

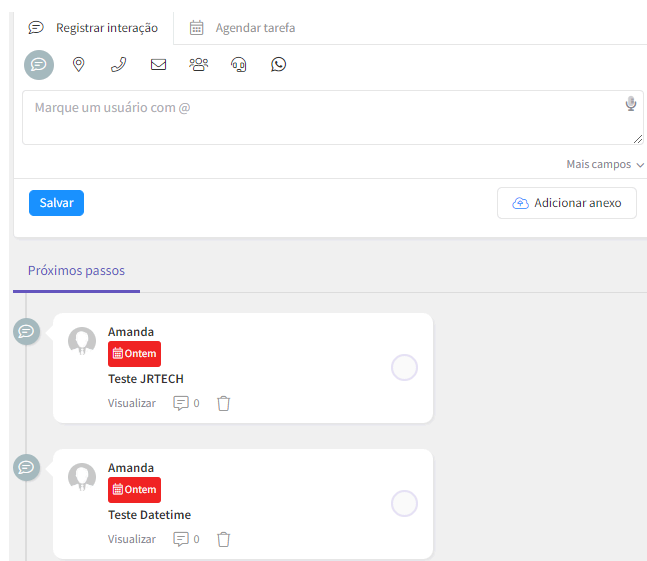
Introdução

Esta documentação fornece uma visão geral do código Python desenvolvido para facilitar o processo de envio de clientes, interações e tarefas do sistema Ploomes para o sistema Milvus. O código implementa uma solução automatizada para integrar dados de clientes entre esses dois sistemas, utilizando requisições HTTP para enviar os dados formatados para o Milvus.

A seguir, serão apresentadas informações sobre a funcionalidade principal do código, detalhes da implementação, dependências externas necessárias, observações importantes e exemplos de uso.

Importante

Tarefas e interações serão enviados somente se criados dentro de um perfil de cliente do Ploomes, como no exemplo:



Isso ocorre porque o Milvus atualmente não aceita tickets sem identificação de clientes.

Clientes, interações e tarefas são enviados por padrão a partir de 24 de fevereiro de 2024, para minimizar ao máximo a possibilidade de duplicatas.

Envio de Clientes do Ploomes para o Milvus

Este script em Python processa e envia dados de clientes do sistema Ploomes para o sistema Milvus. Ele realiza as seguintes operações:

Conexão com o MongoDB:

O script se conecta a um banco de dados MongoDB local.

Funções Auxiliares:

cliente_enviado(cnpj_cpf): Verifica se um cliente já foi enviado com base no seu CNPJ ou CPF.

marcar_cliente_enviado(cnpj_cpf): Marca um cliente como enviado no MongoDB.

Iteração sobre Clientes do Ploomes:

Itera sobre os clientes do Ploomes para processamento e envio.

Verificação da Data de Criação:

Verifica se a data de criação do cliente é posterior a 24/02/2024.

Formatação dos Dados do Cliente para o Milvus:

Os dados do cliente do Ploomes são formatados de acordo com as necessidades do sistema Milvus.

Envio dos Dados para o Milvus:

Os dados formatados são enviados para o Milvus via requisição HTTP POST.

O cliente é marcado como enviado no MongoDB após o envio bem-sucedido.

Gestão de Tentativas:

Caso ocorra um erro durante o envio, o script tentará novamente até um máximo de três tentativas.

Dependências:

pymongo: Utilizado para interagir com o MongoDB.

requests: Utilizado para fazer requisições HTTP.

time: Utilizado para pausar a execução do script entre tentativas de envio.

Envio de Clientes para o Ploomes

O código tem a finalidade de enviar clientes do sistema Milvus para o sistema Ploomes utilizando requisições HTTP POST. Ele itera sobre os clientes do sistema Milvus, verifica se cada cliente já existe no sistema Ploomes e, caso contrário, formata os dados do cliente de acordo com os requisitos do Ploomes e envia a requisição.

Funcionalidade Principal:

Este código realiza as seguintes tarefas:

Itera sobre os clientes do sistema Milvus.

Verifica se o cliente já existe no sistema Ploomes.

Formata os dados do cliente de acordo com os requisitos do Ploomes.

Envia a requisição HTTP POST para o Ploomes.

Lida com respostas de sucesso, erros e limites de taxa (status 429) da API do Ploomes.

Detalhes da Implementação:

Clientes a Enviar:

Os clientes são obtidos do sistema Milvus e armazenados na lista *clientes_milvus*.

Antes de enviar um cliente para o Ploomes, verifica-se se o CNPJ/CPF desse cliente já existe no Ploomes para evitar duplicatas. Isso é feito mantendo um conjunto *existing_clients* que armazena os CNPJs/CPFs dos clientes já existentes no Ploomes.

Formatação de Dados do Cliente:

Os dados do cliente são formatados de acordo com os requisitos do Ploomes.

O tipo de pessoa (física ou jurídica) é determinado com base no comprimento do CNPJ/CPF.

Os dados do cliente são organizados em um dicionário *cliente_ploomes* conforme necessário pelo Ploomes, incluindo informações como nome, endereço, telefone, etc.

Envio da Requisição:

Após a formatação dos dados, uma requisição HTTP POST é enviada para o Ploomes contendo os detalhes do cliente formatados.

O código lida com respostas de sucesso (status 200), erros (outros códigos de status) e limites de taxa (status 429) da API do Ploomes.

Tentativas de Reenvio:

Se ocorrer um erro de limite de taxa (status 429), o código tentará novamente o envio da requisição até um máximo de 3 tentativas, com um intervalo de 5 segundos entre as tentativas.

Dependências Externas:

O código depende da biblioteca *requests* para fazer requisições HTTP.

A biblioteca *time* é utilizada para introduzir um intervalo entre as tentativas de reenvio em caso de limite de taxa.

Envio de Dados de Tarefas do Ploomes para o Milvus

Esta parte do código é responsável por processar dados de tarefas do sistema Ploomes e enviar informações relevantes para o sistema Milvus. O código realiza a verificação de tarefas específicas do Ploomes, identifica clientes associados a essas tarefas e envia os dados formatados para o Milvus, sujeito a algumas condições de tempo e presença de marcadores.

Funcionalidade Principal:

- Itera sobre as tarefas do sistema Ploomes (*tasks_data_ploomes*).
- Verifica se as tarefas contêm os marcadores necessários (*INSTALAÇÃO* ou *ENTREGA*).
- Calcula a diferença de tempo entre a criação da tarefa e a hora atual.
- Se a diferença de tempo for menor ou igual a 10 minutos, o código busca por clientes relacionados à tarefa no sistema Ploomes.

- Obtém os detalhes do cliente no sistema Ploomes e procura pelo cliente correspondente no sistema Milvus.
- Se um cliente correspondente for encontrado, os dados relevantes da tarefa são formatados e enviados para o Milvus.

Detalhes da Implementação:

Processamento de Tarefas do Ploomes:

As tarefas são iteradas uma a uma para verificar sua relevância e associar clientes específicos.

Tarefas que não contêm os marcadores necessários são ignoradas.

Verificação de Tempo:

A diferença de tempo entre a criação da tarefa e a hora atual é calculada para determinar a recenticidade da tarefa.

Associação de Clientes:

Os dados dos clientes são obtidos do sistema Ploomes.

Cada cliente é associado a uma tarefa com base no CPF/CNPJ presente nos dados da tarefa.

Envio de Dados para o Milvus:

Os dados da tarefa são formatados de acordo com os requisitos do Milvus.

A requisição HTTP POST é enviada para o Milvus com os dados formatados.

Dependências Externas:

O código depende das bibliotecas `pytz` e `datetime` para lidar com fusos horários e manipulação de datas e horas.

A biblioteca `requests` é utilizada para fazer requisições HTTP para o Milvus

Envio de Dados de Interações do Ploomes para o Milvus

Tem a finalidade de processar dados de interações do sistema Ploomes e enviar informações relevantes para o sistema Milvus. Ele itera sobre as interações do sistema Ploomes (`tickets_data_ploomes`), verifica a presença de marcadores específicos e calcula a diferença de tempo entre a criação da interação e a hora atual. Com base nessas verificações, o código associa clientes às interações e envia os dados formatados para o Milvus, sujeito a algumas condições.

Funcionalidade Principal:

- Itera sobre as interações do sistema Ploomes.
- Verifica se interações contêm os marcadores necessários (`INSTALAÇÃO` ou `ENTREGA`).
- Calcula a diferença de tempo entre a criação da interação e a hora atual.
- Se a diferença de tempo for menor ou igual a 10 minutos, o código busca por clientes relacionados à interação no sistema Ploomes.

- Obtém os detalhes do cliente no sistema Ploomes e procura pelo cliente correspondente no sistema Milvus.
- Se um cliente correspondente for encontrado, os dados relevantes da interação são formatados e enviados para o Milvus.

Detalhes da Implementação:

Processamento de Tickets do Ploomes:

Os tickets são iterados um a um para verificar sua relevância e associar clientes específicos.

Tickets que não contêm as tags necessárias são ignorados.

Tickets sem clientes associados também são ignorados.

Verificação de Tempo:

A diferença de tempo entre a criação da interação e a hora atual é calculada para determinar a recenticidade da interação.

Associação de Clientes:

Os dados dos clientes são obtidos do sistema Ploomes.

Cada cliente é associado a uma interação com base no CPF/CNPJ presente nos dados da interação.

Envio de Dados para o Milvus:

Os dados relevantes da interação são formatados de acordo com os requisitos do Milvus.

A requisição HTTP POST é enviada para o Milvus com os dados formatados.

Dependências Externas:

O código depende das bibliotecas *pytz* e *datetime* para lidar com fusos horários e manipulação de datas e horas.

A biblioteca *requests* é utilizada para fazer requisições HTTP para o Milvus.

Observações:

Os marcadores em tarefas e interações são valores numéricos. Caso seja necessário realizar adições, exclusões e alterações de tickets a serem enviados, o valor numérico se encontra em *TagID* dentro do campo *Tag* em ambos.

Não há uma correspondência exata atualmente entre todos os campos das plataformas Milvus e Ploomes, podendo ser realizadas futuras adições conforme as APIs de ambas receberem novas versões.

O MongoDB está sendo utilizado para armazenar os dados de CPF/CNPJ de clientes. A medida de comparação com os dados do Milvus podem não estar totalmente acuradas, visto que o Milvus retorna apenas 1000 itens.