

Shopify

と

楽天

Yahoo!ショッピング

ネクストエンジン

を連携

業務効率化10の事例



9.14



16:00 - 17:00

WEBセミナー

テーブス株式会社
マーケティング責任者

堤 健一郎



登壇者紹介



テープス株式会社

マーケティング責任者

堤 健一郎 Tsutsumi Kenichiro

▼経歴

市場調査会社、CRM マーケティング支援会社を経て、前職ではヤフー株式会社で Yahoo! ショッピングのマーケティングを担当。

現在は、TēPs 事業のマーケティングを担当

▼趣味

- ・本（読書、一箱本棚オーナー、古本市のイベント開催）
- ・バドミントン（高校の部活でコーチ）

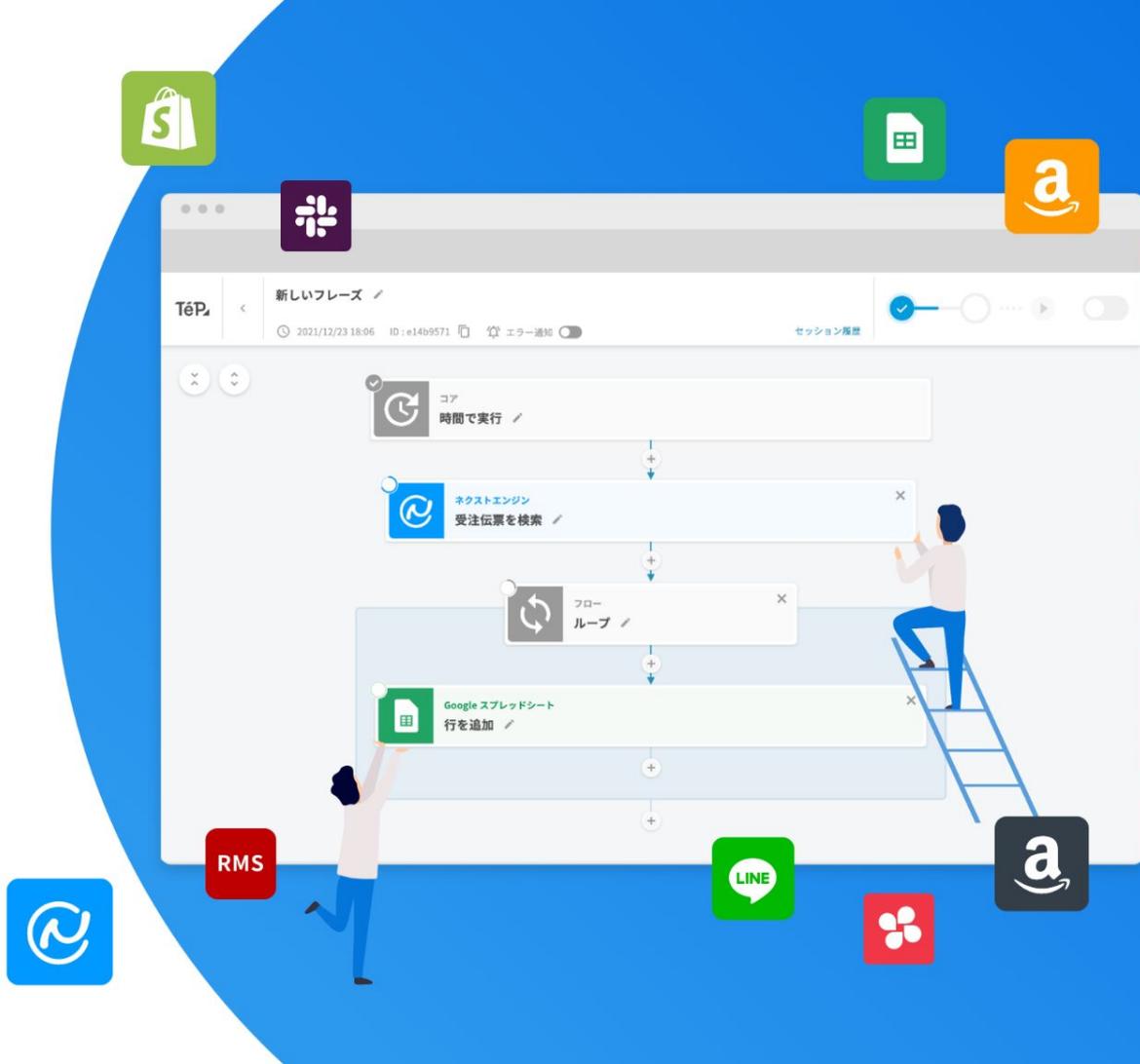
第 1 部

TēPs (テープス) の概要

TéP

ECに特化した ノーコードツール

独自の業務フローを
自動化するツールをすぐに作成



ECの運営業務でよくある悩み



システムが
うまく連携できていない

- ✔ モール間のシステム連携がされていない
- ✔ 基幹システムとの連携が大変



利用しているシステムの
機能が足りない

- ✔ カスタマイズは高額
- ✔ 適切なアプリが見つからない



販路や商流が変わるたび
新しい作業が発生する

- ✔ 都度システムの切り替えに追われる
- ✔ 頻繁にエンジニアに頼めない

「EC特化のノーコード」で、 それぞれの悩みを解決できます



プログラミングなしで
複数のシステムを連携

- ✓ モール・カート間の各種データを同期
- ✓ 各店舗の売上状況をリアルタイム管理



利用しているシステムの
機能を拡張

- ✓ 処理の条件を自由に設定
- ✓ API を介して細かなコントロール



業務フローに合わせて
新しいツールを作成

- ✓ システムを切り替えずに対応
- ✓ 必要なツールを自分でつくる

ノーコードツールとは | プログラミングの知識がなくても、画面の操作のみで誰でもWEBサービスやアプリなどをつくれるツールです。

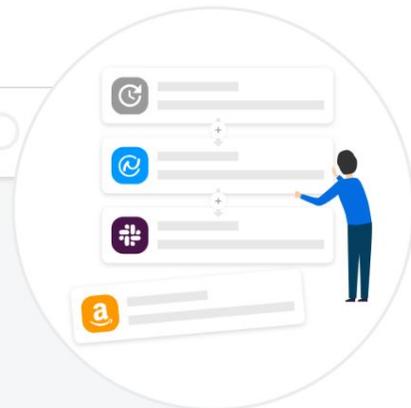
EC 運営で利用するシステムの機能を 自由に組み合わせ独自の業務フローをつくれるノーコードツールです



プログラミングの知識がなくても、 オリジナルのツールを自由につくることができます

ノート(ブロック)を自由に組み合わせることのできる業務フローは無限大

The screenshot displays the TéP workflow editor interface. At the top, the title bar reads "商品分類タグをもとに伝票を分割する" (Split invoices based on product category tags). The interface includes a header with the TéP logo, a back arrow, a timestamp "2022/04/06 16:43", an ID "ID: 42a6893e", a trash icon, a bell icon for "エラー通知" (Error notification), and a "セッション履歴" (Session history) link. The main workspace contains a vertical sequence of blocks. The first block is a "コア | 時間で実行" (Core | Execute by time) block, set to "15分ごとに実行" (Execute every 15 minutes). Below it is a "ネクストエンジン" (Next Engine) block titled "受注伝票を検索" (Search for order invoices). This block has a "メモを残す" (Save memo) option and a "コネクション" (Connection) dropdown menu with the text "選択してください。" (Please select). Below the connection menu are sections for "伝票の検索条件" (Invoice search conditions) and "明細行の検索条件" (Detail row search conditions). The search conditions section includes a red error message "1項目以上設定してください。" (Please set at least 1 item) and a "+ 項目を追加" (Add item) button.



接続しているサービス

2022年7月22日時点の情報。順次拡大予定。



Amazon
セラーセントラル



Amazon
バンダーセントラル



楽天市場
RMS



Yahoo!ショッピング



Shopify



ネクストエンジン



Google
スプレッドシート



Chatwork



Slack



LINE



Google Map



Mailgun



Sendgrid



Dropbox



ノート一覧 (できること)

 レポートドキュメントを取得	 注文を検索 (β)	 注文をフルフィルメント	 伝票を更新	 行を追加
 出品中商品の価格情報を取得	 POを検索	 注文を更新	 配送情報出力ステータス更新	 ワークシートを更新
 商品のカート獲得価格を取得	 商品の在庫情報の取得	 商品の在庫数を更新	 伝票を出荷済みに更新	 ワークシートをクリア
 出荷レポートを取得	 商品の在庫情報を更新	 商品の在庫情報を更新	 在庫の入出庫を更新	 ワークシートの内容を CSV ファイルとして取得
 トランザクションステータスの取得	 受注を検索	 商品の在庫情報を取得	 受注 CSV ファイルをアップロード	 Amazon 商品レビューを取得
 在庫フィードの更新	 商品情報を更新	 商品データをアップロード	 伝票の明細を更新	 ファイルをアップロード
 注文情報の取得	 発送完了報告	 すべての未反映項目の反映をリクエスト	 伝票分割	 通知を送信
 注文の納品書を取得	 配送方法を更新	 注文を検索	 伝票に明細行を追加	 Chatwork にメッセージを送信
 注文の発送ステータスを更新	 サブステータスを更新	 受注伝票を検索	 セット商品を含む伝票の不整合を解消	 Slack にメッセージを送信
 注文の送り状 (配送ラベル) の生成をリクエスト	 注文を検索	 商品マスタを取得	 アップロードキューの状態を取得	 メールを送信する
 注文の送り状 (配送ラベル) を取得	 注文をキャンセル	 在庫を取得	 行を検索	 ジオコーディング

約 70 種類のテンプレートで作成もかんたん

テンプレート
画面

TéP

テンプレート

テンプレート プレイグラウンド

Shopify | 注文情報を取得し、Google スプレッドシートに書き出す

Amazon バンダーセントラル | ダイレクトフルフィルメント注文の納品書を取得し、Dropbox にアップロードする

楽天市場 RMS | Google スプレッドシートをもとに在庫数と「あす楽設定」を更新する

Yahoo!ショッピング | Google スプレッドシートに書き出す

Yahoo!ショッピング | 在庫情報を取得

Amazon バンダーセントラル | ダイレクトフルフィルメント注文の納品書を取得し、Dropbox にアップロードする

テンプレートは最低限の入力ですぐにつかえる

The screenshot shows a workflow automation interface with the following steps:

- コア | 時間で実行**
15分ごとに実行
- Google スプレッドシート | 行を検索**
Google スプレッドシート上で「在庫不足」と記載されている行を取得
- ユーティリティ | エコーを作成**
通知文面に挿入するための空のエコーを作成
- フロー | ループ**
行ごとに繰り返す
- ユーティリティ | エコーを変更**
在庫不足の商品があるたびに空のエコーに追記
- LINE Notify | 通知を送信**
在庫不足の商品を通知

テンプレート
詳細画面

デモンストレーション

デモンストレーション

- TēPs の基本操作説明
- Shopify 注文を Google スプレッドシートに書き出し
- Shopify の注文をフルフィルメント

Shopify Flow との違い

- 時間をトリガーにして実行
- すでに存在するデータの利用が可能
- 1つのアプリで複数の Shopify アカウントで利用可能
- 国内の主要モール、サービスとの連携
- 日本語による分かりやすい条件設定

第 2 部

Shopify と楽天・Yahoo!ショッピング・

ネクストエンジンを連携 業務効率化 10 の事例

Shopify と楽天・Yahoo!ショッピング・ ネクストエンジンを連携 業務効率化 10 の事例

1. Shopify 注文を Google スプレッドシートに書き出し
2. Shopify 注文をフルフィルメント
3. 入金催促メールの自動送信
4. Shopify と各サービスの在庫情報を Google スプレッドシートで一元管理
5. 楽天市場と Shopify の在庫情報を連動
6. Yahoo!ショッピングと Shopify の在庫情報を連動
7. Google スプレッドシートをもとに、Shopify と各サービスの在庫情報を更新
8. Shopify 注文が支払済みになったらネクストエンジンの受注伝票に反映
9. Shopify 注文のキャンセルをネクストエンジンの受注伝票に反映
10. Shopify の e ギフト（ソーシャルギフト）注文に配送先住所が入力されたらネクストエンジンに自動で反映

Shopify と楽天・Yahoo!ショッピング・ ネクストエンジンを連携 業務効率化 10 の事例

1. Shopify 注文を Google スプレッドシートに書き出し
2. Shopify 注文をフルフィルメント
3. 入金催促メールの自動送信
4. Shopify と各サービスの在庫情報を Google スプレッドシートで一元管理
5. 楽天市場と Shopify の在庫情報を連動
6. Yahoo!ショッピングと Shopify の在庫情報を連動
7. Google スプレッドシートをもとに、Shopify と各サービスの在庫情報を更新
8. Shopify 注文が支払済みになったらネクストエンジンの受注伝票に反映
9. Shopify 注文のキャンセルをネクストエンジンの受注伝票に反映
10. Shopify の e ギフト（ソーシャルギフト）注文に配送先住所が入力されたらネクストエンジンに自動で反映

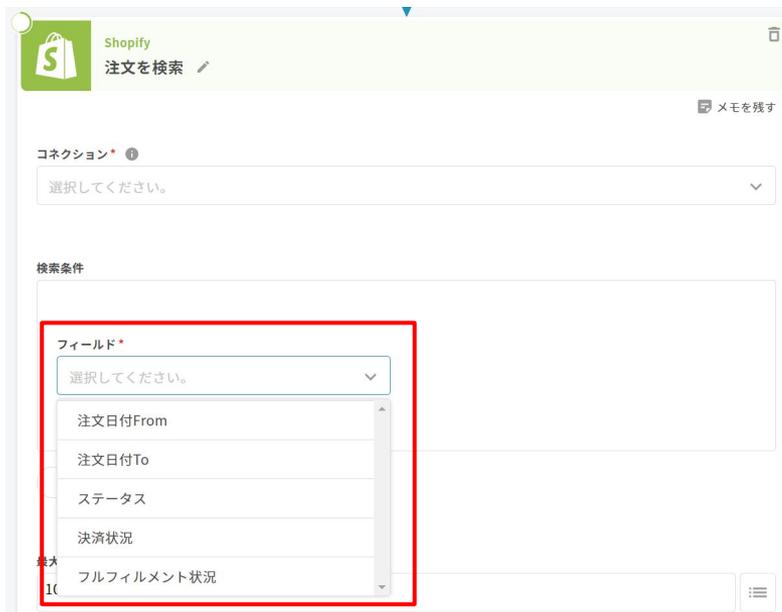
Shopify 注文を Google スプレッドシートに書き出し

- Shopify の注文の必要な項目だけを Google スプレッドシートに書き出し
- 住所や注文金額の他に、購入された商品の情報も書き出し可能

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	注文ID	注文番号	注文日	郵便番号	住所	姓	名	電話番号	決済状況	総合計・最終	プロダクトID	数量	SKU	商品名	バリエーション	バリエーション	単価 (ショ)
2	4928728924	1219	2022/09/06	253-0055	Kanagawa	タデモ	セミナー	+81467385	pending	1610							
3	4928728924	1219	2022/09/06								7668969996	10	SENBEI_SH	せんべい	429577007	しお	100
4	4928728924	1219	2022/09/06								7668969996	5	SENBEI_SH	せんべい	429577007	しょうゆ	100
5	4928728924	1219	2022/09/06								7668969996	1	SENBEI_CC	せんべい	429577008	ごま	100
6	4928728924	1219	2022/09/06								7668969996	1	SENBEI_ZA	せんべい	43060288	ざらめ	10
7	4928724992	1218	2022/09/06	253-0055	Kanagawa	タデモ	セミナー	+81467385	pending	15800							
8	4928724992	1218	2022/09/06								7561126281	1	Smartglasse	スマートグラス	4253114567		15800
9	4928720568	1217	2022/09/06	253-0055	Kanagawa	タデモ	セミナー	+81467385	pending	380							
10	4928720568	1217	2022/09/06								7746122055	1	pen_kuro	ボールペン	4323791801	黒	120
11	4928720568	1217	2022/09/06								7746122055	1	pen_aka	ボールペン	432379323	赤	140
12	4928720568	1217	2022/09/06								7746122055	1	pen_aka	ボールペン	432381546	赤被り	120

Shopify 注文を Google スプレッドシートに書き出し

- 日付やステータスを指定して、特定の注文だけに絞り込むこともできる
- 特定の SKU だけ別のワークシートに書き出すなどの設定も可能



注文を Google スプレッドシートに書き出せる接続先

- Shopify
- 楽天市場
- Yahoo!ショッピング
- Amazon ベンダーセントラル (PO、ダイレクトフルフィルメント)
- Amazon セラーセントラル (決済レポート V2、FBA 出荷レポート)
- ネクストエンジン

ブログ記事 : <https://teps.io/info/append-shopify-order-to-google-spreadsheets/>

Shopify と楽天・Yahoo!ショッピング・ ネクストエンジンを連携 業務効率化 10 の事例

1. Shopify 注文を Google スプレッドシートに書き出し
2. Shopify 注文をフルフィルメント
3. 入金催促メールの自動送信
4. Shopify と各サービスの在庫情報を Google スプレッドシートで一元管理
5. 楽天市場と Shopify の在庫情報を連動
6. Yahoo!ショッピングと Shopify の在庫情報を連動
7. Google スプレッドシートをもとに、Shopify と各サービスの在庫情報を更新
8. Shopify 注文が支払済みになったらネクストエンジンの受注伝票に反映
9. Shopify 注文のキャンセルをネクストエンジンの受注伝票に反映
10. Shopify の e ギフト（ソーシャルギフト）注文に配送先住所が入力されたらネクストエンジンに自動で反映

Shopify と楽天・Yahoo!ショッピング・ ネクストエンジンを連携 業務効率化 10 の事例

1. Shopify 注文を Google スプレッドシートに書き出し
2. Shopify 注文をフルフィルメント
3. **入金催促メールの自動送信**
4. Shopify と各サービスの在庫情報を Google スプレッドシートで一元管理
5. 楽天市場と Shopify の在庫情報を連動
6. Yahoo!ショッピングと Shopify の在庫情報を連動
7. Google スプレッドシートをもとに、Shopify と各サービスの在庫情報を更新
8. Shopify 注文が支払済みになったらネクストエンジンの受注伝票に反映
9. Shopify 注文のキャンセルをネクストエンジンの受注伝票に反映
10. Shopify の e ギフト（ソーシャルギフト）注文に配送先住所が入力されたらネクストエンジンに自動で反映

入金催促メールの自動送信

件名*

商品代金・未入金のご連絡（●●ストア）

本文*

姓 名 様

この度は、●●ストアで商品をお買い上げいただき、ありがとうございます。
商品代金の入金が確認できなかったため、ご連絡させていただきました。
以下の内容をご確認のうえ、入金をお願いいたします。

<お買い上げ商品の明細>

商品明細

<未入金の金額>

残高（お客様の未払い額） 円

<振込先情報>

XX銀行 XX支店
普通：XXXXXXXXXX

入れ違いでご入金の手続きがお済の場合は、大変失礼いたしました。
今後とも●●ストアをよろしくお願いいたします。

●●ストア
電話番号：XXX-XXX-XXXX
メール：hogehoge@sample.com

商品代金・未入金のご連絡（●●ストア） [受信トレイ](#)

K

@teps.io

To 自分

姓 名 様

この度は、●●ストアで商品をお買い上げいただき、ありがとうございます。
商品代金の入金が確認できなかったため、ご連絡させていただきました。
以下の内容をご確認のうえ、入金をお願いいたします。

<お買い上げ商品の明細>

商品名：せんべい
バリエーション：しお
数量：10
単価：100円

商品名：せんべい
バリエーション：しょうゆ
数量：5
単価：100円

商品名：せんべい
バリエーション：ごま
数量：1
単価：100円

商品名：せんべい
バリエーション：ざらめ
数量：1
単価：10円

<未入金の金額>
1610円

<振込先情報>
XX銀行 XX支店
普通：XXXXXXXXXX

入れ違いでご入金の手続きがお済の場合は、大変失礼いたしました。
今後とも●●ストアをよろしくお願いいたします。

●●ストア
電話番号：XXX-XXX-XXXX
メール：hogehoge@sample.com

入金催促メールの自動送信

- メール送信に利用できるサービスは以下の4つ

+ 新しいSMTPコネクションを追加

+ 新しいGmailコネクションを追加

+ 新しいSendGridコネクションを追加

+ 新しいMailgunコネクションを追加

- To には Shopify の注文者の連絡先メールアドレスを指定できる
- メールアドレスは直接入力もできるので、CC に店舗のメールアドレスを入れることも可能

The screenshot shows an email composition window with the following elements highlighted by red boxes:

- コネクション***: A dropdown menu with the text "選択してください。" (Please select).
- To**: A field containing a contact labeled "連絡先email" (Contact email).
- CC**: A field containing the email address "sample@teps.io".

Other visible elements include "メール" (Email), "メールを送信する" (Send email), "メモを残す" (Save memo), and "項目を追加" (Add item) buttons for each recipient field.

入金催促メールの自動送信

- 入金催促メールを送る条件を細かく設定可能
- たとえば、
 - ・ 決済状況：未払い
 - ・ 入金催促メール未送信
 - ・ 注文から 7 日以上経過
- 入金催促メールを送った注文にタグを付与し、何度もメール送信することを防止

The screenshot displays the Shopify admin interface for configuring automatic email sending. The top navigation bar includes the Shopify logo, the text "Shopify 注文を検索", and a "メモを残す" button. Below this, a "コネクション" dropdown menu is set to "Shopify".

The main section is titled "検索条件" (Search Conditions) and is enclosed in a red border. It contains three rows of conditions, each with a "フィールド" (Field) and a "値" (Value) dropdown menu, and a "かつ" (AND) connector between rows.

- Row 1: Field "決済状況" (Payment Status) with value "未払い" (Not Paid).
- Row 2: Field "タグ (含まない)" (Tag (Exclude)) with value "入金催促メール送信済み" (Payment Reminder Email Sent).
- Row 3: Field "注文日付To" (Order Date To) with value "実行時に計算" (Calculate at Execution Time).

At the bottom, there is a "基準とする日時" (Date and Time to Use as Reference) section. It includes a "現在時刻" (Current Time) button, a "よりも" (More) button, and a date selector set to "7" days, "前" (Previous).

入金催促メールの自動送信が可能な接続先

- Shopify
- 楽天市場
- Yahoo!ショッピング
- ネクストエンジン

ブログ記事：<https://teps.io/info/send-payment-reminder-email-on-shopify/>

Shopify と楽天・Yahoo!ショッピング・ ネクストエンジンを連携 業務効率化 10 の事例

1. Shopify 注文を Google スプレッドシートに書き出し
2. Shopify 注文をフルフィルメント
3. 入金催促メールの自動送信
4. Shopify と各サービスの在庫情報を Google スプレッドシートで一元管理
5. 楽天市場と Shopify の在庫情報を連動
6. Yahoo!ショッピングと Shopify の在庫情報を連動
7. Google スプレッドシートをもとに、Shopify と各サービスの在庫情報を更新
8. Shopify 注文が支払済みになったらネクストエンジンの受注伝票に反映
9. Shopify 注文のキャンセルをネクストエンジンの受注伝票に反映
10. Shopify の e ギフト（ソーシャルギフト）注文に配送先住所が入力されたらネクストエンジンに自動で反映

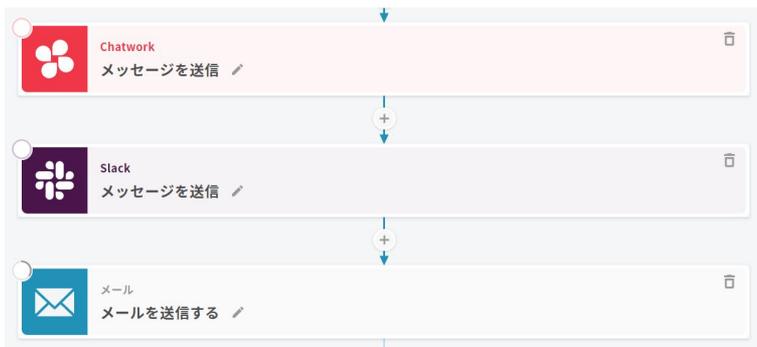
Shopify と各サービスの在庫情報を Google スプレッドシートで一元管理

- Shopify、楽天市場、Yahoo!ショッピングなどの最新の在庫数を Google スプレッドシートに書き出し、在庫の一元管理が可能
- 商品ごとに発注が必要な在庫数のしきい値を決めておき、発注フラグをたてることもできます

	A	B	C	D	E	F	G	H	I	J
1	商品コード・SKU	ロケーションID	Shopify	楽天市場	Yahoo!ショッピング	FBA	ネクストエンジン	合計	しきい値	発注フラグ
2	sample-001	111111111111	10	50	20	30	100	210	500	発注必要
3	sample-002	222222222222	20	100	40	60	200	420	500	発注必要
4	sample-003	333333333333	30	150	60	90	300	630	500	
5	sample-004	444444444444	40	200	80	120	400	840	500	
6	sample-005	555555555555	50	250	100	150	500	1050	500	

在庫数がしきい値を下回ったら メールやチャットツールに自動で通知

H	I	J
合計	しきい値	発注フラグ
210	500	発注必要
420	500	発注必要
630	500	
840	500	
1050	500	



在庫情報が取得可能な接続先

- [Shopify](#)
- [楽天市場](#)
- [Yahoo!ショッピング](#)
- [ネクストエンジン](#)
- [Amazon セラーセントラル \(FBA 在庫管理レポート\)](#)
- [オープンロジ](#)

Shopify と楽天・Yahoo!ショッピング・ ネクストエンジンを連携 業務効率化 10 の事例

1. Shopify 注文を Google スプレッドシートに書き出し
2. Shopify 注文をフルフィルメント
3. 入金催促メールの自動送信
4. Shopify と各サービスの在庫情報を Google スプレッドシートで一元管理
5. **楽天市場と Shopify の在庫情報を連動**
6. Yahoo!ショッピングと Shopify の在庫情報を連動
7. Google スプレッドシートをもとに、Shopify と各サービスの在庫情報を更新
8. Shopify 注文が支払済みになったらネクストエンジンの受注伝票に反映
9. Shopify 注文のキャンセルをネクストエンジンの受注伝票に反映
10. Shopify の e ギフト（ソーシャルギフト）注文に配送先住所が入力されたらネクストエンジンに自動で反映

楽天市場と Shopify の在庫情報を同期

- Shopify で新規登録した商品は、楽天市場と在庫が同期される
- 楽天市場に登録済みの商品は Shopify と在庫情報が同期されない
- TēPs を利用することで、15 分間隔で在庫の同期が可能

ブログ記事

<https://teps.io/info/sync-shopify-inventory-from-rakuten-rms/>

	A	B	C	D	E
1	楽天市場 RMS	横軸選択枝子番	縦軸選択枝子番	Shopify SKU	ロケーション
2	r-test-001			s-test-001	11111111111
3	r-test-002	s	wh	s-test-002-s-wh	22222222222
4	r-test-002	m	bk	s-test-002-m-bk	33333333333
5	r-test-002	l	pk	s-test-002-l-pk	44444444444
6					



Shopify と楽天・Yahoo!ショッピング・ ネクストエンジンを連携 業務効率化 10 の事例

1. Shopify 注文を Google スプレッドシートに書き出し
2. Shopify 注文をフルフィルメント
3. 入金催促メールの自動送信
4. Shopify と各サービスの在庫情報を Google スプレッドシートで一元管理
5. 楽天市場と Shopify の在庫情報を連動
6. **Yahoo!ショッピングと Shopify の在庫情報を連動**
7. Google スプレッドシートをもとに、Shopify と各サービスの在庫情報を更新
8. Shopify 注文が支払済みになったらネクストエンジンの受注伝票に反映
9. Shopify 注文のキャンセルをネクストエンジンの受注伝票に反映
10. Shopify の e ギフト（ソーシャルギフト）注文に配送先住所が入力されたらネクストエンジンに自動で反映

Yahoo!ショッピングと Shopify の在庫情報を同期

- Yahoo!ショッピングと Shopify の在庫を同期するには OMS などのツール導入が必要
- 楽天市場と同様に TēPs と使うと 15 分間隔で在庫の同期が可能

	A	B	C	D
1	Yahoo!ショッピ	個別商品コード	Shopify SKU	ロケーション
2	r-test-001		s-test-001	11111111111
3	r-test-002	s-wh	s-test-002-s-wh	22222222222
4	r-test-002	m-bk	s-test-002-m-bk	33333333333
5	r-test-002	l-pk	s-test-002-l-pk	44444444444



Shopify と楽天・Yahoo!ショッピング・ ネクストエンジンを連携 業務効率化 10 の事例

1. Shopify 注文を Google スプレッドシートに書き出し
2. Shopify 注文をフルフィルメント
3. 入金催促メールの自動送信
4. Shopify と各サービスの在庫情報を Google スプレッドシートで一元管理
5. 楽天市場と Shopify の在庫情報を連動
6. Yahoo!ショッピングと Shopify の在庫情報を連動
7. Google スプレッドシートをもとに、Shopify と各サービスの在庫情報を更新
8. Shopify 注文が支払済みになったらネクストエンジンの受注伝票に反映
9. Shopify 注文のキャンセルをネクストエンジンの受注伝票に反映
10. Shopify の e ギフト（ソーシャルギフト）注文に配送先住所が入力されたらネクストエンジンに自動で反映

Google スプレッドシートをもとに Shopify と各サービスの在庫情報を更新

- Shopify、楽天市場、Yahoo!ショッピングなどの在庫数を Google スプレッドシートをもとに更新が可能
- あらかじめ各モール・カートへの在庫の配分率を決めておけば、Google スプレッドシートに最新の在庫数を記入するだけで自動更新ができる

商品コード・SKU	ロケーションID	Shopify	楽天市場	Yahoo!ショッピング	最新の在庫数
sample-001	11111111111	50	300	150	500
sample-002	22222222222	100	600	300	1000
sample-003	33333333333	150	900	450	1500
sample-004	44444444444	200	1200	600	2000
sample-005	55555555555	250	1500	750	2500

在庫情報が更新可能な接続先

- [Shopify](#)
- [楽天市場](#)
- [Yahoo!ショッピング](#)
- [ネクストエンジン](#)
- [Amazon ベンダーセントラル | ダイレクトフルフィルメント](#)

Shopify と楽天・Yahoo!ショッピング・ ネクストエンジンを連携 業務効率化 10 の事例

1. Shopify 注文を Google スプレッドシートに書き出し
2. Shopify 注文をフルフィルメント
3. 入金催促メールの自動送信
4. Shopify と各サービスの在庫情報を Google スプレッドシートで一元管理
5. 楽天市場と Shopify の在庫情報を連動
6. Yahoo!ショッピングと Shopify の在庫情報を連動
7. Google スプレッドシートをもとに、Shopify と各サービスの在庫情報を更新
8. Shopify 注文が支払済みになったらネクストエンジンの受注伝票に反映
9. Shopify 注文のキャンセルをネクストエンジンの受注伝票に反映
10. Shopify の e ギフト（ソーシャルギフト）注文に配送先住所が入力されたらネクストエンジンに自動で反映

Shopify 注文はネクストエンジンに取り込まれると その後の変更が同期されない

- 銀行振り込みなどの Shopify 注文で、ステータスを「支払い済み」にしてもネクストエンジンに反映されない
- Shopify 注文をキャンセルしても、ネクストエンジンの受注伝票がキャンセルされない
- Shopify で e ギフト（ソーシャルギフト）アプリを利用した場合、入力された配送先住所がネクストエンジンに反映されない

TēPs を利用することで、これらを全て自動化できます！

Shopify と楽天・Yahoo!ショッピング・ ネクストエンジンを連携 業務効率化 10 の事例

1. Shopify 注文を Google スプレッドシートに書き出し
2. Shopify 注文をフルフィルメント
3. 入金催促メールの自動送信
4. Shopify と各サービスの在庫情報を Google スプレッドシートで一元管理
5. 楽天市場と Shopify の在庫情報を連動
6. Yahoo!ショッピングと Shopify の在庫情報を連動
7. Google スプレッドシートをもとに、Shopify と各サービスの在庫情報を更新
8. **Shopify 注文が支払済みになったらネクストエンジンの受注伝票に反映**
9. Shopify 注文のキャンセルをネクストエンジンの受注伝票に反映
10. **Shopify の e ギフト（ソーシャルギフト）注文に配送先住所が入力されたらネクストエンジンに自動で反映**

Shopify 注文が支払済みになったら ネクストエンジンの受注伝票に反映

検索条件

フィールド*	値*
決済状況	支払い済み

更新内容 (基本項目)

フィールド*	値*
入金状況区分	2: 入金済み



Shopify と楽天・Yahoo!ショッピング・ ネクストエンジンを連携 業務効率化 10 の事例

1. Shopify 注文を Google スプレッドシートに書き出し
2. Shopify 注文をフルフィルメント
3. 入金催促メールの自動送信
4. Shopify と各サービスの在庫情報を Google スプレッドシートで一元管理
5. 楽天市場と Shopify の在庫情報を連動
6. Yahoo!ショッピングと Shopify の在庫情報を連動
7. Google スプレッドシートをもとに、Shopify と各サービスの在庫情報を更新
8. Shopify 注文が支払済みになったらネクストエンジンの受注伝票に反映
9. **Shopify 注文のキャンセルをネクストエンジンの受注伝票に反映**
10. Shopify の e ギフト（ソーシャルギフト）注文に配送先住所が入力されたらネクストエンジンに自動で反映

Shopify 注文がキャンセルされたら ネクストエンジンの受注伝票をキャンセルする

検索条件

フィールド*	値*
ステータス	キャンセル済み

更新内容 (基本項目)

フィールド*	値*
受注キャンセル区分	1: 顧客依頼によりキャンセル



Google スプレッドシートをもとに Shopify 注文をキャンセル

	A	B	C
1	オーダーID	キャンセルの理由	キャンセル処理
2	4854580379860	customer	済
3	4854579953876	fraud	済
4	4854579134676	inventory	
5	4853291745492	declined	
6	4853238726868	other	
7			
8			

Shopify
注文をキャンセル

メモを残す

コネクション *

選択してください。

注文 ID *

支払済み金額を返金する *

はい

アイテムの在庫を補充する *

はい

キャンセルの理由 *

選択してください。

お客様に通知を送信する *

はい

Google フォームを使って Shopify 注文のキャンセル処理を自動化

- 注文のキャンセルを Google フォームで受け付け、Google スプレッドシートに自動で反映することができる
- これにより、キャンセル処理の自動化が可能
- 注文から 〇 日以内の場合だけキャンセルし、それ以外の場合はキャンセルできないことをメールで連絡することもできる

ブログ記事： <https://teps.io/info/cancel-shopify-order-requested-via-cancellation-form/>

注文キャンセル申請フォーム

kenichiro.tsutsumi@teps.io (共有なし)
アカウントを切り替える

*必須

注文ID *
回答を入力

注文番号 *
回答を入力

注文日 *
日付
年 / 月 / 日

送信

	A	B	C	D	E
1	タイムスタンプ	注文ID	注文番号	注文日	キャンセル処理
2	2022/09/09 15:48:20	9999999999	#9999	2022/09/09	
3	2022/09/09 15:49:25	8888888888	#8888	2022/08/15	
4	2022/09/09 15:50:08	7777777777	#7777	2022/08/31	
5					

Liquid を活用することで、 お客様が必要な情報を入力する手間を省くことも可能

- Shopify の自動返信メールに Google フォームの URL を埋め込む際に、Liquid で注文 ID や注文番号が事前に入力されたフォームの表示が可能
- お客様の入力の手間を省き、入力ミスによる確認のやりとりを削減することができる

https://docs.google.com/forms/d/e/フォームのID/viewform?usp=pp_url&entry.664697793=9999999999999999 注文 ID
&entry.128361470=9999 注文番号
&entry.1610311798=2022-08-01 日付

https://docs.google.com/forms/d/e/フォームのID/viewform?usp=pp_url&entry.664697793={{ id }}&entry.128361470={{ number }}&entry.1610311798={{ created_at|date:'%Y-%m-%d' }}

注文キャンセル申請フォーム

kenichiro.tsutsumi@teps.io (共有なし)
アカウントを切り替える

*必須

注文ID *
9999999999999999

注文番号 *
9999

注文日 *
日付
2022/08/01

送信

フォームをクリア

Shopify と楽天・Yahoo!ショッピング・ ネクストエンジンを連携 業務効率化 10 の事例

1. Shopify 注文を Google スプレッドシートに書き出し
2. Shopify 注文をフルフィルメント
3. 入金催促メールの自動送信
4. Shopify と各サービスの在庫情報を Google スプレッドシートで一元管理
5. 楽天市場と Shopify の在庫情報を連動
6. Yahoo!ショッピングと Shopify の在庫情報を連動
7. Google スプレッドシートをもとに、Shopify と各サービスの在庫情報を更新
8. Shopify 注文が支払済みになったらネクストエンジンの受注伝票に反映
9. Shopify 注文のキャンセルをネクストエンジンの受注伝票に反映
10. Shopify の e ギフト（ソーシャルギフト）注文に配送先住所が入力されたらネクストエンジンに自動で反映

Shopify の e ギフト（ソーシャルギフト）注文に 配送先住所が入力されたらネクストエンジンに自動で反映

- Shopify の配送先住所が入力されたらネクストエンジンに自動で反映
- 同時に「確認チェック区分」に確認チェックをつけて、ステータスを印刷待ちに移動することも可能

ブログ記事

<https://teps.io/info/update-next-engine-shipping-address-from-shopify-egift-order/>

更新内容（基本項目）

フィールド*	値*
確認チェック区分	1: 確認チェックをつける

ネクストエンジン
受注伝票を更新

メモを残す

コネクション*
選択してください。

伝票番号*
伝票番号

更新内容（基本項目）

フィールド*	値*
送り先住所1	都道府県（漢字） 市区町村
送り先住所2	住所1 住所2
送り先郵便番号	郵便番号変換