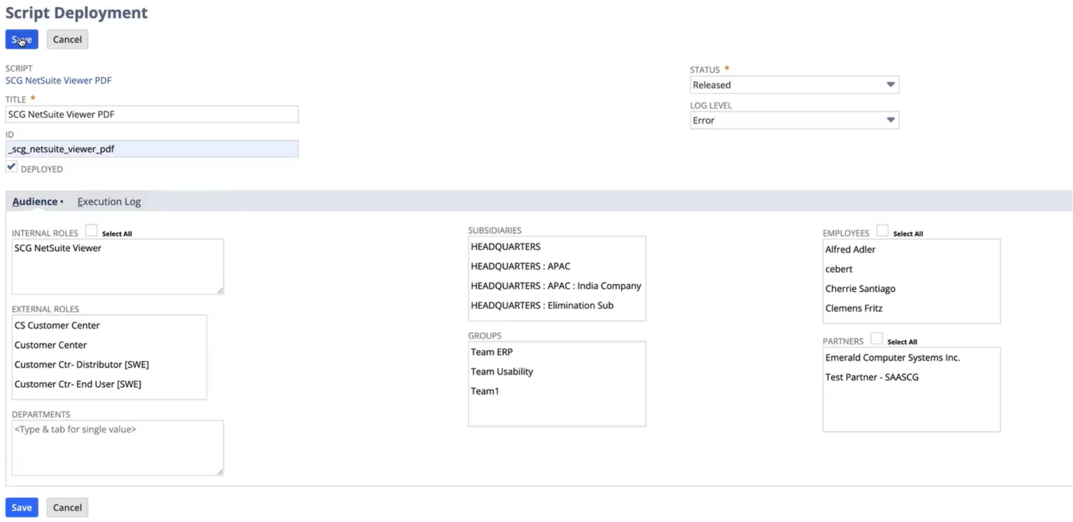
- Click Save.
- Now click on Create Script Record



- Enter a Name (e.g., "SCG NetSuite Viewer") and ID (e.g., "_scg_netsuite_viewer").
- Click Save and Deploy (From the drop down beside Save.)



- The next screen is Script Deployment, re-enter your ID
- Change the Status to 'Released'
- Add the role we created earlier under Audience and Internal Roles.



- Click Save.

- On the following screen, we are going to need some information to later use in the Salesforce setup.
  - Find the URL
    - Invoice Script Number - This is the number in the URL after script=
      - In the screenshot below the Invoice Script Number would be 1269
    - Invoice Deploy Number - This is the number in the URL after deploy=
      - In the screenshot below the Invoice Deploy Number would be 1

2.4 Salesforce Setup and Configuration

2.4.1 Pre-Installation Setup

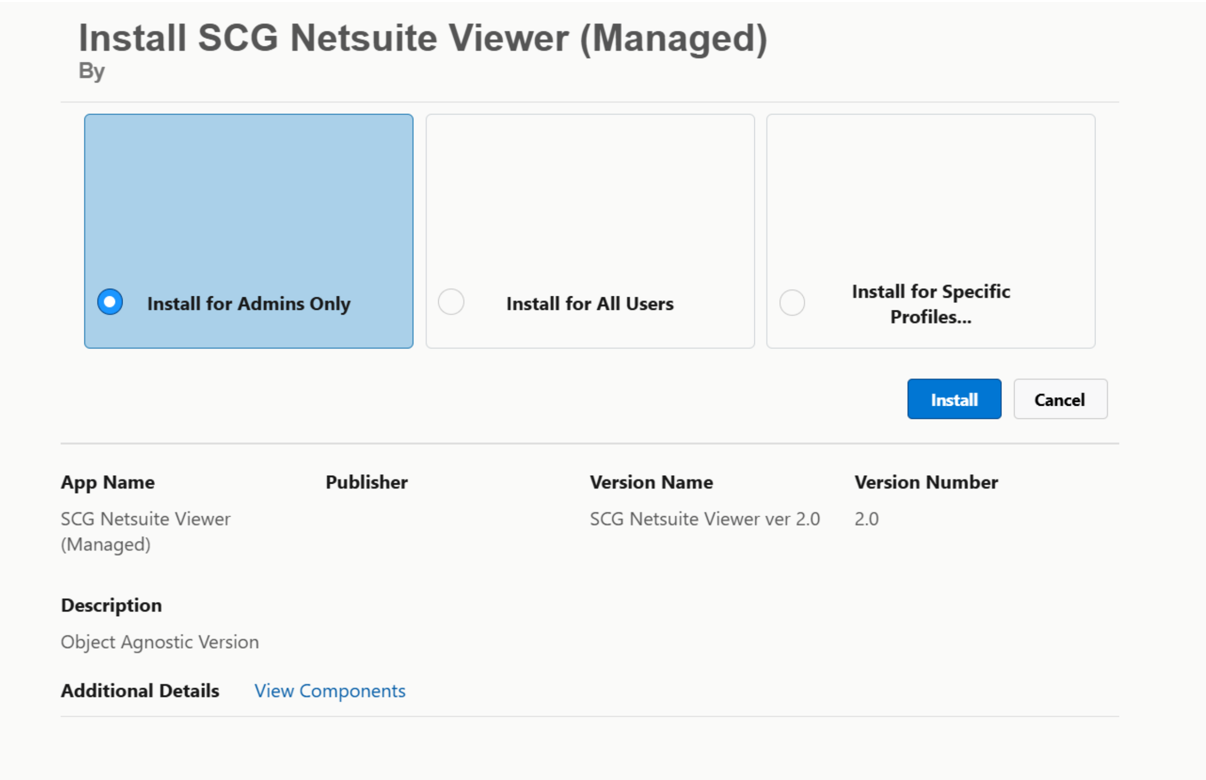
Identify the NetSuite user

2.4.2 Installation

The NetSuite Viewer Package currently consists of one installation package.

> **Note**: if you're installing into a sandbox, change the http://login.salesforce.com part of the url to http://test.salesforce.com . Alternatively, you can always change the first part of the url to your production or sandbox My Domain url.

When deploying each of the packages below, be sure to select **Install for Admins Only.** Permissions will set be set afterwards via permission set assignments.



1. NetSuite Viewer Package
   a. https://login.salesforce.com/packaging/installPackage.apexp?p0=04tVX000003RKYnYAO

2.4.3 Post-Installation Setup

2.4.3.1 Salesforce Configuration

The following steps will need to be executed by a Salesforce Administrator.

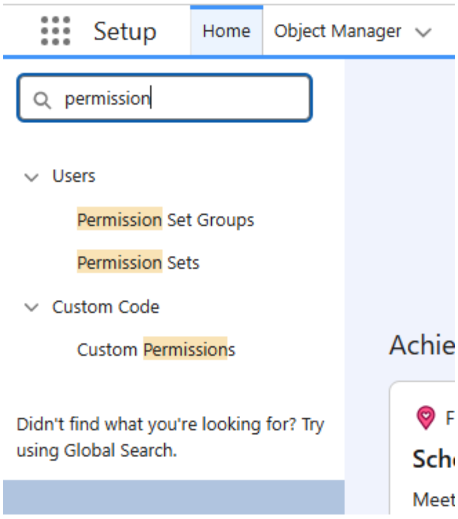
2.4.3.1.1 Assign Permissions Sets

The app has the following two permission sets that can be assigned:

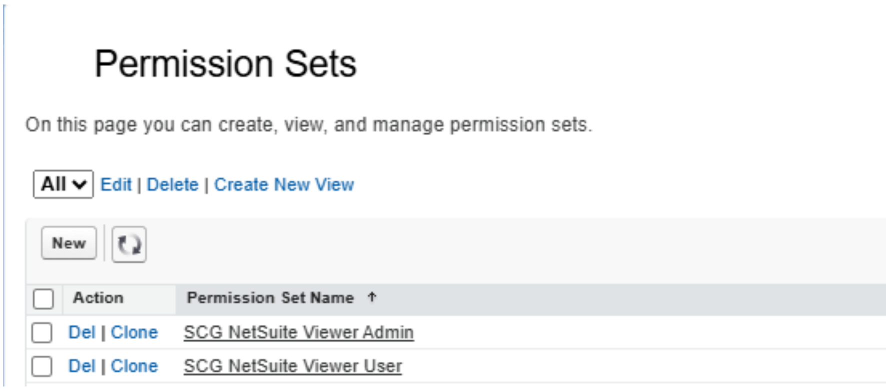
- **SCG NetSuite Viewer User -** Grants permissions to view and interact with NetSuite Viewer data components. This functionality is included as part of the NetSuite Viewer product.
- **SCG NetSuite Viewer Admin -** Grants permission to configure settings for NetSuite Viewer data components within your Salesforce environment.

You can assign permission sets via

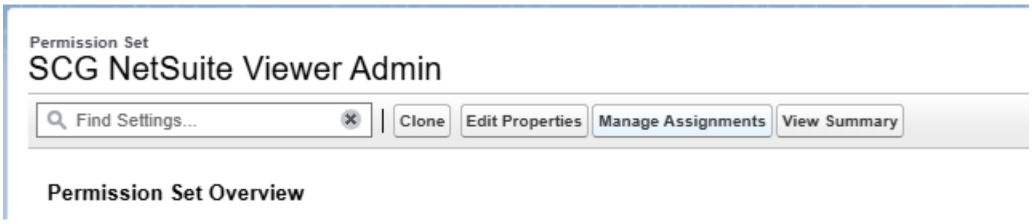
1. Navigate to **Setup** → **Permission Sets**.



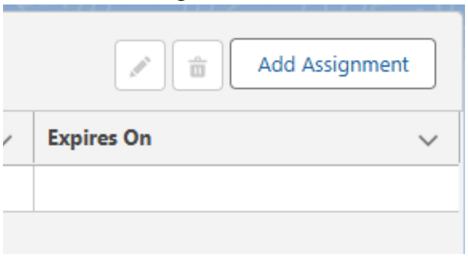
2. Select the desired **Permission Set**.

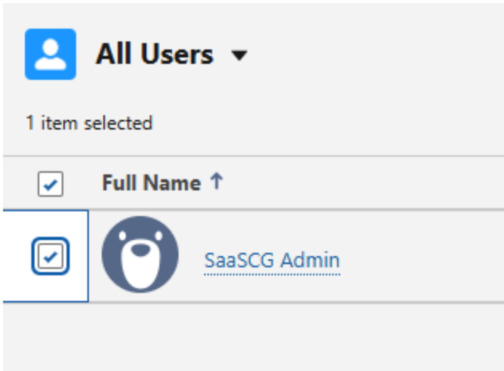


3. Click **Manage Assignments**.

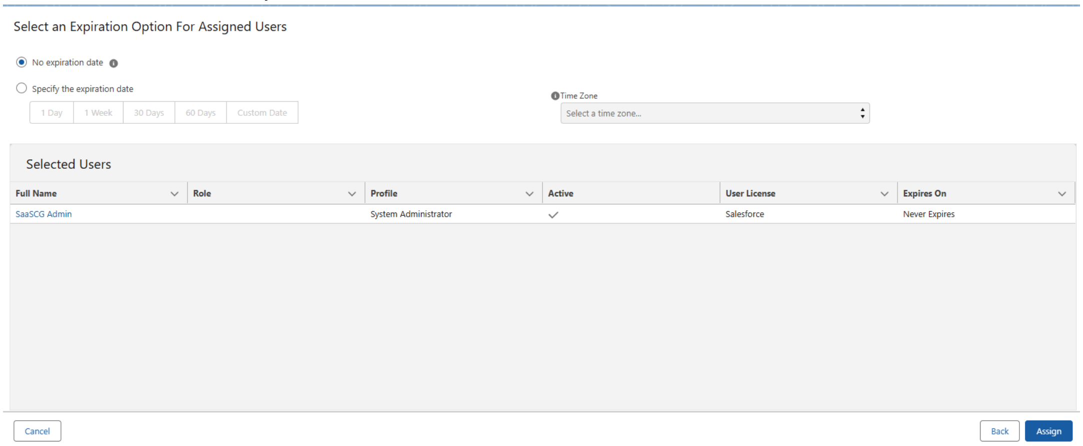


4. Click **Add Assignments** and select the users you wish to assign it to.





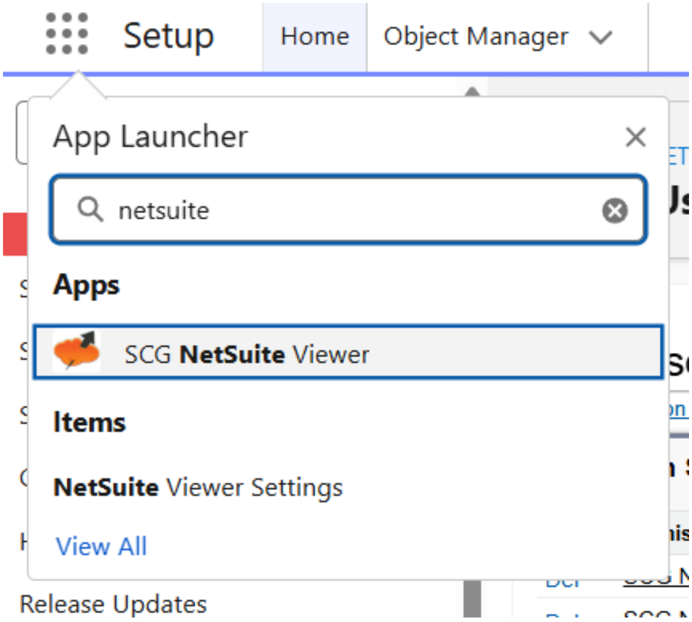
5. Click **Next**.
6. Choose an **Expiration Option** for the assigned users.
   - **Recommended:** *No Expiration Date*



7. Click **Save**.

2.4.3.2 Package Setup

Click on the app launcher and navigate to the **SCG NetSuite Viewer** app.


Setup
Home
Object Manager
App Launcher
netsuite
Apps
SCG NetSuite Viewer
Items
NetSuite Viewer Settings
View All
Release Updates

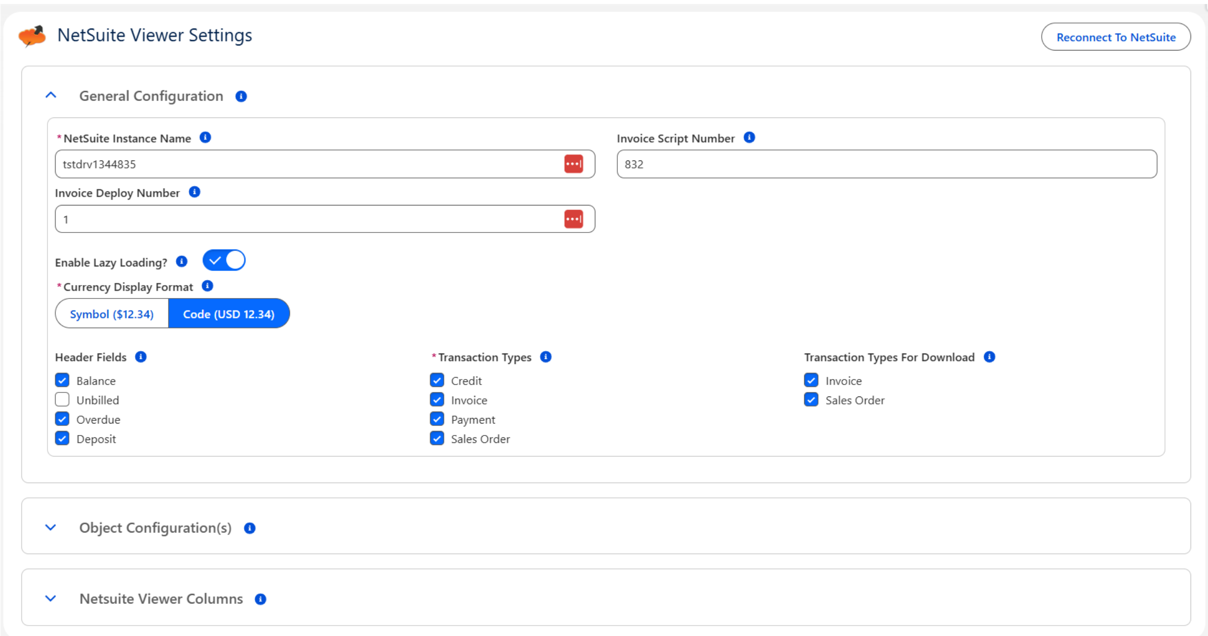
2.4.3.2.1 NetSuite Viewer Settings

This app contains a tab labeled **NetSuite Viewer Settings** which contains the UI to be able to configure the settings for the viewer. Populate the details based on the following

- NetSuite Instance Name - This is the Account ID found by going to Setup>Company>Company Information
- Invoice Script Number - This is the number in the URL after 'script='
  - In the screenshot below the Invoice Script Number would be 1269
- Invoice Deploy Number - This is the number in the URL after 'deploy='
  - In the screenshot below the Invoice Deploy Number would be 1

1 This is from the RESTlet script deployment that was done in NetSuite.

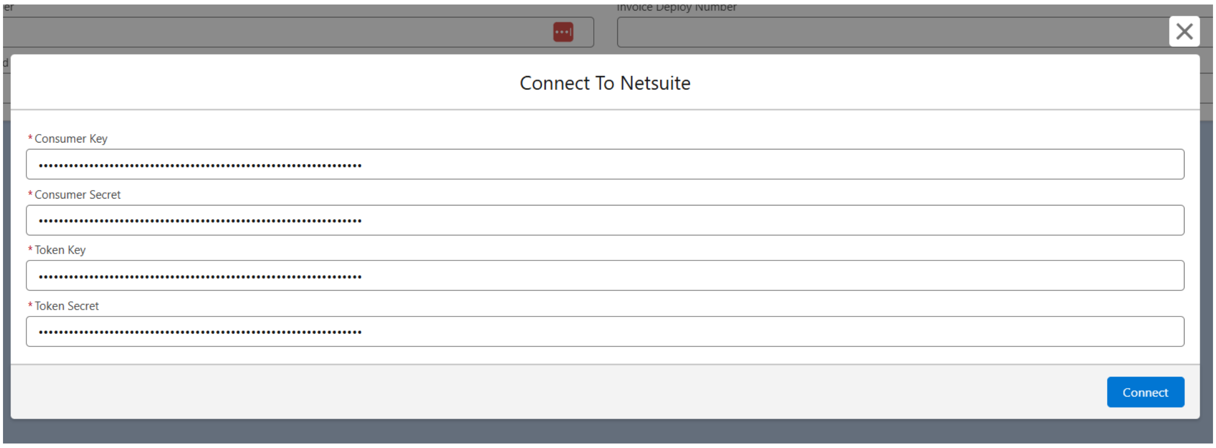
- Enable Lazy Loading - This toggle allows the user to enable lazy loading for the viewer table. If unchecked, the results are paginated.
- Currency Display Format - The format of how currency values are displayed in the viewer.
- Header Fields - Select the customer level fields that will appear as tiles above the viewer table.
- Transaction Types - Select the transactions types that will appear as rows in the viewer table.
- Transaction Types for Download - Select the transaction types that will have the PDF download button enabled in their rows.



After the details are entered, click **Save**.

2.4.3.2.2 Connect to NetSuite

To set the NetSuite credentials, click on **Connect to NetSuite**. A modal will appear where you can enter or copy/paste the Consumer Key, Consumer Secret, Token Key, and Token Secret from your NetSuite environment. (These were available after setting up your Integration Record, and Access Tokens. Create Integration Record and Create Access Tokens)



Click **Connect** to save the credentials.

After the credentials are set up, you will need to configure the remote site settings by going to **Setup > Remote Site Settings**. You will need to create two custom settings: One for the suitetalk api, and one for the restlet. Name the settings something memorable in case you need to update them later. These can be found on the Company Information page in NetSuite under Company URLs.

- Setup>Company>Company Information

For the urls, follow the following formats:

- SuiteTalk:

```
https://{ns_instance_domain}.suitetalk.api.NetSuite.com
```

- Restlet:

```
https://{ns_instance_domain}.restlets.api.NetSuite.com
```

Make sure that the settings are active, and **Save**.

2.4.3.2.3 Reconnect to NetSuite

If the values are already populated, the button displayed will read **Reconnect to NetSuite**.

You may adjust any of these fields, and then you will be directed to the same modal from above, where you may enter or copy/paste the Consumer Key, Consumer Secret, Token Key, and Token Secret from your NetSuite environment.

Click **Connect** to save the credentials and proceed with the steps specified above.

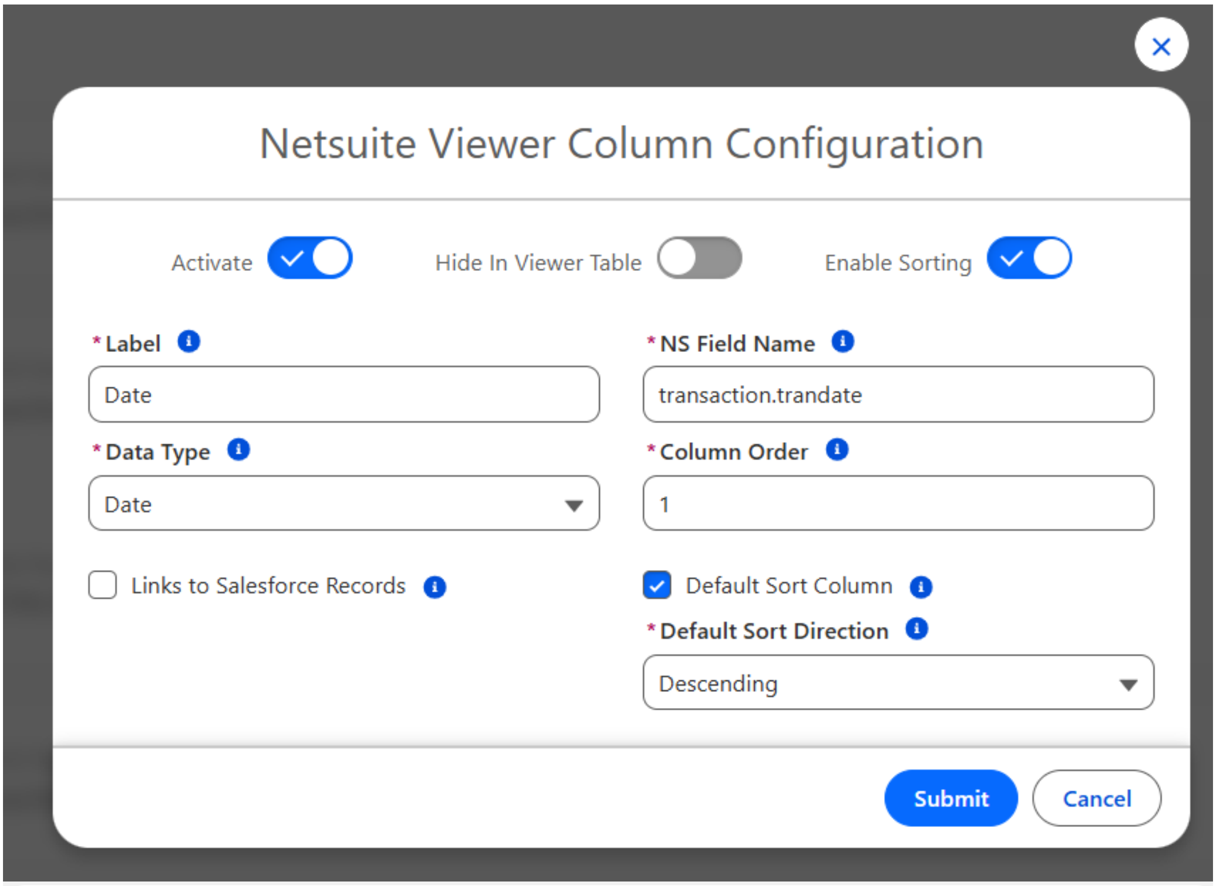
2.4.3.2.4 Table Configuration

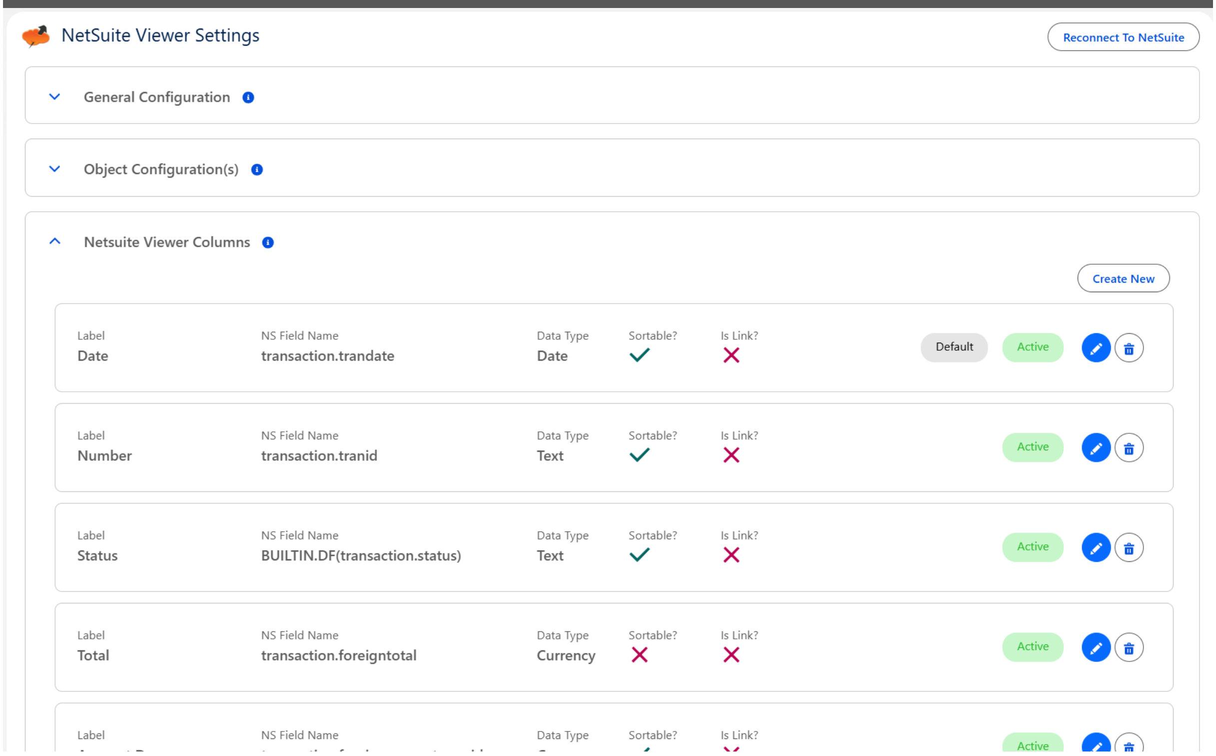
To configure columns for the table, expand the Column Configuration section at the bottom of the settings screen.

To create a new column, click the **Create New** button.

Enter the details for the new column:

- Activate - The column must be active in order to appear in the viewer table or as a filter.
- Hide In Viewer Table - When checked, the column will not appear in the viewer table. If the column is active, it will still be available for use as a filter.
- Enable Sorting - When checked, enables sorting capability for the column.
- Label - The column’s label.
- NS Field Name - The valid transaction field name from Netsuite.
- Data Type - The data type that will be returned for the column. Controls how the data is formatted within the viewer table.
- Column Order - Controls the order in which columns will appear in the viewer table from left to right.
- Links to Salesforce Records - When checked, denotes the column as having a valid Salesforce id. An sobject type as well as its name field must be specified in order to successfully link to the records within Salesforce.
- Default Sort Column - When checked, tells the viewer how to sort results initally. A sort direction must be specified, and there can only ever be 1 default sort column in the system at a time.





To find the NS Field Name, you will need to get with your NS Admin and do the following in NetSuite:

- Go to Setup>Records Catalog>go to 'T' and find Transactions>SuiteScript and REST Query API

- Once loaded, this will give you all the available fields.
  - For Picklists and Lookups you will need to use the BUILTIN.DF(NSfieldname)
  - For most other field types, you should be able to use transaction.NSfieldname
  - If you want to use other field functions, please use the appropriate one from the NetSuite documentation: SuiteQL Supported and Unsupported Functions[2]

As an example, the standard columns below should be available in the transaction table for every NetSuite instance and can serve as a starting point.

Label	NS Field Name	Data Type	Column Order
Date	transaction.trandate	Date	1
Number	transaction.tranid	Text	2
Status	BUILTIN.DF(transaction.status)	Text	3
Total	transaction.foreigntotal	Currency	4
Amount Due	transaction.foreignamountunpaid	Currency	5
Due Date	transaction.duedate	Date	6
Days Overdue	transaction.daysoverduesearch	Number	7

2.4.3.2.5 Page Layouts

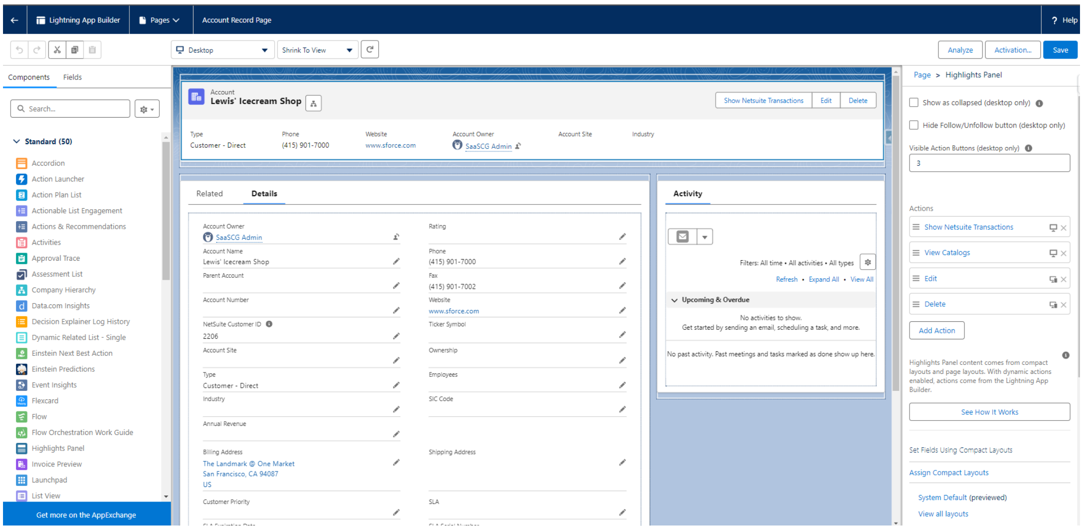
Finally, navigate to Object Manager in Salesforce Setup to ensure all components are properly displayed on the Account record page.

- Locate the **Lightning Record Page** and verify that the **'Show NetSuite Transactions'** quick action is included.
- Locate the **page layout** and confirm that the **'NetSuite Customer ID'** field has been added.

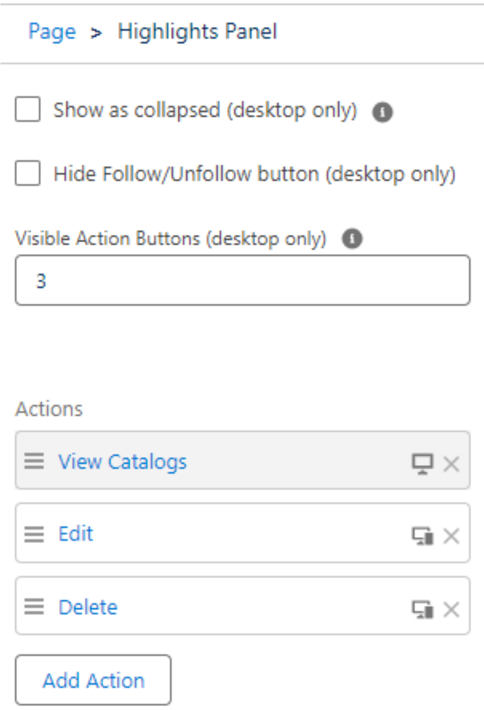
*If your organization uses **Dynamic Forms**, you can add the field and the action directly to the Lightning Record Page instead of the standard page layout.*

- Here's how to add the action to the Lightning Record page layout using Dynamic Forms.
  - From the Account record page, click the gear icon at the top right, and select edit page.
  - Click on the highlights panel at the top of the page layout.

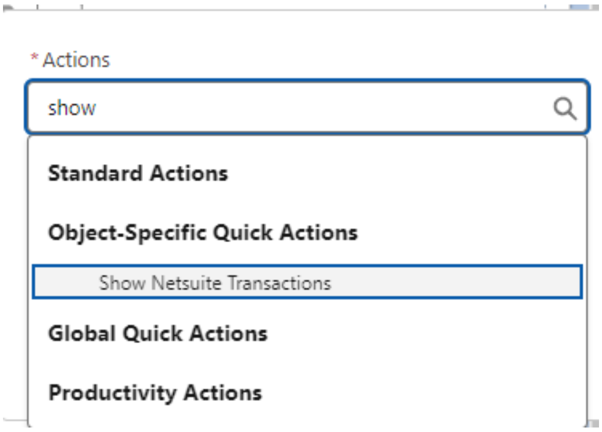
2. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/section_158513731864.html#subsect_156330166816



- On the right side, click on the ‘Add Action’ button.



- In the new window start typing ‘Show’, then select ‘Show Netsuite Transactions’, then click done.



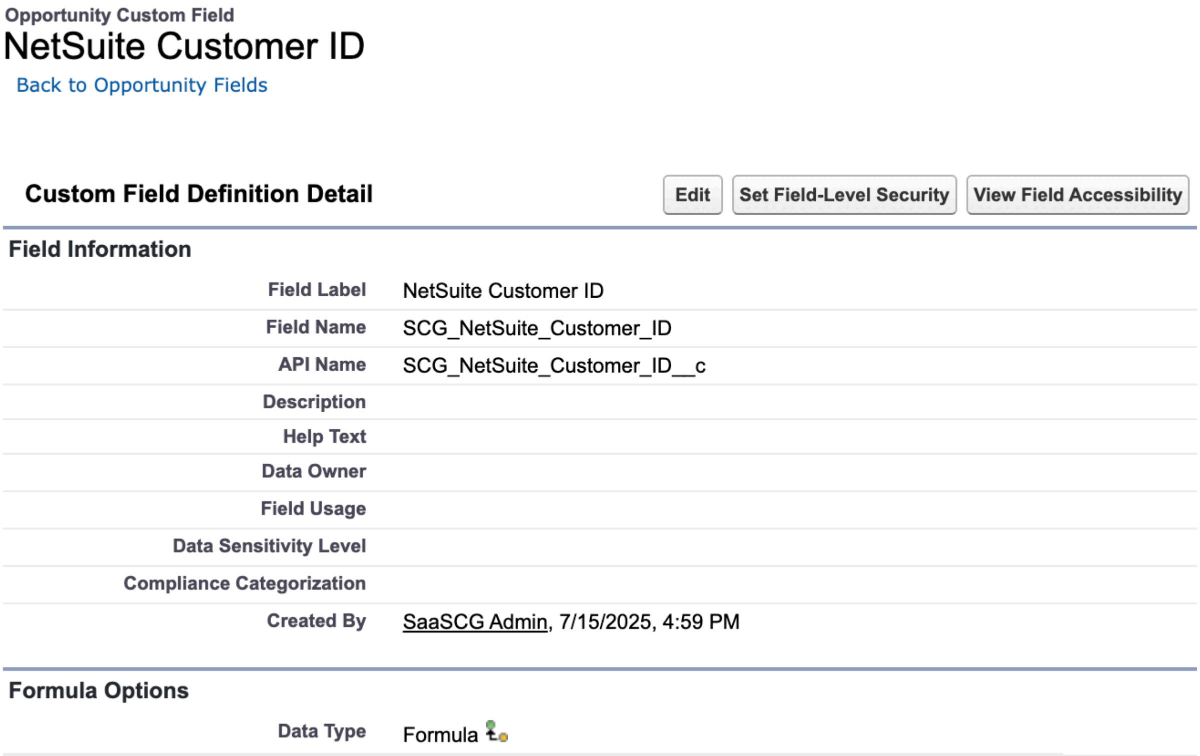
- Click Save at the top right, then use the back arrow to return to the Account record page.
  - *If your organization has enabled Dynamic Layouts, you can control visibility to the action by creating a Custom Permissions and assign it to the correct permission set.*

2.4.3.3 Connecting to Other Objects

The NetSuite Viewer plugin also allows you to connect to other Salesforce objects via a customer Id field.

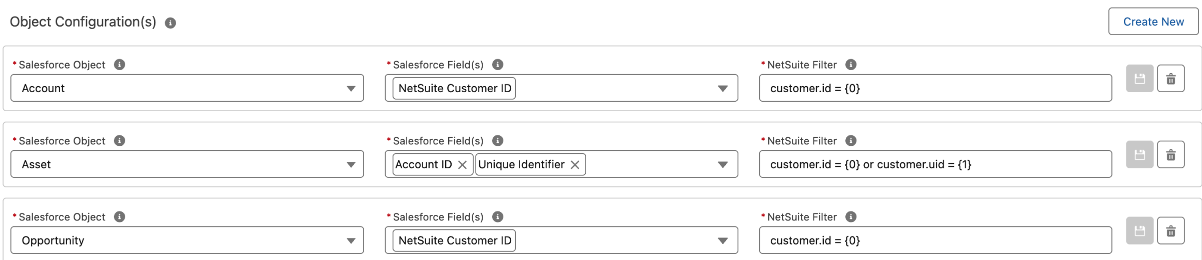
On the **NetSuite Viewer Settings** tab in the app, you will see the Object Configuration(s) section displayed at the bottom of the page. Follow these steps to set up a new object connection:

1. In Object Manager under Setup, you will need to add a **NetSuite Customer Id** field onto the object where you want to setup the NetSuite connection. This field should be a Formula (Text) type that references the NetSuite Id field on the related account.

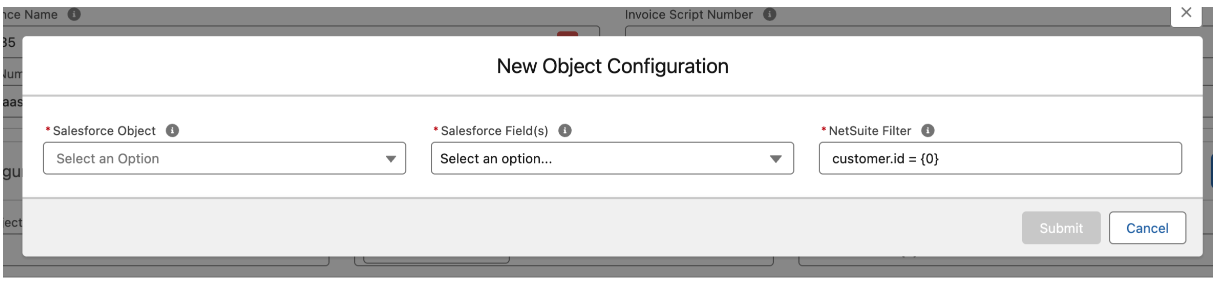


Account.SCG_NetSuite_Customer_ID__c

2. Navigate back to the **NetSuite Viewer Settings** tab in the SCG NetSuite Viewer app and scroll down to the Object Configuration(s) section:



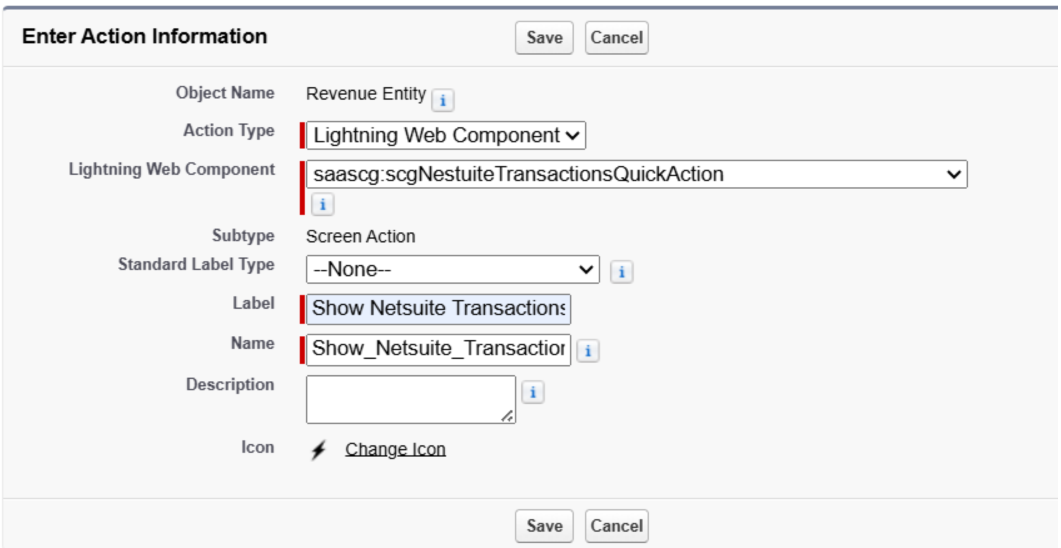
3. Click the **Create New** in the upper right corner of the component.



2. Select whichever object you'd like to connect in the far left **Salesforce Object** field (listed alphabetically).
3. Select the **Salesforce Field** whose value is used to find the correct NetSuite record. Optionally, you can add more than one field here by holding the *Shift* key by clicking multiple fields.

4. Finally, update the **NetSuite Filter** field so it matches the filter for the **Salesforce Field(s)** within the NetSuite query's WHERE clause. Remember that all fields selected on the **Salesforce Field** section must be included.
5. Click **Submit**.
6. Once created, the object relationships can be updated and saved via the Object Configurations user interface directly on the page. The far right icons show the *Save* and *Delete* actions.

The "Show Netsuite Transactions" quick action is only included for Account as part of the package. For any other objects, the quick action will need to be manually created. Go to Setup > Object Manager > {Object} > Button, Links, and Actions. Create a new action with the following configuration:



2.4.3.4 Data Staging

The final step in the setup process is to ensure your data is configured correctly.

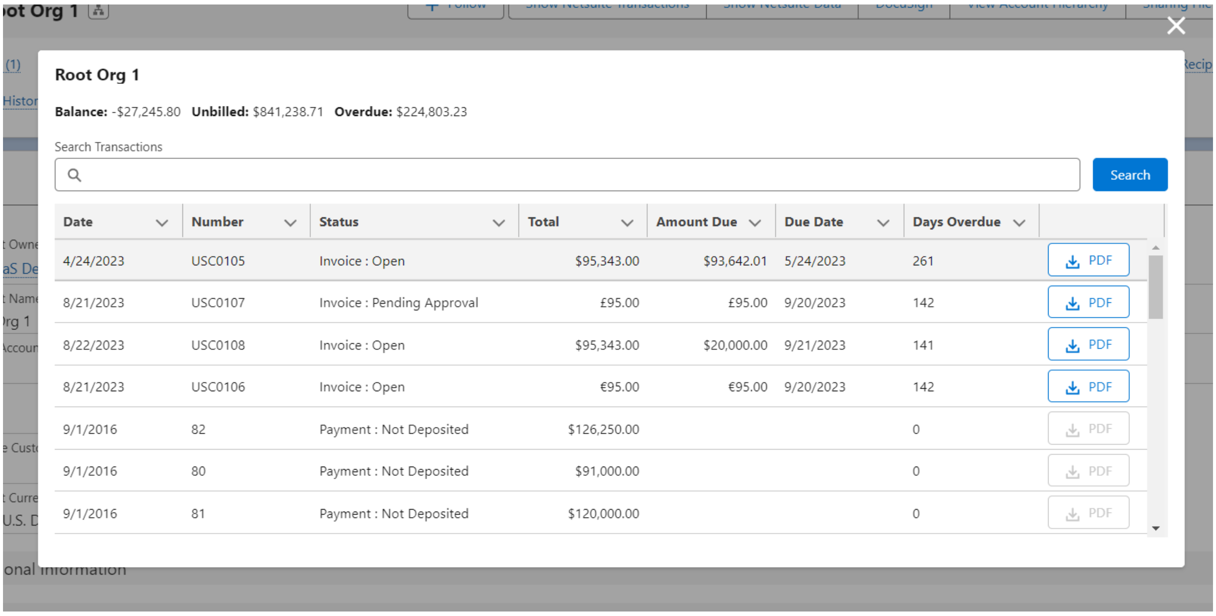
To integrate your Salesforce Account records with your NetSuite Customer records, verify that the data is properly configured within Salesforce. By default, the integration uses the **NetSuite Customer ID (SCG_NetSuite_Customer_ID__c)** field on the Account record. If a different unique field is already being used, you can update the **NetSuite Customer ID Field** setting within the **NetSuite Viewer Settings** section with the API field name found on the Account that contains the NetSuite Customer Internal Id.

Once the correct field is set, use a data management tool (e.g., Data Loader, Data Import Wizard) to populate the field with the corresponding NetSuite Customer ID. This ID is typically a 6- to 9-digit number, which can be found in your NetSuite customer list.

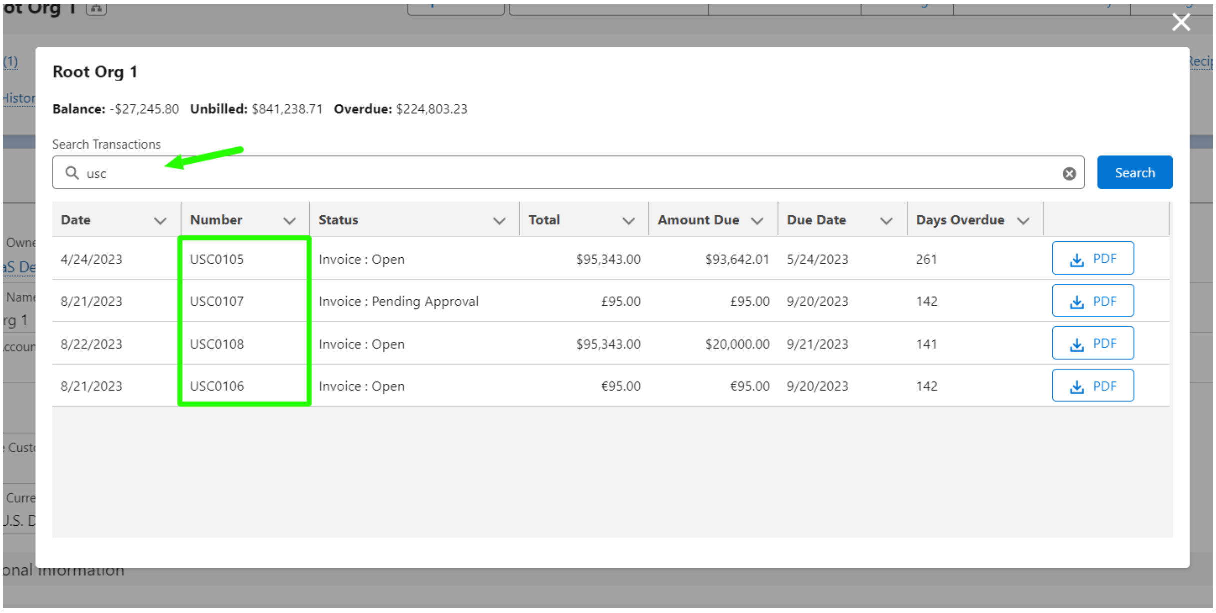
2.5 General Use

Now that the components are configured, you can now navigate to an account record and view the NetSuite data for that account. By clicking on **Show NetSuite Transactions**, we can see the transactions for the account in NetSuite.

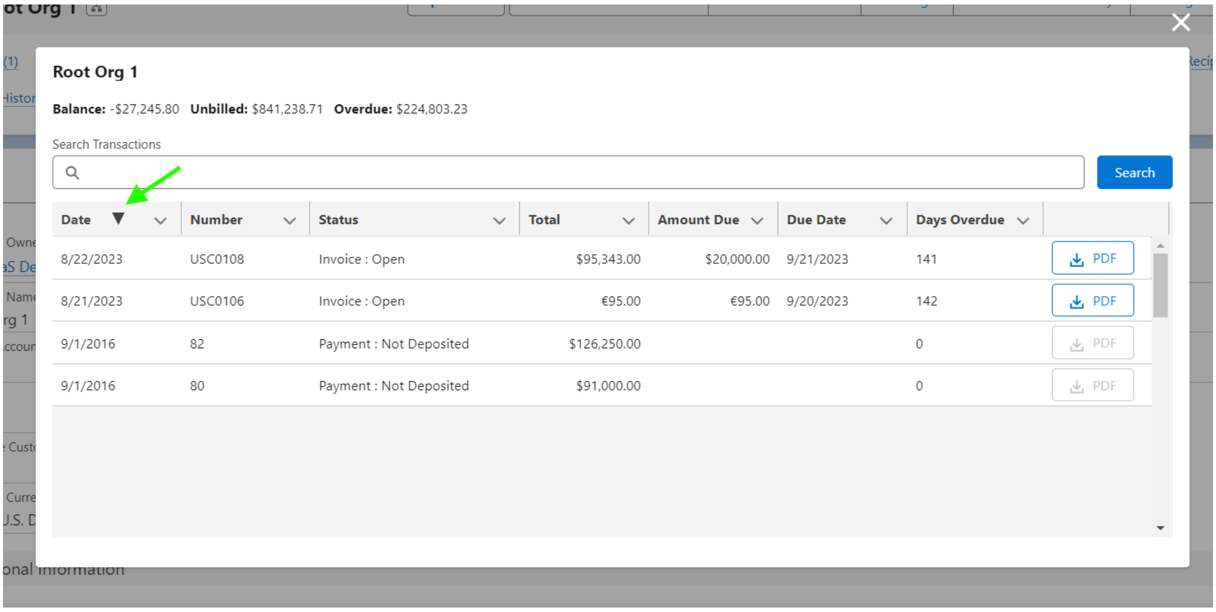
FYI, currencies are displayed as the NetSuite customer's currency.



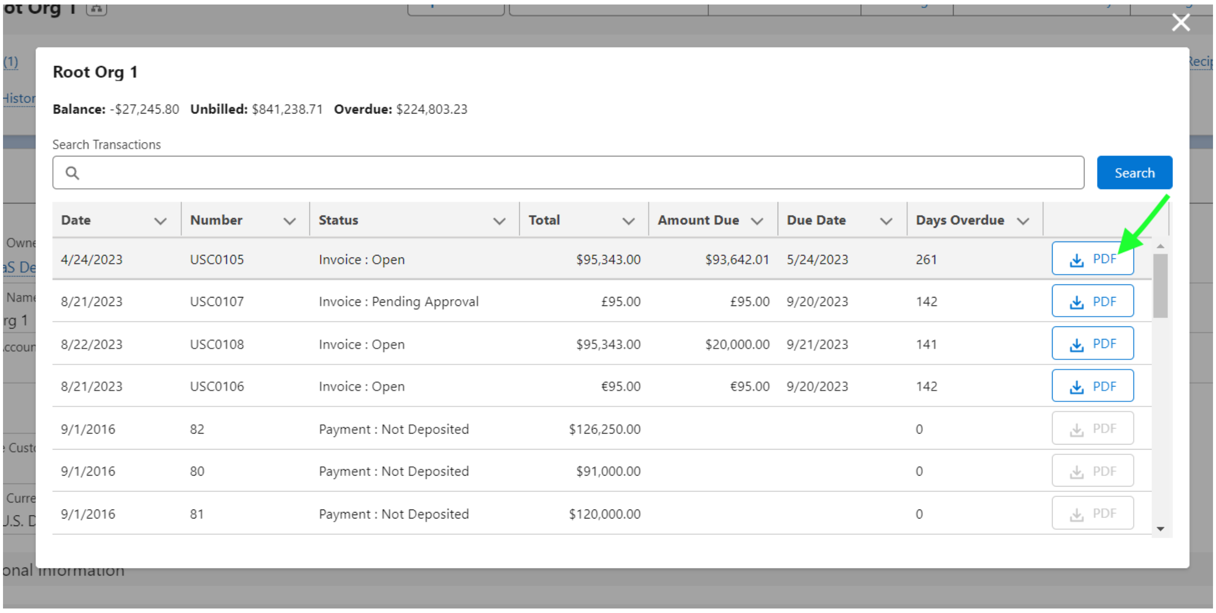
To search for a specific row, type what you are looking for in the search bar and click search to filter the results.

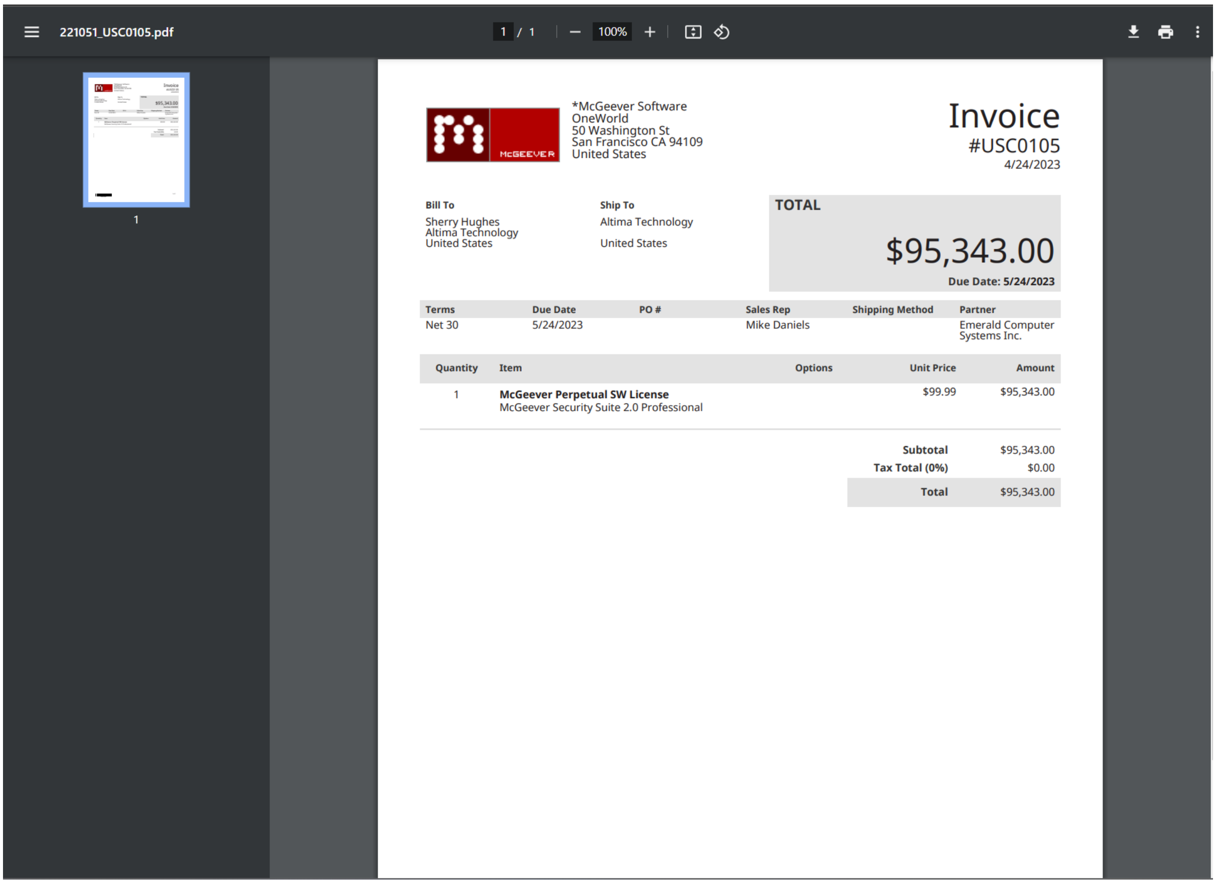


To sort by a specific column, hover over the column header and click the sort arrow.



To download a PDF for a row, simply click the button on the far right. The pdf file will be downloaded to the system, and can then be opened directly in the browser.





2 The PDF template is part of NetSuite's **Advanced PDF Templates**[3] feature and is determined by the form assigned to the invoice. Different forms can reference different templates, all of which can be customized. For more information, please contact your NetSuite administrator.

To add additional filters to the results, click on the blue filter tab in the top right of the window.

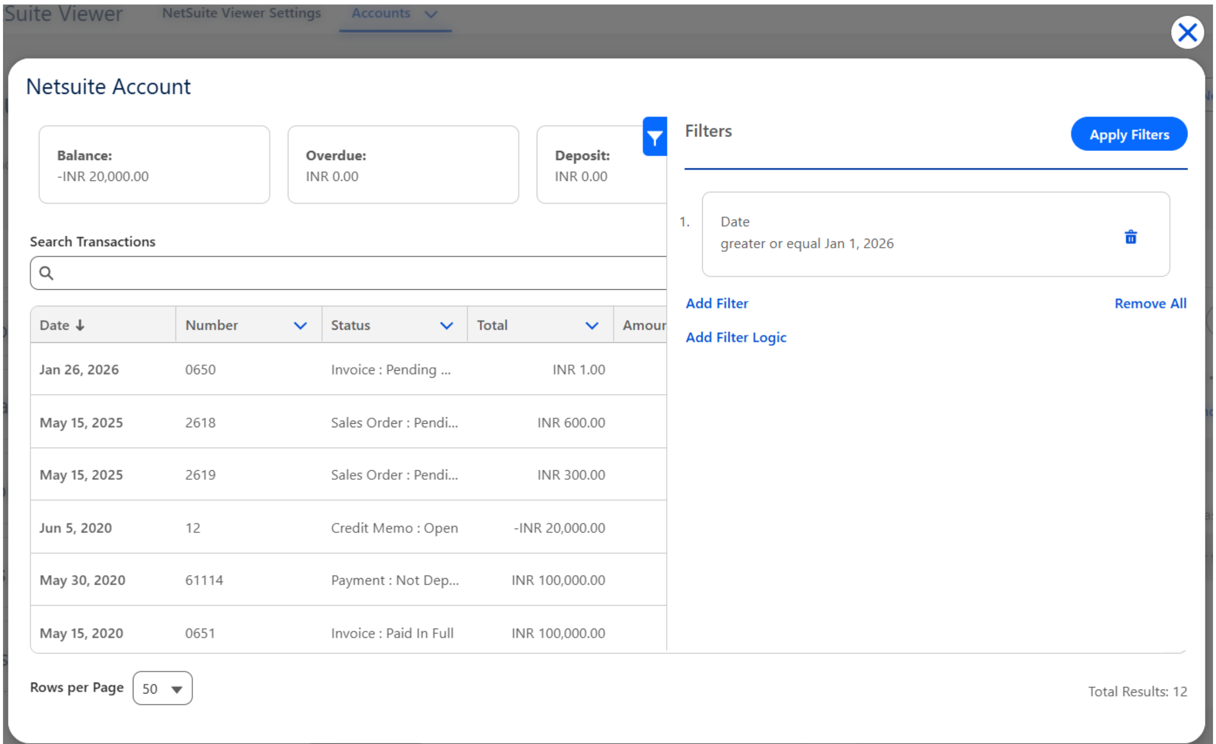
Click **Add Filter**.

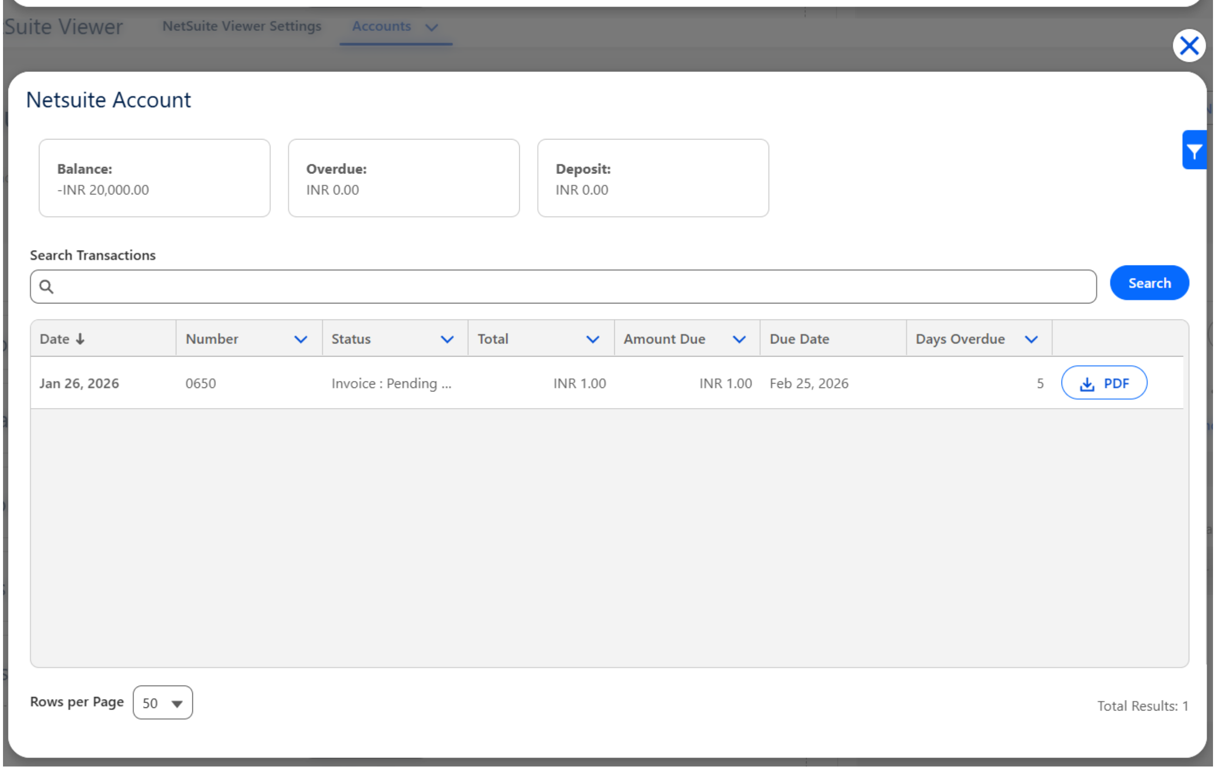
Select the field, operator, and value, then click **Save**.

Once all your filters are defined, click **Apply Filters** to refresh the table with the new criteria.

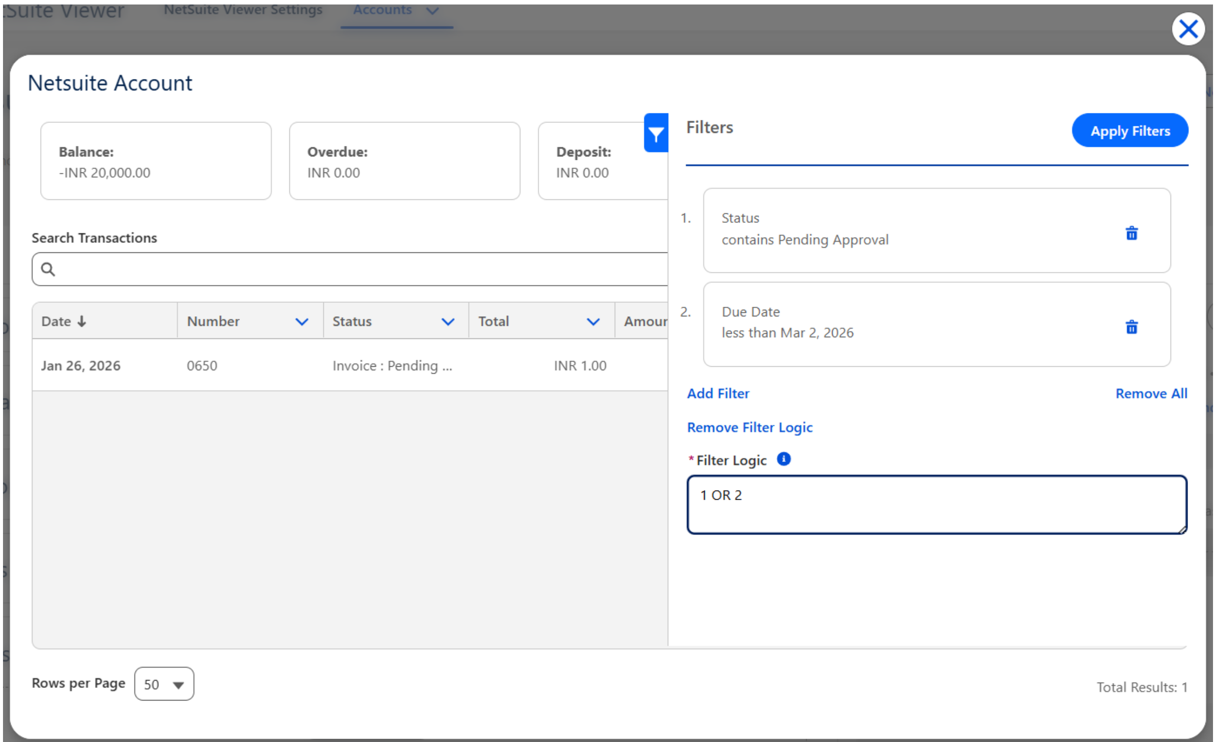
In this example, we are going to only retrieve transactions that have a date for 2026.

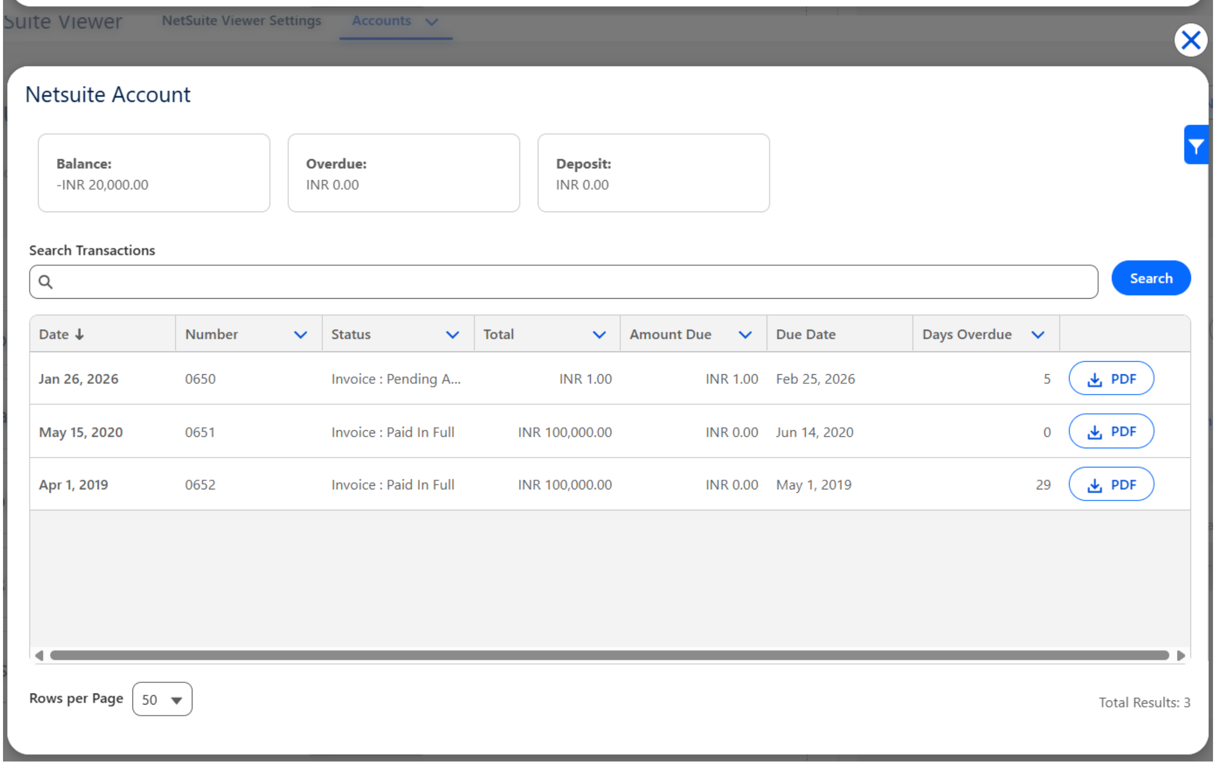
3. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/chapter_4453550706.html





You can also add additional filter logic for more complex queries. For example, let's return records that have a status of Pending Approval or their due date has past.





The viewer window will always initialize with no additional filters. If there are any filters defined when the viewer window is closed, they will be lost.

2.6 Uninstalling

To properly uninstall the managed package, make sure to complete the following pre-uninstallation steps:

1. Remove permission set assignments from users for the following permission sets:
   - **SCG NetSuite Viewer Admin**
   - **SCG NetSuite Viewer User**
2. Remove the **Show NetSuite Transactions** button from all Account page layouts and Lightning record pages where it is referenced.
3. If no longer needed, delete the remote site settings for the NetSuite URLs.
4. Go to **Installed Packages** in Setup and uninstall the package.

2.7 Troubleshooting

Issue or Question	**Resolution Steps**
"Invalid login attempt. For more details, see the Login Audit Trail in the NetSuite UI at Setup > Users/Roles > User Management > View Login Audit Trail."	• Check that the client id/secret is correct. • The NetSuite user's role may not be configured correctly.
"That Suitelet is invalid, disabled, or no longer exists."	• Check that the script/deploy numbers are correct.
"Something went wrong. Please check the console for more info." after clicking on the Show Netsuite Transactions button	• Right click on the page and open up the browser Dev Tools • Go to the Console tab, and find the Netsuite data error::: • Expand Object>body>message to view the error message • If the error mentions a field is not found, go ensure that field is not referenced in the custom metadata (or correct it if it has a typo), SCG NetSuite Viewer Table Column • If the error is of type 403, there is likely a connection issue, use the Reconnect To Netsuite button on the Netsuite Viewer Settings page

3 NetSuite Viewer - Technical Documentation

3.1 Table of Contents

3.2 Overview

This document outlines the components relevant to the Netsuite Viewer package.

This entire package is **managed.**

3.3 Permissions

The package contains two permission sets:

Name	Description
SCG Netsuite Viewer User	Contains permissions to be able to to use the SCG NetSuite Viewer Plugin components
SCG Netsuite Viewer Admin	Contains permissions to be able to configure the settings for the SCG NetSuite Viewer Plugin components.

3.4 Objects

3.4.1 Custom Objects

SCG_Netsuite_Viewer_Column__c		
Field	**Type**	**Description**
Active__c	Checkbox	When checked, the column is active and will appear in the viewer table.
Column_Order__c	Number	The order in which the column will appear in the viewer table.
Data_Type__c	Picklist	The data type for which the value will be formatted as in the viewer table.

SCG_Netsuite_Viewer_Column__c		
Default_Sort_Column__c	Checkbox	When checked, will denote the column as the default to sort by on initial load of the viewer table.
Hidden__c	Checkbox	When checked, the column will be included in the query from Netsuite, but will not show in the viewer table.
Label__c	Text(255)	The label of the column in the viewer table.
Links_To_Salesforce_Records__c	Checkbox	When checked, denotes the column as having a valid Salesforce Id value that can be linked to.
NS_Field_Name__c	Text(255)	The name of the field in netsuite.
SObject_Name_Field__c	Text(255)	Defines the field for the subject to be used as the label for a column that is marked as Links to Salesforce Records.
SObject_Type__c	Text(255)	If the Links to Salesforce Records box is checked, defines the subject type for the column.
Sort_Direction__c	Picklist	When the Default Sort Column box is checked, this value determines the initial direction for which the column values are sorted.
Sortable__c	Checkbox	When checked, the column will be sortable within the viewer table.

SCG_NS_Viewer_Object_Config__c		
Field	**Type**	**Description**
Field__c	Text(80)	The selected sobject field(s).

SCG_NS_Viewer_Object_Config__c		
NetSuite_Filter__c	Text(80)	The conditional for the where clause in the SQL query against the NetSuite database, which can use substitution notation (i.e. {0}) to indicate where to insert the field specified.
Object__c	Text(80)	The selected object.

3.4.2 Standard Objects

Account		
Field	**Type**	**Description**
SCG_Netsuite_Customer_ID__c	Text(255)	Stores the customer id from Netsuite.

3.5 Quick Actions

Account	
Name	**Description**
SCG_Show_Netsuite_Transactions	Action to display NetSuite transaactions in a table.

3.6 Apex

3.6.1 Triggers

Trigger API Name	Object	Fires On
SCG_NetsuiteViewerColumnTrigger	SCG_Netsuite_Viewer_Column__c	After Insert After Update

3.6.2 Classes

SCG_NetSuiteViewerColumnsHelper		
Method Name	**Parameters**	**Description**
doInit	n/a	Retrieves necessary data to be used in the column config ui.
getNetsuiteColumnsLightning	activeOnly	@AuraEnabled method invokes getNetsuiteViewerColumns()
getNestuiteViewerColumns	activeOnly	Returns a list of ns viewer columns.
wrapColumn	col	Wraps the ns viewer column sobject record to be used in the column config ui.
getActivePicklistValues	objectApiName, fieldApiName	Returns a list of active picklist values given and sobject type and its picklist field.
getOrgObjectOptions	n/a	Returns a sorted list of options for all objects within the org.
getNameFieldOptions	sobjectType	Returns a list of name fields for a given sobject type.

SCG_NetSuiteViewerColumnTriggerHandler		
Method Name	**Parameters**	**Description**
checkForMultipleDefaults	newColumns	Blocks record modification if multiple columns marked as default are discovered.

SCG_NetSuiteViewerUtils		
Method Name	**Parameters**	**Description**

SCG_NetSuiteViewerUtils		
getNetSuiteTransactionData	params	Params is a stringified object passed from javascript with the attributes requiered for the method logic. Returns the NetSuite transaction rows for the current account.
getInvoicePdfUrl	invoiceId	Returns the url of the downloaded pdf.
doInit	accountId	Retrieves the admin configuration and account information to be used for retrieving NetSuite data rows.
buildQuery	customerId, sortBy, sortDirection	Builds the transaction query based on the defined columns in SCG_NetSuite_Viewer_Table_Column__c.
saveToFiles	theName, pdfData	Inserts a ContentVersion record for the pdf data retrieved from NetSuite.
getAccessToken	n/a	Generates the jwt to retrieve the access token from NetSuite with values stored in the NetSuite Integration Credentials record.
base64URLEncode	input	Takes a blob as input to convert to a base64 encoded string.
doCallout	endpoint, accessToken, httpMethod, payload, contentType	Performs an http callout to NetSuite from Salesforce and returns the response.

3.6.3 Test Classes

Name	**Provides Coverage For**
SCG_TestNetsuiteViewerColumnsHelper	• SCG_NetsuiteViewerColumnsHelper
SCG_TestNetSuiteViewerUtils	• SCG_NetSuiteViewerUtils
SCG_TestNetsuiteViewerColumnTrigger	• SCG_NetsuiteViewerColumnTriggerHandler • SCG_NetsuiteViewerColumnTrigger

3.7 Lightning Components

3.7.1 Lightning Web Components

Name	Description
scgDisplayNetSuiteData	This component renders a data table with information queried from NetSuite. PDF Download is available for customer invoice rows.
scgModalEventDispatcher	Used to be able to fire custom events from a lightning modal.
scgNetSuiteTransactionsQuickAction	Wrapper component for Show NetSuite Transactions quick action.
scgNetSuiteObjectConfig	Represents a single object config in the viewer settings.
scgNetSuiteObjectConfigManager	Renders a list of created object configs in the viewer settings. Allows for edit and delete of object config records.
scgNetsuiteServiceUtils	A shared javascript utility for all components in the package.
scgNetsuiteViewerColumn	Edit form for ns viewer columns.
scgNetsuiteViewerColumnManager	Renders a list of created ns viewer columns. Allows for edit and delete of column records.
scgNetsuiteViewerFilters	Renders the ui to set additional filters within the viewer table.
scgNetsuiteViewerSettings	Renders the General Configuration ui in the ns viewer settings.
scgNsCredentialsModal	A custom lightning modal to input credential details to connect to Netsuite.
scgNsViewerStyles	A shared stylesheet for all components in the package.
scgPillContainer	A custom pill container to display selected values for object fields within an object config.

3.8 Custom Settings

SCG_NS_Viewer_Settings__c		
Field	**Type**	**Description**
Consumer_Key__c	Text(255)	The consumer key/client id value of the NetSuite instance to connect.
Conusmer_Secret__c	Text(255)	The consumer secret/client secret value of the NetSuite instance to connect.
Currency_Display_Format__c	Text(255)	The format for which currency values will display in the viewer table.
Customer_Currency_Code__c	Text(255)	The currency code to use for formatting the transaction table summary items.
Enable_Lazy_Loading__c	Checkbox	When checked, enables lazy loading for the viewer table.
Header_Fields__c	Text(255)	The fields that will show above the table in the viewer.
Invoice_Deploy_Number__c	Text(255)	The deploy number from NetSuite to use in the integration url.
Invoice_Script_Number__c	Text(255)	The script number from NetSuite to use in the integration url.
NetSuite_Customer_Id_Field__c	Text(255)	Api name of NetSuite customer id field on the account.
NetSuite_Instance_Name__c	Text(255)	The domain name of the NetSuite instance.
Token_key__c	Text(255)	The token key value of the NetSuite instance to connect.
Token_Secret__c	Text(255)	The token secret value of the NetSuite instance to connect.
Transaction_Types__c	Text(255)	Stores the selected types of transactions that will be queried from NetSuite.

SCG_NS_Viewer_Settings__c		
Transaction_Types_For_Download__c	Text(255)	Selected values will have have the PDF download button enabled for the row type.