

解説書：解約自動返信システムの全像

作成	Senko
種別	

全体処理フロー

1. Zendeskフォームでの流入
2. Zendeskトリガでの処理発火
 - ・ フォーム・ステータスなどを元に3のAPIリクエストを含め処理
3. パイプライン（Dify）での処理の実施
 - ・ 応対内容の判断、Zendeskへのコメント登録
4. Zendeskから顧客へメール通知

▼ Zendesk処理

Difyへの通信処理の登録

解約自動返信 アクティブ

詳細

イベントのサブスクリプション

アクティビティ

エンドポイント

https://api.dify.ai/v1/workflows/run

リクエスト方法

POST

リクエスト形式

JSON

作成者:

 専光建志 2025年11月13日, 5:25

最終更新

 専光建志 2025年11月13日, 5:28

認証

ベアラートークン

シークレットキー

WebhookデータがZendeskから取得されているかどうかを検証します [検証について](#)

LL2zD.....

コピー

トリガの発火条件とアクション内容の設定

1. 自動処理対象の判定

▼ 条件

条件

トリガを実行するための必要条件

以下の条件をすべて満たす：

カテゴリ	演算子	値
チケット > チケットフォーム	=	消費者用_B.定期便のスキップ...
チケット > 自動返信_ステータス	=	-
チケット > チケット	=	更新された
チケット > チケットのステータス	≠	自動処理待ち
チケット > AI要約 (30文字)	あり	

条件を追加

▼ アクション

アクション

上記条件が満たされた場合に実行されるアクション

カテゴリ	値
チケット > 自動返信_ステータス	処理待ち
チケット > 自動返信_内容種別	解約停止希望
チケット > チケットのステータス	自動処理待ち

2.自動対応のための通信処理

▼ 条件

条件

トリガを実行するための必要条件

以下の条件をすべて満たす：

カテゴリ	演算子	値	
チケット > 自動返信_内容種別	=	解約停止希望	🗑
チケット > チケットのステータス	=	自動処理待ち	🗑
チケット > 自動返信_ステータス	=	処理待ち	🗑

▼ アクション

アクション

上記条件が満たされた場合に実行されるアクション

カテゴリ	値	🗑
チケット > 自動返信_ステータス	処理済み	
カテゴリ	値	🗑
チケット > タグを追加	初回自動返信を実施 ×	
カテゴリ	値	🗑
チケット > タグを追加	自動返信を実施 ×	
カテゴリ	値	🗑
チケット > 自動処理への改善要望/違和感	未判定	
カテゴリ	値	🗑
チケット > チケットのステータス	自動処理済_未対応	
カテゴリ	値	🗑
通知方法 > アクティブなWebhook	解約自動返信	
エンドポイント https://api.dify.ai/v1/workflows/run		

カテゴリ

通知方法 > アクティブなWebhook

値

解約自動返信

エンドポイント

https://api.dify.ai/v1/workflows/run

説明

-

方法

POST

1

{

2

"user": "gs_cs_API",

3

"response_mode": "streaming",

4

"inputs": {

5

"email": "{{ticket.requester.email}}",

6

"tel": "{{ticket.ticket_field_33787008920599}}",

7

"ticket_id": "{{ticket.id}}",

8

"text": "{{ticket.latest_comment_html}}",

9

"needs": "{{ticket.ticket_field_35512850954263}}",

10

"data_delete_flag": "{{ticket.ticket_field_35512888196375}}"

使用可能なプレースホルダを表示

▼ パイプライン処理

環境URL

▼ DifyのURL

<https://cloud.dify.ai/app/59b9e3cb-87a4-46ca-bbdf-f4c082b2c069/workflow>

▼ GUIでのテスト環境

<https://udify.app/workflow/FD8knwBIQgIU2ZIM>

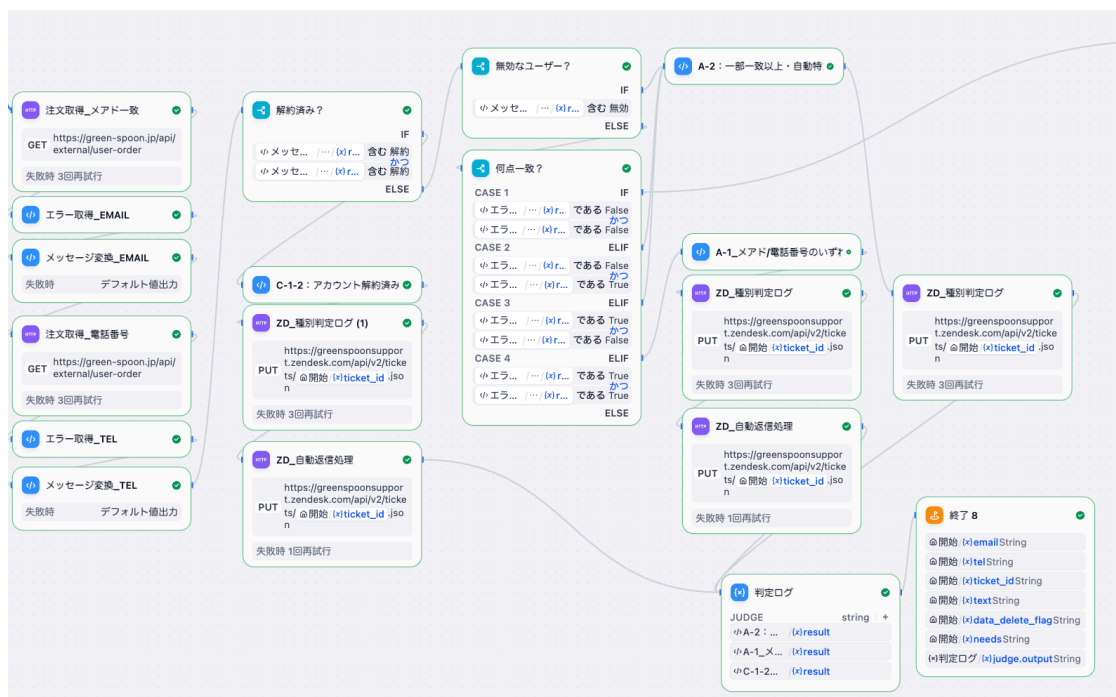
詳細な処理解説

▼ 1.環境変数と入力値

- 環境変数
 - `gs_api_get_volume` : API検索時の取得件数制限（例: 10件）
 - `gs_api_token` : Green Spoon APIアクセス用トークン
 - `zendesk_basic` : Zendesk API Basic認証情報
- 開始ノードの入力値（問い合わせフォームから送信される）
 - メールアドレス (`email`)

- 電話番号 (`tel`)
- ZendeskチケットID (`ticket_id`)
- 問い合わせテキスト (`text`)
- データ削除希望フラグ (`data_delete_flag`)
- その他要望 (`needs`)

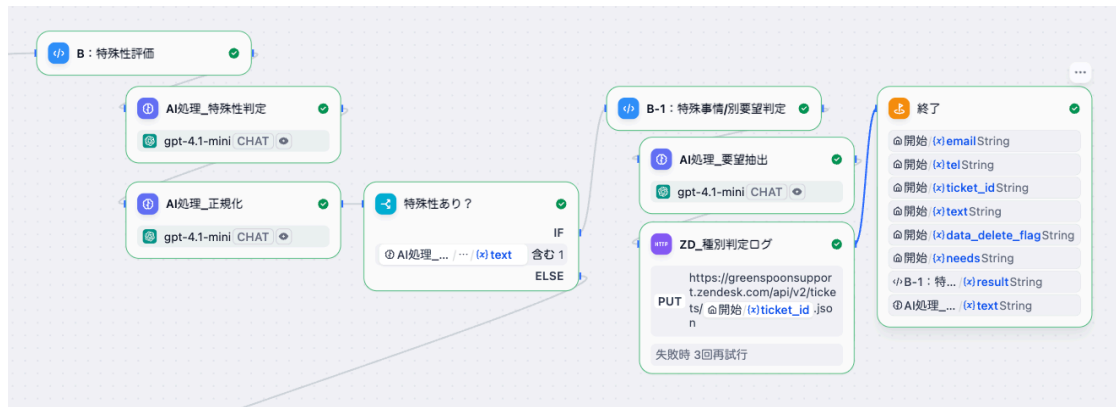
▼ 2. 注文情報取得とエラーチェック



- メールアドレスで注文情報取得
 - `https://green-spoon.jp/api/external/user-order` にメールアドレス、APIトークン、取得件数制限をパラメータとしてGETリクエストを送る
 - 取得結果のJSONを受け取る
- 電話番号で注文情報取得
 - 同じAPIに電話番号とAPIトークンをパラメータとしてGETリクエストを送る
- APIレスポンスのエラー判定
 - 返却JSONに `"message"` キーが含まれていれば取得失敗と判定（例：「メール未登録」など）
- 注文取得成功の総合判定

- メールアドレス検索と電話番号検索の結果のいずれかが成功していれば注文特定成功と判断する
- 複数顧客一致の判定
 - 複数の注文が該当した場合は自動返信せずオペレーター対応推奨とする

▼ 3. 問い合わせテキストのAI分析



- 特殊事情判定
 - GPT-4 miniモデルで問い合わせテキストを解析し、緊急対応が必要なクレームや体調不良、返金要求などを判定する
 - 出力は「1」（個別対応推奨）か「2」（自動返信OK）の半角数字1文字
- AI判定結果の正規化
 - AIの出力が半角数字1文字でない場合に厳密に修正する
- 判定理由の短縮化
 - AIによる特殊事情判定の理由文を20文字以内に簡潔にまとめ、オペレーターが確認しやすくする

▼ 参考：AI処理_特殊性判定のプロンプト

依頼内容

問い合わせ文を解析し、自動返信ではなく個別対応が必要かを判定してください。

具体的な事情やお怒りの表現がない、かつ解約・一時停止希望のみであれば自動返信で構いません。

状況

食品の定期配送サービスに対する、解約や一時停止の希望が主な内容です。

判断基準

以下のいずれかに該当する場合は「個別対応」(1) としてください。
それ以外は「自動返信OK」(2) とします。

- 具体的な不満やトラブルが明確に書かれている（例：配送トラブル、品質問題、誤配など）
- 謝罪や返金など顧客対応が必要と推測される内容が含まれる
- 「解約・停止」以外の要望（商品の再送、注文確認、発送遅延など）が含まれる
- 体調不良、入院、身内の不幸など緊急性や配慮が必要と考えられる事情に触れている
- 明確に感情的な怒りや困惑の表現がある

出力形式

個別対応推奨 → 1

自動返信で良い → 2

判断が難しい場合は2を出力してください。

出力は必ず半角数字の1文字のみとしてください。

判定対象テキスト

{{#1761111669370.text#}}

あなたが過去に誤って個別回答不要と判断した問い合わせ文章
「解約金発生とのことですがいくらの請求がかかりますでしょうか？」

→ これは具体的な金銭に関係する質問が含まれてるため個別回答と

判断すべきでした。

あなたが過去に誤って個別回答すべきとした問い合わせ文章

「この前商品が来たばかりなのにまた本日配達されて冷凍庫にはいりませんでした。とりあえず冷蔵庫にいらしていますが、多分ダメになってしまうかと思います。多すぎなのでしばらくお休みします。再会時に再度ご連絡いたします。」 → 2

「解約希望 添加物が多く入っていてとても残念でした。」 → 2

「長期で不在なのでしばらくお休みしたいです。再開日は未定です。」 → 2

「美味しくなくない」 → 2

「解約希望 添加物が多く入っていてとても残念でした。」 → 2

「少ししんどくなってきたので無期限で停止したい」 → 2

参考知識

(1) 「先日Amazonにてキャンセル致しましたが発送連絡があります。確認をお願いします」

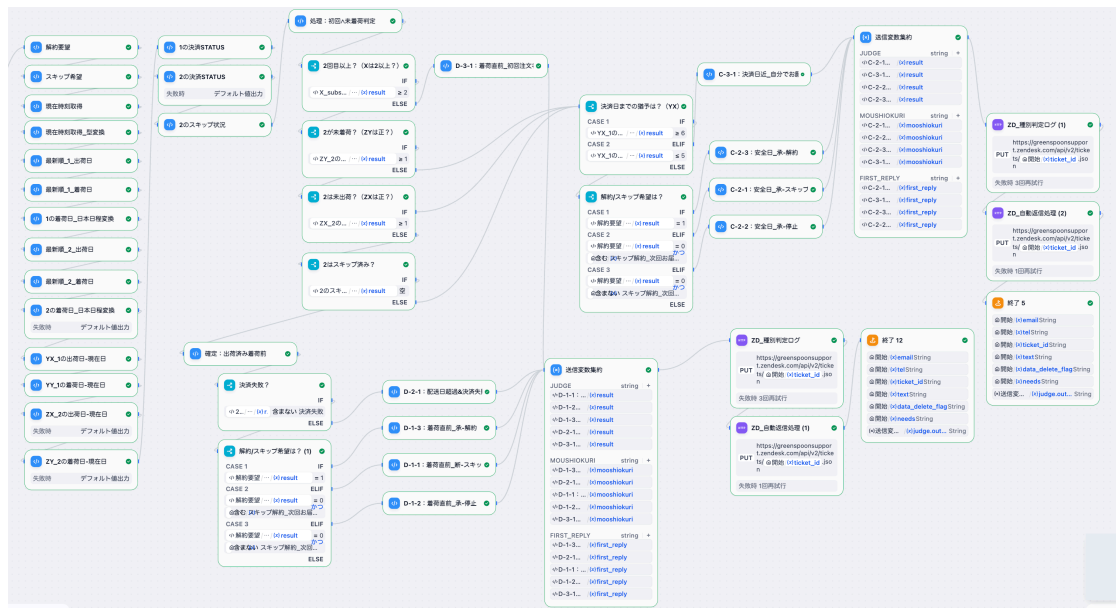
こういった場合は個別対応です。

自社サイト以外のamazonなどでの販売をしておりその場合のデータは確認ができません。

そのため自動回答ができないためそういった主旨が見られる場合は個別対応です。

以上です。

▼ 4. 注文・契約状況の解析



- 定期注文有無判定
 - 取得注文データから `sub_subscription_count` が1以上かつ `order_deleted_at` が無い注文を抽出し、定期契約の有無を判定
- 最新の出荷日・配送日の取得
 - 有効な注文の中で最新の `shipped_at` (出荷日) と `delivery_date` (配送日) を抽出
- APIレスポンス日時の現在時刻へ変換
 - APIレスポンスヘッダーのHTTP形式日時をISO8601形式に変換し扱いやすくする
- 日付差分計算
 - 「出荷日－現在時刻」「配送日－現在時刻」の日数差を計算し、停止・解約猶予期間を判定に利用
- 解約希望判定
 - フォームの `data_delete_flag` が「アカウント削除希望」かどうか判定し、0/1の数値で管理
- 決済状態取得
 - 最新注文の決済状態（成功、未処理、失敗など）を取得する
- 停止済み判定
 - ユーザーデータの `user_deleted_at` の有無で停止済みか判定

- その他条件判定
 - 「注文不明」「複数一致」「定期なし」「停止済み」「決済状態」「出荷日直近」「解約希望」などを多段階で条件分岐する