

gusuku Customine

カスタマイズの歩き方

ルックアップ編



なまえ

はじめに（この冊子の目的）

使いこなすとても便利な gusuku Customine（以降カスタミンと表記）。しかし、実現できることが多すぎるがゆえに使っていただいているお客様より「難しい」というお声をいただくことがあります。そこで、実際に役立つカスタマイズを作成しながらカスタミンの使い方にも慣れていただくための教材を用意しました。

この冊子では、kintone でも特によく使われる機能である「ルックアップ」の機能を取り上げて、カスタミンの使い方やコツを解説しています。この冊子に沿ってカスタマイズを作成いただくことで、カスタマイズを作成しながらカスタミンの使い方もレベルアップしていただければと思います。

この冊子で理解できること

- 1, 他のアプリからレコードを取得する方法
- 2, 他のアプリにレコードを追加する方法
- 3, ダイアログによってユーザーの入力を補助する方法

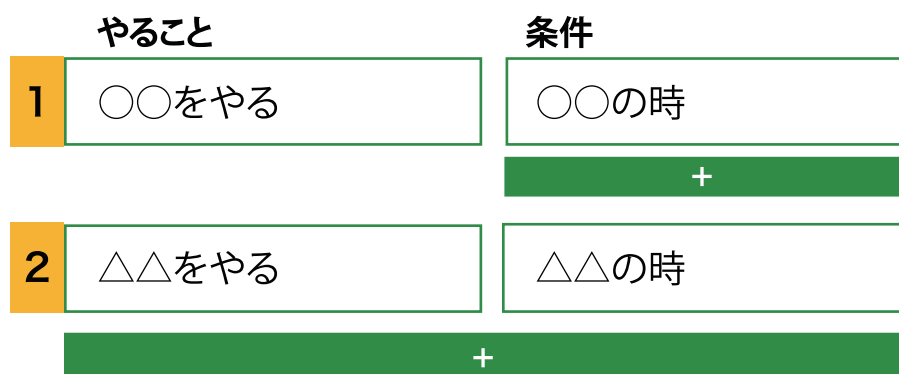
目次

1, はじめに理解していただきたいこと -----	4
2, 実例に学ぶカスタマイズ作成のポイント -----	11
3, うまく動かないときは？ -----	28

1. はじめに理解していただきたいこと

1-1. カスタマインの基本的な構造

カスタマインの基本的な構造は「～の時に～ならば～する」を「やること」と「条件」を組み合わせるという形で設定します。



この「条件」のドキュメント (<https://docs-customine.gusuku.io/ja/conditions/>) には「(追加条件)」と記載されている一覧があります。これは「～ならば」に該当するもので、「条件を追加」の箇所で指定します。

「条件」と「条件を追加」は指定する場所が違うのでご注意ください。

1-2. カスタマインで作成したカスタマイズが動く仕組み

カスタマインで作成したカスタマイズは、ユーザーの画面操作に基づいて動く仕組みとなっています。

1-2-1. 画面ごとの特性

kintone にはいくつかの種類の画面があり、その画面ごとにできることが異なります。この画面ごとの特性を理解することでよりスムーズにカスタマイズを作成できるようになります。

一覧画面は表示が主な目的なので 1 レコード単位の編集しかできず、編集後は「条件：一覧画面を表示した時」に該当しないのでカスタマイズが動作しません。また、一覧の編集画面以外の方法で値を変更した場合はリロードしない限り表示は変わりません。

追加・編集画面はフィールド値を変更することが出来ますが、保存ボタンを押すまでは確定していないのでキャンセルボタンを押すと変更は破棄されます。

詳細画面は kintone の仕様ではフィールド値の更新は出来ませんが、Customine が裏側でレコード更新処理をする事でフィールド値を更新しています。この場合は一覧画面と同じく値は変わっていますが表示は変わらないのでリロードが必要になります。

1-2-2. 同じやることでも画面によって動作が違う

例として、[やること：フィールドに値をセットする]というやること1つをとっても、以下のよう
に違いがあります。

追加画面の場合

フィールドに値がセットされて、表示も変わる

編集画面の場合

フィールドに値がセットされて、表示も変わる

詳細画面の場合

フィールドに値はセットされているが、表示は変わらない

表示を更新するためにはリロードする必要がある

一覧画面の場合

画面に表示されている全レコードのフィールドに値がセットされて、表示は変わらない

表示を更新するためにはリロードする必要がある

一覧画面の鉛筆を押した編集画面

編集状態になっているレコードのフィールドに値がセットされて、表示も変わる

フィールド値がリアルタイムに変わるのは、kintone での操作と同じく追加画面・編集画面のみと
いうことにご注意ください。

コラム：フィールド名とフィールドコード

kintoneの仕組みとして、フィールド値を参照する場合は「フィールドコード」を指定し
ます。

kintoneのフィールドには「フィールド名」と「フィールドコード」があるのでご注意ください。

簡単な見分け方としては、フィールド設定で上にあるのが「フィールド名」で下にあるの
が「フィールドコード」です。

「フィールドコード」を明示的に指定しないと「文字列_1行_1」のようにどのフィールドを指
すのか判別しづらくなります。そのため、フィールド名と同じ名称をつけたり、「フィールド名
_アプリ名」のようにフィールドコードからフィールド名を類推しやすい名前にしておくこと
をお勧めします。

文字列 (1行) の設定

? ヘルプ

フィールド名 *

文字列 (1行)

☐ フィールド名を表示しない

☐ 自動計算する

☐ 必須項目にする

☐ 値の重複を禁止する

文字数 (整数で指定)

最小 最大

初期値

フィールドコード *

文字列_1行 [Z]

キャンセル 保存

1-2-3. 自分が操作しているアプリでカスタマイズは動く

カスタマイズを作る際には「どのアプリで操作をしたときにカスタマイズを動かしたいか」を意識して作成する必要があります。

先程の「フィールドに値をセット」だと、表示しているアプリのフィールドに対してセットされますので、別アプリのフィールドにセットしたい場合は「やること」が変わります。（「レコードを更新する（キーの値をフィールドで指定）」など）

コラム：これから「保存」するレコードのみが対象

カスタマイズで作成したカスタマイズは、今後操作するときに適用されるものであるため、過去に登録したレコードには適用されません。

例えば、レコード保存後にルックアップを自動更新するカスタマイズを作成したとしても、レコードを1つずつ保存していかないとルックアップは更新されません。ですので、カスタマイズ作成前に作成済みのレコードに対して更新しようとする、大変な手間がかかり実用的ではありません。そのため別の仕組みで更新する必要があります。

ルックアップの場合は、ルックアップフィールドの値を更新すれば取得できるので、全レコードを取得してルックアップフィールドだけを書き出せば可能です。また、CSV読み込みで一括更新する方法でも可能です。

同じような話で、検索用文字列作成があります。これもレコード保存時に文字列を作成しますが、過去のレコードは再保存が必要です。そして、検索文字列の場合はCSV読み込みでは作成できないので、何らかのカスタマイズが必要になります。

本冊子では詳細な説明を記載しませんが、既に存在しているデータに対して検索用文字列の生成を行いたい場合には、こちらのページをご参照ください。

大量のレコードを安全に処理する方法：検索文字列を作成する事例

（サポートサイトで「検索文字列を作成する事例」で検索してください）

<https://support.gusuku.io/ja-JP/support/solutions/articles/36000268083>

また、kintone にはユーザーの画面操作を通知する「イベント」という仕組みがあります。カスタマイズで作成したカスタマイズは基本的にはこの「イベント」に基づいて動きます。

1-2-4. イベント

イベントとは、「ユーザーの画面操作したタイミングでプログラムを動かすための kintone の仕組み」であり、一例として以下のようなイベントが定義されています

- 1, レコード追加画面を開いた時
- 2, プロセス管理のアクションを実行した時
- 3, フィールドの値が変わった時

kintone の「イベント」についてはサイボウズのヘルプをご確認ください。

<https://developer.cybozu.io/hc/ja/articles/360000361686>

カスタマイズにおける「条件」は主にこの「イベント」を日本語に置き換えたものです。また、より使い勝手を向上するために kintone のイベント以外のものも「条件」として利用が可能です。

カスタマイズを作成するときには「実現したいこと」とともに「どのイベント」でその処理を実行したいかを意識し、「実現したいこと」→「やること」、「イベント」→「条件」に置き換えていただくとスムーズにカスタマイズが作れます。

例) レコード一覧画面表示後に注意すべき案件を色を変えて目立たせたい

実現したいこと：案件を色を変えて目立たせたい

イベント：レコード一覧画面表示後

↓

やること：フィールド文字色を変更する

条件：一覧画面を表示した時

例) 入力時にユーザーに警告を促すために特定のフィールドの背景色を変えたい

実現したいこと：背景色を変えて警告を促したい

イベント：フィールド値変更

↓

やること：フィールド背景色を変更する

条件：フィールドの値を編集して値が変わった時

また、イベントごとの特性や制約を理解することで、kintone の制約上実現できないことでも代替手段を検討することができます。

kintone の制約によって実現できないことの例をあげますと、

- 1, kintone 側でイベントがない場合
- 2, イベントが発生しない場合
- 3, そのイベントでは実行できないカスタマイズの場合

などがあり、例えば「フォームブリッジでデータが登録されたことを契機にカスタマイズを動かしたい」という要件は、「2. イベントが発生しない場合」に該当するためカスタマイズが動きません。

また、上記以外でもファイルから読み込んだ際にもカスタマイズが動きませんが、これも kintone の特性によるものです。

1-3. 実行順序

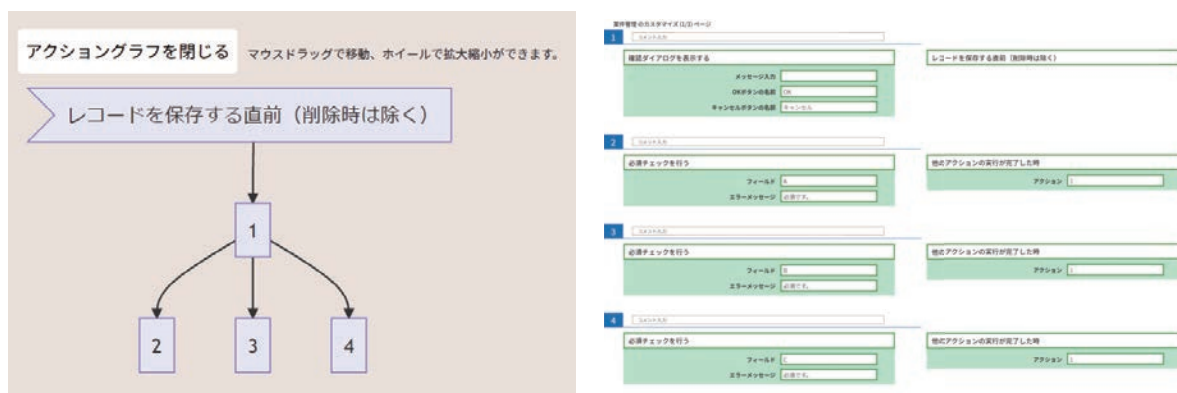
カスタマイズでアクションを作成すると、番号が自動的に付与されます。
これはアクション番号と呼ばれるもので、カスタマイズの実行順とは関係ありません。

全てのアクションは何かの「イベント」を起点に処理が開始され、
「他のアクションの実行が完了した時」や「確認・入力ダイアログで「OK」を押した時」などで繋がっている順に処理されます。

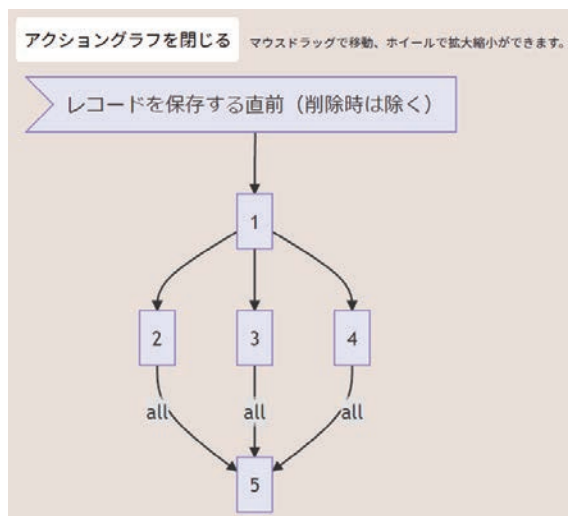
アクションの実行順を確認するにはアクショングラフを表示すると分かりやすいので、
カスタマイズ作成途中や作成後に表示して確認することをおすすめします。
よくあるのが、線で繋がっていないアクションの結果を参照したり、アクションの繋ぎ先を間違えているケースです。

特にアクションを複製した場合に設定を変え忘れているというケースは良くあります。

以下のように＜条件：他のアクションの実行が完了した時＞で同じアクション番号を指定したアクションが複数ある場合は、並行して処理されます。



また、これら複数のアクションが全て終わった時に次のアクションを続ける場合は、＜条件：他のアクションの実行が完了した時＞で全てのアクションを指定してください。



案件管理のカスタマイズ (2/2) ページ(2)

1. コマンド入力

確認ダイアログを表示する

メッセージ入力
OK
キャンセルボタンを配置

2. コマンド入力

必須チェックを行う

フィールド
エラーメッセージ

3. コマンド入力

必須チェックを行う

フィールド
エラーメッセージ

4. コマンド入力

必須チェックを行う

フィールド
エラーメッセージ

5. コマンド入力

フィールドに値を設定する

フィールド
値

レコードを保存する直前（削除時は除く）

他のアクションの実行が完了した時

アクション

その他にも <条件：いずれかのアクションの実行が完了した時> を使うと、指定したアクションのいずれかが完了した時にアクションが処理されます。

例えば、入力した金額によって処理を分ける場合などのように、どれか1つが実行されてその後は同じ処理が続くような場合に使います。

案件管理のカスタマイズ (2/2) ページ(2)

6. コマンド入力

なにもしない

7. コマンド入力

なにもしない

8. コマンド入力

フィールドやグループを表示する

フィールド テーブル

9. コマンド入力

フィールドやグループを非表示にする

フィールド テーブル

レコード追加・編集画面を表示した時またはフィールドの値を編集して値が変わった時

フィールド 状態

詳細画面を表示した時

いずれかのアクションの実行が完了した時

アクション 6,7

フィールド値が特定の値ならば

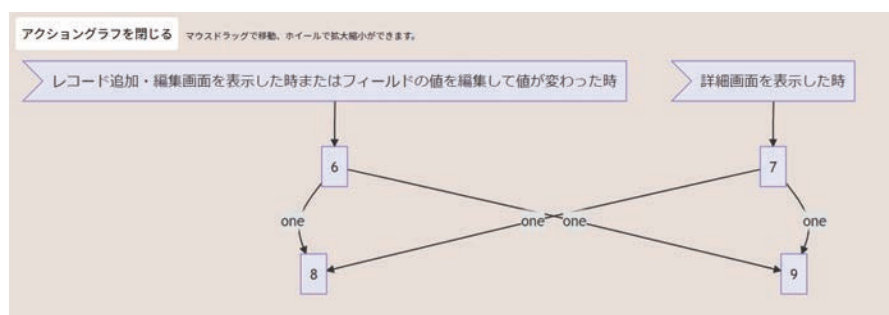
フィールド 状態
条件 等しい
比較値 対応

いずれかのアクションの実行が完了した時

アクション 6,7

フィールド値が特定の値でないならば

フィールド 状態
条件 等しい
比較値 対応



1-4. 本書で使用する用語

本書ではカスタマインの画面や用語を以下のように記載します。

1 コメント入力

アクション

フィールドを無効化する

フィールド 会社名

やること

追加画面・編集画面を表示した時

一覧画面の鉛筆を押した時 一覧画面の鉛筆を押した時は含めない

条件

条件を追加

アクション

カスタマインの設定における 1 単位です。「やること」と「条件」を組み合わせた 1 セットが 1 つのアクションとなります。

やること

カスタミン設定画面の左側で選択する「やること」は、本書では [やること：フィールドを無効化する] のように記載します。

条件

カスタミン設定画面の右側で選択する「条件」は、本書では <条件：一覧画面を表示した時> のように記載します。

追加条件

「条件」の下部で「条件を追加」を選択して設定する「追加条件」は、本書では <追加条件：フィールド値が特定の値ならば> のように記載します。

アクション番号

カスタミン設定画面で各アクション（設定）の左上の数字です。本書内で【アクション：1】と記載した場合には、アクション番号 1 番の設定を指します。

1 コメント入力

アクション番号 1: 無効化する

フィールド 会社名

2. 事例に学ぶカスタマイズ作成のポイント

2-1. ルックアップとは

「ルックアップ」とは kintone の標準機能で配置できるフィールドの 1 つで、ルックアップフィールドを配置すると、他のアプリにあるレコードの情報を参照してデータを取得（コピー）することができます。同じ内容のデータを繰り返し入力せずに済むことで、入力ミスを防ぎ、入力時間を短縮できます。

たとえば、案件管理アプリにルックアップフィールドを配置して、関連付けるアプリとして顧客管理アプリを指定します。これにより、案件管理アプリでレコードを登録するときに、顧客管理アプリのデータをコピーして入力できるようになります。

顧客名	部署名	担当者名
金都運総研	情報システム部	下山 達士
戸田ネットソリューションズ	開発本部	川崎 丈史
田端食品	総務部	川越 憲一
株式会社中川工業	営業部	戸塚 瑞子
池田学院大学	情報システム部	小寺 樹里
寺島総合法律事務所	営業部	星野 長利

この例では、案件管理アプリにおける「顧客名」は「ルックアップ」フィールドで、「部署名」「担当者名」は「ほかのフィールドのコピー」として設定されており、ユーザーは入力する必要がなく顧客名をコピーする時に同時に顧客管理アプリからコピーされます。

フィールド名 *

顧客名

☐ フィールド名を表示しない

☒ 必須項目にする

関連付けるアプリ *

顧客管理 (営業支援パック)

コピー元のフィールド *

顧客名

ほかのフィールドのコピー

部署名 < 顧客管理 (営業支援パック) ...

担当者名 < 顧客管理 (営業支援パック) ...

2-2. ルックアップのよくある課題・注意点

「2-1. ルックアップとは」で紹介したように非常に便利なルックアップの機能ですが、使っていると少し物足りないと感じるポイントや注意しなければいけないポイントがいくつかあります。以下に具体例とともに紹介します。

課題 i. ルックアップで取得した値はその時点のコピーである

ルックアップの機能を使用して他のアプリから取得した値は、「取得」ボタンを押した時点のコピーです。そのため、コピー元のデータに変更があった場合、再度「取得」ボタンを押さない限り、コピー先では反映されません。

案件管理アプリ

顧客名
金都運総研

部署名
情報システム部

担当者名
下山 達士

登録時

顧客管理アプリ

顧客名	部署名	担当者名
金都運総研	情報システム部	下山 達士
戸田ネットソリューションズ	開発本部	川崎 丈史
田端食品	総務部	川越 憲一

↓

案件管理アプリ

顧客名
金都運総研

部署名
情報システム部

担当者名
下山 達士

担当者名変更時

担当者名が書き換わらない

顧客管理アプリ

顧客名	部署名	担当者名
金都運総研	情報システム部	柏田 舞
戸田ネットソリューションズ	開発本部	川崎 丈史
田端食品	総務部	川越 憲一

課題 ii. ルックアップは存在するデータしか取得できない

ルックアップでデータを取得するときには、必ずコピー元のアプリにデータが存在している必要があります。

顧客名 *

取得 クリア

データがありません。

部署名

担当者名

コピー元のアプリに存在しない場合には上のようなエラーとなり、一度コピー元アプリを開いて該当のレコードを登録した後に、改めて「取得」ボタンを押す必要があり非常に手間がかかります。

課題 iii.1 つの条件でしか検索できない

ルックアップでレコードを取得する場合、ルックアップフィールドに設定した値で検索して絞り込むことが可能です。ただし、それ以外の条件を設定できないため一致するレコードが大量に存在する場合は以下のようにたくさんのレコードが候補として表示されます。

顧客管理 (営業支援パック) ×

🔍
全 6 件

	顧客名	部署名	住所
選択	金都運総研	情報システム部	岐阜県岐阜市 ××××
選択	金都運総研ソリューションズ	開発本部	栃木県宇都宮市 ××××
選択	金都運総研商会	総務部	埼玉県浦和市 ××××
選択	金都運	営業部	静岡県熱海市 ××××
選択	金都運事務所	情報システム部	神奈川県小田原市 ××××

キャンセル

例えば上の例では「金都運」が含まれる全ての顧客が表示されてしまいます。ルックアップの機能では複数条件の絞り込みができないため、このままだと自身の担当顧客を探すのに手間がかかります。

2-3. カスタマインを用いた解決策

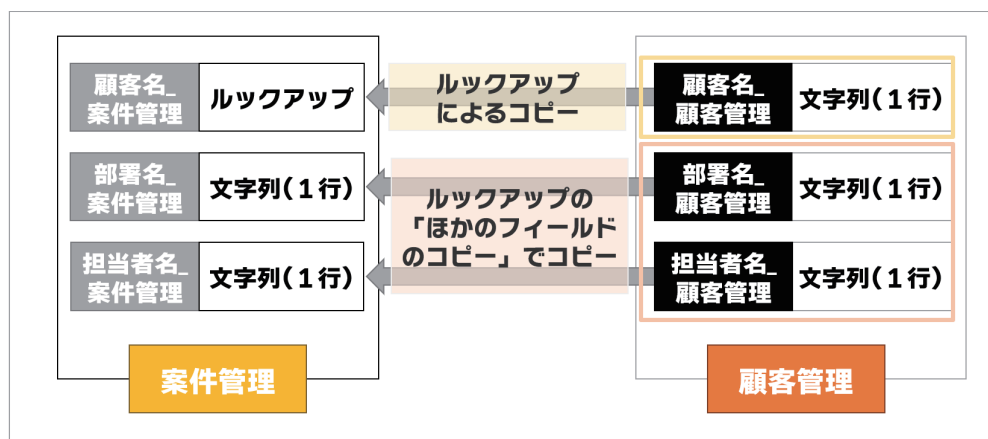
「2-2. ルックアップのよくある課題・注意点」で紹介した課題ですが、カスタマインを使うと簡単に解決することが可能です。課題別に解決策を見ていきましょう。

課題 i. 「ルックアップで取得した値は「その時点のコピー」である」の解決策

想定するアプリ構成

課題 i では、kintone アプリストアにある「営業支援パック」を使用してカスタマイズを作成します。「営業支援パック」では各アプリ間には以下のような関係となっています。

※フィールドコードを分かりやすくするため、フィールドコードは「フィールド名_アプリ名」に変更しています。



この課題を解決するためにはコピー元のアプリを変更した時に、「コピー先アプリで『取得』ボタンを再度押す」ことをカスタマイズで実現する必要があります。

まずは、これを実現するためのカスタマイズを紹介します。

カスタマイズはたったこれだけです。このカスタマイズによって、「顧客管理」アプリ側でレコード保存した時に、ルックアップフィールドが存在している「案件管理」アプリ側でルックアップフィールドの「取得」ボタンを押したのと同じ動きになります。

設定自体は非常に簡単ですが、このカスタマイズを作成するにもいくつかのポイントがあります。

POINT1. どのアプリにカスタマイズを入れるか？

カスタマインでカスタマイズを行う場合、どのアプリにカスタマイズを入れるか？を意識して作成する必要があります。これは、『1-2-3. 自分が操作しているアプリでカスタマイズは動く』に記載の通り、カスタマイズはユーザーが開いているアプリで動くためです。

今回実現したいカスタマイズは「案件管理」と「顧客管理」の2つのアプリが関連するため、どちらにカスタマイズを適用するべきか悩むかもしれません。ここでどちらのアプリにカスタマイズを入れるべきか考えてみます。

まず、実現したいことを図で示すと以下のようになります。



では、これを踏まえて考えてみましょう。

「案件管理」アプリにカスタマイズを入れる場合

これは、＜条件：詳細画面を表示した時＞を使用してユーザーが詳細画面を開いた時に動くカスタマイズを作成することで実現可能です。ただ、これだと一覧画面を開いた時点ではまだレコードが書き換わっていません。また、カスタマイズを動かすために書き換わった「顧客名_案件管理」を使用する全レコードを1件1件開くのは現実的ではありません。

では、＜条件：一覧画面を開いた時＞に変更するとどうでしょうか。この場合でも、どのレコードのルックアップコピー元が書き換わったかを判別する手段がないため、全レコードを更新するカスタマイズを作成する必要があります。全レコードの更新は処理的に時間がかかるだけでなくAPIの消費も多くなる懸念もあるため最適な方法とは言えません。

また、「案件管理」アプリの「顧客名_案件管理」フィールドを関連レコードで他アプリに表示している場合や表やグラフを使っている場合には、一覧画面を開くまでは古いレコードのまま更新されないため、問題となるケースがあります。このことから、「案件管理」アプリにカスタマイズを入れるのは良い方法とは言えません。

「顧客管理」アプリにカスタマイズを入れる場合

反対に、「顧客管理」アプリはユーザーが「顧客名_顧客管理」を変更する時に必ず操作するアプリです。そのため、「顧客管理」アプリ側でユーザーが操作したタイミングで動くカスタマイズを実装すれば、漏れがなくなります。

このように複数のアプリが関連するカスタマイズを作成する場合には、「どちらのアプリで操作した時にデータを反映したいか」を考えて、それに合わせたカスタマイズを作成いただくのが良いと良いです。

POINT2. どの条件を使うか？

POINT 1 で記載したとおり、今回のカスタマイズは「顧客管理」アプリに入れます。次に考えるのは、「どの条件を使うか？」です。

使えそうな選択肢としてイメージできるものは

- ・フィールドの値が変わった時
- ・保存前
- ・保存後

があると思います。このうち、どれを使うのが最適か考えてみます。

●フィールドの値が変わった時

<条件：フィールドの値が変わった時> で実行するメリットは、ほぼリアルタイムに処理が実行される点があります。そのため、これが一番良いのでは？と思ってしまいがちです。

しかし、この条件で動かすと

- ・入力した際に即時処理が実行されるため、誤って入力した際などに無駄な更新が発生する
- ・レコードを保存せずキャンセルした場合でも更新されてしまう
- ・上記による API 消費が増大する

といった問題があるため、適しているとは言えません。

●レコードを保存する直前

<条件：レコードを保存する直前（削除時は除く）>だと全ての入力が終わっているためこれの良いような気がします。ただし、この場合にも注意点があります。必須チェックのエラー、型のエラー、同時編集のエラーなどがあったときレコードの保存ができていないのにカスタマイズが動いてしまうため、こちらも適していません。

●レコードを保存した直後

今回のカスタマイズのように他のアプリを更新する用途で作成するカスタマイズは<条件：レコードを保存した直後（削除時は除く）>を使うのがオススメです。先述のように<条件：レコードを保存する直前（削除時は除く）>はエラーなどで保存に失敗した時にも動いてしまうため、確実に保存に成功したときのみ動かすために使うのが<条件：レコードを保存した後（削除時は除く）>です。

上記で記載した3つの条件の特性を簡単に纏めます。

条件	メリット	デメリット	適している用途
フィールドの値が変わった時	即座に処理を実行できる	誤入力時やキャンセル時など不要な場合にも処理が実行される	フィールドの表示・非表示や必須チェックなどレコードを操作せず、入力を補助するような処理
レコードを保存する直前	保存時に行いたい一連の処理をまとめて実行できる	エラー発生時にも実行されてしまう	自アプリ内での計算処理
レコードを保存した直後	レコード保存が完了した後のみに確実に処理を実行できる	自レコードに対する更新処理には向いていない	他アプリへのレコード更新

このカスタマイズにおけるその他注意点

i. 「値の重複を禁止する」にチェックが入っている必要がある

ルックアップフィールドを更新する場合、コピー元のフィールドで「値の重複を禁止する」の設定を行う必要があります。今回の場合、「顧客管理」アプリの「顧客名_顧客管理」フィールドで「値の重複を禁止する」の設定が必要です。

ii. ルックアップのキーが変わった場合にはカスタマイズは動かない

このカスタマイズは「顧客名_顧客管理」に入力されている値と同じ値を持つ「案件管理」アプリ内のレコードで、ルックアップの「取得」ボタンを押すことを自動で行うためのカスタマイズです。

そのため、「顧客名_顧客管理」の値自体が変更になった場合にはこのカスタマイズを使用することはできません。

ページを分けましょう

次のカスタマイズに移る前に、「ページ」を分けましょう。カスタマイズの設定は数が増えるとページ自体が縦長になり視認性が悪くなるだけでなく、メンテナンス時にもどこを修正すべきかわかりづらくなるため、異なる要件のカスタマイズを作成するときにはページを分けることをお勧めします。

画面右上の「+」ボタンを押すと新しいページが作成されます。また、ページごとに名前をつけておくことでページ数が増えたときにも目的のカスタマイズを探しやすくなると思います。この例では「ルックアップ自動更新」という名前をつけています。

[gusuku 共通管理](#) [ドキュメント](#) [サポートサイト](#)

[サインアウト](#)



※注意点

ページを跨ったカスタマイズを作成することはできません。例として設置したボタンや他のアクションを参照する場合には、同一ページ内でアクションを作成する必要があります。

課題 ii. 「ルックアップは存在するデータしか取得できない」の解決策

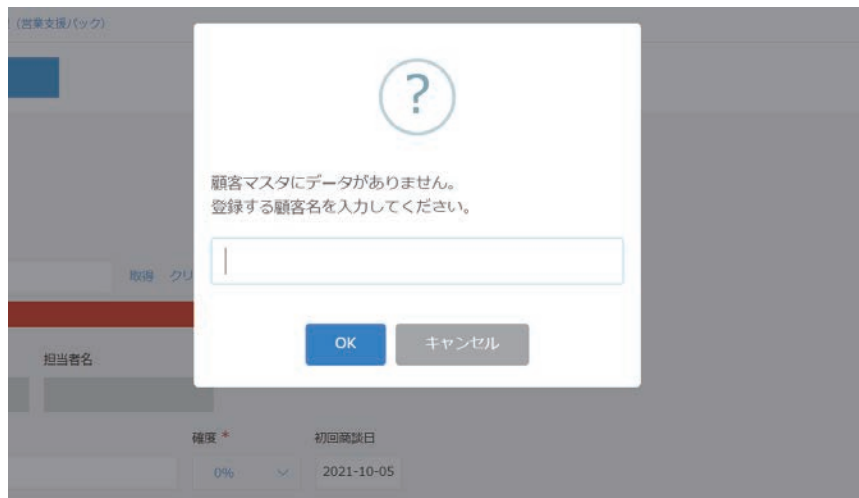
想定するアプリ構成

課題 ii では、課題 i に引き続き「営業支援パック」を使用します。

次に「ルックアップは存在するデータしか取得できない」という課題を解決するためのカスタマイズを作成していきましょう。この課題を解決するためには「取得時にマスタに存在しなかったら、レコードを追加する」というカスタマイズを行います。

これを実現するカスタマイズ例は以下のとおりです。

このカスタマイズによって、ルックアップでエラーとなった場合には以下のようなダイアログを表示し、ダイアログに入力された内容に従って「顧客管理」アプリへのレコード追加を実現することが可能です。



それではこのカスタマイズにおけるポイントを順に見ていきましょう。

POINT1. どの条件を使うか？

今回の要件は「マスタに無い時にレコードを追加する」です。これをカスタマイズの条件に置き換えると <条件：ルックアップで取得が失敗した時> になります。これを使うことで取得に成功したときには動かず、取得に失敗したとき (= マスタにデータが無い時) だけに動くカスタマイズとなります。条件に関する詳しい説明は『1-2-4. イベント』をご参照ください。

ルックアップフィールドでは使用できない条件

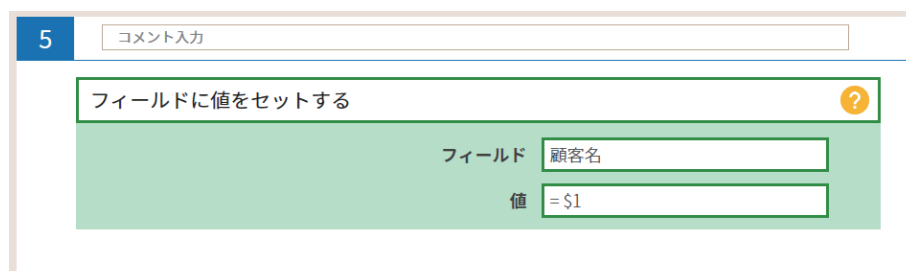
kintone の制約によりルックアップフィールドでは<条件：フィールドの値が変わった時>を使用することはできません。ルックアップフィールドの変更をきっかけに処理を動かしたい場合は、今回のように <条件：ルックアップで取得が失敗した時> や <条件：ルックアップで取得が成功した時> を使う点に注意してください。

POINT2. ダイアログを使用する

カスタマイズでは入力補助の仕組みとしてダイアログを作成することが可能です。ダイアログは全部で8種類あり、単純に情報を出力するための「情報ダイアログ」や OK/ キャンセルを選択させる「確認ダイアログ」などがあります。

今回使用するものは「入力ダイアログ」です ([やること：入力ダイアログを表示する])。これを使用するとユーザーが任意のテキストを入力することができ、入力された内容は \$1 のようにアクション番号を指定することで参照が可能です。

今回のカスタマイズでは【アクション：5】にてフィールドにセットする値が「\$1」となっていますが、これは【アクション：1】の入力ダイアログで入力された内容を参照しています。



POINT3. マッピング画面の使用法

他アプリへレコードを追加・更新する時や自アプリにおいても複数のフィールドに一度に値をセットしたい場合には、「マッピング」という設定項目で設定を行います。

マッピング画面は以下のような構成になっています。

フィールドマッピング (顧客管理 (営業支援バック))

① セットする値

② セット先のフィールド

③

顧客名 文字列 (1行)

部署名 文字列 (1行)

担当者名 文字列 (1行)

郵便番号 文字列 (1行)

住所 文字列 (1行)

TEL リンク

FAX リンク

メールアドレス リンク

備考 文字列 (複数行)

{式}記法はサポートしていません。 ? フィールドマッピングの書き方 ? 式の書き方

OK キャンセル

画面の要素を簡単に説明しますと以下の通りです。

- ①：入力補助用プルダウン
- ②：入力エリア
- ③：適用先フィールド

基本的な動きとしては「②に入力された値が③のフィールドにセットされる」です。それぞれの動きについてももう少し詳細に説明します。

③に表示されているフィールド名は使用するやることによって異なりますが大きく2つのパターンに分かれます。レコードの追加や更新で使用する際には「アプリ名」で指定したアプリのフィールドが表示されます。今回だと、レコード追加先として指定している「顧客管理」アプリのフィールド一覧が表示されています。[やること：フィールド値をまとめてセットする] など自アプリに適用するやることを使用する場合には、③には自アプリのフィールド一覧が表示されます。

②は③で表示されているフィールドに何をセットするか?を指定するための領域です。また、①は②への入力を補助するためのプルダウンメニューになっています。

例として、以下のようにプルダウンで「フィールド」を選択した場合には、アプリのフィールドに既に入力されている値を参照することができます。今回だと「案件管理」アプリで現在開いているレコードのフィールドの値を使用できます。

セットする値

挿入

フィールド

他のアクションの結果

関数

挿入

セット先のフィールド

顧客名 文字列 (1行)

部署名 文字列 (1行)

担当者名 文字列 (1行)

郵便番号 文字列 (1行)

上の画面で「フィールド」を選択すると、以下のようにフィールド一覧が表示されるため、使用したいフィールドを選択するとマッピング画面にセットされます。

検索			
フィールド名	フィールドコード	テーブル	フィールドタイプ
☐ レコード番号	レコード番号		レコード番号
☒ 顧客名	顧客名_案件管理		ルックアップ（文字列）
☐ 部署名	部署名_案件管理		文字列（1行）

セットする値		セット先のフィールド	
挿入 v	= 顧客名_案件管理	➤ 顧客名	文字列（1行）
挿入 v		➤ 部署名	文字列（1行）
挿入 v		➤ 担当者名	文字列（1行）

この状態で、「案件管理」アプリの「顧客名」フィールドに入っている値が、顧客管理アプリの顧客名にセットされます。

挿入プルダウンで「他のアクションの結果」を使用すると、アクション番号を指定して他のアクションの実行結果を参照することができます。今回のカスタマイズのように [やること：入力ダイアログを表示する] のアクションを指定すると、ユーザーがダイアログで入力した値を参照できます。

セットする値		セット先のフィールド	
挿入 v		➤ 顧客名	文字列（1行）
フィールド		➤ 部署名	文字列（1行）
他のアクションの結果		➤ 担当者名	文字列（1行）
関数		➤ 郵便番号	文字列（1行）
挿入 v			

選択したアクションの結果値を式に追加します。

	アクション番号	やること	コメント / パラメーター
☒	1	入力ダイアログを表示する	メッセージ入力=顧客マスタにデータがありません。登録する顧客名を入力してください。 ,OKボタンの名前=OK,キャンセルボ...
☐	2	入力ダイアログを表示する	メッセージ入力=部署名を入力してください。 ,OKボタンの名前=OK,キャンセルボタンの名前=キャンセル,空を許可するかど...
☐	3	入力ダイアログを表示する	メッセージ入力=担当者名を入力してください。 ,OKボタンの名前=OK,キャンセルボタンの名前=キャンセル,空を許可するか...
☐	4	レコードを追加する	追加先アプリ=顧客管理（営業支援パック） ,マッピング=顧客名_顧客管理=\$1 部署名_顧客管理=\$2 担当者名_顧客...

今回のカスタマイズでは以下のように設定されており、これによって【アクション：1,2,3】の結

果を参照してレコード追加を行うことが可能です。

セットする値		セット先のフィールド	
挿入	= \$1	顧客名	文字列 (1行)
挿入	= \$2	部署名	文字列 (1行)
挿入	= \$3	担当者名	文字列 (1行)
挿入		郵便番号	文字列 (1行)
挿入		住所	文字列 (1行)
挿入		TEL	リンク
挿入		FAX	リンク
挿入		メールアドレス	リンク
挿入		備考	文字列 (複数行)

\$[式]記法はサポートしていません。 ? フィールドマッピングの書き方 ? 式の書き方

OK キャンセル

また、今回のカスタマイズでは使用しませんが = “文字列” と入力することで固定値をセットすることも可能です。

セットする値		セット先のフィールド	
挿入	= "アールスリーインスティテュート"	顧客名	文字列 (1行)
挿入		部署名	文字列 (1行)
挿入		担当者名	文字列 (1行)

上記の例では、常に「アールスリーインスティテュート」という値が顧客名としてセットされます。ラジオボタンやドロップダウンなど選択肢の中の値を指定したい場合では有用です。

その他、式の使用方法についてはこちらをご覧ください

(ドキュメントで「カスタマイズ」 - 「やること一覧」 - 「式の記述方法」を御覧ください)

<https://docs-customine.gusuku.io/ja/actions/expression/>

関数についてはこちらをご覧ください

(ドキュメントで「関数一覧」を御覧ください)

<https://docs-customine.gusuku.io/ja/functions/>

POINT4. アクションごとの繋げ方

複雑なカスタマイズを作成する場合には、1つのやることのみでは完結せず複数のやることを繋げてアクションを作成する必要があります。『1-3. 実行順序』に記載の通り、複数のアクションを繋げて動かす場合には、<条件：他のアクションの実行が完了した時> や <条件：いずれかのアクションの実行が完了した時> を使います。

また、[やること：入力ダイアログを表示する] など、OK とキャンセルのいずれかを選択できるやることの後には「OK」が押された時と「キャンセル」が押された時で動きを変えたいケースが多いため、<条件:確認・入力ダイアログで「OK」を押した時> や <条件:確認・入力ダイアログで「キャンセル」を押した時> を使用します。

POINT5. ルックアップを取得し直す

kintone の仕様により、ルックアップフィールドは新たに値を入れた場合には必ず「取得」ボタンを押した後に保存する必要があります。これを自動化するためのやることが【アクション：6】で使用している [やること：ルックアップを取得し直す] です。

やってみましょう：[やること：ルックアップを取得し直す] を無効にして動かしてみましょう。

試しに【アクション：6】を無効にした状態で動作確認をしてみましょう。そうすると、値のセットまでは問題なく動きますが以下のようなエラーが表示され、保存ができません。一度ルックアップの「取得」ボタンを押すと保存できるようになります。

[取得]を押し、参照先からデータを取得してください。

POINT6. 意図したとおりに動かない場合には

今回のカスタマイズのように複数のアクションから構成される場合、意図したとおりに動かないケースが出てくると思います。意図したとおりに動かない場合には闇雲に設定を変更するのではなく、『3. うまく動かないときは?』に従って解決を目指しましょう。

やってみましょう

練習問題：1

顧客管理アプリへの登録時に、「登録日」フィールドに本日の日付を入力するカスタマイズを作成してみましょう。「登録日」フィールドは顧客管理アプリに「日付型」で予め追加しておいてください。

ヒント：ドキュメントで today 関数を調べてみましょう

練習問題：2

顧客管理アプリへレコードを追加した後、自動的に別タブで顧客管理アプリを開くカスタマイズを作成してみましょう。

ヒント：他アプリを開くやることは、ドキュメントで「遷移」で探してみましょう。

レコードを追加するのアクション番号を指定することで、追加したレコードのレコード番号の取得が可能です。

練習問題の解答は巻末に記載しています。

課題 iii. 「1つの条件でしか検索できない」の解決策

この課題は標準のルックアップフィールドでは解決できません。そのため、ルックアップフィールドと同等（以上）の機能を持つ、「ボタンを押したら他アプリから指定した条件でレコードを取得する」仕組みをカスタマイズで作成していきます。

まずは、これを実現するためのカスタマイズを紹介します。

1	コメント入力
ボタンをスペースに配置する	追加画面・編集画面を表示した時
場所 lookup	一覧画面の鉛筆を押した時
ラベル 顧客情報取得	一覧画面の鉛筆を押した時は含めない
上下位置合わせ 指定なし	

2	コメント入力
全レコードを取得する	ボタンを押した時
取得先アプリ 顧客管理	ボタン 1

3	コメント入力
取得したレコードを絞り込む	他のアクションの実行が完了した時
元のレコード 2	アクション 2
条件になるフィールド 顧客区分	
条件 等しい	
比較値 = 顧客区分	

4	コメント入力
レコードの一覧をポップアップで表示する	他のアクションの実行が完了した時
レコード 3	アクション 3
表示するフィールド 顧客区分, 顧客名, 部署名	
選択方法 単一選択（ラジオボタン）	
メッセージ入力 <入力されていません>	
OKボタンの名前 OK	
キャンセルボタンの名前 キャンセル	
未選択を許可するかどうか 未選択でもOKを許可する	
テーブルの幅（ピクセル） <入力されていません>	

5	コメント入力
フィールド値をまとめてセットする	確認・入力ダイアログで「OK」を押した時
マッピング 顧客名= \$4(0).顧客名 部署名= \$4(0).部署名 担当者名= \$4(0).担当者名	確認・入力ダイアログアクション 4

上記のカスタマイズによって以下のように顧客区分の選択内容に応じてポップアップを表示し、ポップアップで選択された値をセットする動きとなります。

顧客区分 *

☒ エンドユーザー ☐ パートナー

顧客情報取得

顧客名 部署名 担当者名

顧客名 部署名 担当者名

顧客区分 顧客名 部署名

☐	エンドユーザー	新山物産	営業本部第一営業部
☐	エンドユーザー	岡崎商会	ソリューション営業グループ
☐	エンドユーザー	鈴木土地建物	第二営業部
☐	エンドユーザー	株式会社井上技研	情報システム部
☐	エンドユーザー	仲間建材工業	営業部
☐	エンドユーザー	千葉コールセンター	総務部
☐	エンドユーザー	中丸建設	総務部
☒	エンドユーザー	池田学院大学	情報システム部
☐	エンドユーザー	株式会社中川工業	営業部
☐	エンドユーザー	戸田ネットソリューションズ	開発本部

OK キャンセル

顧客名 部署名 担当者名

池田学院大学 情報システム部 小寺 樹里

それではこのカスタマイズにおけるポイントを順に見ていきましょう。

POINT1. 他アプリからのレコード取得

『1-2-3. 自分が操作しているアプリでカスタマイズは動く』で記載の通り、カスタマイズはユーザーが開いているアプリで動きます。そのため、ユーザーが操作していないアプリのデータを使用したい場合にはレコードを取得する必要があります。他アプリからレコードを取得するためのやることは大きく以下の4つがあり、それぞれ特徴が異なります。

やることの名称	用途
全レコードを取得する	指定したアプリから全てのレコードを取得できます。
キーを指定してレコードを取得する	キーとなるフィールドを指定し、キーが一致したレコードに絞り込んで取得できます。
関連レコード一覧の条件でレコードを取得する	関連レコード一覧の「表示するレコードの条件」で指定した条件で絞り込んでレコードを取得できます。[やること：キーを指定してレコードを取得する]との大きな違いとして、取得先として自アプリを指定した場合に「自レコードを除外する」設定をすることで自レコードを除外できる点が挙げられます。
クエリで条件を指定してレコードを取得する	kintone のクエリ記法に従って記載することで複雑な条件でレコードを取得できます。使いこなすと非常に便利ですが、フィールドごとに使用可能な演算子や記法を理解する必要があるため上級者向けのやることになります。

ボタンの設置

【アクション:1】でボタンを設置していますが、ここでボタンを設定する場所は「スペース」フィールドの場所になります。スペースフィールドを設置する時に「要素 ID」を付け、その指定した要素 ID の場所にボタンを配置するという流れになります。



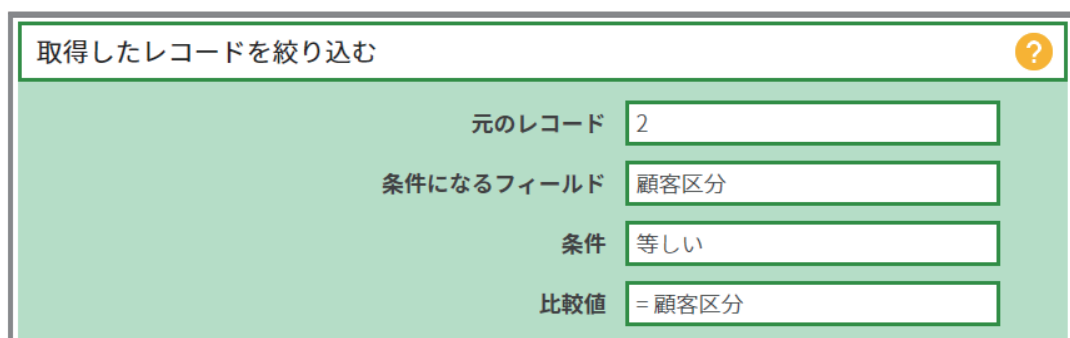
ボタン以外でも「kintone のフィールドには存在しないもの」を描画するときには基本的にスペースフィールドを用います。カスタマイズによるカスタマイズでは、ボタンの他にタブ、ドロップダウン、地図の表示などでもスペースフィールドを用います。

POINT2. 取得したレコードを絞り込む

POINT1 で記載の通り、レコード取得には様々な方法があります。今回はまず【アクション:

1】で全てのレコードを取得しました。このままでは選択しているラジオボタンの項目に関わらず全てのレコードが表示されてしまうため、「ラジオボタンで選択した内容と一致するレコードのみ」に絞り込む必要があります。このように取得したレコードから条件に合致したレコードだけを取り出すときには [やること: 取得したレコードを絞り込む] を使用します。

[やること: 取得したレコードを絞り込む] の設定項目は以下の通り 4 つあります。



設定項目名	設定すべき内容
元のレコード	絞り込む前のレコードを指定します。予め他のアクションによってレコードを取得しておく必要があります。 今回のカスタマイズだと【アクション：2】の[やること：全レコードを取得する]でレコードを取得済みなのでこれを指定します。
条件になるフィールド	ここで指定した条件に従って元のレコードを絞り込みます。 条件になるフィールドには元のレコードで絞り込みの条件としたいフィールドを指定。比較値には現在のレコードのフィールドや固定値、他のアクションの結果を指定し、条件で指定した内容に従って絞り込みます。 例として今回のカスタマイズでは 条件になるフィールド：顧客区分 _ 顧客管理 条件：等しい 比較値：= 顧客区分 _ 案件管理 と設定しており、この場合元のレコードの「顧客区分 _ 顧客管理」フィールドが現在のレコードの「顧客区分 _ 案件管理」で選択されている値と等しいレコードのみに絞り込まれます。 もう1つの例で記載します。 条件になるフィールド：数値系のフィールド 条件：大きい 比較値：20 と指定した場合、数値型のフィールドが 20 以上のレコードのみに絞り込まれます。
条件	
比較値	

やってみましょう

練習問題 3

上記カスタマイズに加えて、「最終更新日が 2021 年 1 月 1 日以降ならば」という条件を追加してレコード取得してみましょう

ヒント：アクションを 1 つ追加し、[やること：取得したレコードを絞り込む]を使用します。

練習問題の解答は巻末に記載しています。

POINT3. ポップアップで指定したものを後続のアクションで使用する

[やること：レコードの一覧をポップアップで表示する]を使用した場合、ラジオボタンやチェックボックスで指定したレコードを後続のアクションで使うことが可能です。今回の例では【アクション：5】のマッピング画面で「他のアクション結果」を指定することでポップアップで選択したレコードを指定しています。

5

コメント入力

フィールド値をまとめてセットする

?

マッピング

顧客名= \$4[0].顧客名
部署名= \$4[0].部署名
担当者名= \$4[0].担当者名

POINT4. 意図したとおりに動かない場合には

今回のカスタマイズのように指定したい条件でレコードを取得する場合、条件の設定間違いなどにより意図したレコードが取得できないことが多くあります。そういった場合には、アクションごとに取得した情報(レコードやフィールドの値)がどんな値になっているかを表示して確認していきましょう。

詳しくは『3. うまく動かないときは?』に記載していますが、例えばフィールドの値や計算した結果を一時的に確認するためには、[やること : 情報ダイアログを表示する] や [やること : コンソールにメッセージを出力する] を使って表示するのが良いでしょう。

3. うまく動かないときは？

3-1. カスタマイズは少し作って確認、少し作って確認の繰り返し

カスタマインのカスタマイズは複数のアクションを組み合わせで作成することが多く、慣れてくると多くのアクションによって構成される大きなカスタマイズを、一度も kintone アプリでの実行を試さずに作ってしまいがちです。

一度も実行結果を試さずに作成した大きなカスタマイズは、うまく動かない場合にどのアクションに原因があるかを見つけるのが非常に大変です。

複数のアクションを組み合わせたカスタマイズでは、少し作ったら実行して動きを確認する→問題があれば直す→動きを確認する→問題なければ次のカスタマイズを追加する、という手順の繰り返しで作成することをおすすめします。

この章では、うまく動かない時に原因を見つけるための方法と、便利なカスタマインの機能について紹介します。

3-2. うまく動かない時に最初に確認するポイント

カスタマイズを「kintone アプリへ登録」して実行してみると、エラーが出たり、想定とは異なる結果になったりと、うまく動かないことがよくあります。

うまく動かないときは、ひとつひとつ原因となりそうな箇所を確認し、どこに原因があるのかを絞り込んでいくことになります。

カスタマインのほかに JavaScript やプラグインが入っている時

JavaScript やプラグインは、カスタマインのカスタマイズの動作に影響を与える場合があります。

そのためカスタマイズがうまく動かないときは、まずは JavaScript やプラグインを削除・無効化し、kintone アプリにカスタマインのカスタマイズのみが設定されている状態にして動作を確認してみてください。これは、原因がカスタマインのカスタマイズにあるのか、それとも JavaScript やプラグインが影響して動かないのかを判別するために行います。

kintone アプリにカスタマインのカスタマイズのみが登録されている状態で問題なく動作するのであれば、JavaScript やプラグインが影響してカスタマイズが動いていない可能性が高いです。

逆に、JavaScript やプラグインを削除・無効化してもカスタマイズがうまく動かない時はカスタマインのカスタマイズが誤っている可能性が高いです。カスタマイズを見直してみてください。

※ドキュメントの「[外部サービス連携](#)」で連携している以外のプラグイン、及び自作 JavaScript ファイルが含まれた状態では、カスタマインの動作は保証外となります。

外部サービス連携：

<https://docs-customine.gusuku.io/ja/actions/external/>

(ドキュメントで「外部サービス連携」を御覧ください)

※ JavaScript は後でもとに戻せるように、削除する前にダウンロードするなどして保存しておいてください。

3-3. エラーが表示されたとき

エラーが発生した際に表示されるメッセージには、原因を究明するための情報が含まれています。エラーダイアログが表示されると驚いてしまう事もあると思いますが、エラーの原因がそのまま記載されている事もありますので、一度心を落ち着けて読んでみてください。

カスタマインのカスタマイズ画面で表示されるエラー

カスタマインのカスタマイズ画面で表示されるエラーです。

下記の例では「ページ (3): アクション (11) : アクション: 無効なアクションです。 (10)」とあります。

冒頭の「ページ (3): アクション (11) :」の部分がエラーとなっているアクション、その後がエラーの内容を示しています。

今回はアクション番号 11 で条件「他のアクションの実行が完了した時」に指定されたアクションが使用できない、というエラーです。

カスタマイズを見てみると、アクション番号 10 が無効化されているため、使用できない状態になっていることがわかります。

カスタマイズ画面に表示されるエラーは、エラーが解消するまで「kintone アプリへ登録」できません。メッセージの内容を確認して、アクションの設定を確認してみてください。

kintone アプリで表示されるエラーダイアログ

このような白いダイアログは、カスタマインによって表示されるダイアログです。

カスタマインのカスタマイズで、やることや条件の使い方が誤っている場合に表示されるエラーメッセージです。

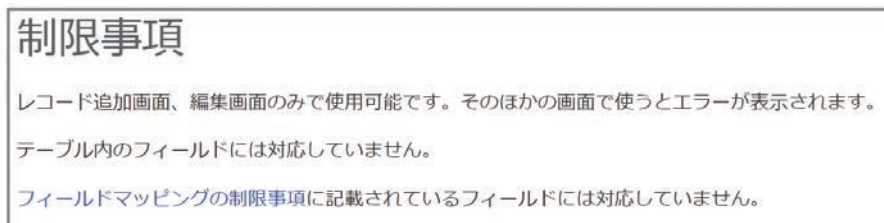


このエラーは、[やること：ルックアップを取得しなおす] を、レコード保存前・保存後のタイミングで実行した時に表示されるエラーです。

エラーメッセージの前 (25) のように、() で囲まれた数字が記載されている場合は、このアクション番号のアクションでエラーが起きている事を意味しています。(25) ですと、【アクション：25】のアクションでエラーが起きています。



このようなエラーが表示されたら、ドキュメントの「[制限事項](#)」の項目を確認してみてください。
[やること： [フィールド値をまとめてセットする](#)] の制限事項は下記のように記載されています。



今回のエラーの原因は、[やること： [ルックアップを取得しなおす](#)] は、レコードを保存するタイミングの < 条件： [レコードを保存する直前（削除時は除く）](#) > では使用できないため起こりました。使用するやることを変更するか、条件を変える必要があります。

次のエラーは先ほどのエラーダイアログと同様に白いダイアログですが、こちらのメッセージはカスタマイズでカスタマイズを実行した結果、kintone からエラーが返ってきた場合のダイアログです。



見分け方は [kintone からの応答] というメッセージが記載されているかどうかです。[kintone からの応答] とあれば、kintone からエラーが返ってきた場合のダイアログであることがわかります。

この場合は、kintone 側からエラーが返ってきているため、kintone のプロセス管理の設定とカスタマイズで指定したステータスの整合性が取れていない、カスタマイズを実行したユーザーが kintone のアプリやレコードに対して権限がないなど、kintone の設定とカスタマイズに齟齬がないか確認する必要があります。

このエラーの場合、[kintone からの応答] に、「入力内容が正しくありません。Code:CB_VA01 record. 氏名 .value: 必須です。」とありますので、レコードを追加・更新する際に、kintone アプリで必須と指定されているフィールドに値をセットしていないためエラーが発生しています。

このように、エラーメッセージを読むとエラーの原因が判明する、または糸口が見つかることはよくあります。一度心を落ち着けて、エラーメッセージを読んでみてください。

また、[サポートサイト](https://support.gusuku.io/)では、カスタマイズでカスタマイズしているとよく出会うエラーメッセージとその原因を記事としてまとめています。

サポートサイト

(カスタマイズの設定画面右上「サポートサイト」からも開くことができます)

<https://support.gusuku.io/>

サポートサイト上部の検索窓で、エラーと検索してみてください。



検索すると、エラーに関する記事が表示されます。例えば、先ほどご紹介した必須エラーの例は下記の「[入力内容が正しくありません。records\[0\].〇〇.value: 必須です](#)」というエラーが出ました。に詳しく記載していますので、ぜひ確認してみてください。

- [「kintone API呼び出しがエラーになりました」という場合の対応方法](#)
<https://support.gusuku.io/ja-JP/support/solutions/articles/36000243836>
Customizeを使ったカスタマイズで、「kintone API呼び出しがエラーになりました」というエラーダイアログが出る場合があります。これを解決するには、kintone側・Customize側...
- [フィールドのエラーをクリアする](#)
https://docs-customize.gusuku.io/ja/actions/error_check/clear_field_error/
概要 フィールドのエラーをクリアします。もともとエラーがない場合は何も起こりません。使い方 「フィールドエラーを表示する」などフィールドにエラーを設定する「やること」を実行したときや、kintone...
- [「入力内容が正しくありません。records\[0\].〇〇.value: 必須です」というエラーが出ました。](#)
<https://support.gusuku.io/ja-JP/support/solutions/articles/36000220365>
こちらは、追加もしくは更新するレコードの中の入力必須のフィールドが空になる場合に発生するエラーです。解決手順 1. エラーが出るアクションの候補を探す まずはこのエラーが出そうなアクションを探します...

検索キーワードは、ある程度の長い文で検索して想定する記事がヒットしない場合は、単語をスペースで区切って指定いただくのがお勧めです。

例えば今回の必須エラーであれば、「必須 エラー」のように区切って検索してみてください。文

の状態を検索するより目当ての記事が検索されやすい場合が多いです。

3-4. 原因究明に役立つカスタマイズの便利な機能

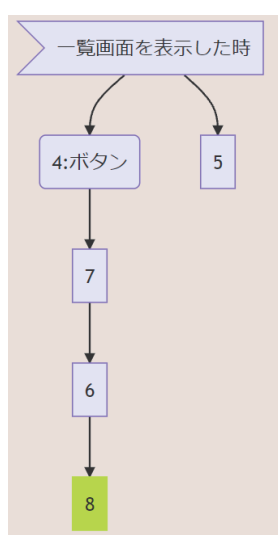
カスタマイズは、うまく動かないときの原因究明に役立つ機能を提供しています。

アクショングラフ

アクショングラフは、<条件:[他のアクションの実行が完了した時](#)>などで順番を指定したアクションが、どの順番で実行されるのかを図で表示してくれる機能です。

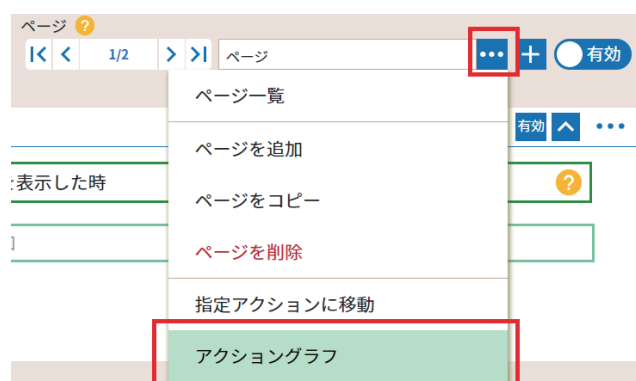
アクションが想定通りにつながっているかを確認したい場合に便利です。

カスタマイズを実行した時、順番に実行されるアクションが途中で動かない場合は、アクショングラフを確認してみてください。矢印が途切れていたり、順番が間違っているなど、一目でわかってとても便利です。



アクショングラフ上に表示されている番号は、アクション番号です。アクション番号をクリックすると、該当するアクションの設定にジャンプすることができます。

アクショングラフは、ページメニューの「アクショングラフ」を選択すると表示されます。



アクションの無効化

一時的にアクションが実行されないようにしたいときに使用します。

アクションを無効化した後で「kintone アプリへ登録」を行った際に、無効化したアクションは

kintone アプリに登録されず、実行されなくなります。

アクションの左上のアクション番号または右上の有効ボタンをクリックすると、そのアクションを無効化できます。

右上のメニューから 無効にする を選択する方法もあります。

無効化したアクションは、左上のアクション番号がグレーになり、右上には無効と表示されます。

他のアクションの動きを確認するために無効化したり、アクションの設定を変更して動きを確認する際に、変更前の設定を残しておきたい場合にも使用できます。

その際は、アクションを複製して、複製元のアクションを無効化して残しておく、という方法がおすすめです。

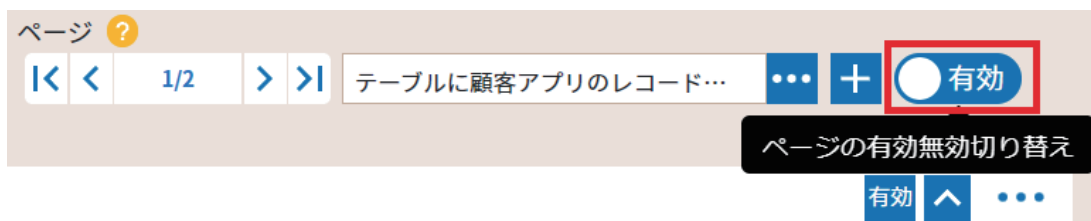
アクションの複製は下記のメニューから行ってください。

ページの無効化

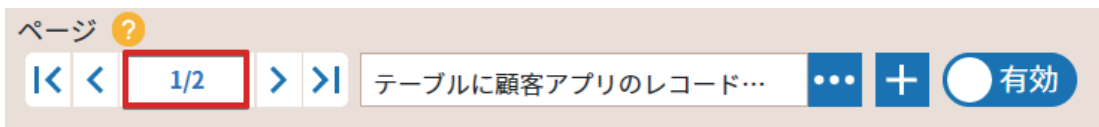
個別のアクションではなく、ページをまるごと無効化・有効化することができます。

無効化したページは、「kintone アプリへ登録」する際に kintone アプリに登録されず、実行されません。

ページの無効化は、カスタマイズ画面右上の 有効 をクリックすると無効に切り替ええます。



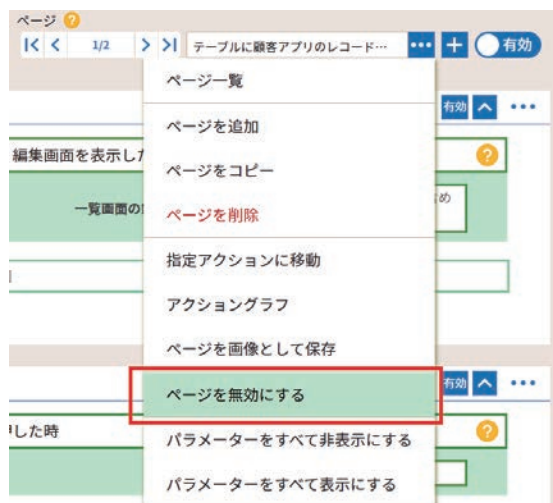
または、ページの選択・並べ替え をクリックすると表示される ページ選択 画面でも無効化できます。



この画面では、複数ページをまとめて設定できます。ページが一覧されますので、どのページが有効・無効になっているか確認する場合にも便利です。



ページメニューから無効化することもできます。



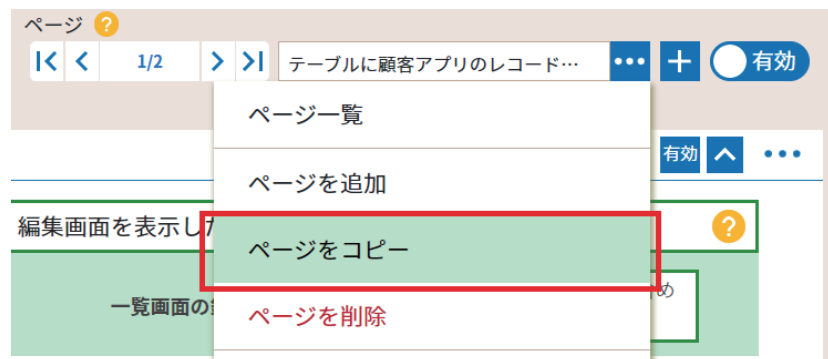
ページの無効化は、他のページの動作を確認したいときに、一旦ページ全体を無効化して置きたい場合などに使用します。この機能を効果的に使うためには、機能ごと、画面ごとにこまめにページを分けてカスタマイズを作成しておく必要があります。

ページの無効化は、次に記載するページのコピーと組み合わせて使用することが多い機能です。

ページのコピー

ページ内のカスタマイズに変更を加える場合は、ページをコピーして無効化し、変更前のカスタマイズの状態を保存しておくことをお勧めします。

変更前の状態を保存しておけば、カスタマイズの変更に失敗した場合などに、コピーしておいたページを有効化し、変更したページを無効化することでカスタマイズを変更前の状態に簡単に戻すことができます。



3-5. レコードが想定通り取得・絞り込みできているかポップアップ表示して確認する

[やること：[キーを指定してレコードを取得する](#)] や [やること：[クエリで条件を指定してレコードを取得する](#)] のように、条件を指定してレコードを取得するやることを使用した場合や、取得したレコードを [やること：[取得したレコードを絞り込む](#)] で絞り込んだ場合、想定通りのレコードが取得・絞り込みできているか確認したい事がよくあります。

そういった場合は、取得・絞り込んだ後のレコードを [やること：[レコードの一覧をポップアップで表示する](#)] で確認してみてください。



このように、絞り込んだ結果を一覧で表示できるため、想定通りの絞り込みが行われているかを一目で確認することが可能です。

	都道府県名	市町村名
☐	静岡県	川根本町
☐	大阪府	島本町
☐	奈良県	田原本町
☐	和歌山県	串本町
☐	島根県	川本町

確認のための仮のアクションですので、確認が終わったらアクションを無効にしておけば、条件が変わった場合などの再確認の際に役立ちます。

3-6. テスト用アプリ作成

ガンガン試してと言われても、既にユーザーが使っている kintone アプリ (以下 本番用アプリ) にカスタマイズすると、誤ったレコードを更新するリスクがあったり、フィールドの値やステータスなど、カスタマイズを試したい状態のレコードがないなど実行確認が難しいことがよくあります。

そういった時のために、実際にユーザーが使うアプリとは別に、そのアプリをコピーしたテスト用アプリを作成することをお勧めします。

練習問題の解答例

練習問題 1

問題

顧客管理アプリへの登録時に、「登録日」フィールドに本日の日付を入力するカスタマイズを作成してみましょう。

回答例

【アクション：4】のフィールドマッピング画面にて「挿入」 - 「関数」を選択すると today 関数を選択できます。

セットする値	セット先のフィールド
挿入 	> 登録日 日付
フィールド	> 顧客名 文字列 (1行)
他のアクションの結果	> 部署名 文字列 (1行)
関数	> 担当者名 文字列 (1行)

これによって、【アクション：4】は以下のような設定となり、登録日に本日の日付を入力することができます。

4

コメント入力

レコードを追加する

追加先アプリ

顧客管理 (営業支援バック)

マッピング

登録日= today()
顧客名_顧客管理= \$1
部署名_顧客管理= \$2
担当者名_顧客管理= \$3

練習問題 2

問題

顧客管理アプリへレコードを追加した後、自動的に別タブで顧客管理アプリを開くカスタマイズを作成してみましょう。

回答例

以下のように【アクション:6】を追加し、[やること:レコード一覧画面に遷移する] を使用します。

The screenshot shows a custom action configuration for 'Record List View' (レコード一覧画面に遷移する). The 'Action' (アクション) field is set to '5'. The 'When the action is completed' (他のアクションの実行が完了した時) section is empty. The 'Conditions' (条件を追加) section is also empty.

この他にも、[やること:レコード詳細画面に遷移する]、[やること:レコード追加画面に遷移する]、[やること:レコード編集画面に遷移する] もありますので要件に合わせて使い分けましょう。

練習問題 3

問題

上記カスタマイズに加えて、「最終更新日が 2021 年 1 月 1 日以降ならば」という条件を追加してレコード取得してみましょう

回答例

【アクション:6】を追加して、< やること:取得したレコードを絞り込む > を使用します。【アクション:4】も赤枠の部分の修正が必要となるので注意しましょう。

The screenshot shows two custom action configurations. The top configuration is for 'Record List View' (レコード一覧画面に遷移する) with Action 5. The bottom configuration is for 'Record Detail View' (レコードの詳細画面に遷移する) with Action 6. In the bottom configuration, the 'Record' (レコード) field is highlighted with a red box, and the 'Action' field is also highlighted with a red box. The 'Conditions' (条件を追加) section is empty.

