Automating Quotation Document Generation

Exploring Power Automate

- Initially attempted to automate document generation using Power Automate.
- Faced challenges in understanding the exact workflow and configuring automation correctly.
- Due to the complexity of implementation, did not proceed further.

Extracting Data from Excel Using Flask

- Developed a Flask-based script to process Excel data and insert it into a DOC template.
- Successfully generated documents.

Workflow

- Reads and processes data from quotations.xlsx using pandas.
- Uses python-docx to replace placeholders ({{key}}) in paragraphs and tables.
- Dynamically finds and inserts customer and industry logos.
- Generates Quotation_<Customer Name>.docx in the generated_quotations folder.
- Processes all rows and creates individual quotations with success messages.

Fetching Data from Odoo Using Webhooks

- Attempted to retrieve data directly from Odoo using webhooks.
- Encountered issues with data not being sent while running on localhost.
- Used Pipedream to verify if data was being received correctly.
- Deployed on Replit but faced document corruption issues.
- Finally deployed the solution on Railway, resolving document corruption problems.

Final Workflow

1. Data Collection & Transfer

- Odoo automation collects data from a record.
- Sends the data to the Flask app via a webhook.

2. Document Generation (Flask App)

- Processes received data and fills a Word template (.docx).
- Replaces placeholders with actual values.
- Inserts customer and industry logos.
- Saves the generated document.

3. Download & Access

- Generates a download link for the document.
- Sends the link back to Odoo.
- User clicks the "Generate Quotation" button in Odoo.
- Redirects to the Flask app’s download URL, allowing users to download the document.

STEEL:

Bharat Heavy Electrical Limited (BHEL)

Bhushan Power & Steel

Jindal SAW

Jindal Steel & Power Limited (JSPL)

Narsingh Ispat Limited

CEMENT:

Infinity Automation Solutions

Meghalaya Cements Limited

NMDC Limited

SEB ENERGY BUSINESS

These customers names can be for demo

Flask App Functionality

- Receives data from Odoo via a webhook (`/odoo-webhook`).
- Generates a Word document by replacing placeholders with received data.
- Inserts customer and industry logos.
- Saves the generated quotation in the `generated_quotations` folder.
- Provides a downloadable link for the generated document.
- Allows users to download the quotation via `/download/<filename>`.

  https://github.com/shivani8naik/flask-webhook-app

Updates Made in Odoo

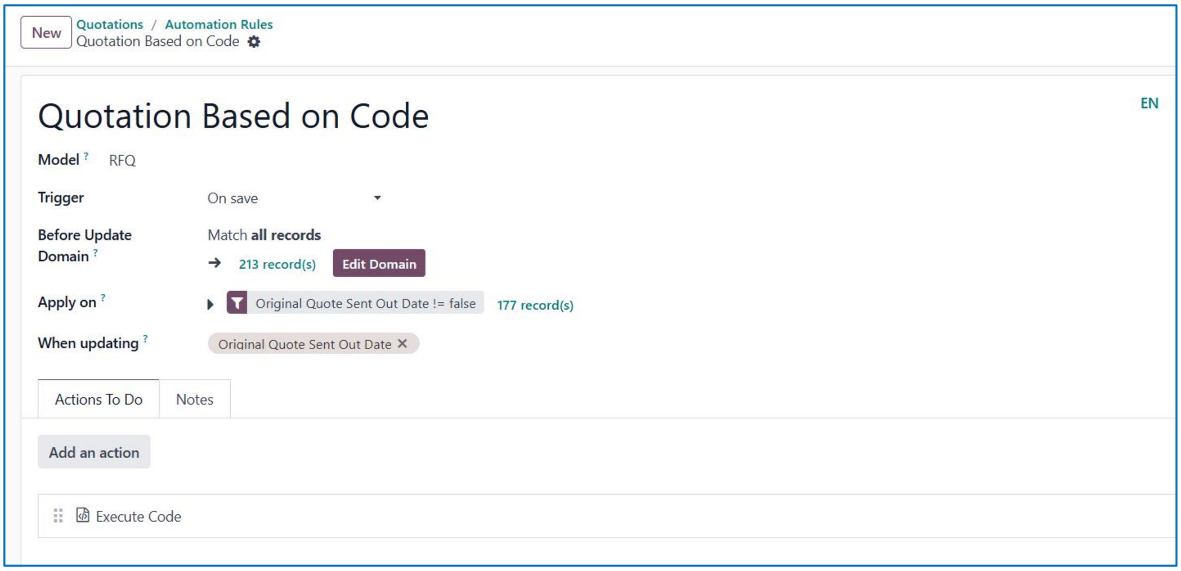
Automation Action

- Sends data to the Flask app for document generation.

- Initially attempted the "Webhook Notification" action but found it returned IDs instead of names.
- Implemented a workaround to send correct names instead of IDs.

Server Action

- Provides the "Generate Quotation" button.
- Allows users to download the generated document by redirecting them to the Flask app’s download link.



Quotation Based on Code

Model: RFQ

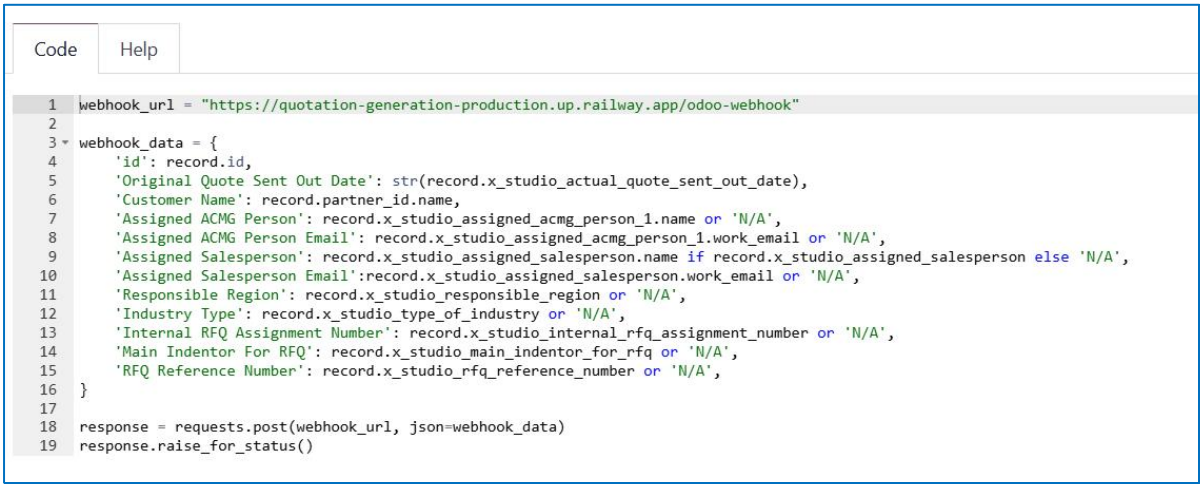
Trigger: On save

Before Update Domain: Match **all records** → 213 record(s)

Apply on: Original Quote Sent Out Date != false 177 record(s)

When updating: Original Quote Sent Out Date ✕

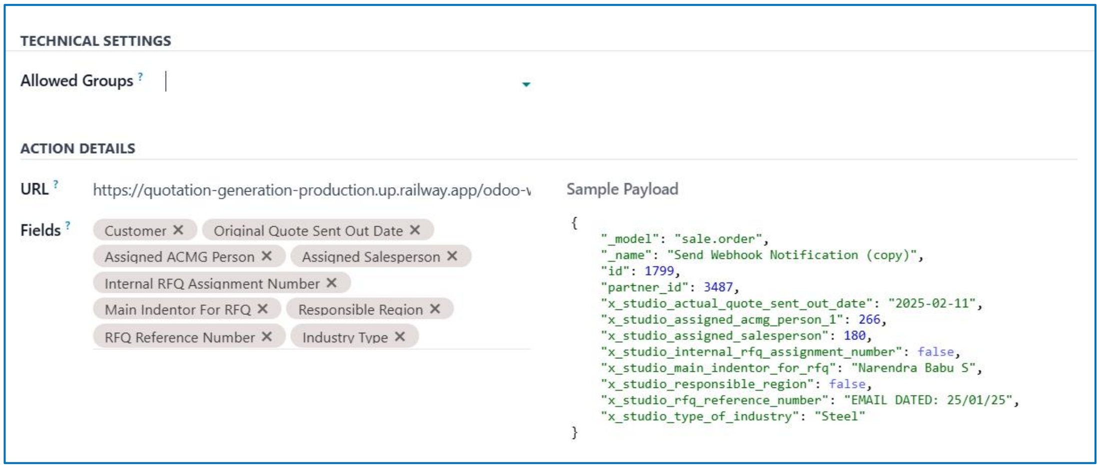
Execute Code



```python
webhook_url = "https://quotation-generation-production.up.railway.app/odoo-webhook"

webhook_data = {
    'id': record.id,
    'Original Quote Sent Out Date': str(record.x_studio_actual_quote_sent_out_date),
    'Customer Name': record.partner_id.name,
    'Assigned ACMG Person': record.x_studio_assigned_acmg_person_1.name or 'N/A',
    'Assigned ACMG Person Email': record.x_studio_assigned_acmg_person_1.work_email or 'N/A',
    'Assigned Salesperson': record.x_studio_assigned_salesperson.name if record.x_studio_assigned_salesperson else 'N/A',
    'Assigned Salesperson Email':record.x_studio_assigned_salesperson.work_email or 'N/A',
    'Responsible Region': record.x_studio_responsible_region or 'N/A',
    'Industry Type': record.x_studio_type_of_industry or 'N/A',
    'Internal RFQ Assignment Number': record.x_studio_internal_rfq_assignment_number or 'N/A',
    'Main Indentor For RFQ': record.x_studio_main_indentor_for_rfq or 'N/A',
    'RFQ Reference Number': record.x_studio_rfq_reference_number or 'N/A',
}

response = requests.post(webhook_url, json=webhook_data)
response.raise_for_status()
```



TECHNICAL SETTINGS

Allowed Groups ?

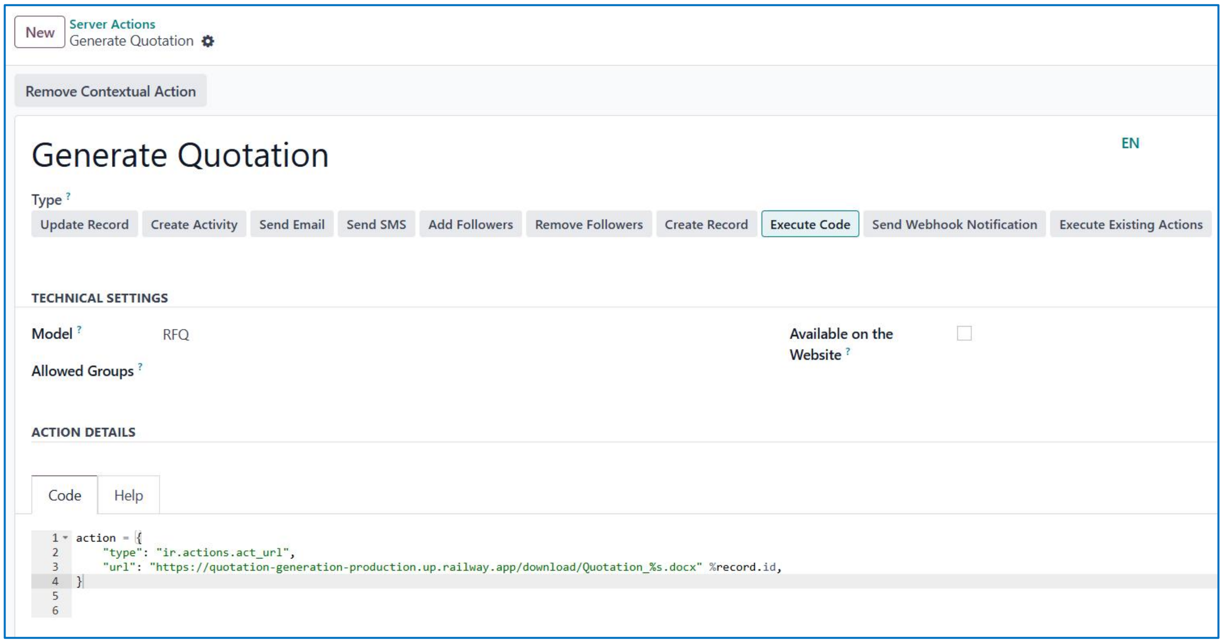
ACTION DETAILS

URL ? https://quotation-generation-production.up.railway.app/odoo-v

Fields ? Customer ✕ Original Quote Sent Out Date ✕ Assigned ACMG Person ✕ Assigned Salesperson ✕ Internal RFQ Assignment Number ✕ Main Indentor For RFQ ✕ Responsible Region ✕ RFQ Reference Number ✕ Industry Type ✕

Sample Payload

```
{
    "_model": "sale.order",
    "_name": "Send Webhook Notification (copy)",
    "id": 1799,
    "partner_id": 3487,
    "x_studio_actual_quote_sent_out_date": "2025-02-11",
    "x_studio_assigned_acmg_person_1": 266,
    "x_studio_assigned_salesperson": 180,
    "x_studio_internal_rfq_assignment_number": false,
    "x_studio_main_indentor_for_rfq": "Narendra Babu S",
    "x_studio_responsible_region": false,
    "x_studio_rfq_reference_number": "EMAIL DATED: 25/01/25",
    "x_studio_type_of_industry": "Steel"
}
```



Generate Quotation

Type ?

TECHNICAL SETTINGS

Model ? RFQ

Available on the Website ? □

Allowed Groups ?

ACTION DETAILS

```
action = {
    "type": "ir.actions.act_url",
    "url": "https://quotation-generation-production.up.railway.app/download/Quotation_%s.docx" %record.id,
}


```

System Workflow

1. Odoo Automation Trigger

- Triggered when the "Original Quote Sent Out Date" is filled and the record is saved.
- Collects relevant data from the Odoo record (Customer Name, Assigned ACMG Person, Industry Type, Assigned Salesperson, and other necessary fields).
- Sends the data as a JSON payload via an HTTP POST request to the Flask app's webhook endpoint (`/odoo-webhook`).

2. Flask App - Receiving Webhook Data

- Listens for incoming data at the `/odoo-webhook` endpoint.
- Parses the JSON to extract key fields.
- Validates the received data:
  - Ensures required fields (e.g., `id`) are present.
  - If missing, sends an error response to Odoo.
- Logs the received data for debugging and processing.

3. Processing the Document Template

Replacing Text Placeholders

- Opens the quotation template (`quotation_template.docx`).
- Loops through:
  - Paragraphs
  - Tables
  - Headers and Footers
- The `replace_text_in_doc` function replaces placeholders (`{{key}}`) with actual values from the received data.

Inserting Customer Logo

- Checks if a customer logo needs to be inserted.
- Searches for the logo in the `Logos` folder, using case-insensitive matching based on the customer's name.
- If found, the `insert_customer_logo` function inserts the logo where the placeholder `[Customer Logo Will Be Inserted Here]` exists.

Inserting Industry Logos

- Checks if industry-specific logos need to be inserted.
- Retrieves the **Industry Type** (e.g., Steel, Cement, Power, Oil & Gas, etc.).
- Searches for a matching folder under `Logos/Industry Type` and inserts all logos found in that folder into the document.
- The `insert_industry_logos` function replaces the placeholder `[Industry Logos Will Be Inserted Here]`.

4. Saving the Generated Document

- Once all placeholders are replaced and logos inserted, the final document is saved.
- Filename format: `Quotation_<record_id>.docx` (e.g., `Quotation_1771.docx`).

- Stored in the `generated_quotations` directory.

5. Generating & Returning the Download Link

- Constructs a download URL for the generated document.
- Sends the URL back to Odoo in the webhook response.

6. User Interaction in Odoo

Clicking "Generate Quotation" Button

- The download URL is embedded in the "Generate Quotation" button.
- When clicked, Odoo triggers a server action (`ir.actions.act_url`), redirecting the user to the Flask app's download endpoint.

Downloading the Document

- The user is redirected to the `/download/<filename>` endpoint.
- The Flask app serves the file as a **direct download**, allowing the user to save or open the document.