Zoho Custom Function Documention

Release 0.1

BlinkSwag

Jan 23, 2023

CONTENTS

Zoho Books is online accounting software that manages your finances, automates business workflows, and helps you work collectively across departments. Visit <https://www.zoho.com/books/>`_

We use custom functions help in automation where procedural logic is required, which cannot be implemented with the default actions such as, Alerts/Tasks/WebHooks, etc. With custom functions you can automatically update the data in the related Books modules or third-party applications by executing simple program scripts.

Check out the Functions section for further information, including how to *Workflow Rule: Convert Estimate To SO* the project.

Note: This project is under active development.

CHAPTER

ONE

CONTENTS

1.1 Zoho Books

1.1.1 Estimate

Workflow Rule: Convert Estimate To SO

Module
Estimate

Function
Convert_estimate_to_so

Workflow Rule
Convert Estimate To SO

Description : This code is used to take information from an estimate in Zoho Books and create a new sales order using that information. The script starts by retrieving the estimate ID and organization ID and then goes on to get more detailed information about the estimate, such as the project ID, contact persons, and other details. It then takes the line items from the estimate and updates their custom fields. The script also updates the custom fields of the estimate itself. Finally, it creates a new sales order using the information gathered from the estimate.

It basically is a way to automate the process of converting an estimate to a sales order in Zoho Books, it eliminates the need for manual entry of data and saves time.

- The script starts by getting the "estimate_id", "organization_id" and "customer_id" from the estimate object.
- Then, it uses the "zoho.books.getRecordsByID" method to retrieve detailed information about the estimate using the organization ID and estimate ID. It stores this detailed information in the "estimate_detailed" variable.
- Next, the script checks if the estimate is associated with a project and if so, it gets the project ID and assigns it to the "project_id" variable.
- The script then gets information about the estimate such as whether the discount is applied before tax, contact persons, estimate number, discount amount, discount type, and salesperson ID.
- The script also retrieves the line items of the estimate and stores it in the "estimate_line_items" variable.
- For each line item, it retrieves the custom fields and updates the label and value of each custom field. It also adds the project ID to the line item.
- Then, the script makes a GET request to the Zoho Books API to retrieve the custom fields of sales orders.

- It then iterates through each custom field of the estimate and for each custom field that is active, has the same data type and label as the sales order custom field, it adds the custom field to the list of updated custom fields for the sales order.
- Finally, the script creates a new sales order using the updated information such as line items and custom fields.

1.1.2 Sales Order

Workflow Rule: Update SO Data on Zoho Project

Module
Sales Order

Function
update_so_data_on_zoho_project

Workflow Rule
Update SO Data on Zoho Project

Description : Certainly, the code you provided is a program that helps businesses keep track of their projects and their progress. The program automatically checks the status of sales orders in an accounting system, and if it finds a sales order related to a specific project, it retrieves more information about that project from a project management system. This information includes details such as the expected completion date, the names of the project manager and production manager, and other relevant details. Once it has all this information, it updates the project management system with the new details, and sends an email to the team members working on the project to let them know that the information has been updated.

This program is designed to automate and streamline the process of keeping track of project details, and helps businesses stay informed and make better decisions. It makes it easy to see the progress of a project, as well as any updates or changes that have been made, by pulling all this information into one place.

- The code starts by using the try keyword, which is used to handle exceptions that may occur during the execution of the code.
- The first step is to retrieve a specific sales order record from Zoho Books using the getRecordsByID method, passing in the salesorders module, the ID of the sales order (666479573), and the value of the salesorder_id field from the salesorder object. The response of this request is stored in the response variable.
- The code then retrieves the current_sub_status field from the salesorder object in the response variable and stores it in the Current_Sub_Status variable.
- Next, the code checks if the salesorder object's custom_field_hash field contains the key "cf_organization" and it is equal to "Blink Signs" and also if it contains a key "cf_project_unformatted" using containKey() method, returns true.
- If the above conditions are met, the code retrieves the value of the "cf_project_unformatted" field from the salesorder object's custom_field_hash field, and stores it in the cf_project_id variable.
- The code then makes a request to the Zoho Projects API using the invokeurl method, passing in the URL of the project, the request type (GET), and the connection (zoho_book). The response is stored in the res variable.
- The code checks if the res object contains a key "zprojects_project_id" using containKey() method, returns true.
- If the above condition is met, the code retrieves the value of the "zprojects_project_id" field from the res object and stores it in the zproject_id variable.

- The code then makes another request to the Zoho Projects API, this time to retrieve information about the project using the zproject_id variable, passing in the URL of the project, the request type (GET), and the connection (zoho_projects). The response is stored in the Get_proj variable.
- The code then initializes the Expected_Install_Date, owner_name, zproject_name, Production_Manager, Project_Manager, updated_by variables with empty values.
- The code then converts the Get_proj object's custom_fields field to a list and stores it in the custom_fields variable.
- The code then uses a for loop to iterate over each index in the custom_fields list, and for each index, it checks if the corresponding field contains the keys "Expected Install Date", "Project Managers", "Production Managers" using containKey() method. If the condition is true, it retrieves the corresponding value and assigns it to the Expected_Install_Date, Project_Manager, Production_Manager variables respectively.
- The code then creates a new map called update_map and adds the key-value pairs to it.
- The code then checks the status of the sales order, it checks if the status field of the salesorder object in the response variable is equal to "open" and the Current_Sub_Status variable is also equal to "open". If these conditions are met, it adds the key-value pair "UDF_CHAR23":"Approved" and "custom_status":"1264527000033851429" to the update_map.
- The code then checks if the status field of the salesorder object in the response variable is equal to "void". If this condition is met, it adds the key-value pair "UDF_CHAR23":"Declined" and "custom_status":"1264527000036144823" to the update_map.
- The code then checks if the Current_Sub_Status variable is equal to "cs_product" or "cs_orderma". If this condition is met, it adds the key-value pair "UDF_CHAR23":"Production in Progress" to the update_map.
- The code then adds the key-value pair "UDF_CHAR24":salesorder.get("reference_number") to the update_map.
- The code then uses the update_map to update the project fields in Zoho Projects using the API.
- The code then uses the catch keyword to handle any exceptions that may occur during the execution of the code.
- The code then ends with the finally keyword, which is used to execute any code that needs to run regardless of whether an exception occurred or not.
- The code initializes a variable called owners as a list of dictionaries, each containing a name and email address.
- The code uses a for loop to iterate over each item in the owners list.
- Inside the loop, the code creates a variable called str which contains a string of HTML formatted text that will be used as the body of the email. The text includes placeholders for various pieces of information, such as the recipient's name, project name, project manager, production manager, and expected installation date.
- The code uses the getprefix(":") method to extract the name of the owner from the current item in the owners list, and uses the remove() method to remove certain characters from the name.
- The code then uses the zproject_id and zproject_name variables, and the Project_Manager, Production_Manager, and Expected_Install_Date variables to replace the placeholders in the str variable with the appropriate values.

*The code then uses the sendMail() function to send the email, passing in the email address of the recipient (which is extracted from the current item in the owners list), the subject of the email, and the body of the email (which is the str variable).

Workflow Rule : Update Production Items by SO

Module
Sales Order

Function
cu_so_production_items

Workflow Rule
Update Production Items by SO

Description :This code is used to take a sales order in Zoho Books and create or update records in a custom module called “cm_production_control” using the information from the sales order. It starts by retrieving the organization ID, sales order ID, and other details such as the sales order number, status, and line items. Then it goes on to retrieve the date and custom field for the sales order user.

For each line item in the sales order, it retrieves the name, quantity, ID, and SKU of the item and creates a data map with the item’s information, including the sales order number, date, and user. It then searches for existing production items in the organization using the line item ID. If no existing items are found, it creates a new record for the item in the “cm_production_control” module with the data map and status of the sales order. If existing items are found, it updates the existing records with the data map. It automates the process of creating or updating records in “cm_production_control” module based on the sales order and saves time.

- Retrieving the organization ID and sales order ID, as well as the sales order number, status, and line items.
- Retrieving the date and custom field for the sales order user.
- For each line item in the sales order:
  - Retrieving the name, quantity, ID, and SKU of the item.
  - Creating a data map with the item’s information, including the sales order number, date, and user.
  - Searching for existing production items in the organization using the line item ID.
  - If no existing items are found, create a new record for the item in the “cm_production_control” module with the data map and status of the sales order.
  - If existing items are found, update the existing records with the data map.

1.2 Zoho CRM