

gusuku Customine

カスタマイズの歩き方

エラーチェック編



はじめに（この冊子の目的）

使いこなすとともに便利な 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もあります。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 での入力を補助するために使われる機能である「エラーのチェック」や「入力内容のチェック」、「重複チェック」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、便利な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、必須チェックの方法
- 2、条件（フィールド条件）の使用方法
- 3、他のレコードとの比較による重複チェックの方法

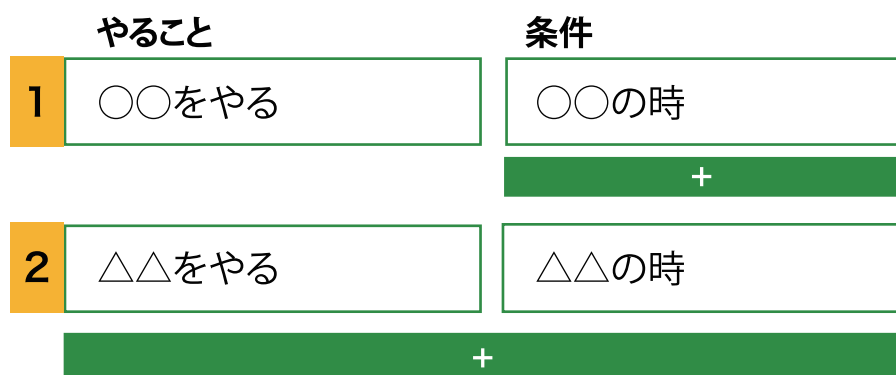
目次

1. はじめに理解していただきたいこと -----	4p
2. 実例に学ぶカスタマイズ作成のポイント -----	11p
2-1. 必須チェック -----	11p
2-2. 入力チェック -----	15p
2-3. 重複チェック -----	17p
3. 上手く行かないときの試行錯誤の方法 -----	21p

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。

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」についての詳細は Cybozu developer Network をご確認ください。

(Cybozu developer Network 内「API ドキュメント」 - 「kintone API」 - 「kintone JavaScript API (イベント) 一覧」)

<https://developer.cybozu.io/hc/ja/articles/3600000361686>

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

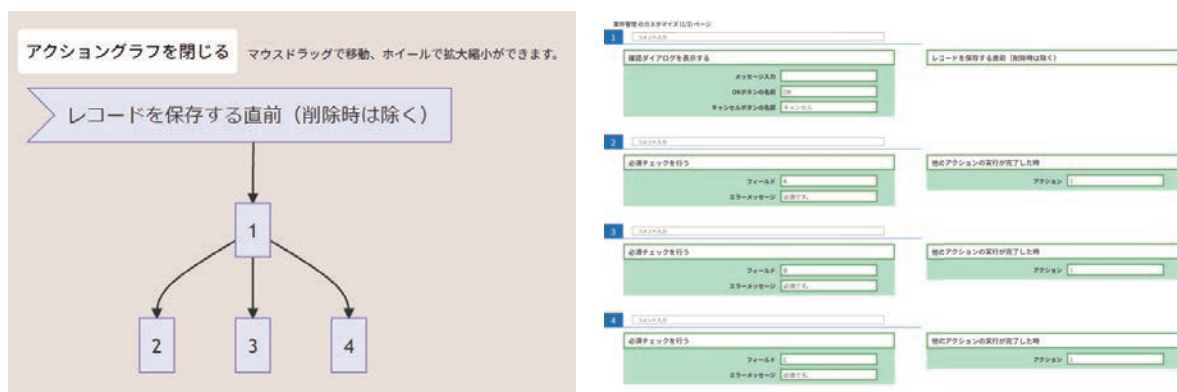
カスタマイズでアクションを作成すると、番号が自動的に付与されます。
これはアクション番号と呼ばれるもので、カスタマイズの実行順とは関係ありません。

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。

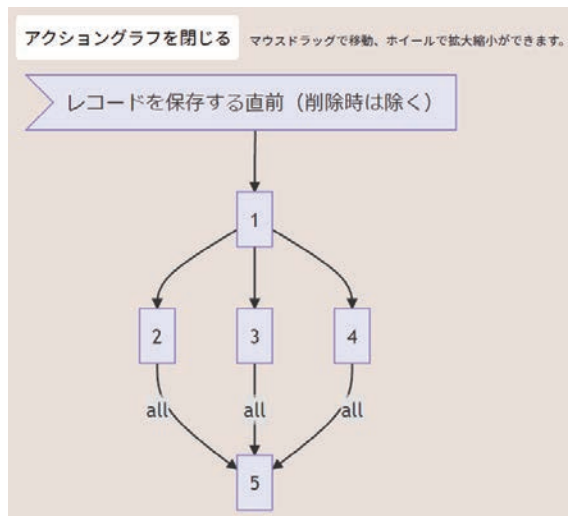
アクションの実行順を確認するにはアクショングラフを表示すると分かりやすいので、
カスタマイズ作成途中や作成後に表示して確認することをおすすめします。
よくあるのが、線で繋がっていないアクションの結果を参照したり、アクションの繋ぎ先を間違えているケースです。

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。



1. コマンド入力

確認ダイアログを表示する

メッセージ本文
メッセージ本文
メッセージ本文
メッセージ本文

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

フィールドに値をセットする

フィールド
値

レコードを保存する条件（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。

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

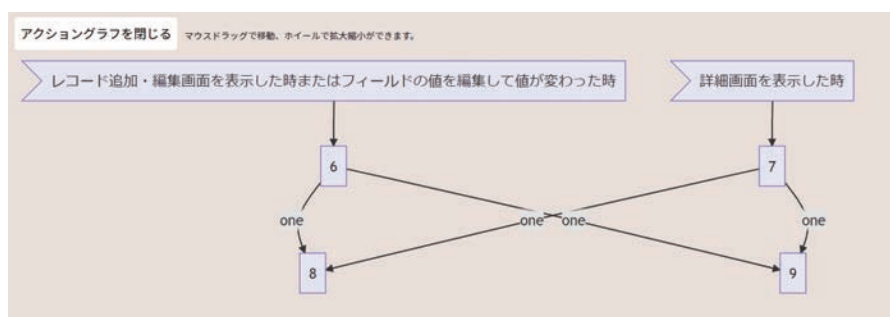
フィールド
状態
条件
等しい
比較値
判定

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
判定



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。

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

カスタマイ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

カスタマイ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

カスタマイ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。

2. 事例に学ぶカスタマイズ作成のポイント

2-1. 必須チェック

2-1-1. 標準機能での必須チェックの課題

kintone の標準機能で以下のように「必須項目にする」にチェックを入れると、レコード保存時にそのフィールドが未入力の状態だとエラーとなり保存できません。

しかし、この設定を行うと常に該当のフィールドが必須となるため、「特定のステータスのときのみ必須にしたい」「別のフィールドの値が A の場合は必須にしたい」など特定の条件下で必須にすることはできません。



ABC 文字列（1行）の設定

? ヘルプ

フィールド名 *

会社名

☐ 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

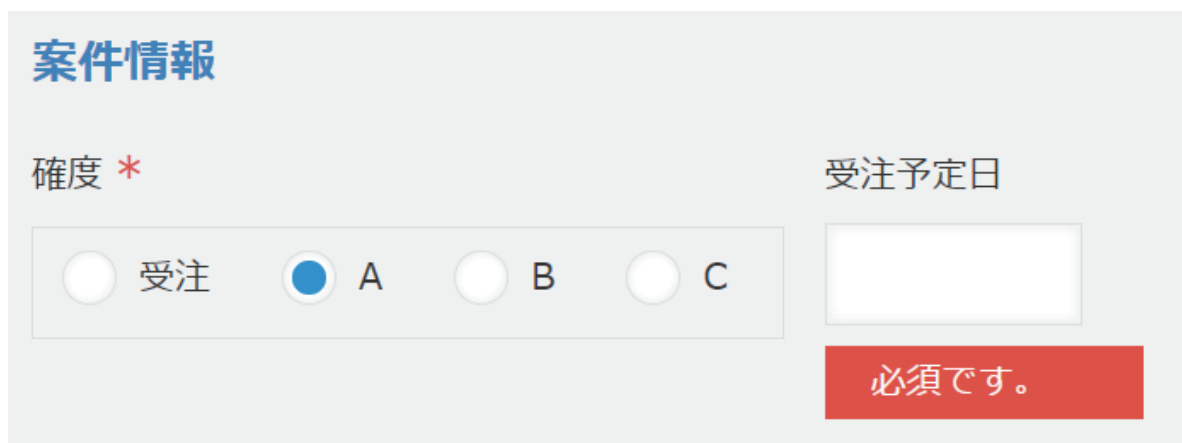
☐ 値の重複を禁止する

カスタマイズの [やること: 必須チェックを行う] などを使用したカスタマイズを行うことで、様々な条件に合致したときのみ必須にすることができます。

2-1-2. 条件付き必須チェック

必須チェックの設定自体は非常にシンプルで [やること: [必須チェックを行う](#)] を使います。ここに追加条件を組み合わせることで様々な条件下での必須チェックが実現できます。

例 1: レコード保存時に「確度」フィールドが「A」だった場合には「受注予定日」を必須にする



案件情報

確度 *

受注 A B C

受注予定日

必須です。

上記のカスタマイズを実現するための設定は以下の通りです。

1 コメント入力

必須チェックを行う

フィールド 受注予定日

エラーメッセージ 必須です。

レコードを保存する直前（削除時は除く）

フィールド値が特定の値ならば

フィールド 確度

条件 等しい

比較値 A

条件を追加

必須チェックの設定自体は上記のように非常に簡単です。ただし、このカスタマイズは条件を「レコードを保存する直前（削除時は除く）」にしており、ユーザーが「保存」ボタンを押すまでその項目が必須であることに気づけません。「確度」フィールドやを変更した時点で「受注予定日」が必須であることに気づかせるためには、以下のように「フィールド変更して値が変わった時」を条件とするカスタマイズも追加する必要があります。

4 コメント入力

必須チェックを行う

フィールド 受注予定日

エラーメッセージ 必須です。

フィールドの値を編集して値が変わった時

フィールド 確度

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

フィールド値が特定の値ならば

フィールド 確度

条件 等しい

比較値 A

条件を追加

※「フィールド変更して値が変わった時」だけだと保存ができてしまうので、必ず「レコードを保存する直前（削除時は除く）」と同時に使用します。

また、一度必須チェックのエラーになった後に「受注予定日」を入力しても、保存ボタンを押すまでエラーは消えません。そのため、「受注予定日」を入力したときや「確度」がA以外になったときに即時にエラーをクリアしたい場合には以下の設定を追加します。

5 コメント入力

フィールドのエラーをクリアする

フィールド 受注予定日

フィールドの値を編集して値が変わった時

フィールド 受注予定日

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

フィールド値が特定の値でないならば

フィールド 受注予定日

条件 等しい

比較値 A

条件を追加

6 コメント入力

フィールドのエラーをクリアする

フィールド 受注予定日

フィールドの値を編集して値が変わった時

フィールド 確度

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

フィールド値が特定の値でないならば

フィールド 確度

条件 等しい

比較値 A

条件を追加

例 2：承認者がプロセス管理の「承認」アクションを実行する際に必須チェックを行う

続いて、プロセス管理のアクションを実行する際に必須チェックを行うカスタマイズです。プロセス管理のアクションを実行する際には [やること： [必須チェックを行う](#)] は使用することができません。そのため、以下のように <追加条件： [フィールド値が空ならば](#)> を使うことで必須チェックを実現します。

The screenshot shows a process management interface. On the left, a green box labeled 'レコードにエラーをセットする' (Set error on record) contains an 'エラーメッセージ' (Error message) field with the text '承認する際には、承認者コメントは必須です。' (When approving, the approver comment is required). On the right, a green box labeled 'プロセス管理のアクションを実行した時' (When executing the process management action) contains a 'プロセス管理アクション' (Process management action) dropdown set to '承認' (Approve). Below it are two '変更前プロセス管理ステータス' (Before process management status) and '変更後プロセス管理ステータス' (After process management status) dropdowns, both set to '<選択されていません>' (Not selected). A 'フィールド値が空ならば' (If field value is empty) condition is set to '承認者コメント' (Approver comment). A '条件を追加' (Add condition) button is at the bottom.

[レコードにエラーをセットする] のエラーは画面上部に以下のように表示されます。



例 3：複数フィールドの必須チェック

例 1, 例 2 で紹介したものは 1 つのフィールドのみの必須チェックですが、複数あるフィールドのうち、いずれか 1 つのフィールドを必須にするという要件もあるかと思います。

The form is titled '顧客情報' (Customer Information). It has three input fields for '会社名' (Company Name), '部署名' (Department Name), and '先方担当者' (Counterparty Contact). Below these are three input fields for 'TEL', 'FAX', and 'メールアドレス' (Email Address). Arrows point from the text 'TEL、FAX、メールアドレスのうちどれか 1 つは入力する必要がある' (At least one of TEL, FAX, or email address must be entered) to the three input fields.

このような要件の場合には、[やること：[複数フィールドのうち少なくとも1つを必須にする](#)]を使用します。設定例は以下の通りです。

The screenshot shows the Kintone configuration page for a 'コメント入力' (Comment Input) field. A green box highlights the configuration for the rule '複数フィールドのうち少なくとも1つを必須にする' (Multiple Fields, at least one is required). The fields listed are 'TEL, FAX, メールアドレス' (TEL, FAX, Email Address). The error message is 'いずれかのフィールドは必須です。' (One of the fields is required). To the right, there is a configuration for the 'Record Save Before' rule, which is set to '条件を追加' (Add Condition).

上記設定により、以下のように 3つのフィールドいずれかが必須フィールドとしてチェックが行われます。

The screenshot shows the '顧客情報' (Customer Information) form. It has three rows of fields: '会社名' (Company Name), '部署名' (Department Name), and '先方担当者' (Counterparty Contact) in the first row; 'TEL', 'FAX', and 'メールアドレス' (Email Address) in the second row. Below each field is a red error message box stating 'いずれかのフィールドは必須です。' (One of the fields is required).

フィールドエラーとレコードエラーの違い

ここまでのカスタマイズで[やること：[必須チェックを行う](#)]や[やること：[レコードにエラーをセットする](#)]を使用してきましたが、それぞれの場合でエラーの表示方法が異なることに気づかれたかと思います。

[やること：[必須チェックを行う](#)]では、以下のようにフィールドの下部にエラーが表示されます。これを kintone では「フィールドエラー」と呼び、[やること：[フィールドエラーを表示する](#)]でも同様にエラーを出すことができます。

The screenshot shows a 'TEL' field with a red error message box below it stating 'いずれかのフィールドは必須です。' (One of the fields is required).

また、[やること：[レコードにエラーをセットする](#)]では、以下のようにレコードの上部にエラーが表示されます。これを kintone では「レコードエラー」と呼びます。

エラー

承認する際には、承認者コメントは必須です。



これら 2 つのエラーは表示する場所の違いだけではなく、使用できるシーンが異なります。

フィールドエラーは該当のフィールド直下にエラーが表示されるため、ユーザーが気づきやすいという利点がありますが、例2で記載したようにプロセス管理の承認時には[やること：[必須チェックを行う](#)]や[やること：[フィールドエラーを表示する](#)]を使用することはできません。そのため、用途やシーンに応じて使い分ける必要があります。

2-2. 入力チェック

「必須チェック」では、データが入力されているか否かのチェックを行っていましたが、入力されたデータが正しい値であるかをチェックするための入力チェックの仕組みもあります。例えば、メールアドレスであれば xxx@yyyyy.zzz (半角英数字で @ が含まれる) が期待される値であり、@ が無かったり英数字以外の値が入っている場合には適切ではないと言えます。

kintone の標準機能ではこの値のチェックまで実現できません。

カスタミンでは多くのデータ形式に対する入力チェックの「やること」がありますが、すべての「やること」は紹介しきれないためいくつかの例を挙げて使用方法を説明します。その他の「やること」はカスタミンのドキュメント内の「エラーチェック」を参照してください。

2-2-1. メールアドレス形式をチェックする

メールアドレス形式をチェックする基本的なカスタマイズは以下の通りです。

The screenshot shows a Kintone form titled "コメント入力" (Comment Input). It features two configuration panels. The left panel, titled "メールアドレス形式をチェックする" (Check email address format), has a "フィールド" (Field) set to "メールアドレス" (Email Address), a "チェック精度" (Check precision) set to "厳密" (Strict), and an "エラーメッセージ" (Error message) that reads "メールアドレスの書式に誤りがあります。" (There is an error in the email address format). The right panel, titled "レコードを保存する直前 (削除時は除く)" (Just before saving the record (except when deleting)), has a "条件を追加" (Add condition) button.

「条件付き必須チェック」のどこかで触れたように、上記の設定だとレコード保存するタイミングのみでチェックが行われます。ユーザーがフィールドに値を入力したタイミングでチェックを行いたい場合には、以下のように「フィールド値」を条件に設定します。

2-2-2. 時刻が時間内に入っているかチェックする

[やること : [時刻が時間内に入っているかチェックする](#)] を使用すると、例えば出勤・退勤時刻が会社で定められた勤務時間内（例：9:00 ～ 18:00）であるかチェックを行い、それ以外の時間が登録された場合にはエラーすることができます。

例として、9:00 ～ 18:00 以外の時間が「出勤」「退勤」フィールドに入力された場合にエラーとするカスタマイズは以下の通りです。

これにより、指定した時間外のデータが入力されると以下の通りエラーとなります。

出勤	退勤
07:00	22:00
時間の範囲外です。	時間の範囲外です。

また、条件を組み合わせることによってチェックの有無を変えることができますので、例えば残業申請が既にされている場合にはチェックを行わないということも可能です。

2-3. 重複チェック

2-3-1. 標準機能での重複チェックと課題

kintone の標準機能でもフィールドの設定画面で「値の重複を禁止する」にチェックを入れると、フィールドの値が重複するレコードの登録を禁止することができます。

しかし、「値の重複を禁止する」の設定はフィールドごとに行うため、以下のように複数のフィールドが重複している時に登録を禁止することはできません。

標準機能で重複禁止を実現できるケース

型番	商品名	単価
R-0001	gusuku Deploit スタンダード	¥120,000
R-0002	gusuku Customine 年額10	¥648,000

型番が重複している ので登録を 禁止したい	R-0002	gusuku Customine 年額4	¥216,000
-----------------------------	--------	----------------------	----------

標準機能では実現できないケース

工場	型番	商品名	単価
大阪工場	R-0001	gusuku Deploit スタンダード	¥120,000
大阪工場	R-0002	gusuku Customine 年額10	¥648,000
南足柄工場	R-0002	gusuku Boostone ライト	¥99,000

型番・工場とも に重複している ので登録を禁止 したい	大阪工場	R-0002	gusuku Customine 年額4	¥216,000
--------------------------------------	------	--------	----------------------	----------

実現するための考え方

このような要件を実現するためには、すでに登録されているレコードとこれから登録するレコードの値を比較し、重複していたらエラーとするカスタマイズを作成します。

2-3-2. テンプレートを使用したカスタマイズ作成

一見難しそうなカスタマイズですが、カスタマイズ自体はカスタマイズの「テンプレート」を使用し、フィールドコードを変更するのみで実現可能です。まずは、難しいことを考えずにテンプレートを適用してみましょう。

想定するアプリとフィールド構成

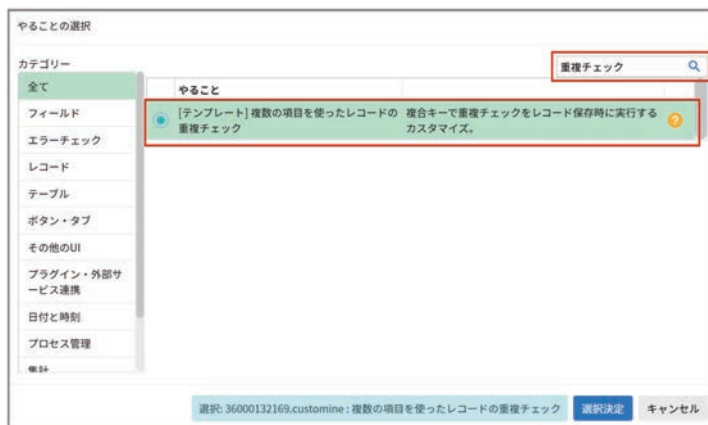
kintone アプリストアの「顧客リスト」アプリを使用します。「顧客リスト」アプリは以下のようなフィールドがあります。

	レコード番号	会社名	部署名	担当者名	住所	
	20	金都運総研	情報システム部	下山 達士	岐阜県岐阜市××××	 
	19	林田商会	ソリューション営業グループ	森 博	埼玉県浦和市××××	 
	18	板橋電子株式会社	経理部	末永 妃里	神奈川県小田原市××××	 
	17	新山物産	営業本部第一営業部	金子 真帆	大阪府大阪市北区梅田×…	 
	16	岩下税理士事務所	情報システム部	佐々木 樹里	兵庫県神戸市須磨区××××	 
	15	岡崎商会	ソリューション営業グループ	清田 誠一	東京都杉並区××××	 
	14	草刈組	総務部	原田 ちえみ	京都府京都市××××	 

今回は、「会社名」と「担当者名」が重複しているときにはエラーとするカスタマイズを作成します。

カスタマイズ作成

使用するテンプレートは「複数の項目を使ったレコードの重複チェック」です。



The screenshot shows the 'Customization' screen in kintone. On the left, a sidebar lists various customization options: 'Category', 'Field', 'Error Check', 'Record', 'Table', 'Button/Tab', 'Other UI', 'Plugin/External Service Connection', 'Date and Time', and 'Process Management'. The 'Record' option is selected. In the main area, a search bar at the top right contains the text '重複チェック' (Duplicate Check). Below it, a list of templates is shown, with the first one, '[テンプレート] 複数の項目を使ったレコードの重複チェック' (Template: Duplicate Check using multiple items), highlighted with a red box. The description for this template reads: '複合キーで重複チェックをレコード保存時に実行するカスタマイズ。' (Customization that executes a duplicate check using a composite key when saving records).

テンプレートを適用すると以下の設定が入ります。



The screenshot shows the configuration screen for the 'Duplicate Check' template. It is divided into two main sections. The top section, labeled '1', is titled 'クエリで条件を指定してレコードを取得する' (Retrieve records by specifying conditions with a query). It includes fields for '取得先アプリ' (Source app) set to '顧客リスト' (Customer List), '検索条件' (Search conditions) set to 'レコード番号: "[レコード番号]" and 日付: "[日付]" and ラジオボタン: "[ラジオボタン]"', and '最大取得件数' (Maximum number of records to retrieve) set to '<入力されていません>' (Not entered). The bottom section, labeled '4', is titled 'エラーダイアログを表示して保存をキャンセルする' (Display error dialog to cancel saving). It includes a 'メッセージ入力' (Message input) field with the text '日付とラジオボタンの複合キーが一致するレコードが既にあります' (A record with a matching composite key of date and radio button already exists), and an 'OKボタンの名前' (OK button name) field set to 'OK'. To the right of these sections, there are additional configuration options for 'レコードを保存する直前 (削除時は除く)' (Just before saving the record (except when deleting)) and '他のアクションの実行が完了した時' (When other actions are completed), each with a '条件を追加' (Add condition) button.

このテンプレートは 2 つのアクションで構成されています。まず、【アクション：1】で[やること：[クエリで条件を指定してレコードを取得](#)]を使用して条件に合致するレコードを取得しています。【アクション：1】でレコードを取得した結果、該当のレコードが 1 件以上存在する場合には**重複するレコード**が存在することになるため、【アクション：1】でアクション 1 のレコードが 1 件以上ならば、を条件としてエラーダイアログを表示して保存をキャンセルするカスタマイズとなります。

テンプレートでカスタマイズの流れはできていますが、検索条件やダイアログに表示する文字はお使いの環境（アプリ・フィールド）に合わせて修正する必要があります。

まず、【アクション：1】の[やること：[クエリで条件を指定してレコードを取得する](#)]の検索条件について見てみましょう。テンプレートで適用されている検索条件では以下のように設定されています。

1 コメント入力

クエリで条件を指定してレコードを取得する

取得先アプリ 顧客リスト

検索条件 レコード番号 != "\${レコード番号}" and 日付 = "\${日付}" and ラジオボタン in ("\${ラジオボタン}")

最大取得件数 <入力されていません>

レコード番号 != "\${レコード番号}" and 日付 = "\${日付}" and ラジオボタン in ("\${ラジオボタン}")

これは、「日付」というフィールドコードと「ラジオボタン」というフィールドコードが一致しているレコードを抽出するための検索条件の記載例です。今回は「会社名」と「担当者名」が一致するレコードを抽出したいため、以下のように変更します。

1 コメント入力

クエリで条件を指定してレコードを取得する

取得先アプリ 顧客リスト

検索条件 レコード番号 != "\${レコード番号}" and 会社名 = "\${会社名}" and 担当者名 = "\${担当者名}"

最大取得件数 <入力されていません>

レコード番号 != "\${レコード番号}" and 会社名 = "\${会社名}" and 担当者名 = "\${担当者名}"

※「レコード番号 != "\${レコード番号}"」は「現在保存しようとしているレコード以外のレコード」を示す条件で、この条件は変更する必要はありません。

なお、「検索条件」パラメーターは kintone のクエリ記法に従っており、詳細な記法については、Cybozu developer Network をご参照ください。

kintone API のクエリの書き方の基本

(Cybozu developer Network 内「TIPS」 - 「kintone TIPS」 - 「カスタマイズ」 - 「開発ノウハウ」)

<https://developer.cybozu.io/hc/ja/articles/900001057206>

続いて、【アクション：4】を見てみます。【アクション：4】ではエラーダイアログのメッセージのみを変更します。今回はメッセージ入力を「会社名と担当者名が一致するレコードが既にあります」に変更します。



以上でカスタマイズは完了です。kintone アプリへ登録した後に動作確認してみましょう。

既に登録されているレコードと同じ「会社名」と「担当者名」で登録すると以下のようにエラーダイアログが表示されて保存ができなくなります。



やってみましょう：練習問題

問題：顧客リストアプリへの登録時に、「会社名」「部署名」「担当者名」が重複していたらエラーとなるカスタマイズを作成してみましょう。

ヒント：複数の条件を組み合わせる際は and を使用してクエリを記載します。

練習問題の解答は巻末に記載しています。

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/>

※ 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。

制限事項

レコード追加画面、編集画面のみで使用可能です。その他の画面で使うとエラーが表示されます。

テーブル内のフィールドには対応していません。

フィールドマッピングの制限事項に記載されているフィールドには対応していません。

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>

Customine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・Customine側...

- フィールドのエラーをクリアする

https://docs-customine.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、kinton...

- 「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

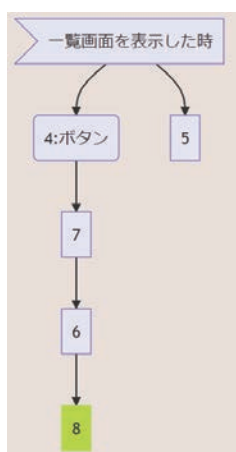
カスタマイズは、うまく動かないときの原因究明に役立つ機能を提供しています。

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

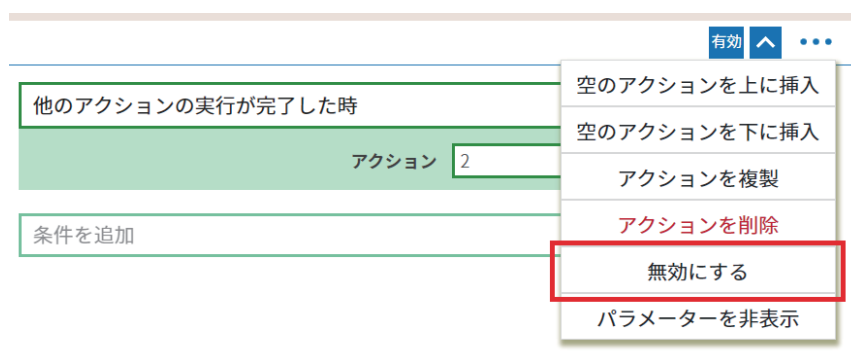
一時的にアクションが実行されないようにしたいときに使用します。

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。



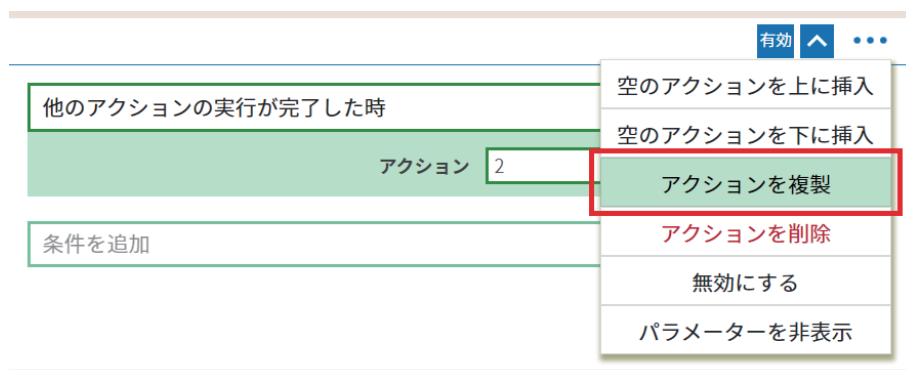
無効化したアクションは、左上のアクション番号がグレーになり、右上には無効と表示されます。



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。

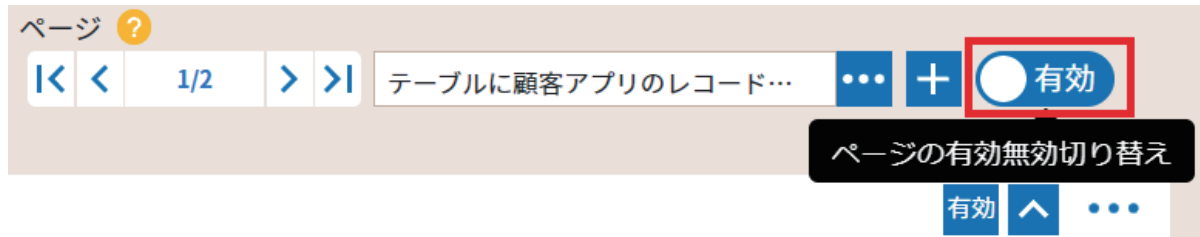


ページの無効化

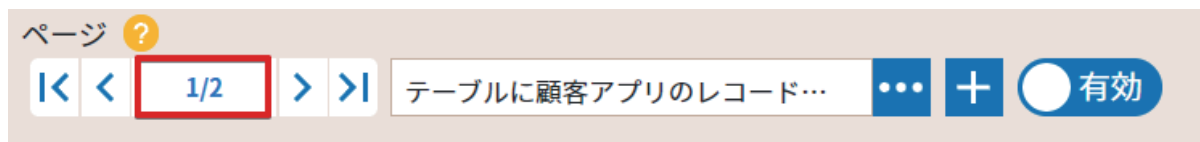
個別のアクションではなく、ページをまるごと無効化・有効化することができます。

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。

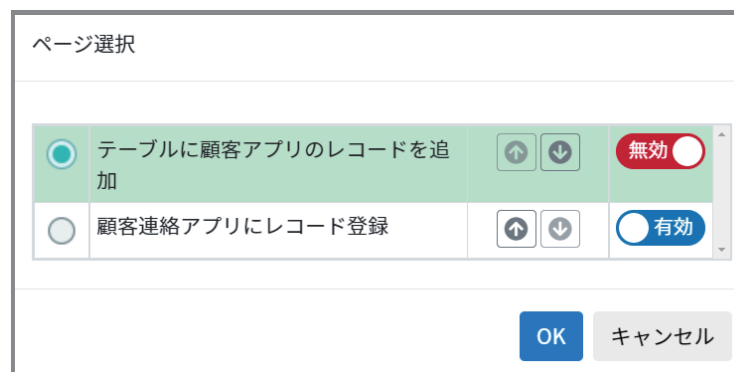
ページの無効化は、カスタマイズ画面右上の 有効 をクリックすると無効に切り替ええます。



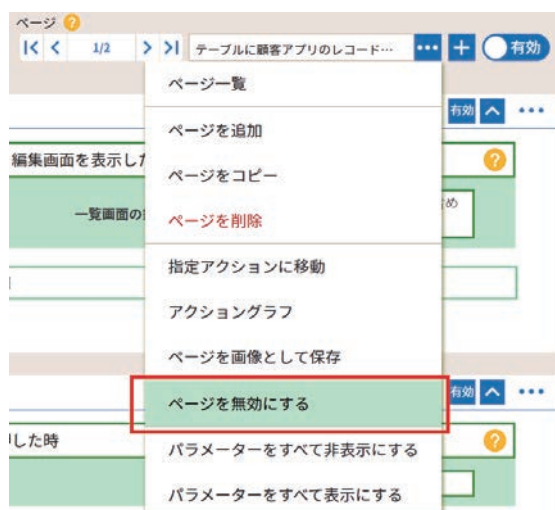
または、ページの選択・並べ替え をクリックすると表示される ページ選択 画面でも無効化できます。



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。



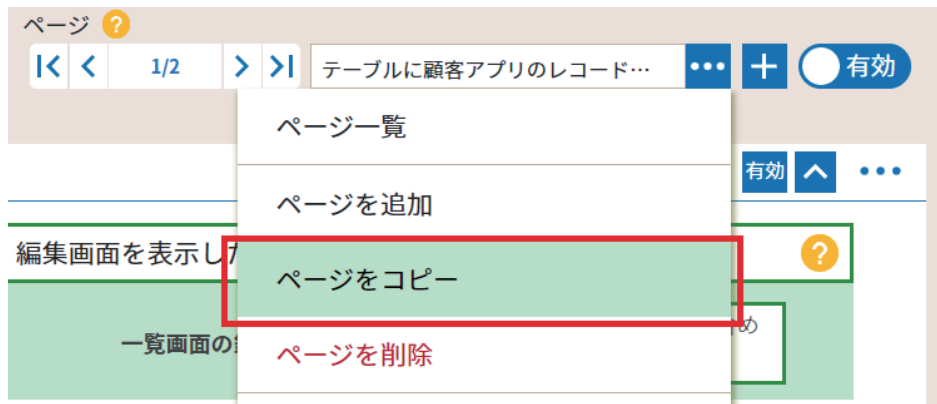
ページの無効化は、他のページの動作を確認したいときに、一旦ページ全体を無効化して置きたい場合などに使用します。この機能を効果的に使うためには、機能ごと、画面ごとにこまめにページを分けてカスタマイズを作成しておく必要があります。

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

問題

顧客リストアプリへの登録時に、「会社名」「部署名」「担当者名」が重複していたらエラーとな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：1】のクエリに以下のように変更します。

レコード番号 != "\${レコード番号}" and 会社名 = "\${会社名}" and 担当者名 = "\${担当者名}" and 部署名 = "\${部署名}"

合わせて【アクション：4】も修正し、以下のようなカスタマイズとして完成です。

The screenshot shows a custom application interface with two actions. Action 1 is a query that checks for duplicates of company name, department name, and manager name. Action 4 is an error dialog that displays a message when a duplicate is found.

アクション：1 クエリで条件を指定してレコードを取得する

取得先アプリ: 顧客リスト

検索条件: レコード番号 != "\${レコード番号}" and 会社名 = "\${会社名}" and 担当者名 = "\${担当者名}" and 部署名 = "\${部署名}"

最大取得件数: (入力されていません)

レコードを保存する直前 (削除時は除く)

条件を追加

アクション：4 エラーダイアログを表示して保存をキャンセルする

メッセージ入力: 会社名と担当者名、部署名が一致するレコードが既にあります

OKボタン名前: OK

他のアクションの実行が完了した時

アクション: 1

レコード件数が1件以上ならば

レコード: 1

条件を追加



エラーチェック編