

# 在庫管理（自動在庫登録～自動発注） リアルタイム実行チュートリアル

～ krewData ～

# テーマ



# 在庫管理の業務改善

**自動在庫登録** 出荷管理アプリの出荷数を在庫管理アプリの在庫数に自動的に反映



## 出荷管理アプリ

レコード更新  
(ユーザー操作)

ステータス	品目名	受注日	受注数	出荷日	出荷数
出荷	品目A	2020/4/20	200	2020/4/25	200



在庫レコード  
自動追加



## 在庫管理アプリ

入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数
2020/4/7	品目A	180			180	180
2020/4/15	品目A	270			450	450
2020/4/20	品目A		200		250	450
2020/4/25	品目A			200	250	250

# crewDataの基本





# krewDataとは

krewDataは、複数のkintoneアプリのデータを集計・加工し、新しいkintoneアプリを作成したり、既存アプリのデータを更新することができるプラグインです。

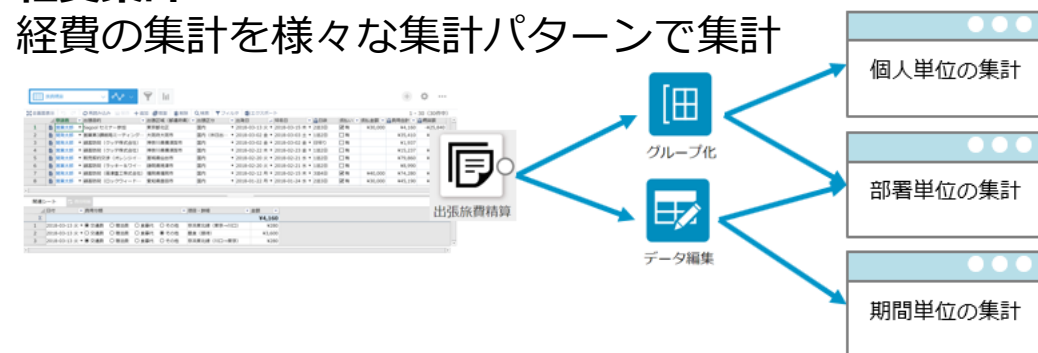
## 予実集計

データ構造の異なるアプリを結合／集計



## 経費集計

経費の集計を様々な集計パターンで集計



## 工数集計

作業工数を様々な集計単位で集計



## 費用の配賦額計算

各部の売上額をもとに複雑な配賦計算を実現



# 実行方法が異なる 2つのライセンスプラン

## スケジュール実行プラン

特定の日時にスケジュールしたタイミングでレコードを集計できます。

### スケジュール



スケジュール設定

- ・毎日・毎週・毎月・毎年
- ・時刻
- etc

スケジュール #1

パターン: 毎月

日: 31 ☐ 月の最終日を設定

曜日: 最終 ☐ 月曜日 ☐

時刻: 12:00



スケジュールされた  
日時になったら

自動的に実行



## 今回紹介

## リアルタイム実行プラン

kintoneの[Webhook](#)またはAPIを通した任意のタイミングでレコードを集計できます。

### Webhook

レコードの追加  
レコードの編集  
レコードの削除  
コメントの追加  
ステータスの更新  
(プロセス管理)



Webhookによる  
通知

通知を受信した  
タイミングで実行



### API

発注依頼  
ボタンClick!

分類 商品名  
文房具 4B鉛筆

※入出荷数：出荷の場合は「-」をつけて入力してください。

在庫確認・入荷日	入出荷数	備考
2020-08-21	30 入荷	

現在の在庫数  
30



APIで呼び出し

呼び出しを受信した  
タイミングで実行



\*1 いずれのプランでもプラグイン設定画面からの手動による即時実行は可能です。

\*2 リアルタイム実行プランではWebhookとAPI実行の両方を利用可能です。

\*3 API呼び出しの利用はJavaScriptによるカスタマイズが必要になります。

# krewDataリアルタイム実行の構成

krewDataを動作させるアプリはアプリテンプレートとして提供しています。データの集計・加工方法を定義するデータ編集フローをアプリ設定のプラグイン設定画面で行います。



## プラグイン設定画面

ドラッグ&ドロップで集計フローを作成



アプリ管理者 ([アプリ設定] - [プラグイン])



リアルタイム実行のログは別アプリで管理

実行結果（ログ）を表示

レコード番号	実行単位ID	実行単位	実行方法	実行ユーザー	アプリID	実行結果	開始日時	終了日時	ログメッセージ
2	1601359955611					Success	2020-10-02 15:31	2020-10-02 15:31	
1	1601359955611					Success	2020-10-02 13:20	2020-10-02 13:20	

アプリの一覧画面

このチュートリアルでは、krewDataのリアルタイム実行を利用した業務改善例を紹介します。



**实行单位**

フロ-2

- ① **アプリを操作**
- ・レコード追加
  - ・レコード編集
  - ・レコード削除
  - ・ステータス変更

## ②Webhook通知 任意実行API

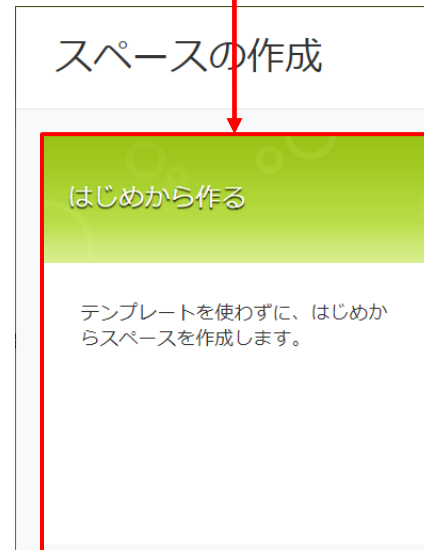
### ③実行単位の実データ編集フローを実行

# 事前準備

1. チュートリアルを配置する用スペースの作成
2. サンプルアプリの作成
3. 出荷管理サンプルデータの読み込み
4. 在庫管理サンプルデータの読み込み
5. 品目管理サンプルデータの読み込み

# 1. チュートリアルを配置するスペースの作成

チュートリアルを配置するスペースを作成します。



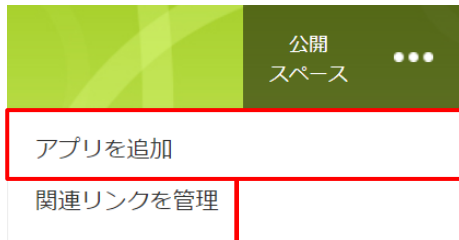
「はじめて作る」を選択



任意のスペース名を入力し、[保存]をクリックします。

## 2. サンプルアプリの作成

チュートリアルで使用するサンプルアプリを作成します。スペースのメニューから「アプリを追加」を選択



「テンプレートファイルを読み込んで作成」を選択



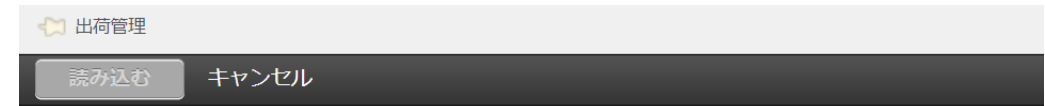
「参照」ボタンをクリックして、このファイルに同封している「krewData リアルタイム実行チュートリアルアプリテンプレート.zip」ファイルを選択し、「アプリを作成」をクリック

### 3. 出荷管理サンプルデータの読み込み

追加した出荷管理アプリを開く



「ファイルから読み込む」を選択



1 記載形式に従ってファイルのデータを整形し、読み込みます。(.xlsx / .csv)

フィールドの種類ごとに読み込める記載形式 (kintone ユーザーヘルプ)

参照 (Excelブック形式：最大1MB、1,000行まで CSV形式：最大100 MB、10万行まで)

[参照] をクリックし、このファイルに同封している**出荷管理.csv** ファイルを選択



1 記載形式に従ってファイルのデータを整形し、読み込みます。(.xlsx / .csv)

フィールドの種類ごとに読み込める記載形式 (kintone ユーザーヘルプ)

出荷管理.csv 100 bytes

参照 (Excelブック形式：最大1MB、1,000行まで CSV形式：最大100 MB、10万行まで)

ファイルを読み込んでアプリに反映されたら完了



## 4. 出荷管理レコードのステータス変更

サンプルデータのステータスを変更します。  
一覧画面から受注日が「2020-04-01」のレコード  
の詳細画面を表示

スペース: krewData リアルタイム実行チュートリアル アプリ: 出荷管理

一覧

1 - 3 (3件中)

	ステータス	品目名	受注日	受注数	出荷日	出荷数	
	未処理	品目A	2020-04-01	50			 
	未処理	品目A	2020-04-02	100	2020-04-02	100	 
	未処理	品目A	2020-04-20	200			 

1 - 3 (3件中)

引当 出荷 現在の作業者を変更

次のステータス

引当

品目名

品目A

受注日

2020-04-01

主数

50

キャンセル 実行

ステータス: 引当 ステータスの履歴

次のステータスに「引当」を選択し「実行」ボタンをクリック

同様に一覧画面から受注日が「2020-04-02」のレコードの詳細画面を表示

スペース: krewData リアルタイム実行チュートリアル アプリ: 出荷管理

一覧

1 - 3 (3件中)

	ステータス	品目名	受注日	受注数	出荷日	出荷数	
	引当	品目A	2020-04-01	50			 
	未処理	品目A	2020-04-02	100	2020-04-02	100	 
	未処理	品目A	2020-04-20	200			 

1 - 3 (3件中)

引当 出荷 現在の作業者を変更

次のステータス

出荷

品目名

品目A

受注日

2020-04-02

主数

100

キャンセル 実行

ステータス: 出荷 ステータスの履歴

次のステータスに「出荷」を選択し「実行」ボタンをクリック

## 5. 在庫管理サンプルデータの読み込み

追加した在庫管理アプリを開く



「ファイルから読み込む」を選択



1 記載形式に従ってファイルのデータを整形し、読み込みます。(.xlsx / .csv)

フィールドの種類ごとに読み込める記載形式 (kintone ユーザーヘルプ)

参照 (Excelブック形式：最大1MB、1,000行まで CSV形式：最大100 MB、10万行まで)

[参照] をクリックし、このファイルに同封している**在庫管理.csv** ファイルを選択



1 記載形式に従ってファイルのデータを整形し、読み込みます。(.xlsx / .csv)

フィールドの種類ごとに読み込める記載形式 (kintone ユーザーヘルプ)

在庫管理.csv

203 bytes

参照 (Excelブック形式：最大1MB、1,000行まで CSV形式：最大100 MB、10万行まで)

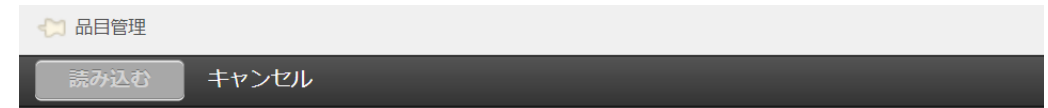
ファイルを読み込んでアプリに反映されたら完了

## 6. 品目管理サンプルデータの読み込み

追加した品目管理アプリを開く



「ファイルから読み込む」を選択



1 記載形式に従ってファイルのデータを整形し、読み込みます。(.xlsx / .csv)

フィールドの種類ごとに読み込める記載形式 (kintone ユーザーヘルプ)

**参照** (Excelブック形式：最大1MB、1,000行まで CSV形式：最大100 MB、10万行まで)

[参照] をクリックし、このファイルに同封している**品目管理.csv** ファイルを選択



1 記載形式に従ってファイルのデータを整形し、読み込みます。(.xlsx / .csv)

フィールドの種類ごとに読み込める記載形式 (kintone ユーザーヘルプ)

品目管理.csv 60 bytes

**参照** (Excelブック形式：最大1MB、1,000行まで CSV形式：最大100 MB、10万行まで)

ファイルを読み込んでアプリに反映されたら完了

# ご利用開始の準備



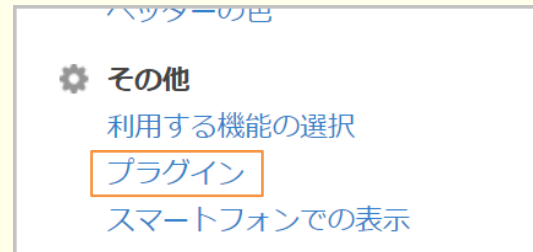
# ご利用開始前の準備

## kintoneにプラグインをインストール

各kintone環境で 1 回だけ実施  
(システム管理者)

<https://help.cybozu.com/ja/k/admin/plugin.html>

kintoneシステム管理

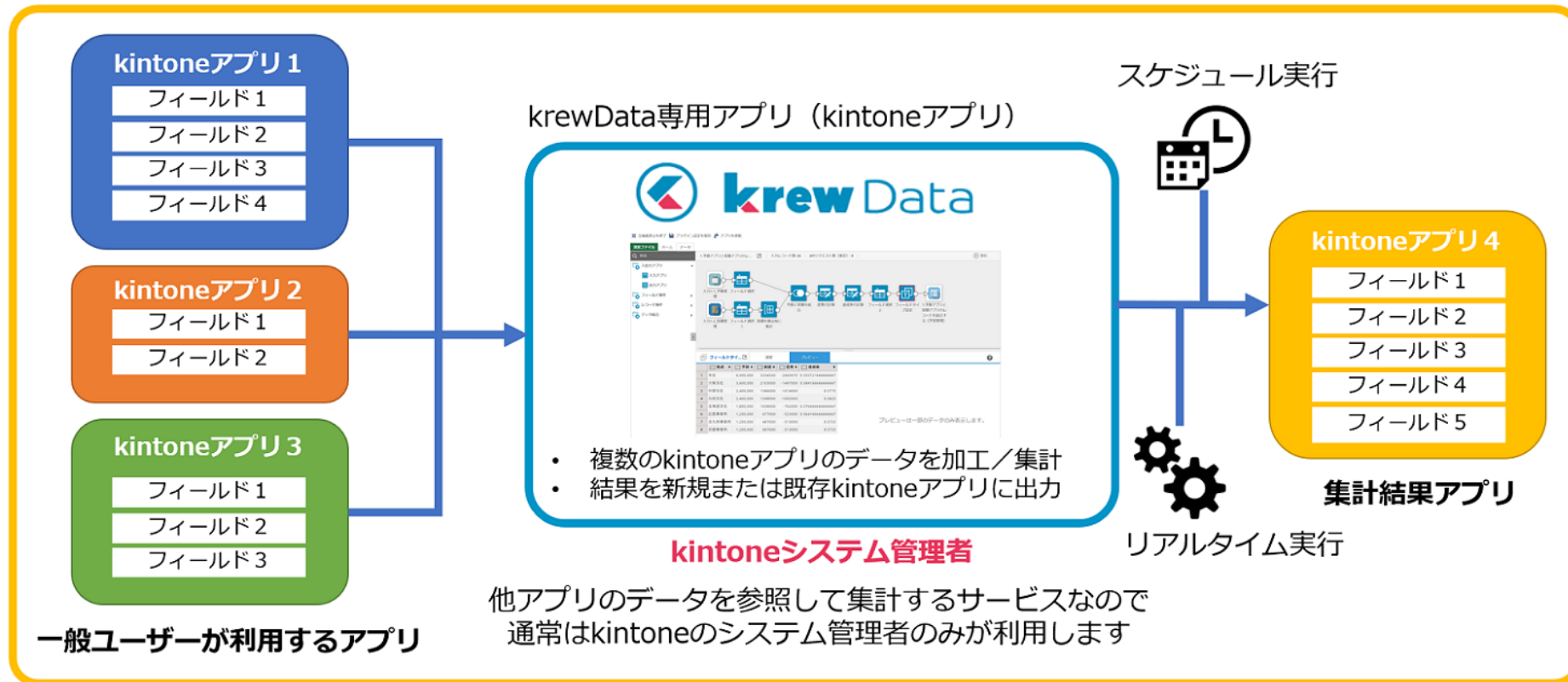


# crewData専用アプリの作成



# krewData専用アプリ

krewDataは、krewData専用アプリにkrewDataプラグインを追加して利用します。



# krewData専用アプリの作成

これらの操作は**アプリ管理者**が各アプリごとに行います。

## 専用アプリを使う理由

- ログをkintoneのレコードとして登録可能にする（必要なフィールド設定済）
- リマインド通知（実行後に通知されるように設定済）
- krewDataの全ての処理を一元管理する

(\*) krewData専用アプリには、スケジュール実行されたデータ編集フローの実行ログが登録されます。本チュートリアルで使用するリアルタイム実行の実行ログは、以降の手順で作成する「ログ出力先アプリ」に登録されます。

## 作成手順（次ページ以降の手順）

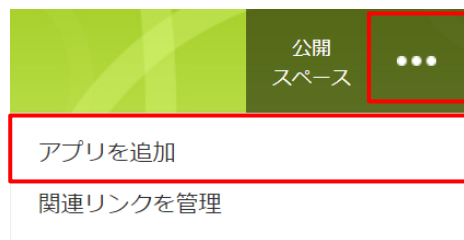
1. アプリテンプレートからkrewData専用アプリを作成
2. krewData専用アプリのアクセス権を設定
3. krewData専用アプリへのkrewDataプラグインの追加
4. krewDataプラグインの設定



# 1. krewData専用アプリの作成

アプリテンプレートからkrewData専用アプリを作成します。

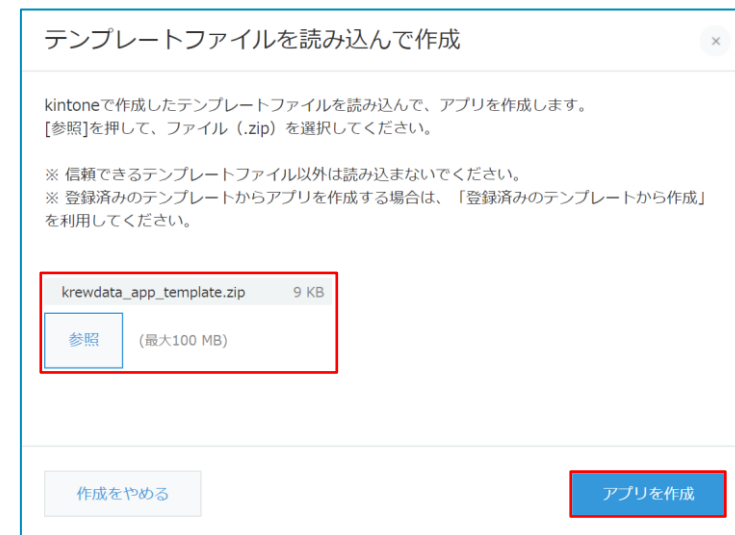
①スペースのメニューから「アプリを追加」を選択



②「テンプレートファイルを読み込んで作成」を選択



③ krewData専用アプリのアプリテンプレートファイル（krewdata\_app\_template.zip）を選択し「アプリを作成」をクリック※



※ krewData専用アプリのテンプレートファイルは下記よりダウンロードしてください  
[https://download.krew.mescius.jp/products/template/krewdata\\_app\\_template.zip](https://download.krew.mescius.jp/products/template/krewdata_app_template.zip)

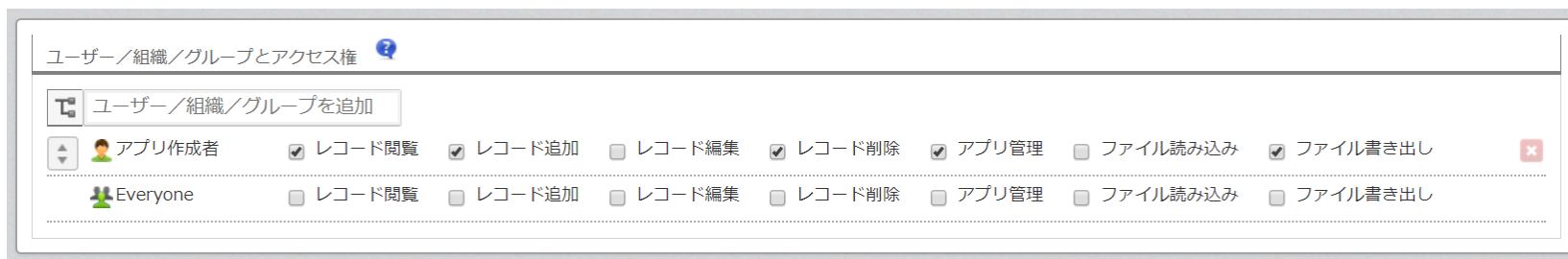
## 2. アクセス権の設定

krewData専用アプリのアクセス権を設定します。

① アプリの設定画面の「設定」タブをクリックし、「アクセス権」の「アプリ」をクリック



② アプリ管理者、Everyoneのアクセス権を以下に設定



※ krewDataは単一または複数の業務アプリを対象にデータを加工／集計できるため  
特定の管理者に限定して利用することを推奨します。

### 3. krewDataプラグインの追加

krewData専用アプリにkrewDataプラグインを追加します。

- ① アプリの設定画面の「設定」タブをクリックし、「カスタマイズ／サービス連携」の「プラグイン」をクリック



- ② 「プラグインの追加」をクリック



- ③ krewDataプラグインのチェックボックスを選択し、[追加]ボタンをクリック



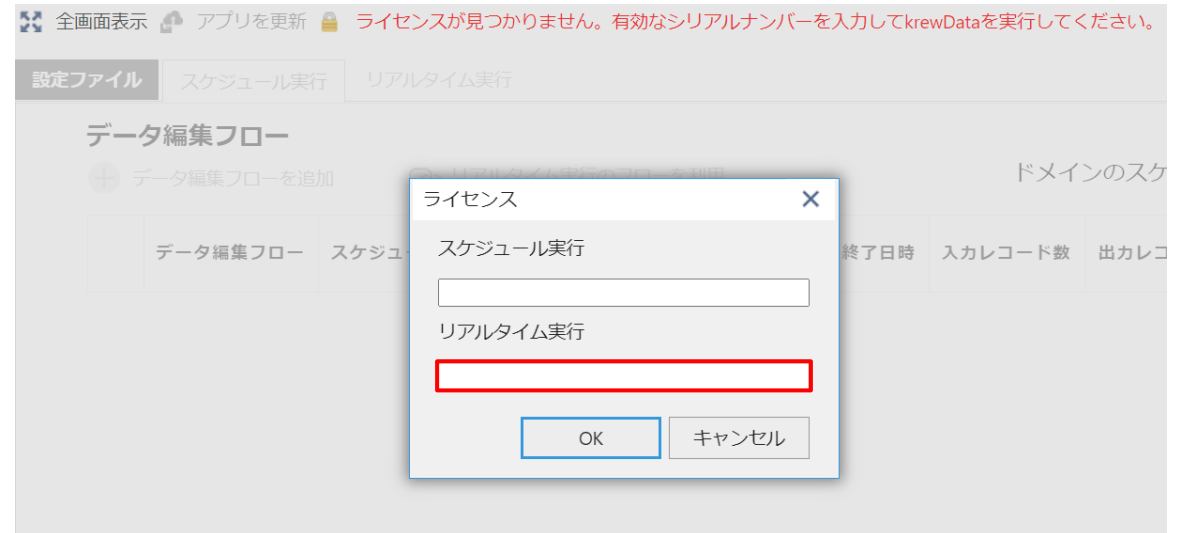
## 4. ライセンスの設定

krewDataのライセンスを設定します。

① krewDataプラグインの設定アイコンをクリック



② シリアルナンバーを入力するポップアップの「リアルタイム実行」に購入時または無料お試し申請時に入手したシリアルナンバーを入力し [OK] をクリック



以上でkrewDataを利用する準備が整いました。

# 完成イメージの確認



# データの集計・加工イメージ

本チュートリアルでは、以下のデータ集計・加工を行うデータ編集フローを作成します。



出荷管理アプリ

ステータス変更  
(ユーザー操作)

ステータス	品目名	受注日	受注数	出荷日	出荷数
出荷	品目A	2020/4/20	200	2020/4/25	200



在庫管理アプリ

入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数
2020/4/7	品目A	180			180	180
2020/4/15	品目A	270			450	450
2020/4/20	品目A		200		250	450
2020/4/25	品目A			200	250	250

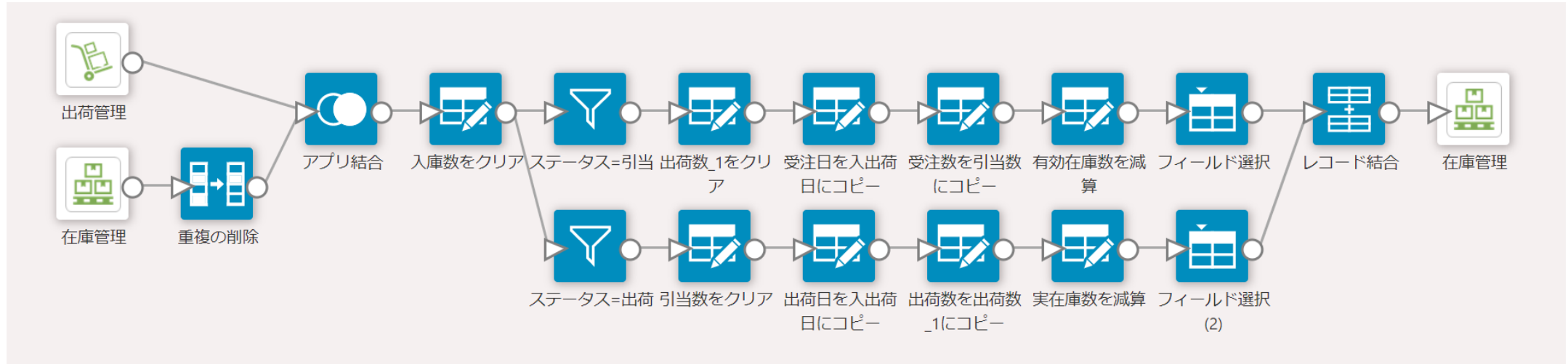
在庫レコード追加

- ステータス変更 → 在庫レコード追加

出荷管理アプリのレコードのステータスが「引当」、または「出荷」になった場合、在庫管理アプリのデータを集計して最新の在庫レコードを追加

# 作成するデータ編集フロー（完成イメージ）

ステータス変更 → 在庫レコード追加



# 実行単位の作成

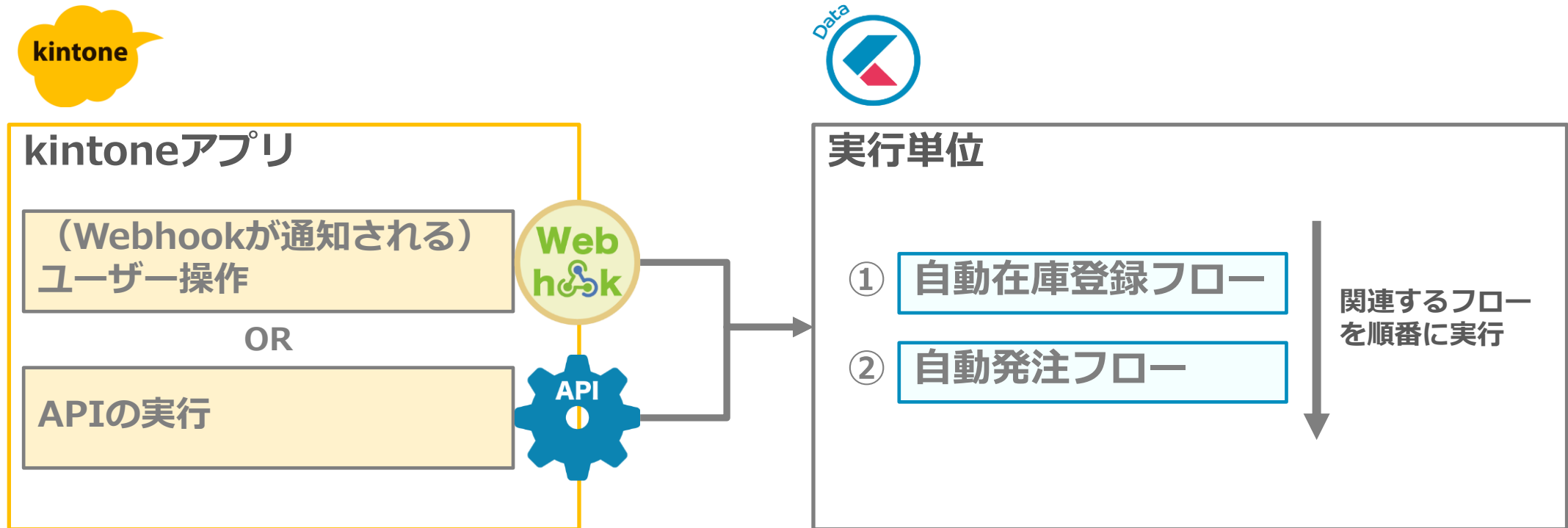




# 実行単位

リアルタイム実行はWebhookの通知やAPIの呼び出しで、複数のデータ編集フローを順番に実行することができます。これにより、関連するデータ編集フローを1度のユーザー操作で実行できます。

「実行単位」は1度の呼び出しで実行する複数のデータ編集フローをまとめた単位です。リアルタイム実行を呼び出すkintoneアプリに対して1つ設定します。（kintoneアプリ:krewData実行単位 = 1 : 1）



# 1. デフォルトのログ出力先アプリの作成と設定

ログ出力先アプリを作成し、デフォルトとして設定します。

① 実行単位一覧の「ログ出力先アプリ」列ヘッダーの▼をクリック

1. 実行単位： フローを実行する契機となるアプリの設定。1つの実行単位に対して1つのアプリを設定で  
+ 実行単位を追加

実行単位	有効	実行結果	呼び出し元アプリ	実行方法	ログ出力先アプリ デフォルト: ▼
------	----	------	----------	------	----------------------


④ [OK] をクリック

② [新しいログ出力先アプリを作成する] を選択

③ アプリ名に「ログ出力先アプリ」と入力し [OK] をクリック

## 2. 新規実行単位の作成と設定

新しい実行単位の作成と設定を行います。


①  実行単位を追加をクリックして、新しい実行単位を作成します。

設定ファイル スケジュール実行 リアルタイム実行

1. 実行単位： フローを実行する契機となるアプリの設定

 実行単位を追加

実行単位	有効	実行結果	呼び出し元アプリ

② 実行単位の  をクリックし、名前を「在庫管理（自動在庫登録～自動発注）」に変更

設定ファイル スケジュール実行 リアルタイム実行

1. 実行単位： フローを実行する契機となるアプリの設定。1

 実行単位を追加

実行単位	有効	実行結果	呼び出し元アプリ
在庫管理（自動在庫登録～自動発注） 	<input type="checkbox"/>		未設定

③ 呼び出し元アプリの▼をクリックし、アプリ選択ダイアログで「出荷管理」アプリを選択

設定ファイル スケジュール実行 リアルタイム実行

1. 実行単位： フローを実行する契機となるアプリの設定。1

 実行単位を追加

実行単位	有効	実行結果	呼び出し元アプリ
在庫管理（自動在庫登録～自動発注） 	<input type="checkbox"/>		未設定 ▼

アプリを選択

アプリの検索

kreisData リアルタイム... ▼ すべての作成者 ▼

- ログ出力先アプリ
- kreisData
- 出荷管理**
- 品目管理
- 在庫管理
- 発注管理

作成者: kreisData

更新日時: 2020-10-19 13:58

フィールド名: レコード番号

品目名: 文字列 (1行)

受注日: 日付

受注数: 数値

更新者: 更新者

スペース名: kreisData リアルタイム実行 チュートリアル

作成日時: 2020-10-19 13:25

フィールドの種類: レコード番号

レコード番号: 文字列 (1行)

日付: 日付

数値: 数値

更新者: 更新者

OK キャンセル

# データ編集フローの作成



# データ編集フロー作成の流れ

データ編集フローは以下の3ステップで作成します。

## 1. 入力アプリを選択



データ取得元のアプリを選択します  
(複数選択可能)

## 2. 編集コマンドを定義



データ編集コマンドを組み合わせ  
てデータを集計・加工します。

## 3. 出力アプリを選択



データ出力（更新）先のアプリを選択  
します。

- データ編集フローは常に「入力アプリ」コマンドで始まり、「出力アプリ」コマンドで終わる
- 入力アプリコマンドと出力アプリコマンドの間に編集コマンドを配置し、データの集計・加工方法を定義
- 配置したコマンドは接続線で結ばれ、編集したデータは矢印の向きに流れる

# パラメータ

krewDataを起動する「きっかけ」を与えたレコードデータの値をパラメータとしてデータ編集フローの処理で利用することができます。



## 出荷管理アプリ

レコード番号	ステータス	品目名	受注日	受注数	出荷日	出荷数
1	出荷	品目A	2020/4/20	200	2020/4/25	200
2	出荷	品目B	2020/4/21	100	2020/4/23	100

このレコードの変更を「きっかけ」に  
krewDataを起動

対象レコードの値を  
パラメータとして利用可能

## パラメータの活用例

フィルタ条件に指定することで、変更したレコードのみをkrewDataの処理対象とする



### 1-2 出荷管理

アプリを選択

標準フィルタ ?

レコード番号 ▼ = (等しい) ▼ %レコード番号% ▼ (x)

使用する値を %[フィールド名] %  
の形式で指定

レコード番号

1

※単一レコードの値のみをパラメータとして利用できます（APIから実行する場合は引数として指定可能）

# 即時実行時のパラメータ

データ編集フロー作成している途中や確認のために実行したい場合、パラメータへ仮の値を設定して、フローを実行したりプレビューで処理結果を確認することができます。

## 即時実行およびプレビューで利用するパラメータの設定

1-2. 在庫管理（不定期定量発注） - Webhook > 引当／出荷（ステータス変更）-> 在庫記録追加 | 入力レコード数: 0 | APIリクエスト数（推定）: 4 | (x) 実行

1-2 出荷管理

### 即時実行用のパラメータ設定

データ編集フローを「実行」ボタンにより即時実行する際に適用するパラメータを設定できます。リアルタイム実行時はWebhook通知、またはAPIパラメータのレコード情報が適用されるため、この設定は無効になります。

#### 1-2 出荷管理

1. 即時実行時に値を設定するフィールドを選択      2. パラメータに設定する値を入力


レコード番号  x


すべてのフィールドを追加    すべてのフィールドを削除    1/12 選択したフィールド

この画面で設定したパラメータの仮の値はプレビューおよび即時実行（画面右上の「実行ボタン」押下）時のみ適用され、WebhookやAPIから呼び出された場合は有効になりません。

# 1. 新規データ編集フローの作成

出荷管理アプリの出荷数を在庫管理アプリの在庫数に反映するデータ編集フローを作成します。

① 2. データ編集フローの  データ編集フローを追加をクリックして、新しいデータ編集フローを作成します。

② データ編集フローの  をクリックし名前を「ステータス変更 -> 在庫レコード追加」に変更

③ 作成したデータ編集フローの設定アイコンをクリックし、「フロー設定」タブを開く

2. データ編集フロー： 実行単位で実行するデータ編集フロー。1つ

 データ編集フローを追加  スケジュール実行のフローを利用  フロー

データ編集フロー			



2. データ編集フロー： 実行単位で実行するデータ編集フロー。1

 データ編集フローを追加  スケジュール実行のフローを利用  フロー

データ編集フロー			
≡	1		ステータス変更 -> 在庫レコード追加 

2. データ編集フロー： 実行単位で実行するデータ編集フロー。1

 データ編集フローを追加  スケジュール実行のフローを利用  フロー

データ編集フロー			
≡	1		ステータス変更 -> 在庫レコード追加 

設定ファイル | スケジュール実行 | リアルタイム実行 | フロー設定

在庫管理 (不定期定数) → ステータス変更 -> 在庫...  | 入力レコード数: 0 | APIリクエスト数 (推定): 0 |    実行

設定ファイル  
 入出力アプリ  
 入力アプリ  
 出力アプリ  
 フィールド操作  
 レコード操作  
 データ結合

1. 入力 2. 加工/集計 3. 出力

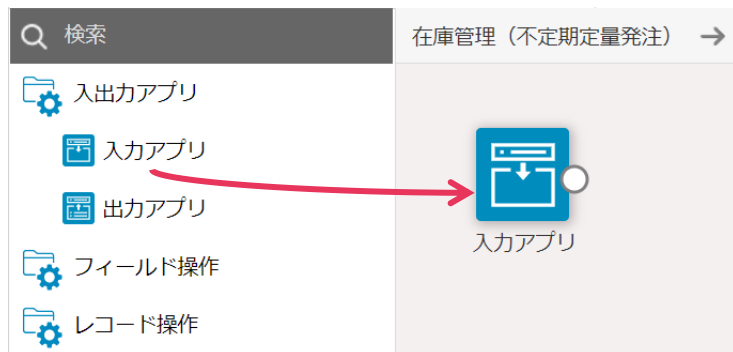
入力アプリコマンドを追加 データ編集コマンドを接続 出力アプリにデータ編集フローの結果を出力します。



## 2. 入力アプリの選択（出荷管理アプリ）

出荷管理アプリのデータを取得する「入力アプリ」コマンドを追加します。

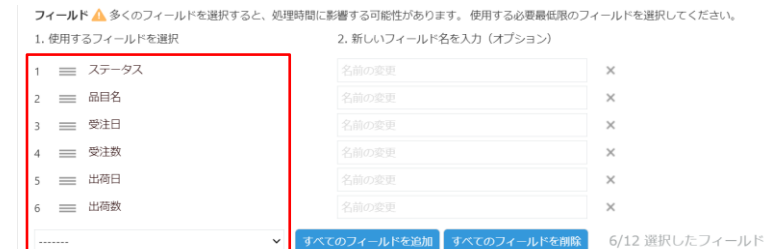
- ① コマンドパネルからフロー作成エリアに「入力アプリ」コマンドを追加



- ② 設定タブで「アプリ選択」ボタンをクリックし、アプリ選択ダイアログで「出荷管理」アプリを選択




- ③ 使用するフィールドを選択
- ・ステータス
  - ・品目名
  - ・受注日
  - ・受注数
  - ・出荷日
  - ・出荷数



- ④ 「プレビュー」タブで入力アプリのデータを確認 ※



※プレビュータブには、最大500件のレコードが表示されます。

 ① 設定タブの標準フィルタで「レコード番号」を選択


アプリを選択

レコード番号 = (等しい)


④ 画面右上のパラメータ設定ボタンをクリック

② パラメータ設定ボタンをクリックして「%レコード番号%」を選択

標準フィルタ ?



レコード番号 ▼ = (等しい) ▼ %レコード番号% ▼ 

+

フィールド  多くのフィールドを選択すると、処理時間に影響する可能性があります。使用する必要最

1. 使用するフィールドを選択

2. 新しいフィールド名を入力 (オプション)


1  ステータス  名前の変更

⑤ 即時実行用のパラメータ設定で  
以下を設定

1.即時実行時に値を設定するフィールドを選択      2.パラメータに設定する値を入力

レコード番号  ×

----- ▼      すべてのフィールドを追加      すべてのフィールドを削除      1/12

 ③ 標準フィルタで「ステータス」を選択し、以下の条件を設定

レコード番号

▼

= (等しい)

▼

%レコード番号%

▼

(x)

ステータス

▼

次のいずれかを含む

▼

未処理

▼

引当

▼

出荷

▼

(x)

ステータス	次のいずれかを含む	未処理 ✓ 引当 ✓ 出荷
-------	-----------	---------------------

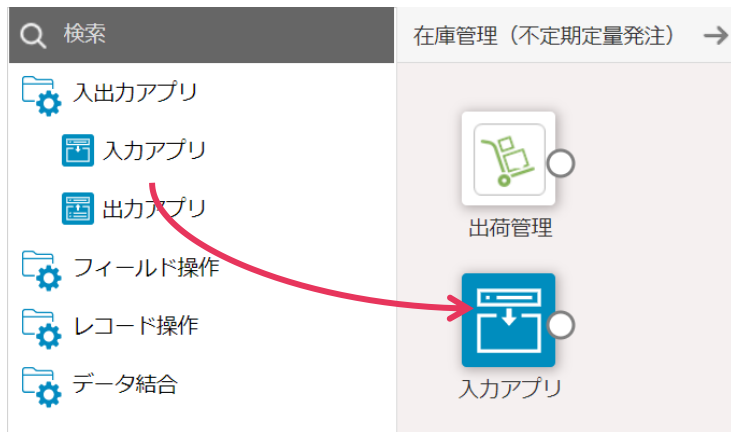
⑥ 「プレビュー」タブで入力アプリのデータを確認

フィルタ条件で絞り込まれたレコードが表示されます。

## 4. 入力アプリの選択（在庫管理アプリ）

在庫管理アプリのデータを取得する「入力アプリ」コマンドを追加します。

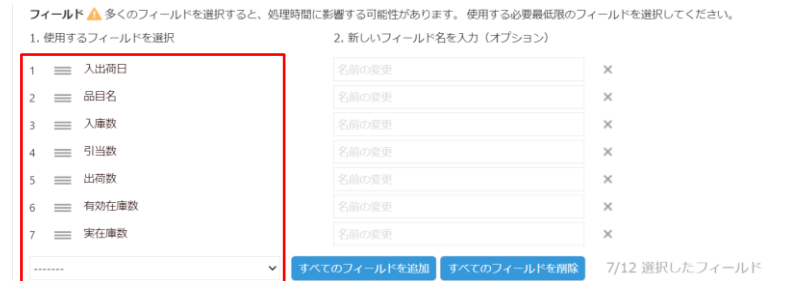
- ① コマンドパネルからフロー作成エリアに「入力アプリ」コマンドを追加



- ② 設定タブで「アプリ選択」ボタンをクリックし、アプリ選択ダイアログで「在庫管理」アプリを選択



- ③ 使用するフィールドを選択
- ・ 入出荷日
  - ・ 品目名
  - ・ 入庫数
  - ・ 引当数
  - ・ 出荷数
  - ・ 有効在庫数
  - ・ 実在庫数



- ④ 「プレビュー」タブで入力アプリのデータを確認 ※



※プレビュータブには、最大500件のレコードが表示されます。

## 5. フィルタとパラメータの設定（在庫管理アプリ）

Webhook通知で送信されるレコードを入力レコードとするためのフィルタ条件を設定します。

① 設定タブの標準フィルタで「品目名」を選択



② パラメータ設定ボタンをクリックして「%品目名%」を選択



### 在庫管理

アプリを選択

標準フィルタ ?

品目名 = (等しい) (x)

標準フィルタ ?

品目名 = (等しい) %品目名% (x)

+

フィールド ⚠ 多くのフィールドを選択すると、処理時間に影響する可能性があります。使用する必要最

1. 使用するフィールドを選択

2. 新しいフィールド名を入力（オプション）

1 出入荷日

名前の変更

③ 画面右上のパラメータ設定ボタンをクリック



④ 即時実行用のパラメータ設定で以下を設定



⑤ 「プレビュー」タブで入力アプリのデータを確認



リアルタイム実行 フロー設定

在庫管理 (不定...) > ステータス変更... 入力レコード数: 4 APIリクエスト数 (推定): 8 (x) 実行

出荷管理

在庫管理

**手動実行用のパラメータ設定**

データ編集フローを「実行」ボタンにより手動実行する際に適用するパラメータを設定できます。リアルタイム実行時はWebhook通知、またはAPIパラメータのレコード情報が適用されるため、この設定は無効になります。

**出荷管理**

1. データ編集フローでユーザーを選択します

2. 「実行」ボタンから実行されるとき、値は適用されます

レコード番号 1

### 出荷管理

1. 即時実行時に値を設定するフィールドを選択

2. パラメータに設定する値を入力

レコード番号 1

品目名 品目A

すべてのフィールドを追加 すべてのフィールドを削除 2/12

項目	設定
1. 即時実行時に値を設定するフィールドを選択	品目名
2. パラメータに設定する値を入力	品目A

在庫管理

在庫管理 設定 プレビュー

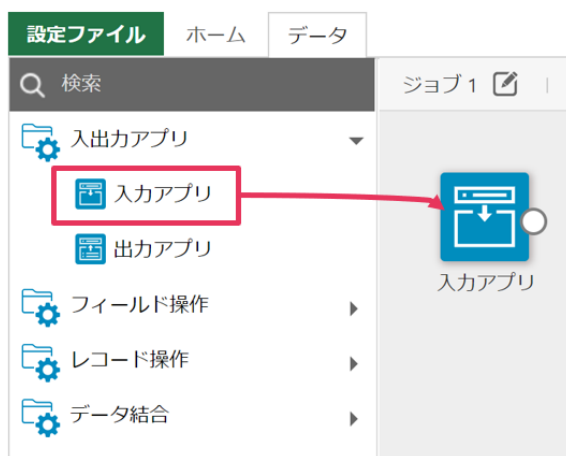
	入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数
1	2020-04-20	品目A		200		250	450
2	2020-04-15	品目A	270			450	450
3	2020-04-07	品目A	180			180	180

フィルタ条件で絞り込まれたレコードが表示されます。

# 次のステップへ

完了

## 1. 入力アプリを選択



データ取得元のアプリを選択します  
(複数選択可能)

## 2. 編集コマンドを定義



データ編集コマンドを組み合わせ  
てデータを集計・加工します。

## 3. 出力アプリを選択

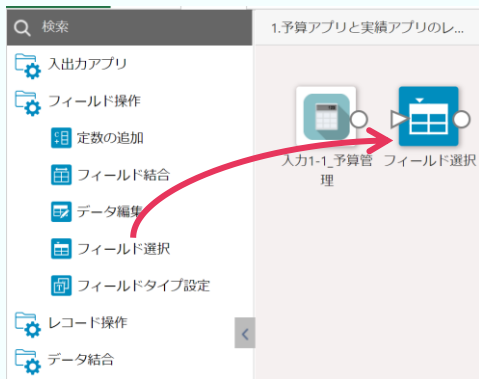


データ出力（更新）先のアプリを選択  
します。

# データ編集フロー作成の基本

フローの作成は基本的に以下の手順を1セットとして繰り返し設定していきます。

## ①コマンドを配置



## ②コマンドをつなぐ



## ③コマンドを設定



必要に応じてコマンド名も設定

## ④プレビューで確認※

	フィールド選択	設定	プレビュー
	ABC 拠点	123 予算	
1	京都事業所	1,200,000	
2	北九州事業所	1,200,000	
3	広島事業所	1,200,000	
4	北海道支社	1,800,000	
5	九州支社	2,400,000	
6	中部支社	2,400,000	
7	大阪支社	3,600,000	
8	本社	6,000,000	

※プレビュータブには、入力アプリから取得した最大500件のレコードが表示されます。

## 6. 重複の削除 完了イメージ

在庫管理アプリの品目名が同じレコードの中から、入出荷日が最も新しいレコードを抽出します。

### Before

	📅 入出荷日 ⚡	ABC 品目名 ⚡	123 入庫数 ⚡	123 引当数 ⚡	123 出荷数 ⚡	123 有効在庫数 ⚡	123 実在庫数 ⚡
1	2020-04-20	品目A		200		250	450
2	2020-04-15	品目A	270			450	450
3	2020-04-07	品目A	180			180	180

### After

	📅 入出荷日 ⚡	ABC 品目名 ⚡	123 入庫数 ⚡	123 引当数 ⚡	123 出荷数 ⚡	123 有効在庫数 ⚡	123 実在庫数 ⚡
1	2020-04-20	品目A		200		250	450

フローの完成イメージ



## 6. 重複の削除 設定方法

### ① コマンドの配置 「重複の削除」をドラッグ



### ② コマンドをつなぐ 線でつなぎます



### ③ 集計方法を設定

※設定タブが表示されない場合はフローのコマンドアイコンをクリックします

重複の削除 設定 プレビュー

1. 重複を検査するフィールド名を選択  
品目名

2. 大文字と小文字を区別するか選択  
☐ 大文字と小文字を区別する

3. ソート  
入出荷日 降順

### ④ プレビューで意図した設定になっているかを確認

 <b>重複の削除</b>			設定		プレビュー		
	 入出荷日 ⇅	 品目名 ⇅	 123 入庫数 ⇅	 123 引当数 ⇅	 123 出荷数 ⇅	 123 有効在庫数 ⇅	 123 実在庫数 ⇅
1	2020-04-20	品目A		200		250	450

#### 項目

#### 設定

1. 重複を検査するフィールド名を選択	品目名
2. 大文字と小文字を区別するか選択	チェックなし
3. ソート	入出荷日 ー 降順



# 7. アプリ結合 完了イメージ

出荷管理アプリと在庫管理アプリを結合します。

## Before

### 出荷管理

	🔒 ステータス	ABC 品目名	📅 受注日	123 受注数	📅 出荷日	123 出荷数
	引当	品目A	2020-04-01	50		

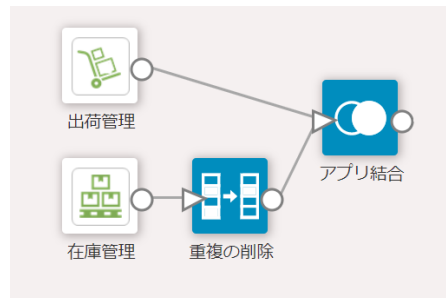
### 在庫管理

	📅 入出荷日	ABC 品目名	123 入庫数	123 引当数	123 出荷数	123 有効在庫数	123 実在庫数
1	2020-04-20	品目A		200		250	450

## After

	🔒 ステータス	ABC 品目名	📅 受注日	123 受注数	📅 出荷日	123 出荷数	📅 入出荷日	ABC 品目名_1	123 入庫数	123 引当数	123 出荷数_1	123 有効在庫数	123 実在庫数
1	引当	品目A	2020-04-01	50			2020-04-20	品目A		200		250	450

### フローの完成イメージ

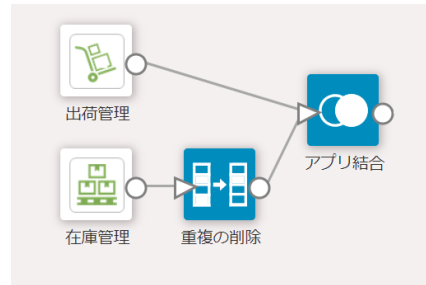


## 7. アプリ結合 設定方法

① コマンドの配置  
「アプリ結合」をドラッグ



② コマンドをつなぐ  
線でつなぎます



③ 結合条件を設定  
[出荷管理] (左外部結合) [重複の削除]  
品目名 品目名

アプリ結合 設定 プレビュー

1. 元になるアプリの条件フィールドを選択

出荷管理 出荷管理 品目名

2. 結合するアプリの条件フィールドを選択

左外部結合 重複の削除 品目名

条件を追加

④ プレビューで意図した設定になっているかを確認

アプリ結合 設定 プレビュー

	ステータス	品目名	受注日	受注数	出荷日	出荷数	入荷日	品目名_1	在庫数	引当数	出荷数_1	有効在庫数	実在庫数
1	引当	品目A	2020-04-01	50			2020-04-20	品目A		200		250	450

※ 結合によりフィールド名が重複する場合は「品目名\_1」のように、名前が自動的に変更されます。

## 8. 在庫数をクリア 完了イメージ

在庫管理アプリから結合した在庫数フィールドに空白を設定し、値をクリアします。

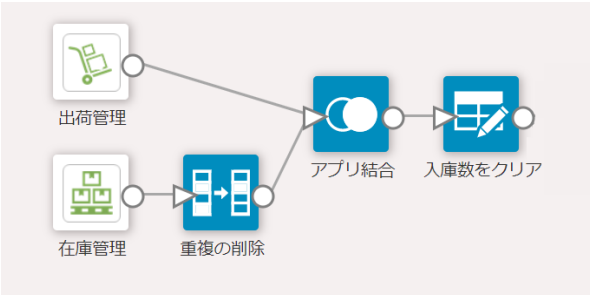
### Before

	🔒 ステータス ◆	ABC 品目名 ◆	📅 受注日 ◆	123 受注数 ◆	📅 出荷日 ◆	123 出荷数 ◆	📅 入出荷日 ◆	ABC 品目名_1 ◆	123 在庫数 ◆	123 引当数 ◆	123 出荷数_1 ◆
1	引当	品目A	2020-04-01	50			2020-04-20	品目A		200	

### After

	🔒 ステータス ◆	ABC 品目名 ◆	📅 受注日 ◆	123 受注数 ◆	📅 出荷日 ◆	123 出荷数 ◆	📅 入出荷日 ◆	ABC 品目名_1 ◆	123 在庫数 ◆	123 引当数 ◆	123 出荷数_1 ◆
1	引当	品目A	2020-04-01	50			2020-04-20	品目A		200	

フローの完成イメージ



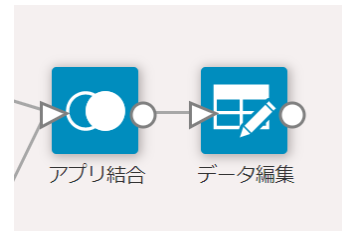
※このチュートリアルサンプルデータには在庫数が設定されていないため、この手順前後のプレビューデータに変化はありません。

## 8. 入庫数をクリア 設定方法

① コマンドの配置  
「データ編集」をドラッグ



② コマンドをつなぐ  
線でつなぎます



④ コマンド名もあわせて設定

入庫数をクリア

③ 編集方法を設定

入庫数をクリア

設定 プレビュー

1. 結果を保存するフィールドを選択	2. 編集方法を選択	3. 編集内容を設定
既存フィールドのデータを置換 フィールドを選択 入庫数	空白を設定	(空白)
4. 検索条件	すべてのレコード	

⑤ プレビューで意図した設定になっているかを確認

入庫数をクリア

設定 プレビュー

	ステータス	品目名	受注日	受注数	出荷日	出荷数	入出荷日	品目名_1	入庫数
1	引当	品目A	2020-04-01	50			2020-04-20	品目A	

項目	設定
1. 結果を保存するフィールドを選択	既存のフィールドのデータを置換
フィールドを選択	入庫数
2. 編集方法を設定	空白を設定
3. 編集内容を設定	—
4. 検索条件	すべてのレコード

## 9. ステータス=引当 完了イメージ

フィルタコマンドでステータスが「引当」のレコードを抽出します。

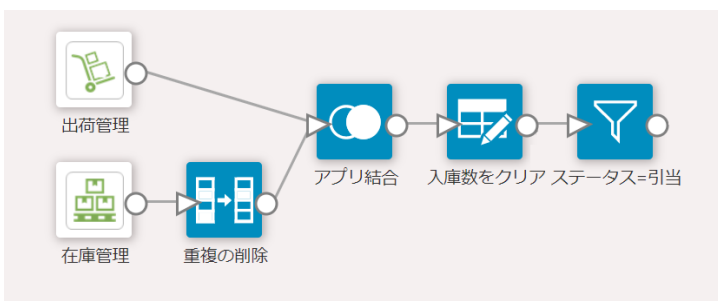
### Before

	🔒 ステータス ◆	ABC 品目名 ◆	📅 受注日 ◆	123 受注数 ◆	📅 出荷日 ◆	123 出荷数 ◆	📅 入出荷日 ◆	ABC 品目名_1 ◆	123 入庫数 ◆	123 引当数 ◆	123 出荷数_1 ◆
1	引当	品目A	2020-04-01	50			2020-04-20	品目A		200	

### After

	🔒 ステータス ◆	ABC 品目名 ◆	📅 受注日 ◆	123 受注数 ◆	📅 出荷日 ◆	123 出荷数 ◆	📅 入出荷日 ◆	ABC 品目名_1 ◆	123 入庫数 ◆	123 引当数 ◆	123 出荷数_1 ◆
1	引当	品目A	2020-04-01	50			2020-04-20	品目A		200	

フローの完成イメージ



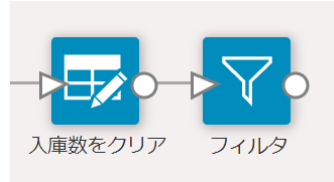
※現在の入力データにはステータスが「引当」のレコードしか存在しないため、この手順前後のプレビューデータに変化はありません。

## 9. ステータス=引当 設定方法

### ① コマンドの配置 「フィルタ」をドラッグ



### ② コマンドをつなぐ 線でつなぎます



### ④ コマンド名もあわせて設定

ステータス=引当

### ③ フィルタ条件を設定

ステータス=引当

標準フィルタ 数式

標準フィルタ

ステータス 次のいずれかを含む

未処理  
✓ 引当  
出荷

### ⑤ プレビューで意図した設定に なっているかを確認

ステータス=引当

設定 プレビュー

	ステータス	品目名	受注日	受注数	出荷日	出荷数	入出荷日	品目名_1	在庫数
1	引当	品目A	2020-04-01	50			2020-04-20	品目A	

標準フィルタ

ステータス 次のいずれかを含む

未処理  
✓ 引当  
出荷

## 10. 出荷数\_1をクリア 完了イメージ

在庫管理アプリから結合した出荷数\_1フィールドに空白を設定し、値をクリアします。

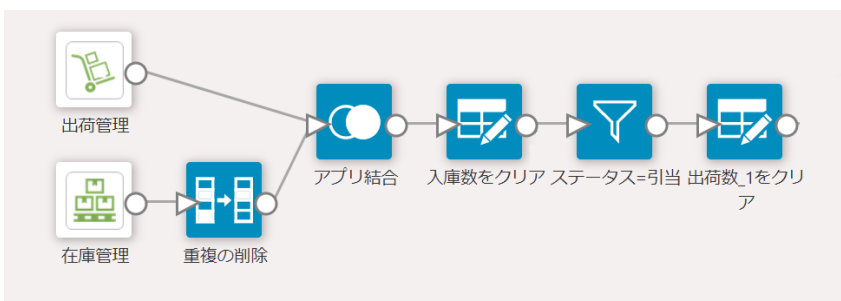
### Before

	🔒 ステータス ◆	ABC 品目名 ◆	📅 受注日 ◆	123 受注数 ◆	📅 出荷日 ◆	123 出荷数 ◆	📅 入出荷日 ◆	ABC 品目名_1 ◆	123 入庫数 ◆	123 引当数 ◆	123 出荷数_1 ◆
1	引当	品目A	2020-04-01	50			2020-04-20	品目A		200	

### After

	🔒 ステータス ◆	ABC 品目名 ◆	📅 受注日 ◆	123 受注数 ◆	📅 出荷日 ◆	123 出荷数 ◆	📅 入出荷日 ◆	ABC 品目名_1 ◆	123 入庫数 ◆	123 引当数 ◆	123 出荷数_1 ◆
1	引当	品目A	2020-04-01	50			2020-04-20	品目A		200	

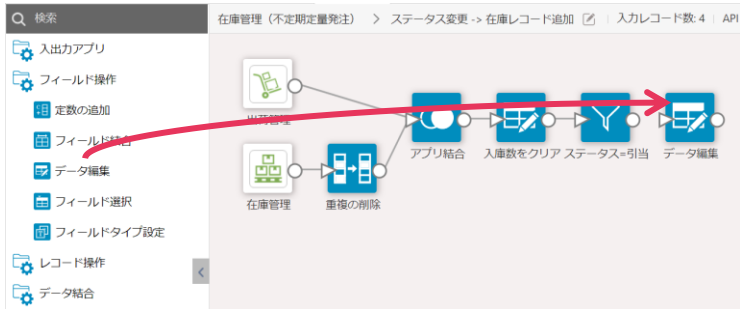
フローの完成イメージ



※このチュートリアルサンプルデータには出荷数が設定されていないため、この手順前後のプレビューデータに変化はありません。

# 10. 出荷数\_1をクリア 設定方法

① コマンドの配置  
「データ編集」をドラッグ



② コマンドをつなぐ  
線でつなぎます



④ コマンド名もあわせて設定

出荷数\_1をクリア

③ 編集方法を設定

出荷数\_1をクリア 設定 プレビュー

1. 結果を保存するフィールドを選択  
既存フィールドのデータを置換  
フィールドを選択  
出荷数\_1

2. 編集方法を選択  
空白を設定

3. 編集内容を設定  
(空白)

4. 検索条件  
すべてのレコード

⑤ プレビューで意図した設定になっているかを確認

	ステータス	品目名	受注日	受注数	出荷日	出荷数	入出荷日	品目名_1	入庫数	引当数	出荷数_1
1	引当	品目A	2020-04-01	50			2020-04-20	品目A		200	

項目	設定
1. 結果を保存するフィールドを選択	既存のフィールドのデータを置換
フィールドを選択	出荷数_1
2. 編集方法を設定	空白を設定
3. 編集内容を設定	—
4. 検索条件	すべてのレコード



# 11. 受注日を入出荷日にコピー 完了イメージ

出荷管理アプリの受注日の値を受注管理アプリの入出荷日にコピーします。

## Before

	🔒 ステータス ◆	ABC 品目名 ◆	📅 受注日 ◆	123 受注数 ◆	📅 出荷日 ◆	123 出荷数 ◆	📅 入出荷日 ◆	ABC 品目名_1 ◆	123 入庫数 ◆	123 引当数 ◆	123 出荷数_1 ◆
1	引当	品目A	2020-04-01	50			2020-04-20	品目A		200	

## After

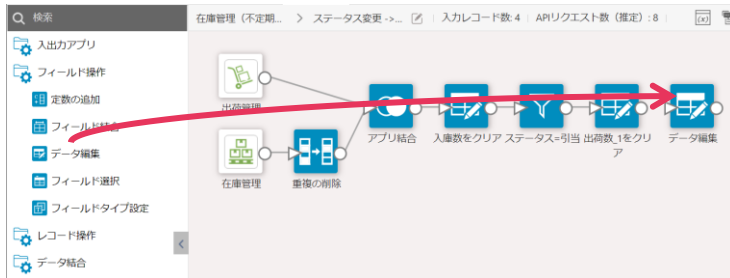
	🔒 ステータス ◆	ABC 品目名 ◆	📅 受注日 ◆	123 受注数 ◆	📅 出荷日 ◆	123 出荷数 ◆	📅 入出荷日 ◆	ABC 品目名_1 ◆	123 入庫数 ◆	123 引当数 ◆	123 出荷数_1 ◆
1	引当	品目A	2020-04-01	50			2020-04-01	品目A		200	

フローの完成イメージ



# 11. 受注日を入出荷日にコピー 設定方法

① コマンドの配置  
「データ編集」をドラッグ



② コマンドをつなぐ  
線でつなぎます



④ コマンド名もあわせて設定

受注日を入出荷...

③ 編集方法を設定

受注日を入出荷... 設定 プレビュー

1. 結果を保存するフィールドを選択	2. 編集方法を選択	3. 編集内容を設定
既存フィールドのデータを置換 フィールドを選択 入出荷日	フィールドをコピー	受注日
4. 検索条件 すべてのレコード		

⑤ プレビューで意図した設定になっているかを確認

	受注日を入出荷...	設定	プレビュー
	ステータス	品目名	受注日
1	引当	品目A	2020-04-01

項目	設定
1. 結果を保存するフィールドを選択	既存のフィールドのデータを置換
フィールドを選択	入出荷日
2. 編集方法を設定	フィールドをコピー
3. 編集内容を設定	受注日
4. 検索条件	すべてのレコード

## 12. 受注数を引当数にコピー 完了イメージ

出荷管理アプリの受注数の値を受注管理アプリの引当数にコピーします。

### Before

	🔒 ステータス	ABC 品目名	📅 受注日	123 受注数	📅 出荷日	123 出荷数	📅 入出荷日	ABC 品目名_1	123 入庫数	123 引当数	123 出荷数_1
1	引当	品目A	2020-04-01	50			2020-04-20	品目A		200	

### After

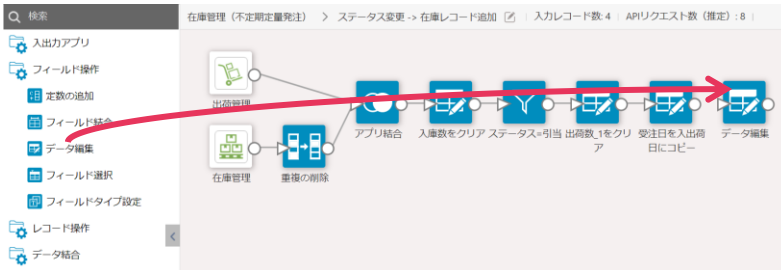
	🔒 ステータス	ABC 品目名	📅 受注日	123 受注数	📅 出荷日	123 出荷数	📅 入出荷日	ABC 品目名_1	123 入庫数	123 引当数	123 出荷数_1
1	引当	品目A	2020-04-01	50			2020-04-01	品目A		50	

フローの完成イメージ



# 12. 受注数を引当数にコピー 設定方法

① コマンドの配置  
「データ編集」をドラッグ



② コマンドをつなぐ  
線でつなぎます



④ コマンド名もあわせて設定

受注数を引当数...

③ 編集方法を設定

受注数を引当数... 設定 プレビュー

1. 結果を保存するフィールドを選択  
既存フィールドのデータを置換  
フィールドを選択  
引当数

2. 編集方法を選択  
フィールドをコピー

3. 編集内容を設定  
受注数

4. 検索条件  
すべてのレコード

⑤ プレビューで意図した設定になっているかを確認

	ステータス	品目名	受注日	受注数	出荷日	出荷数	入出荷日	品目名_1	在庫数	引当数	出荷数_1
1	引当	品目A	2020-04-01	50			2020-04-01	品目A		50	

項目	設定
1. 結果を保存するフィールドを選択	既存のフィールドのデータを置換
フィールドを選択	引当数
2. 編集方法を設定	フィールドをコピー
3. 編集内容を設定	受注数
4. 検索条件	すべてのレコード

# 13. 有効在庫数を減算 完了イメージ

受注管理アプリの有効在庫数から引当数を減算します。

## Before

	🔒 ステータス	ABC 品目名	📅 受注日	123 受注数	📅 出荷日	123 出荷数	📅 入出荷日	ABC 品目名_1	123 入庫数	123 引当数	123 出荷数_1	123 有効在庫数	123 実在庫数
1	引当	品目A	2020-04-01	50			2020-04-01	品目A		50		250	450

## After

	🔒 ステータス	ABC 品目名	📅 受注日	123 受注数	📅 出荷日	123 出荷数	📅 入出荷日	ABC 品目名_1	123 入庫数	123 引当数	123 出荷数_1	123 有効在庫数	123 実在庫数
1	引当	品目A	2020-04-01	50			2020-04-01	品目A		50		200	450

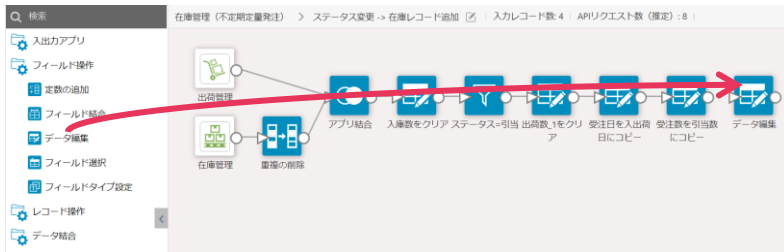
数式を使用して有効在庫数（有効在庫数－引当数）を計算

フローの完成イメージ



# 13. 有効在庫数を減算 設定方法

① コマンドの配置  
「データ編集」をドラッグ



② コマンドをつなぐ  
線でつなぎます



④ コマンド名もあわせて設定

有効在庫数を減算

③ 計算方法を設定

有効在庫数を減算

設定 プレビュー

1. 結果を保存するフィールドを選択

既存フィールドのデータを置換

フィールドを選択

有効在庫数

2. 編集方法を選択

数式

3. 編集内容を設定

=有効在庫数-引当数

4. 検索条件

すべてのレコード

⑤ プレビューで意図した設定になっているかを確認

出荷日	出荷数	入出荷日	品目名_1	入庫数	引当数	出荷数_1	有効在庫数	実在庫数
2020-04-01			品目A		50		200	450

項目	設定
1. 結果を保存するフィールドを選択	既存フィールドのデータを置換
フィールドを選択	有効在庫数
2. 編集方法を設定	数式
3. 編集内容を設定	有効在庫数-引当数
4. 検索条件	すべてのレコード



※数式は関数の挿入  
ダイアログで入力

# 14. フィールド選択 完了イメージ

以降のフローで利用する必要なフィールドのみ抽出します。

## Before

	🔒 ステータス ⇅	ABC 品目名 ⇅	📅 受注日 ⇅	123 受注数 ⇅	📅 出荷日 ⇅	123 出荷数 ⇅	📅 入出荷日 ⇅	ABC 品目名_1 ⇅	123 入庫数 ⇅	123 引当数 ⇅	123 出荷数_1 ⇅	123 有効在庫数 ⇅	123 実在庫数 ⇅
1	引当	品目A	2020-04-01	50			2020-04-01	品目A		50		200	450

## After

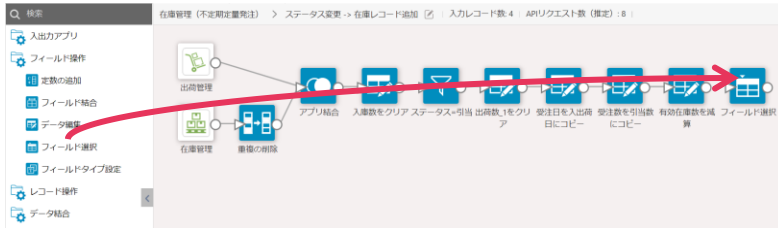
	📅 入出荷日 ⇅	ABC 品目名 ⇅	123 入庫数 ⇅	123 引当数 ⇅	123 出荷数 ⇅	123 有効在庫数 ⇅	123 実在庫数 ⇅
1	2020-04-01	品目A		50		200	450

フローの完成イメージ

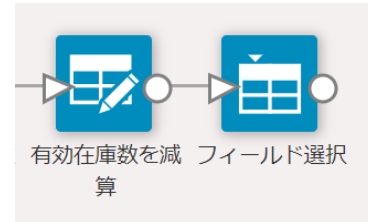


# 14. フィールド選択 設定方法

① コマンドの配置  
「フィールド選択」をドラッグ



② コマンドをつなぐ  
線でつなぎます



③ 必要なフィールドのみ選択する

フィールド選択 設定 プレビュー

1. 使用するフィールドを選択

2. 新しいフィールド名を入力（オプション）

1	入出荷日	名前の変更	×
2	品目名_1	品目名	×
3	入庫数	名前の変更	×
4	引当数	名前の変更	×
5	出荷数_1	出荷数	×
6	有効在庫数	名前の変更	×
7	実在庫数	名前の変更	×

すべてのフィールドを追加 すべてのフィールドを削除 7/13

④ プレビューで意図した設定になっているかを確認

フィールド選択

設定

プレビュー

	<div><div></div><div>入出荷日</div><div>▼</div></div>	<div><div>ABC</div><div>品目名</div><div>▼</div></div>	<div><div>123</div><div>入庫数</div><div>▼</div></div>	<div><div>123</div><div>引当数</div><div>▼</div></div>	<div><div>123</div><div>出荷数</div><div>▼</div></div>	<div><div>123</div><div>有効在庫数</div><div>▼</div></div>	<div><div>123</div><div>実在庫数</div><div>▼</div></div>
1	2020-04-01	品目A		50		200	450

ここまでの手順で、レコードのステータスが「引当」の場合の処理フロー部分が完成しました。引き続き、レコードのステータスが「出荷」の場合の処理フローを作成します。

1.使用するフィールドを選択 2.新しいフィールド名を入力

1 入出荷日

2 品目名\_1

品目名

3 入庫数

4 引当数

5 出荷数\_1

出荷数

6 有効在庫数

7 実在庫数



# 15. パラメータ設定の変更

「3. フィルタとパラメータの設定（出荷管理アプリ）」の手順②で出荷管理アプリのフィルタ条件として「レコード番号=%レコード番号%」を設定し、手順⑤では即時実行用のパラメータ設定としてレコード番号に1を設定しました。

これらの設定により、現在作成中のフローのプレビュー画面には出荷管理アプリの「レコード番号=1」のレコードが表示され、このレコードのステータスは「引当」になっています。

以降の手順では、ステータスが「出荷」のレコードを処理するフローを作成しますので、ここでパラメータ設定を以下のように変更します。

① 画面右上のパラメータ設定ボタンをクリック

② 即時実行用のパラメータ設定でレコード番号の値を「2」に変更

項目	設定
1 .即時実行時に値を設定するフィールドを選択	レコード番号
2 .パラメータに設定する値を入力	2

③ 出荷管理アプリの「プレビュー」タブで入力データを確認

ステータスが「出荷」のレコードが表示されます。

# 16. ステータス=出荷 完了イメージ

フィルタコマンドでステータスが「出荷」のレコードを抽出します。

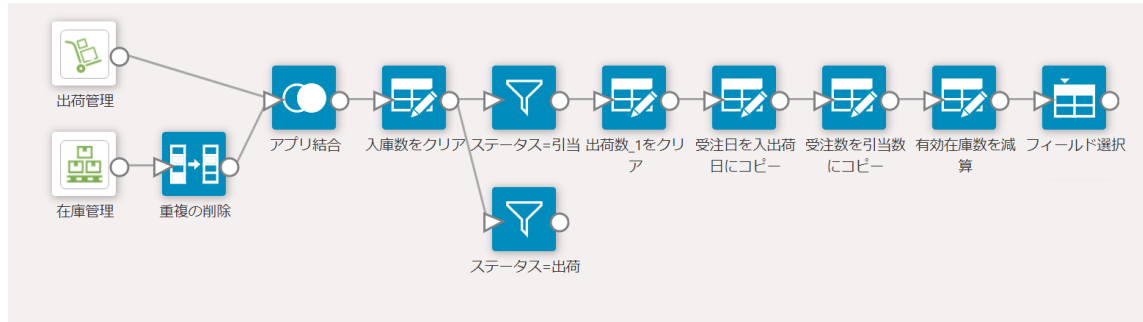
## Before

	🔒 ステータス ◆	ABC 品目名 ◆	📅 受注日 ◆	123 受注数 ◆	📅 出荷日 ◆	123 出荷数 ◆	📅 入出荷日 ◆	ABC 品目名_1 ◆	123 入庫数 ◆	123 引当数 ◆	123 出荷数_1 ◆
1	出荷	品目A	2020-04-02	100	2020-04-02	100	2020-04-20	品目A		200	

## After

	🔒 ステータス ◆	ABC 品目名 ◆	📅 受注日 ◆	123 受注数 ◆	📅 出荷日 ◆	123 出荷数 ◆	📅 入出荷日 ◆	ABC 品目名_1 ◆	123 入庫数 ◆	123 引当数 ◆	123 出荷数_1 ◆
1	出荷	品目A	2020-04-02	100	2020-04-02	100	2020-04-20	品目A		200	

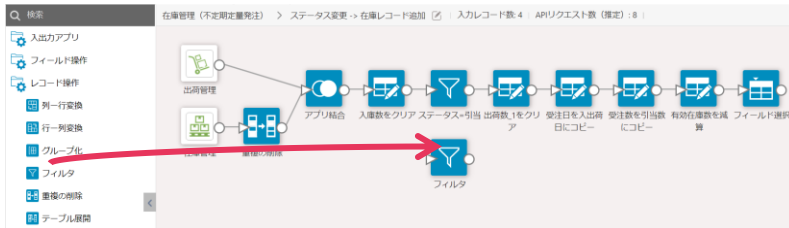
フローの完成イメージ



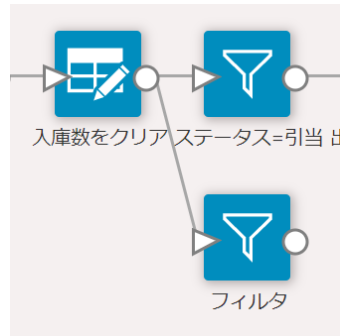
※現在の入力データにはステータスが「出荷」のレコードしか存在しないため、この手順前後のプレビューデータに変化はありません。

# 16. ステータス=出荷 設定方法

① コマンドの配置  
「フィルタ」をドラッグ



② コマンドをつなぐ  
線でつなぎます



④ コマンド名もあわせて設定

ステータス=出荷

③ フィルタ条件を設定

ステータス=出荷

標準フィルタ

ステータス 次のもいずれかを含む

未処理  
引当  
✓ 出荷

⑤ プレビューで意図した設定に  
なっているかを確認

	ステータス	品目名	受注日	受注数	出荷日	出荷数	入出荷日	品目名_1	在庫数
1	出荷	品目A	2020-04-02	100	2020-04-02	100	2020-04-20	品目A	

標準フィルタ  
ステータス 次のもいずれかを含む 未処理  
引当  
✓ 出荷

# 17. 引当数をクリア 完了イメージ

在庫管理アプリから結合した引当数フィールドに空白を設定し、値をクリアします。

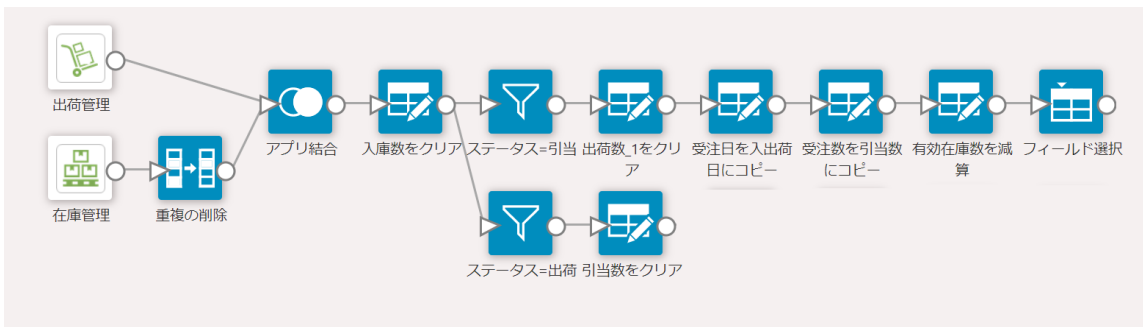
## Before

	🔒 ステータス	ABC 品目名	📅 受注日	123 受注数	📅 出荷日	123 出荷数	📅 入出荷日	ABC 品目名_1	123 入庫数	123 引当数	123 出荷数_1
1	出荷	品目A	2020-04-02	100	2020-04-02	100	2020-04-20	品目A		200	

## After

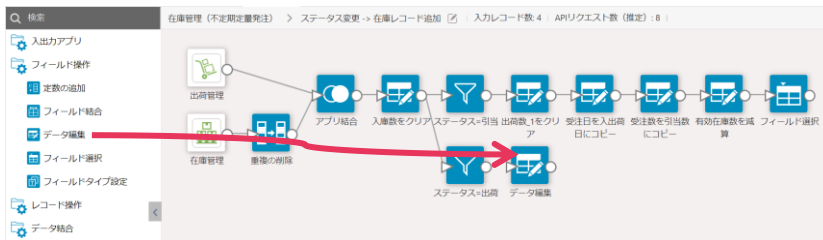
	🔒 ステータス	ABC 品目名	📅 受注日	123 受注数	📅 出荷日	123 出荷数	📅 入出荷日	ABC 品目名_1	123 入庫数	123 引当数	123 出荷数_1
1	出荷	品目A	2020-04-02	100	2020-04-02	100	2020-04-20	品目A			

フローの完成イメージ



# 17. 引当数をクリア 設定方法

① コマンドの配置  
「データ編集」をドラッグ



② コマンドをつなぐ  
線でつなぎます



④ コマンド名もあわせて設定

引当数をクリア

③ 編集方法を設定

引当数をクリア 設定 プレビュー

1. 結果を保存するフィールドを選択  
既存フィールドのデータを置換  
フィールドを選択  
引当数

2. 編集方法を選択  
空白を設定

3. 編集内容を設定  
(空白)

4. 検索条件  
すべてのレコード

⑤ プレビューで意図した設定になっているかを確認

	ステータス	品目名	受注日	受注数	出荷日	出荷数	入出荷日	品目名_1	在庫数	引当数	出荷数_1
1	出荷	品目A	2020-04-02	100	2020-04-02	100	2020-04-20	品目A			

項目	設定
1. 結果を保存するフィールドを選択	既存のフィールドのデータを置換
フィールドを選択	引当数
2. 編集方法を設定	空白を設定
3. 編集内容を設定	—
4. 検索条件	すべてのレコード

# 18. 出荷日を入出荷日にコピー 完了イメージ

出荷管理アプリの出荷日の値を受注管理アプリの入出荷日にコピーします。

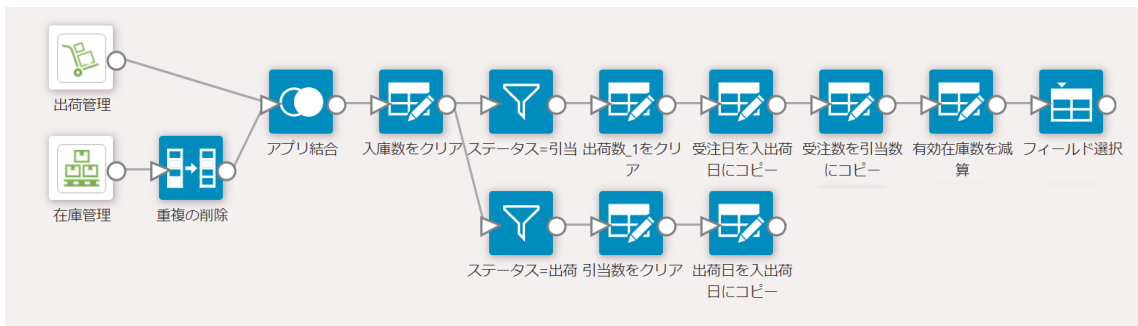
## Before

	🔒 ステータス ◆	ABC 品目名 ◆	📅 受注日 ◆	123 受注数 ◆	📅 出荷日 ◆	123 出荷数 ◆	📅 入出荷日 ◆	ABC 品目名_1 ◆	123 入庫数 ◆	123 引当数 ◆	123 出荷数_1 ◆
1	出荷	品目A	2020-04-02	100	2020-04-02	100	2020-04-20	品目A			

## After

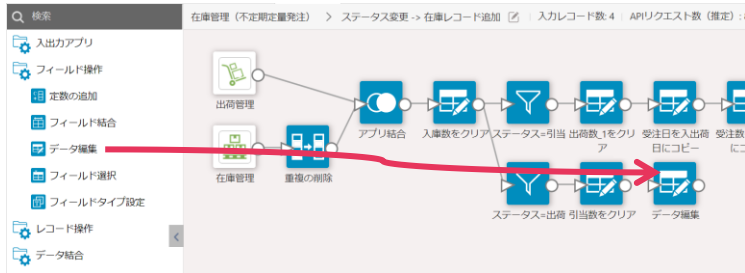
	🔒 ステータス ◆	ABC 品目名 ◆	📅 受注日 ◆	123 受注数 ◆	📅 出荷日 ◆	123 出荷数 ◆	📅 入出荷日 ◆	ABC 品目名_1 ◆	123 入庫数 ◆	123 引当数 ◆	123 出荷数_1 ◆
1	出荷	品目A	2020-04-02	100	2020-04-02	100	2020-04-02	品目A			

フローの完成イメージ

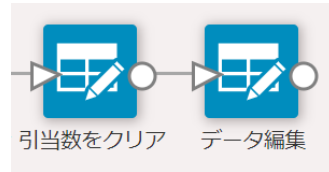


# 18. 出荷日を入出荷日にコピー 設定方法

① コマンドの配置  
「データ編集」をドラッグ



② コマンドをつなぐ  
線でつなぎます



④ コマンド名もあわせて設定

出荷日を入出荷...

③ 編集方法を設定

出荷日を入出荷... 設定 プレビュー

1. 結果を保存するフィールドを選択  
既存フィールドのデータを置換  
フィールドを選択  
入出荷日

2. 編集方法を選択  
フィールドをコピー

3. 編集内容を設定  
出荷日

4. 検索条件  
すべてのレコード

⑤ プレビューで意図した設定になっているかを確認

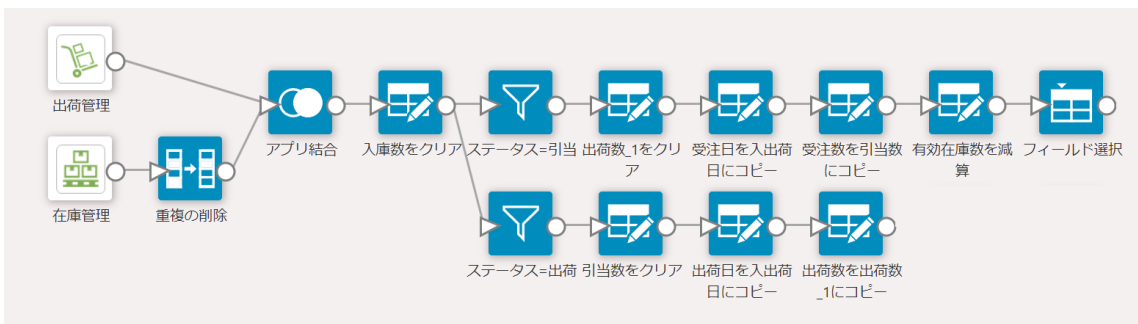
	ステータス	品目名	受注日	受注数	出荷日	出荷数	入出荷日	品目名_1	在庫数	引当数	出荷数_1
1	出荷	品目A	2020-04-02	100	2020-04-02	100	2020-04-02	品目A			

項目	設定
1. 結果を保存するフィールドを選択	既存のフィールドのデータを置換
フィールドを選択	入出荷日
2. 編集方法を設定	フィールドをコピー
3. 編集内容を設定	出荷日
4. 検索条件	すべてのレコード

# Before

# After

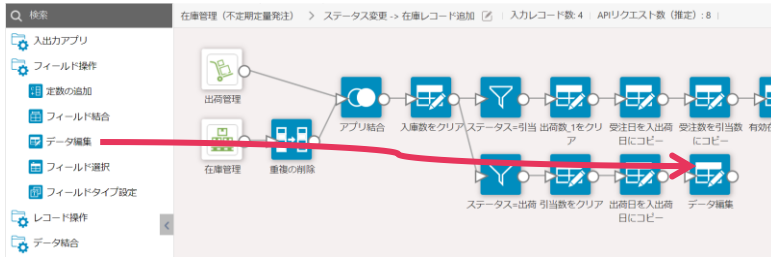
## フローの完成イメージ





# 19. 出荷数を出荷数\_1にコピー 設定方法

① コマンドの配置  
「データ編集」をドラッグ



② コマンドをつなぐ  
線でつなぎます



④ コマンド名もあわせて設定

出荷数を出荷数...

③ 編集方法を設定

出荷数を出荷数... 設定 プレビュー

1. 結果を保存するフィールドを選択  
既存フィールドのデータを置換  
フィールドを選択  
出荷数\_1

2. 編集方法を選択  
フィールドをコピー

3. 編集内容を設定  
出荷数

4. 検索条件  
すべてのレコード

⑤ プレビューで意図した設定になっているかを確認

出荷数を出荷数...

設定

プレビュー

	<div>🔒</div> ステータス	<div>ABC</div> 品目名	<div>📅</div> 受注日	<div>123</div> 受注数	<div>📅</div> 出荷日	<div>123</div> 出荷数	<div>📅</div> 入出荷日	<div>ABC</div> 品目名_1	<div>123</div> 在庫数	<div>123</div> 引当数	<div>123</div> 出荷数_1
1	出荷	品目A	2020-04-02	100	2020-04-02	100	2020-04-02	品目A			100

項目	設定
1. 結果を保存するフィールドを選択	既存のフィールドのデータを置換
フィールドを選択	出荷数_1
2. 編集方法を設定	フィールドをコピー
3. 編集内容を設定	出荷数
4. 検索条件	すべてのレコード

## 20. 実在庫数を減算 完了イメージ

受注管理アプリの実在庫数から出荷数を減算します。

### Before

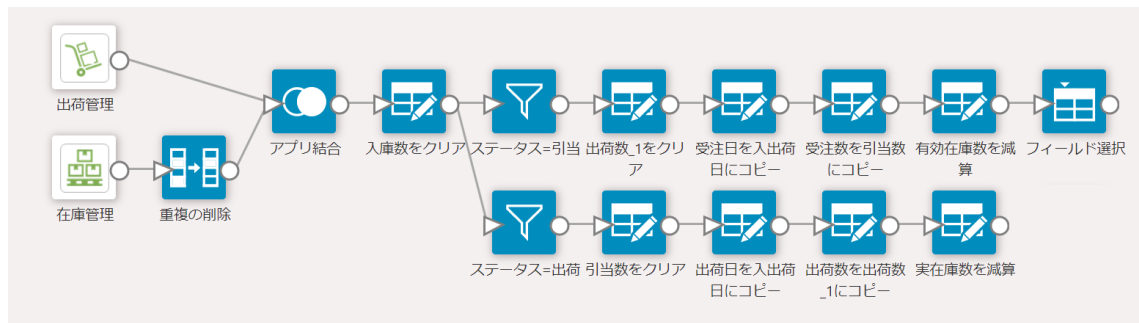
	🔒 ステータス	ABC 品目名	📅 受注日	123 受注数	📅 出荷日	123 出荷数	📅 入出荷日	ABC 品目名_1	123 入庫数	123 引当数	123 出荷数_1	123 有効在庫数	123 実在庫数
1	出荷	品目A	2020-04-02	100	2020-04-02	100	2020-04-02	品目A			100	250	450

### After

	🔒 ステータス	ABC 品目名	📅 受注日	123 受注数	📅 出荷日	123 出荷数	📅 入出荷日	ABC 品目名_1	123 入庫数	123 引当数	123 出荷数_1	123 有効在庫数	123 実在庫数
1	出荷	品目A	2020-04-02	100	2020-04-02	100	2020-04-02	品目A				100	350

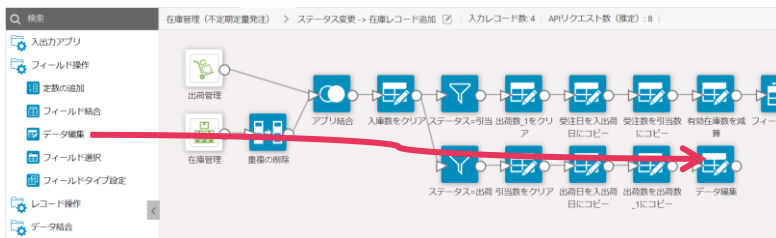
数式を使用して実在庫数（実在庫数－出荷数\_1）を計算

フローの完成イメージ



## 20. 実在庫数を減算 設定方法

① コマンドの配置  
「データ編集」をドラッグ



② コマンドをつなぐ  
線でつなぎます



④ コマンド名もあわせて設定

実在庫数を減算

③ 計算方法を設定

実在庫数を減算

設定 プレビュー

1. 結果を保存するフィールドを選択

既存フィールドのデータを置換

フィールドを選択

実在庫数

2. 編集方法を選択

数式

3. 編集内容を設定

=実在庫数-出荷数\_1

4. 検索条件

すべてのレコード

⑤ プレビューで意図した設定になっているかを確認

出荷日	出荷数	入出荷日	品目名_1	入庫数	引当数	出荷数_1	有効在庫数	実在庫数
2020-04-02	100	2020-04-02	品目A			100	250	350

項目	設定
1. 結果を保存するフィールドを選択	既存フィールドのデータを置換
フィールドを選択	実在庫数
2. 編集方法を設定	数式
3. 編集内容を設定	実在庫数-出荷数_1
4. 検索条件	すべてのレコード



※数式は関数の挿入  
ダイアログで入力

## 21.フィールド選択 (2) 完了イメージ

以降のフローで利用する必要なフィールドのみ抽出します。

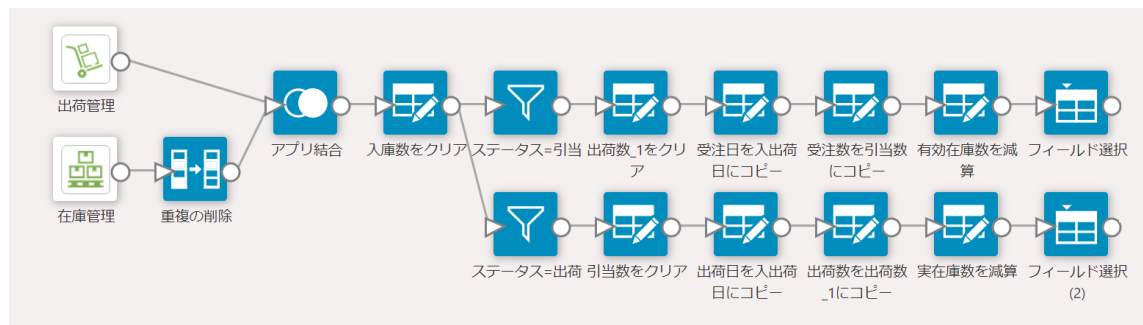
Before

	🔒 ステータス	ABC 品目名	📅 受注日	123 受注数	📅 出荷日	123 出荷数	📅 入出荷日	ABC 品目名_1	123 入庫数	123 引当数	123 出荷数_1	123 有効在庫数	123 実在庫数
1	出荷	品目A	2020-04-02	100	2020-04-02	100	2020-04-02	品目A			100	250	350

After

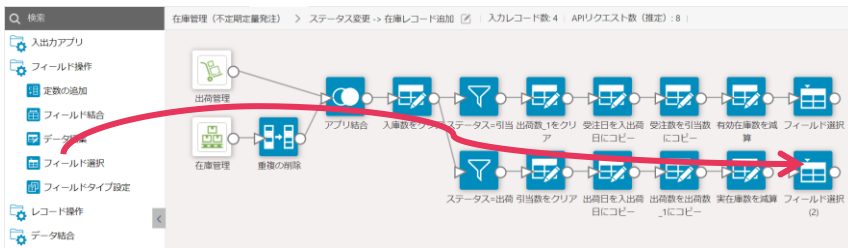
	📅 入出荷日	ABC 品目名	123 入庫数	123 引当数	123 出荷数	123 有効在庫数	123 実在庫数
1	2020-04-02	品目A			100	250	350

フローの完成イメージ



# 21.フィールド選択 (2) 設定方法

① コマンドの配置  
「フィールド選択」をドラッグ



② コマンドをつなぐ  
線でつなぎます



③ 必要なフィールドのみ選択する

フィールド選択... 設定 プレビュー

1. 使用するフィールドを選択		2. 新しいフィールド名を入力 (オプション)	
1	≡ 入出荷日	名前の変更	×
2	≡ 品目名_1	品目名	×
3	≡ 入庫数	名前の変更	×
4	≡ 引当数	名前の変更	×
5	≡ 出荷数_1	出荷数	×
6	≡ 有効在庫数	名前の変更	×
7	≡ 実在庫数	名前の変更	×

すべてのフィールドを追加 すべてのフィールドを削除 7/13

④ プレビューで意図した設定になっているかを確認

フィールド選択... 設定 プレビュー

	入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数
1	2020-04-02	品目A			100	250	350

以上でレコードのステータスが「出荷」の場合の処理フロー部分が完成しました。

1.使用するフィールドを選択 2.新しいフィールド名を入力

1 入出荷日

2 品目名\_1

品目名

3 入庫数

4 引当数

5 出荷数\_1

出荷数

6 有効在庫数

7 実在庫数

## 22. レコード結合 完了イメージ

ステータスに応じて分岐していた処理フローをレコード結合コマンドで結合します。

Before

ステータス=引当

	入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数

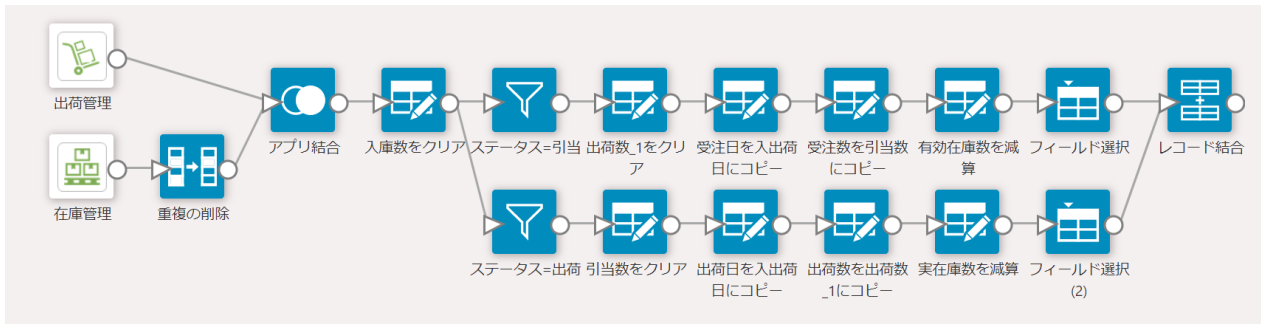
ステータス=出荷

	入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数
1	2020-04-02	品目A			100	250	350

After

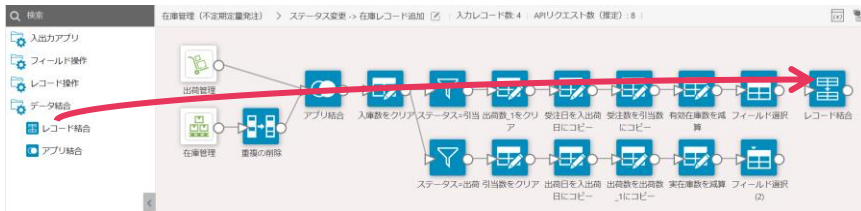
	入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数
1	2020-04-02	品目A			100	250	350

フローの完成イメージ

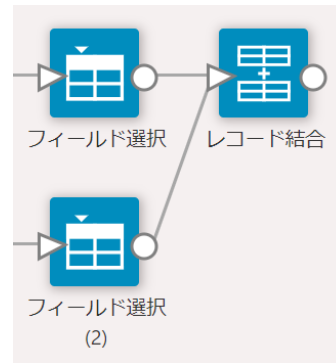


## 22. レコード結合 設定方法

① コマンドの配置  
「フィールド選択」をドラッグ



② コマンドをつなぐ  
線でつなぎます



③ 結合方法を選択する

レコード結合 設定 プレビュー

1. 結合方法を選択

共通するフィールドのみを含める

2. 結合後の変更内容を確認

フィールド選択 変更なし

フィールド選択 (2) 変更なし

④ プレビューで意図した設定になっているかを確認

レコード結合

設定

プレビュー

	<div> <div></div> <div>入出荷日</div> <div></div> </div>	<div> <div>ABC</div> <div>品目名</div> <div></div> </div>	<div> <div>123</div> <div>入庫数</div> <div></div> </div>	<div> <div>123</div> <div>引当数</div> <div></div> </div>	<div> <div>123</div> <div>出荷数</div> <div></div> </div>	<div> <div>123</div> <div>有効在庫数</div> <div></div> </div>	<div> <div>123</div> <div>実在庫数</div> <div></div> </div>
1	2020-04-02	品目A			100	250	350

項目

設定

1. 結合方法を選択

共通するフィールドのみを含める

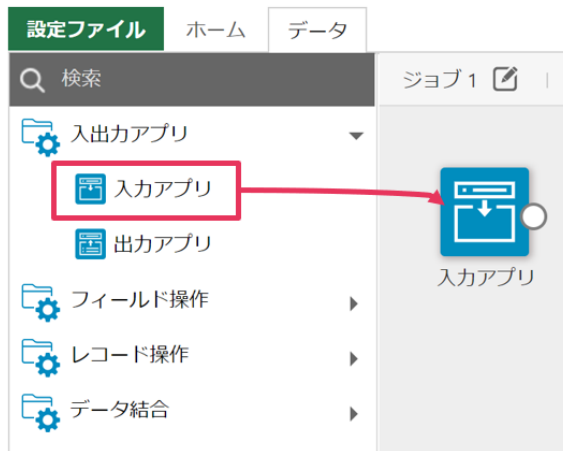
2. 結合後の変更内容を確認

(設定項目なし)

# 最後のステップへ

完了

## 1. 入力アプリを選択



データ取得元のアプリを選択します  
(複数選択可能)

完了

## 2. 編集コマンドを定義



データ編集コマンドを組み合わせ  
てデータを集計・加工します。

最後に出力コマンド  
を設定します

## 3. 出力アプリを選択



データ出力（更新）先のアプリを選択  
します。



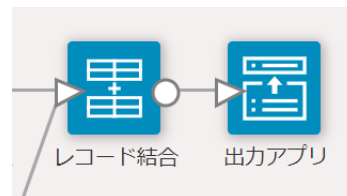
## 23. 出力アプリの選択（在庫管理アプリ）

結果データを出力する在庫管理アプリを選択します。

① コマンドの配置  
「出力アプリ」をドラッグ



② コマンドをつなぐ  
線でつなぎます



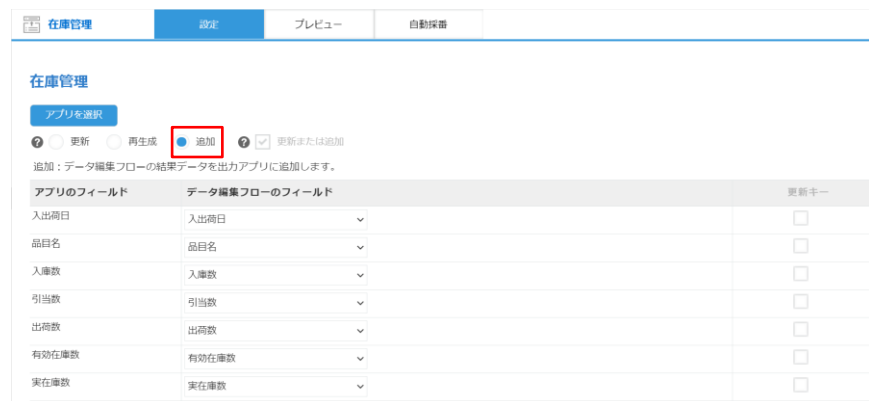
③ 「アプリを選択」をクリック



④ アプリ選択ダイアログで「在庫管理」アプリを選択



⑤ 出力方法に「追加」を指定



⑥ アプリを更新



# 結果データの出力方法

出力アプリコマンドには、更新／再生成／追加の3種類の出力方法があります。

更新前

製品	部門	金額
製品A	第一営業部	100
製品B	第二営業部	200
製品C	第三営業部	300
製品B	第三営業部	400
製品C	第四営業部	500

データ編集フロー  
処理結果

製品	部門	金額
製品A	第一営業部	110
製品B	第二営業部	220
製品C	第三営業部	330
製品D	第三営業部	440

製品	部門	金額
製品A	第一営業部	110
製品B	第二営業部	220
製品C	第三営業部	330
製品D	第三営業部	440
製品C	第三営業部	390

更新キー（製品／部門）が一致するレコードを更新  
（一致するものがない場合に追加するか否かを指定可能）

※更新キー（製品／部門）が一意になっていない場合は更新不可

処理結果

更新

製品	部門	金額
製品A	第一営業部	110
製品B	第二営業部	220
製品C	第三営業部	330
製品B	第三営業部	400
製品C	第四営業部	500
製品D	第三営業部	440

再生成

製品	部門	金額
製品A	第一営業部	110
製品B	第二営業部	220
製品C	第三営業部	330
製品D	第三営業部	440

データ編集フローの結果で入替

追加

製品	部門	金額
製品A	第一営業部	100
製品B	第二営業部	200
製品C	第三営業部	300
製品B	第三営業部	400
製品C	第四営業部	500
製品A	第一営業部	110
製品B	第二営業部	220
製品C	第三営業部	330
製品D	第三営業部	440

データ編集フローの結果を新規追加

# データ編集フローの実行

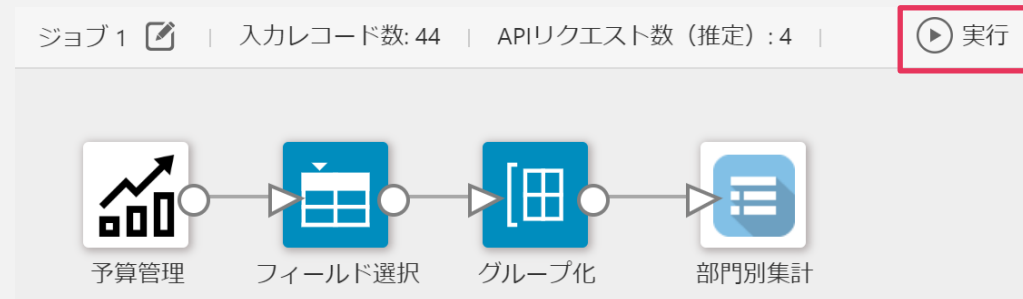


# データ編集フローの実行方法

## ①即時（手動）実行

- データ編集フロー作成画面から手動実行  
フロー作成中の確認時に利用

※処理レコード数の上限は10,000レコード



## ②リアルタイム実行

- Webhook通知またはAPIにより任意のタイ  
ミングで実行

業務アプリのレコード操作時に自動実行が可能

※処理レコード数の上限は10,000レコード

お試し中の処理レコード数の上限は10,000レコード、登録可能なリアルタイム実行フロー数は3個まで

全画面表示を終了 | アプリを更新

設定ファイル | スケジュール実行 | **リアルタイム実行**

1. 実行単位： フローを実行する契機となるアプリの設定。1つの実行単位に対して1つのアプリを設定できます。

ドメインのリアルタイム実行フロー数： **詳細**

実行単位	有効	実行結果	呼び出し元アプリ	実行方法	ログ出力先アプリ デフォルト: ...	更新日時	
1-1. 在庫管理（入荷処理） - Webhook	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1-1 入荷管理	Webhook	ログ出力先アプリ	2020-10-19 14:47	🗑️ 🔄
1-2. 在庫管理（不定期定額発注） - Webhook	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1-2 出荷管理	Webhook	ログ出力先アプリ	2020-10-19 14:47	🗑️ 🔄
1-3. 購買業務（発注） - 任意実行	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1-6 受注管理	API	ログ出力先アプリ	2020-10-19 14:47	🗑️ 🔄
1-4. 購買業務（納品） - Webhook	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1-4 発注管理	Webhook	ログ出力先アプリ	2020-10-19 14:47	🗑️ 🔄

2. データ編集フロー： 実行単位で実行するデータ編集フロー。1つの実行単位から複数のフローを順番に実行することができます。


データ編集フローを追加 | スケジュール実行のフローを利用 | フローをコピー

	データ編集フロー	有効	実行結果	開始日時	終了日時	更新日時	
1	引当/出荷（ステータス変更）-> 在庫レコード追加	<input checked="" type="checkbox"/>	🟢	2020-10-19 14:49	2020-10-19 14:49	2020-10-13 12:08	🗑️ 🔄
2	在庫レコード追加-> 発注レコード追加	<input checked="" type="checkbox"/>	🟢	2020-10-19 14:49	2020-10-19 14:49	2020-10-13 12:09	🗑️ 🔄

## 24. 実行単位の有効化

作成した実行単位のURLを公開し、Webhook通知またはAPIによるリアルタイム実行を有効化します。

- ① 作成した実行単位の有効チェックボックスをチェックし、アプリを更新

全画面表示を終了  アプリを更新

設定ファイル   スケジュール実行   リアルタイム実行


1. 実行単位： フローを実行する契機となるアプリの設定。1つの実行単位に対して1つのアプリを追加

+ 実行単位を追加

実行単位	有効	実行結果	呼び出し元アプリ	実行方法
在庫管理（自動在庫登録～自動発注） 	<input checked="" type="checkbox"/>		出荷管理 ▼	Webhook ▼


- ② ユーザー情報の入力に関するダイアログで [OK] をクリック

情報

 リアルタイム実行のユーザー情報が設定されていません。ユーザー情報を設定してください。

OK

- ③ ユーザー情報画面でkintoneログイン名、パスワードを入力。[アカウントの確認] ボタンをクリックして入力したアカウントが有効なことを確認

全画面表示を終了  アプリを更新

←

ドメイン単位の採番設定

インポート

エクスポート

ライセンス

ユーザー情報

ユーザー情報

リアルタイム実行

リアルタイム実行タブで作成したデータ編集フローを実行するユーザー情報を設定します。

kintone ログイン名

kintone パスワード

Basic認証

☐ Basic認証を使用する

Basic認証ログイン名

Basic認証パスワード

アカウントの確認

セキュアアクセス設定


☐ セキュアアクセス設定を使用する

クライアント証明書ファイル

ファイルを選択 選択されていません

クライアント証明書ファイルのパスワード

- ④ 画面左の戻るボタンをクリック

全画面表示を終了  アプリを更新

←

ドメイン単位の採番設定

インポート

エクスポート

## 25. 出荷管理アプリのWebhook設定（1）

出荷管理アプリでレコードのステータスが変更された際にWebhook通知を送信する設定を行います。

### ① リアルタイム実行タブで実行方法の▼をクリック

設定ファイル スケジュール実行 リアルタイム実行

1. 実行単位： フローを実行する契機となるアプリの設定。1つの実行単位に対して1つのア

+ 実行単位を追加

実行単位	有効	実行結果	呼び出し元アプリ	実行方法
在庫管理（自動在庫登録～自動発注）	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	出荷管理	Webhook ▼

### ② 実行方法ダイアログで、URLをクリップボードにコピーアイコンをクリックし、[OK] ボタンをクリック

実行方法

実行方法: Webhook ▼

URL: https://api.krewdata.grapecity.com/trigger/v1/ /webhook

同時実行時の動作: 要求をキャンセル ▼

① krewDataは複数の実行要求を同時に実行することはできません。[同時実行時の動作]で処理実行中に新たな実行要求が発生した場合の動作を指定できます。

要求をキャンセル: 実行中に新たな実行要求を受け付けません。

実行を待機: 新たな実行要求は実行開始を待機(\*)し、実行中の中の処理が完了した後に実行します。  
(\*) 待機可能な実行数には制限があります。

OK キャンセル

### ③ 呼び出し元アプリの「出荷管理」をクリックして出荷管理アプリを開く

設定ファイル スケジュール実行 リアルタイム実行

1. 実行単位： フローを実行する契機となるアプリの設定。1つの実行単位に対して1つのア

+ 実行単位を追加

実行単位	有効	実行結果	呼び出し元アプリ	実行方法
在庫管理（自動在庫登録～自動発注）	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	出荷管理	Webhook ▼

### ④ 出荷管理アプリの設定メニューで以下を選択 ・カスタマイズ／サービス連携 > Webhook

この一覧の設定を開く

現在の列幅をアプリ設定に保存

受注E

2020-

2020-

2020-

2020-

設定

一般設定 >

通知 >

カスタマイズ／サービス連携 >

アクセス権 >

その他の設定 >

プラグイン

JavaScript / CSSでカスタマイズ

APIトークン

Webhook

## 25. 出荷管理アプリのWebhook設定（2）

出荷管理アプリでレコードのステータスが変更された際にWebhook通知を送信する設定を行います。

### ⑤ Webhook画面の[+]ボタンをクリック

krewData リアルタイム実行チュートリアル > 出荷管理 > アプリの設定 > Webhook

Webhook

Webhookは、レコードの追加や更新などの操作が行われたときに、外部サービスに通知を送信する機能です。  
詳細は [ヘルプ](#) を参照してください。

Webhook URL	説明		
設定済みのWebhookはありません。			

### ⑥ Webhookの追加画面で以下を設定し、[保存] ボタンをクリック

krewData リアルタイム実行チュートリアル > 出荷管理 > アプリの設定 > Webhook > Webhookの追加

Webhookの追加

説明  
在庫管理（自動在庫登録～自動発注）

Webhook URL \*  
https://api.krewdata.grapecity.com/trigger/v1/[ ]/webhook

通知を送信する条件 ②  
☐ レコードの追加
 ☐ レコードの編集
 ☐ レコードの削除
 ☐ コメントの書き込み
 ☒ ステータスの更新

有効化  
☒ このWebhookを有効にする

キャンセル 保存

### ⑦ 「アプリの設定に戻る」をクリックし「アプリを更新」

Webhook

Webhookは、レコードの追加や更新などの操作が行われたときに、外部サービスに通知を送信する機能です。  
詳細は [ヘルプ](#) を参照してください。

Webhook URL	説明		
https://api.krewdata.grapecity.com/trigg...	在庫管理（自動在庫登録～自動発注）	ログを確認	✎ ✕

出荷管理

変更を中止 アプリを更新

フォーム 一覧 グラフ 設定

項目	設定
説明	在庫管理（自動在庫登録～自動発注）
Webhook URL	（②でコピーしたURLをペースト）
通知を送信する条件	ステータスの更新
有効化	チェック

Webhook設定の詳細はkintoneヘルプの「Webhookを設定する」をご参照ください。

[https://jp.cybozu.help/k/ja/user/app\\_settings/set\\_webhook/webhook.html](https://jp.cybozu.help/k/ja/user/app_settings/set_webhook/webhook.html)

## 26. Webhook通知の送信

出荷管理アプリでレコードのステータスを変更し、Webhook通知を送信します。

① 出荷管理アプリのステータスが「未処理」のレコードの「レコードの詳細を表示する」をクリック

スペース: krewData リアルタイム実行チュートリアル アプリ: 出荷管理

一覧

1 - 3 (3件中)

ステータス	品目名	受注日	受注数	出荷日	出荷数
引当	品目A	2020-04-01	50		
出荷	品目A	2020-04-02	100	2020-04-02	100
未処理	品目A	2020-04-20	200		

1 - 3 (3件中)

③ 出荷日、出荷数に以下のデータを入力し [保存] ボタンをクリック

スペース: krewData リアルタイム実行チュートリアル アプリ: 出荷管理 一覧: 一覧 レコード: 品目A

キャンセル 保存

品目名	受注日	受注数	出荷日	出荷数
品目A	2020-04-20	200	2020-04-25	200

- 出荷日 : 2020-04-25
- 出荷数 : 200

② 「レコードを編集する」アイコンをクリック

スペース: krewData リアルタイム実行チュートリアル アプリ: 出荷管理 一覧: 一覧 レコード: 品目A

引当 出荷 現在の作業者を変更

ステータス: 未処理 ステータスの履歴

品目名	受注日	受注数	出荷日	出荷数
品目A	2020-04-20	200		

④ 次のステータスで「出荷」を選択し [実行] ボタンをクリック

スペース: krewData リアルタイム実行チュートリアル アプリ: 出荷管理 一覧: 一覧 レコード: 品目A

引当 出荷 現在の作業者を変更

次のステータス

ステータス: 未 出荷

キャンセル 実行


品目名	受注日	受注数	出荷日	出荷数
品目A	2020-04-20	200	2020-04-25	200

kintoneからWebhook通知が送信され、データ編集フローが実行されます。



## 27. 実行結果の確認

在庫管理アプリにレコードが追加され、ログ出力先アプリに実行ログが登録されていることを確認します。

 ① 在庫管理アプリを開き、レコードが追加されていることを確認

スペース: krewData リアルタイム実行チュートリアル アプリ: 在庫管理


一覧

1 - 4 (4件中)

入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数	
2020-04-07	品目A	180			180	180	 
2020-04-15	品目A	270			450	450	 
2020-04-20	品目A		200		250	450	 
2020-04-25	品目A			200	250	250	 

1 - 4 (4件中)

入出荷日「2020-04-25」、出荷数「200」のレコードが追加されました。実在庫数は通知されたレコードの出荷数を減算し「250」になりました。

 ② ログ出力先アプリを開き、実行ログが登録されていることを確認

スペース: krewData リアルタイム実行チュートリアル アプリ: ログ出力先アプリ

(すべて)

1 - 5 (5件中)

	レコード番号	実行単位ID	実行単位	実行方法	通知ID	イベントの種類	実行ユーザー	アプリID	実行結果	開始日時
	1	1603167...	在庫管理 (不定期定額発注)	Webhook	c6fdfec2-f036...	ステータスの更新		919	成功	2020-10-2

# データ編集フローを即時実行する（参考）

## ① フロー設定画面右上の「実行」をクリック



データ編集フローを即時実行した場合は、呼び出し元アプリから通知されるレコード情報として、即時実行用のパラメータ設定で設定した値が使用されます。

# ご注意点と安全な利用方法

krewDataはアプリのレコードを自動で削除／更新／追加する機能を含んでいます。

## 注意点

krewDataではお客様のアプリのデータを保持しておらず、kintoneではトランザクション機能を有していないため、出力アプリのデータを処理前の状態に復元する機能は提供されていません。処理中に予期しないエラーが発生した場合、利用方法によっては出力アプリのデータの消失や不整合が発生する可能性があるため、以下のような利用方法をお勧めします。

※入力アプリのデータが影響を受けることはありません

## 安全な利用方法

### 1.データを再作成できる状態で運用する

新規作成したアプリを出力アプリに設定する場合、フローを再実行してデータの再作成が可能です  
(入力アプリの対象レコードに変更がない場合)

### 2.実行前にバックアップする

①入出力アプリが同じアプリ、②出力アプリにkrewData以外で追加／更新／削除したレコードが含まれる、いずれかの条件に該当する場合はkrewDataからレコードを再作成できないため、以下の方法をお勧めします。

- 手動で既存データを[外部ファイルに出力](#)しておく
- 自動バックアップ機能を備える[kBackup](#)等の連携製品と併用してバックアップをとる

# リアルタイム実行利用時の重要な注意事項

## 注意点

リアルタイム実行では利用に発生する通信と実行時の認証に関して最大限セキュリティを考慮した仕組みで提供していますが、リアルタイム実行に使用するURLが漏洩した場合はその類ではありません。そのためリアルタイム実行URLについては、お客様ご自身の責任の元、厳重に管理してください。

### ● URL漏洩とリスク

リアルタイム実行URLが漏洩した場合、悪意のある攻撃者によってデータ編集フローを実行される恐れがあります。その場合、攻撃者は出力アプリコマンドに設定しているアプリのレコードをお客様が意図しない内容で登録/更新/削除できます

※攻撃者はレコードの取得はできません。そのためお客様の業務データが漏洩することはありません。

また、攻撃者は対象のリアルタイム実行をお客様が意図しないタイミングと回数で実行できるため、同一のリアルタイム実行を大量に実行できます。krewDataでは安定したパフォーマンスをお客様に提供できるように、同一のリアルタイム実行を1分あたり60回に制限しているため、その上限を超える回数で悪意的に実行された場合、該当のリアルタイム実行を業務上、適切に使用できなくなります。

### ● 悪用された場合の検知

krewDataは大量に同一のリアルタイム実行を実行された場合はそれを検知できる仕組みを提供しています。リアルタイム実行を実行すると、お客様が設定した任意のアプリに実行ログを出力します。大量実行のケースもレコードとしてログアプリに出力し、krewDataのユーザー情報として登録されている管理者に通知します。

- 1分あたり5回を超えて実行した場合  
実行される度に、1分間あたりの制限回数を越えた旨をログ出力し通知します。
- 実行回数上限を超えるリクエストが一定時間内で頻繁に発生した場合  
krewDataは過剰なリクエストとみなし、該当のリアルタイム実行を自動で無効化します。またその旨をログ出力し通知します。無効になったリアルタイム実行URLは呼び出しを行うことができず、それ以降はログも生成されません。

### ● 悪用された場合の対策

漏洩の恐れがある場合や実際に悪用を管理者が検知した場合は、該当のリアルタイム実行を無効にするか、リアルタイム実行URLを再生成してください。

なお、悪用され意図せず登録/更新/削除されたアプリのレコードを悪用前の状態に復元する機能はkrewDataでは提供しておりません。あらかじめご了承ください。

# リアルタイム実行の制限事項

リアルタイム実行には以下の制限事項があります。

- **1分あたりの実行回数**

同一の実行単位における1分あたりの実行回数には上限があります。上限を超えた実行要求はエラーになり、実行単位は実行されません。

- 5回／分

- **処理レコード数**

リアルタイム実行のデータ編集フローの実行時に入力アプリコマンドがkintoneアプリから読み込むことができるレコード数には上限があります。

- お試し版：10,000 レコード
- 製品版
  - 即時実行：10,000 レコード
  - リアルタイム実行：10,000 レコード

※ データ編集フローに複数の入力アプリコマンドを配置している場合は、各アプリの合計レコード数が処理レコード数になります。

※ 出力アプリコマンドによるkintoneアプリへのレコードの書き込み時も上記の上限が適用されます。

- **同時出力可能なレコード数**

複数の実行単位が同時に実行している場合の同時出力可能なレコード数には上限があります。

- 20,000レコード

- **スケジュール実行との同時実行**

スケジュール実行のデータ編集フローが同ドメイン環境で実行中の場合、リアルタイム実行のデータ編集フローは実行されません。リアルタイム実行のデータ編集フローが実行要求された場合は実行を待機し、スケジュール実行が完了した後に実行を開始します。

# + $\alpha$ の学習



# 追加するシナリオ



## 実行単位

作成済

① 自動在庫登録フロー

このフローを追加

② 自動発注フロー

関連するフロー  
を順番に実行

# 在庫管理の業務改善

作成済

**自動在庫登録** 出荷管理アプリの出荷数を在庫管理アプリの在庫数に自動的に反映



出荷管理アプリ

レコード更新  
(ユーザー操作)

ステータス	品目名	受注日	受注数	出荷日	出荷数
出荷	品目A	2020/4/20	200	2020/4/25	200



在庫管理アプリ

入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数
2020/4/7	品目A	180			180	180
2020/4/15	品目A	270			450	450
2020/4/20	品目A		200		250	450
2020/4/25	品目A			200	250	250



在庫レコード  
自動追加

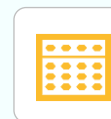
このフローを追加

**自動発注** 品目管理アプリの発注点と在庫を比較して（在庫が不足している場合は）発注レコードを作成



発注管理アプリ

ステータス	発注依頼日	発注日	納品日	品目名	発注数
発注依頼	2020/4/25			品目A	500



品目管理アプリ

品目名	ランク	発注点	発注数
品目A	B	300	500

発注レコード  
自動追加

レコード  
比較





# 完成イメージの確認



# データの集計・加工イメージ

ここからの手順では、以下のデータ集計・加工を行うデータ編集フローを作成します。



## 在庫管理アプリ

入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数
2020/4/7	品目A	180			180	180
2020/4/15	品目A	270			450	450
2020/4/20	品目A		200		250	450
2020/4/25	品目A			200	250	250

## 在庫レコード追加 → 発注レコード追加

在庫管理アプリの実在庫数と品目管理アプリの発注点を比較し、「実在庫数」が「発注点」以下の場合、発注管理アプリに発注レコードを追加



## 発注管理アプリ

ステータス	発注依頼日	発注日	納品日	品目名	発注数
発注依頼	2020/4/25			品目A	500



## 品目管理アプリ

品目名	ランク	発注点	発注数
品目A	B	300	500

レコード  
比較

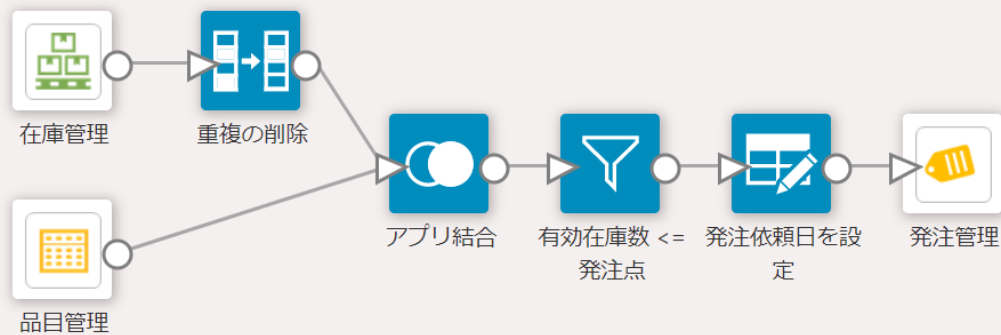


発注レコード  
自動追加



# 作成するデータ編集フロー（完成イメージ）

在庫レコード追加 → 発注レコード追加




# データ編集フローの作成



# 1. データ編集フローの追加作成


在庫管理アプリの実在庫数と品目管理アプリの発注点を比較し、発注管理アプリに発注レコードを追加するデータ編集フローを追加作成します。

① 2. データ編集フローの  データ編集フローを追加をクリックして、新しいデータ編集フローを作成します。

2. データ編集フロー： 実行単位で実行するデータ編集フロー。1つの


 データ編集フローを追加  スケジュール実行のフローを利用  フ

データ編集フロー				
1	≡	⚙	ステータス変更 -> 在庫レコード追加	

② データ編集フローの  をクリックし名前を「在庫レコード追加 -> 発注レコード追加」に変更

2. データ編集フロー： 実行単位で実行するデータ編集フロー。1つの

 データ編集フローを追加  スケジュール実行のフローを利用  フ

データ編集フロー				
1	≡	⚙	ステータス変更 -> 在庫レコード追加	
2	≡	⚙	在庫レコード追加 -> 発注レコード追加	

③ 作成したデータ編集フローの設定アイコンをクリックし、「フロー設定」タブを開く

2. データ編集フロー： 実行単位で実行するデータ編集フロー。1つの

 データ編集フローを追加  スケジュール実行のフローを利用  フ

データ編集フロー				
1	≡	⚙	ステータス変更 -> 在庫レコード追加	
2	≡	⚙	在庫レコード追加 -> 発注レコード追加	

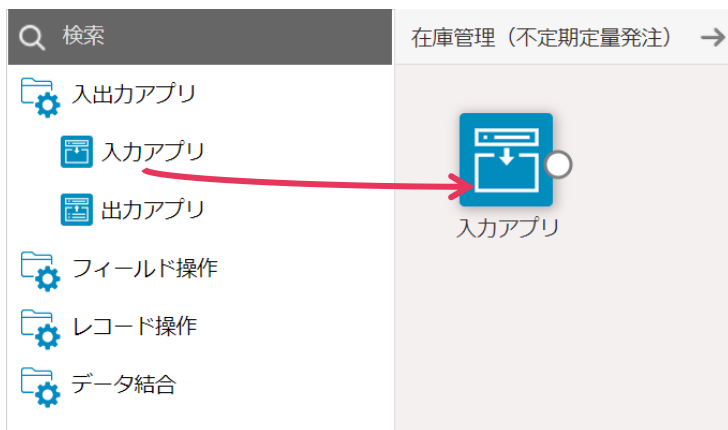


## 2. 入力アプリの選択（在庫管理アプリ）

在庫管理アプリのデータを取得する「入力アプリ」コマンドを追加します。



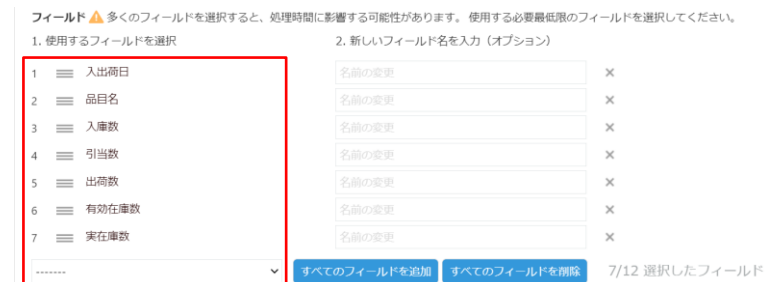
- ① コマンドパネルからフロー作成エリアに「入力アプリ」コマンドを追加



- ② 設定タブで「アプリ選択」ボタンをクリックし、アプリ選択ダイアログで「在庫管理」アプリを選択



- ③ 使用するフィールドを選択
- ・ 入出荷日
  - ・ 品目名
  - ・ 入庫数
  - ・ 引当数
  - ・ 出荷数
  - ・ 有効在庫数
  - ・ 実在庫数



- ④ 「プレビュー」タブで入力アプリのデータを確認 ※

	入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数
1	2020-04-25	品目A			200	250	250
2	2020-04-20	品目A		200		250	450
3	2020-04-15	品目A	270			450	450
4	2020-04-07	品目A	180			180	180

※プレビュータブには、最大500件のレコードが表示されます。

### 3. フィルタの設定（在庫管理アプリ）

Webhook通知で送信されるレコードを入力レコードとするためのフィルタ条件を設定します。

① 設定タブの標準フィルタで「品目名」を選択

在庫管理

アプリを選択

標準フィルタ ?

品目名 = (等しい)

② パラメータ設定ボタンをクリックして「%品目名%」を選択

標準フィルタ ?

品目名 = (等しい) %品目名%

+

フィールド ⚠ 多くのフィールドを選択すると、処理時間に影響する可能性があります。使用する必要最

1. 使用するフィールドを選択

2. 新しいフィールド名を入力（オプション）

1 出入荷日

名前の変更

③ 「プレビュー」タブで入力アプリのデータを確認

在庫管理

在庫管理 設定 プレビュー

	出入荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数
1	2020-04-25	品目A			200	250	250
2	2020-04-20	品目A		200		250	450
3	2020-04-15	品目A	270			450	450
4	2020-04-07	品目A	180			180	180

フィルタ条件で絞り込まれたレコードが表示されます。

## 4. 重複の削除 完了イメージ

在庫管理アプリの品目名が同じレコードの中から、入出荷日が最も新しいレコードを抽出します。

### Before

	📅 入出荷日 ⇅	ABC 品目名 ⇅	123 入庫数 ⇅	123 引当数 ⇅	123 出荷数 ⇅	123 有効在庫数 ⇅	123 実在庫数 ⇅
1	2020-04-25	品目A			200	250	250
2	2020-04-20	品目A		200		250	450
3	2020-04-15	品目A	270			450	450
4	2020-04-07	品目A	180			180	180

### After

	📅 入出荷日 ⇅	ABC 品目名 ⇅	123 入庫数 ⇅	123 引当数 ⇅	123 出荷数 ⇅	123 有効在庫数 ⇅	123 実在庫数 ⇅
1	2020-04-25	品目A			200	250	250

フローの完成イメージ





## 4. 重複の削除 設定方法

### ① コマンドの配置 「重複の削除」をドラッグ



### ② コマンドをつなぐ 線でつなぎます



### ③ 集計方法を設定

※設定タブが表示されない場合はフローのコマンドアイコンをクリックします

### ④ プレビューで意図した設定になっているかを確認

重複の削除

設定

プレビュー

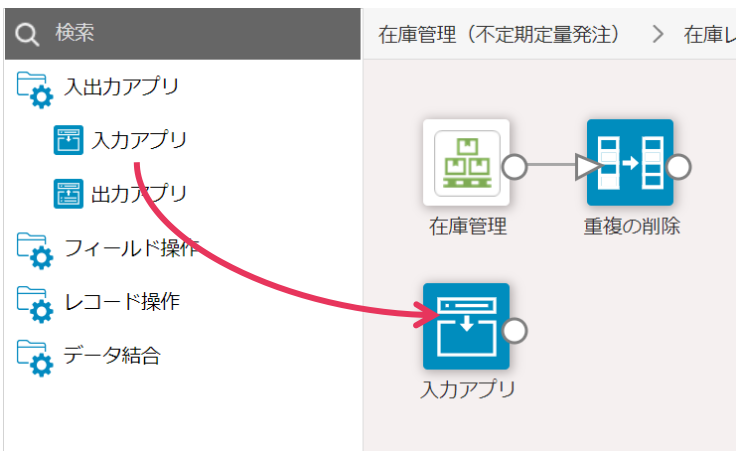
	<div><div></div></div> 入出荷日 ⇅	<div><div>ABC</div></div> 品目名 ⇅	<div><div>123</div></div> 入庫数 ⇅	<div><div>123</div></div> 引当数 ⇅	<div><div>123</div></div> 出荷数 ⇅	<div><div>123</div></div> 有効在庫数 ⇅	<div><div>123</div></div> 実在庫数 ⇅
1	2020-04-25	品目A			200	250	250

項目	設定
1. 重複を検査するフィールド名を選択	品目名
2. 大文字と小文字を区別するか選択	チェックなし
3. ソート	入出荷日 — 降順

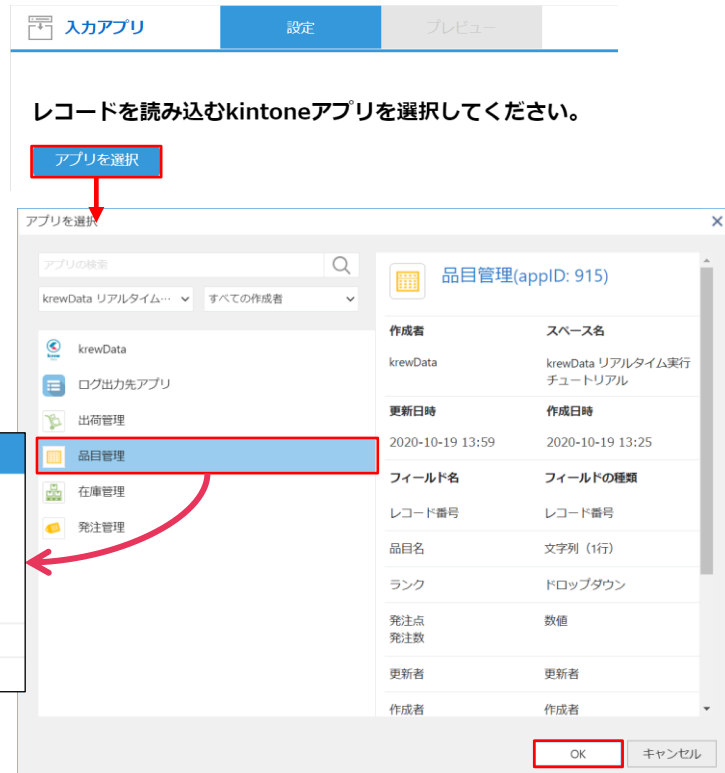
## 5. 入力アプリの選択（品目管理アプリ）

品目管理アプリのデータを取得する「入力アプリ」コマンドを追加します。

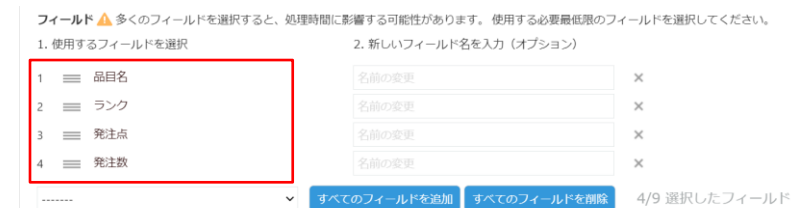
- ① コマンドパネルからフロー作成エリアに「入力アプリ」コマンドを追加



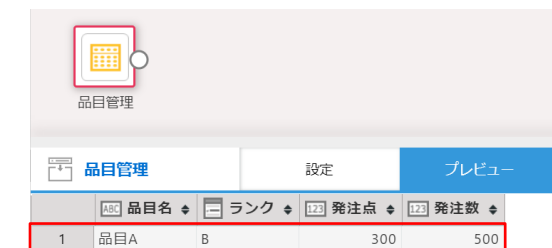
- ② 設定タブで「アプリ選択」ボタンをクリックし、アプリ選択ダイアログで「品目管理」アプリを選択



- ③ 使用するフィールドを選択
- 品目名
  - ランク
  - 発注点
  - 発注数



- ④ 「プレビュー」タブで入力アプリのデータを確認



## 6. フィルタの設定（品目管理アプリ）

Webhook通知で送信されるレコードを入力レコードとするためのフィルタ条件を設定します。

① 設定タブの標準フィルタで「品目名」を選択

品目管理 設定 プレビュー

アプリを選択

標準フィルタ ?

品目名 = (等しい)

② パラメータ設定ボタンをクリックして「%品目名%」を選択

標準フィルタ ?

品目名 = (等しい) %品目名%

フィールド ⚠ 多くのフィールドを選択すると、処理時間に影響する可能性があります。使用する必要最

1. 使用するフィールドを選択 2. 新しいフィールド名を入力（オプション）

1 出入荷日 名前の変更

③ 「プレビュー」タブで入力アプリのデータを確認

品目管理

品目管理 設定 プレビュー

	品目名	ランク	発注点	発注数
1	品目A	B	300	500

フィルタ条件で絞り込まれたレコードが表示されます。

## 7. アプリ結合 完了イメージ

在庫管理アプリと品目管理アプリを結合します。

### Before

在庫管理

	📅 入出荷日	ABC 品目名	123 入庫数	123 引当数	123 出荷数	123 有効在庫数	123 実在庫数
1	2020-04-25	品目A			200	250	250

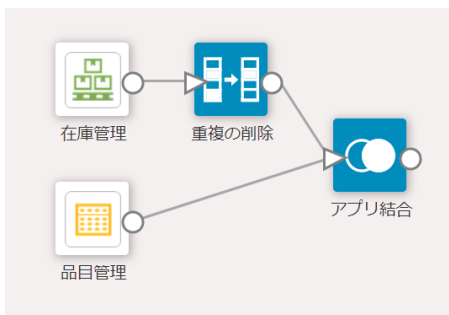
品目管理

	ABC 品目名	📄 ランク	123 発注点	123 発注数
1	品目A	B	300	500

### After

	📅 入出荷日	ABC 品目名	123 入庫数	123 引当数	123 出荷数	123 有効在庫数	123 実在庫数	ABC 品目名_1	📄 ランク	123 発注点	123 発注数
1	2020-04-25	品目A			200	250	250	品目A	B	300	500

フローの完成イメージ

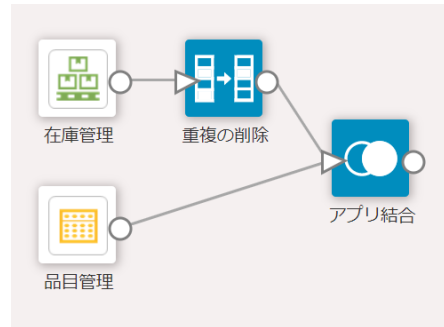


## 7. アプリ結合 設定方法

① コマンドの配置  
「アプリ結合」をドラッグ



② コマンドをつなぐ  
線でつなぎます



③ 結合条件を設定  
[重複の削除] (内部結合) [品目管理]  
品目名

The screenshot shows the 'アプリ結合' (App Link) settings dialog box. It has two tabs: '設定' (Settings) and 'プレビュー' (Preview). The '設定' tab is active. It contains two sections for selecting conditions. The first section, '1. 元になるアプリの条件フィールドを選択' (Select condition field for the source app), has '重複の削除' (Duplicate Elimination) selected. The second section, '2. 結合するアプリの条件フィールドを選択' (Select condition field for the app to be linked), has '品目管理' (Item Management) selected. Both sections show '品目名' (Item Name) as the selected field. A '条件を追加' (Add Condition) button is at the bottom.

④ プレビューで意図した設定になっているかを確認

	入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数	品目名_1	ランク	発注点	発注数
1	2020-04-25	品目A			200	250	250	品目A	B	300	500

※ 結合によりフィールド名が重複する場合は「品目名\_1」のように、名前が自動的に変更されます。

## 8. 有効在庫数 <= 発注点 完了イメージ

フィルタコマンドで「有効在庫数」が「発注点」以下のレコードを抽出します。

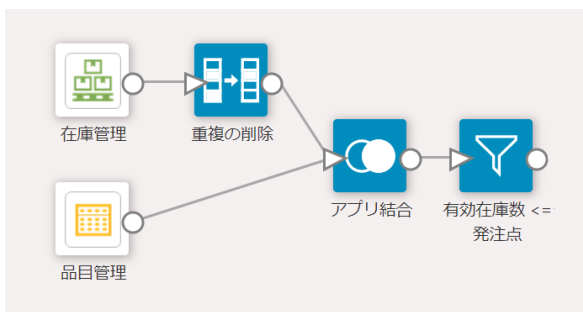
### Before

	入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数	品目名_1	ランク	発注点	発注数
1	2020-04-25	品目A			200	250	250	品目A	B	300	500

### After

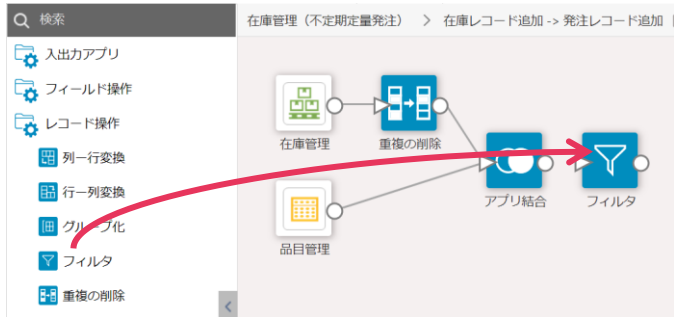
	入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数	品目名_1	ランク	発注点	発注数
1	2020-04-25	品目A			200	250	250	品目A	B	300	500

フローの完成イメージ

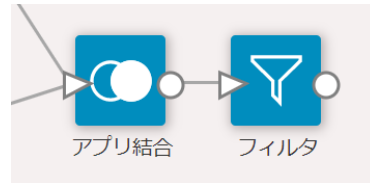


## 8. 有効在庫数 <= 発注点 設定方法

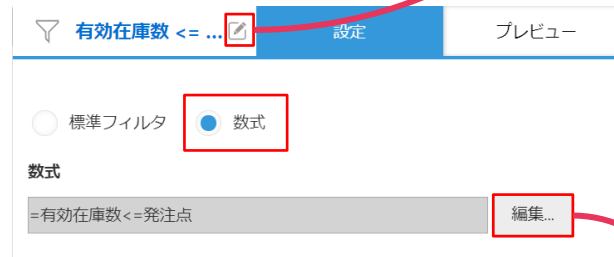
① コマンドの配置  
「フィルタ」をドラッグ



② コマンドをつなぐ  
線でつなぎます



③ 「数式」を選択し、フィルタ条件を数式で設定



数式：  
有効在庫数<=発注点

④ コマンド名もあわせて設定

有効在庫数 <= ...

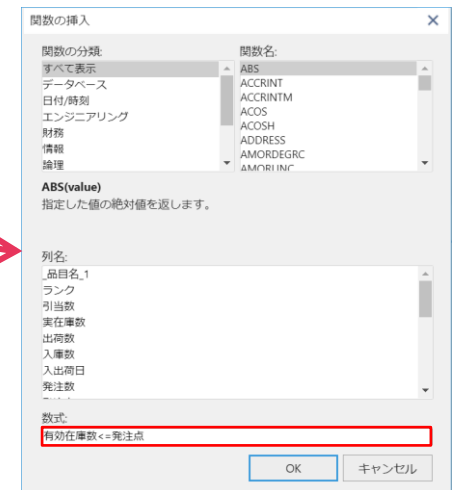
⑤ プレビューで意図した設定になっているかを確認

有効在庫数 <= ...

設定

プレビュー

	<div><div></div><div>入出荷日</div></div>	<div><div>ABC</div><div>品目名</div></div>	<div><div>123</div><div>入庫数</div></div>	<div><div>123</div><div>引当数</div></div>	<div><div>123</div><div>出荷数</div></div>	<div><div>123</div><div>有効在庫数</div></div>	<div><div>123</div><div>実在庫数</div></div>	<div><div>ABC</div><div>品目名_1</div></div>	<div><div></div><div>ランク</div></div>	<div><div>123</div><div>発注点</div></div>	<div><div>123</div><div>発注数</div></div>
1	2020-04-25	品目A			200	250	250	品目A	B	300	500



※数式は関数の挿入  
ダイアログで入力

## 9.発注依頼日を設定 完了イメージ

「発注依頼日」フィールドを作成し、在庫管理アプリの「入出荷日」の値を設定します。

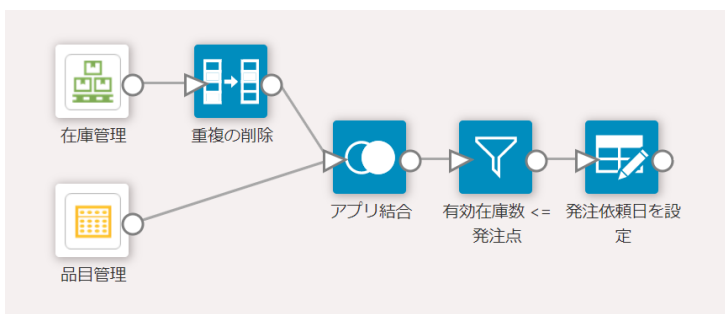
### Before

	📅 入出荷日 ⇅	ABC 品目名 ⇅	123 入庫数 ⇅	123 引当数 ⇅	123 出荷数 ⇅	123 有効在庫数 ⇅	123 実在庫数 ⇅	ABC 品目名_1 ⇅	📊 ランク ⇅	123 発注点 ⇅	123 発注数 ⇅
1	2020-04-25	品目A			200	250	250	品目A	B	300	500

### After

	📅 入出荷日 ⇅	ABC 品目名 ⇅	123 入庫数 ⇅	123 引当数 ⇅	123 出荷数 ⇅	123 有効在庫数 ⇅	123 実在庫数 ⇅	ABC 品目名_1 ⇅	📊 ランク ⇅	123 発注点 ⇅	123 発注数 ⇅	📅 発注依頼日 ⇅
1	2020-04-25	品目A			200	250	250	品目A	B	300	500	2020-04-25

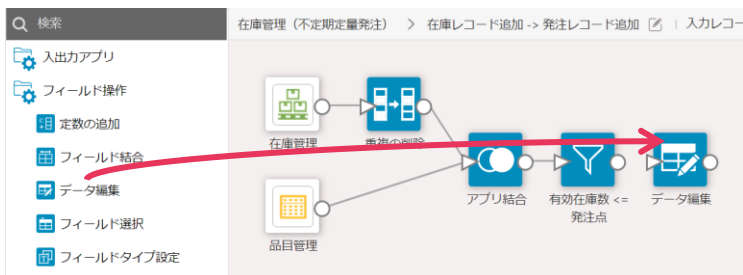
フローの完成イメージ



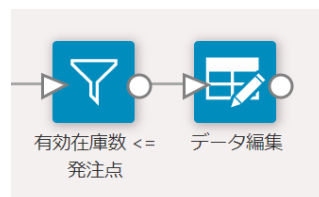


## 9. 発注依頼日を設定 設定方法

① コマンドの配置  
「データ編集」をドラッグ



② コマンドをつなぐ  
線でつなぎます



④ コマンド名もあわせて設定

発注依頼日を設定

③ 編集方法を設定

発注依頼日を設定

1. 結果を保存するフィールドを選択  
新しいフィールドを作成  
新しいフィールド名を入力  
発注依頼日

2. 編集方法を選択  
フィールドをコピー

3. 編集内容を設定  
入出荷日

4. 検索条件  
すべてのレコード

⑤ プレビューで意図した設定になっているかを確認

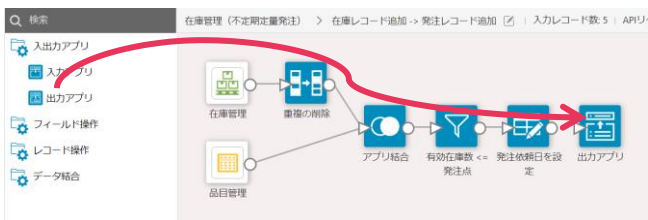
発注依頼日を設定	設定	プレビュー
入出荷日	品目名	在庫数
引当数	出荷数	有効在庫数
実在庫数	品目名_1	ランク
発注点	発注数	発注依頼日
1	2020-04-25	品目A

項目	設定
1. 結果を保存するフィールドを選択	新しいフィールドを作成
新しいフィールド名を入力	発注依頼日
2. 編集方法を設定	フィールドをコピー
3. 編集内容を設定	入出荷日
4. 検索条件	すべてのレコード

# 10. 出力アプリの選択（発注管理アプリ）

結果データを出力する発注管理アプリを選択します。

① コマンドの配置  
「出力アプリ」をドラッグ



② コマンドをつなぐ  
線でつなぎます



③ 「アプリを選択」をクリック



④ アプリ選択ダイアログで「発注管理」アプリを選択



⑤ 出力方法に「追加」を指定



⑥ アプリを更新



# データ編集フローの実行



# 11. 準備事項（在庫管理アプリのレコード削除）

追加したデータ編集フローの動作確認のため、1回目の実行で追加された在庫管理アプリのレコードを削除します。

① 在庫管理アプリの入出荷日が「2020-04-25」のレコードの削除アイコンをクリックし、レコードを削除

スペース: krewData リアルタイム実行チュートリアル アプリ: 在庫管理

一覧

1 - 4 (4件中)

	入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数	
	2020-04-07	品目A	180			180	180	 
	2020-04-15	品目A	270			450	450	 
	2020-04-20	品目A		200		250	450	 
	2020-04-25	品目A			200	250	250	 

削除します。よろしいですか?

キャンセル 削除する

② 同様に入出荷日が「2020-04-20」のレコードの削除アイコンをクリックし、レコードを削除

スペース: krewData リアルタイム実行チュートリアル アプリ: 在庫管理

一覧

1 - 4 (4件中)

	入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数	
	2020-04-07	品目A	180			180	180	 
	2020-04-15	品目A	270			450	450	 
	2020-04-20	品目A		200		250	450	 

削除します。よろしいですか?

キャンセル 削除する

スペース: krewData リアルタイム実行チュートリアル アプリ: 在庫管理

一覧

1 - 2 (2件中)

	入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数	
	2020-04-07	品目A	180			180	180	 
	2020-04-15	品目A	270			450	450	 

1 - 2 (2件中)

最新の有効在庫数が「450」の状態になりました。

## 12. Webhook通知の送信

出荷管理アプリでレコードのステータスを変更し、Webhook通知を送信します。

- ① 出荷管理アプリの受注日が「2020-04-20」のレコードの「レコードの詳細を表示する」をクリック

スペース: krewData リアルタイム実行チュートリアル アプリ: 出荷管理

一覧

1 - 3 (3件中)

ステータス	品目名	受注日	受注数	出荷日	出荷数	
引当	品目A	2020-04-01	50			
出荷	品目A	2020-04-02	100	2020-04-02	100	
出荷	品目A	2020-04-20	200	2020-04-25	200	

1 - 3 (3件中)

- ② レコードのステータスを一度「未処理」に戻します。次のステータスで「未処理」を選択し【実行】ボタンをクリック

スペース: krewData リアルタイム実行チュートリアル アプリ: 出荷管理 一覧: 一覧 レコード: 品目A

未処理 引当 現在の作業者を変更

次のステータス

未処理

キャンセル 実行

注数 200 出荷日 2020-04-25 出荷数 200

- ③ 次のステータスで再度「引当」を選択し【実行】ボタンをクリック

スペース: krewData リアルタイム実行チュートリアル アプリ: 出荷管理 一覧: 一覧 レコード: 品目A

引当 出荷 現在の作業者を変更

次のステータス

引当


キャンセル 実行

注数 200 出荷日 2020-04-25 出荷数 200

kintoneからWebhook通知が送信され、データ編集フローが実行されます。

## 13. 実行結果の確認

在庫管理アプリ、および発注管理アプリにレコードが追加されていることを確認します。

 ① 在庫管理アプリを開き、レコードが追加されていることを確認

スペース: krewData リアルタイム実行チュートリアル アプリ: 在庫管理

一覧


1 - 3 (3件中)

入出荷日	品目名	入庫数	引当数	出荷数	有効在庫数	実在庫数	
2020-04-07	品目A	180			180	180	 
2020-04-15	品目A	270			450	450	 
2020-04-20	品目A		200		250	450	 

1 - 3 (3件中)

入出荷日「2020-04-20」、引当数「200」のレコードが追加されました。有効在庫数は通知されたレコードの引当数を減算し「250」になりました。

このレコードは、1つ目のデータ編集フローの実行結果によって追加されています。

 ② 発注管理アプリを開き、レコードが追加されていることを確認

スペース: krewData リアルタイム実行チュートリアル アプリ: 発注管理

一覧

1 - 1 (1件中)

ステータス	発注依頼日	発注日	納品日	品目名	発注数	仕入先	テーブル	
発注依頼	2020-04-20			品目A	500		表示する▶	 

1 - 1 (1件中)

発注依頼日「2020-04-20」、発注数「500」のレコードが追加されました。発注数は品目管理アプリの「発注数」から取得した値が設定しています。

このレコードは、2つ目のデータ編集フローの実行結果によって追加されています。