

fast開発を体験！ さっぽろkintone (キントーン)ハンズオン祭り

cybozu developer network プロデューサー

Kitagawa Kyohei

今回のハンズオンで利用するもの

■ 本資料

[URL] <https://bit.ly/ki-contents>

■ kintone環境

まだ取得していない方は、試用環境をご利用ください。約5分ほどで取得できます。(開発環境の発行は、即日の発行ができません。)



<https://bit.ly/ki-contents>



ダウンロードがまだの
方はこちら

自己紹介



北川 恭平 (きたがわ きょうへい)

Details

- 埼玉県春日部市出身
- 東京オフィス勤務 @日本橋
- cybozu developer networkサイトのプロデューサー

Favorite

- 筋トレ #百獣の王を目指しています。
- 読売の次に日ハムファンです。



<https://bit.ly/ki-contents>



やること

ダウンロードがまだの方は
こちら

- カスタマイズ用アプリケーションの作成
 - 出張申請アプリの作成
 - 社員マスタと出張申請をルックアップで連携設定
 - 豆知識：kintoneのショートカットキーを紹介
- 完成版の動作を確認
 - カスタマイズの適用手順
 - JavaScript
 - kintoneのイベントと操作
- カスタマイズ①「フィールドの編集不可」
 - コーディング
 - 開発者ツール
 - フィールドコード
 - モバイルアプリ対応
- カスタマイズ②「フィールドの値書き換え」

カスタマイズ用アプリケーションの作成

準備いただいた環境にログイン

A screenshot of the Cybozu login interface. At the top, there is a browser address bar showing a masked URL ending in '.cybozu.com' and a 'ヘルプ' (Help) link. Below this are two input fields: 'ログイン名' (Login Name) and 'パスワード' (Password). At the bottom left, there is a checkbox labeled 'ログイン名を保存する' (Save login name). To the right of the checkbox is a blue 'ログイン' (Login) button. The footer of the page displays 'www.cybozu.com'.

アプリテンプレートの読み込み

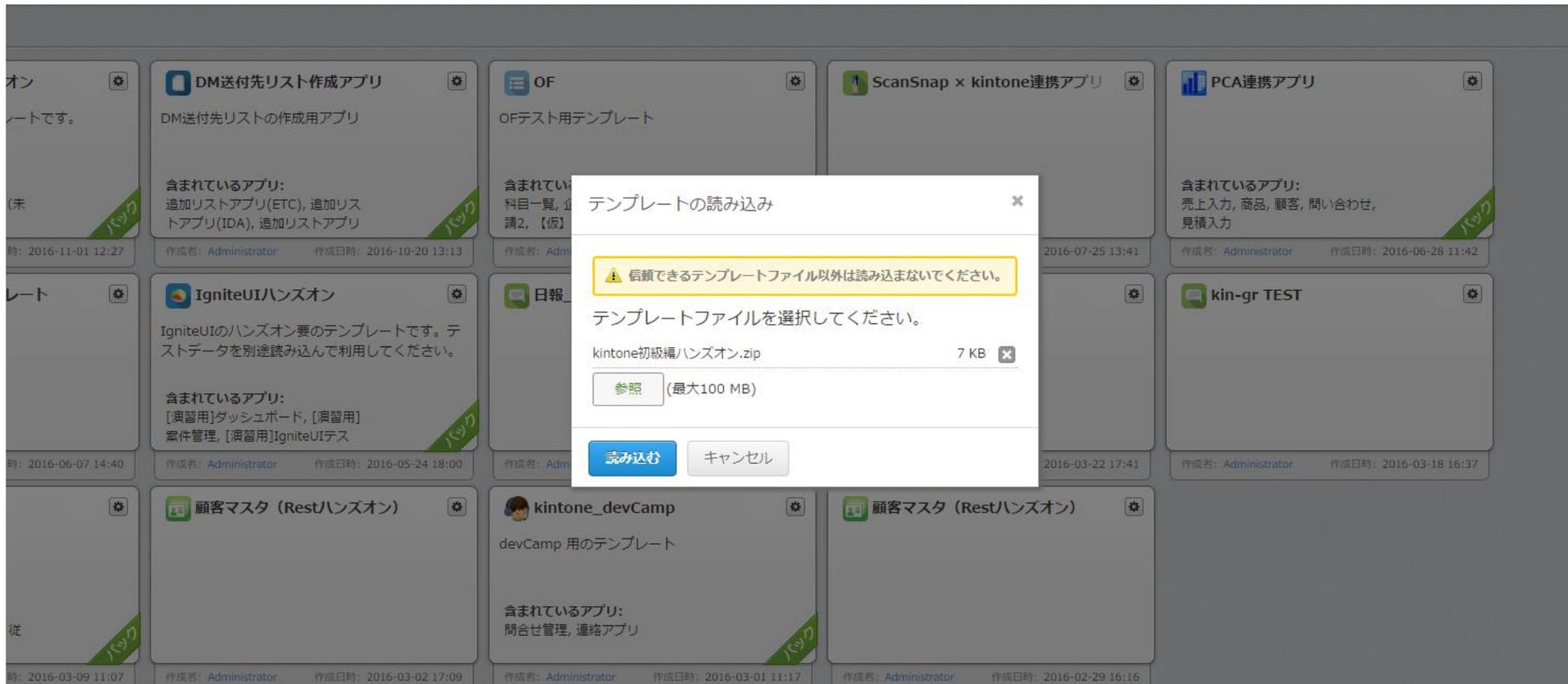
- 歯車アイコンをクリック
- kintoneシステム管理をクリック



■ アプリテンプレートをクリック



「読み込む」よりkintone初級編ハンズオン.zipを読み込む



The screenshot shows a grid of Kintone application templates. A modal dialog titled "テンプレートの読み込み" (Template Upload) is open in the center. The dialog contains the following text:

テンプレートの読み込み

⚠ 信頼できるテンプレートファイル以外は読み込まないでください。

テンプレートファイルを選択してください。

kintone初級編ハンズオン.zip 7 KB

参照 (最大100 MB)

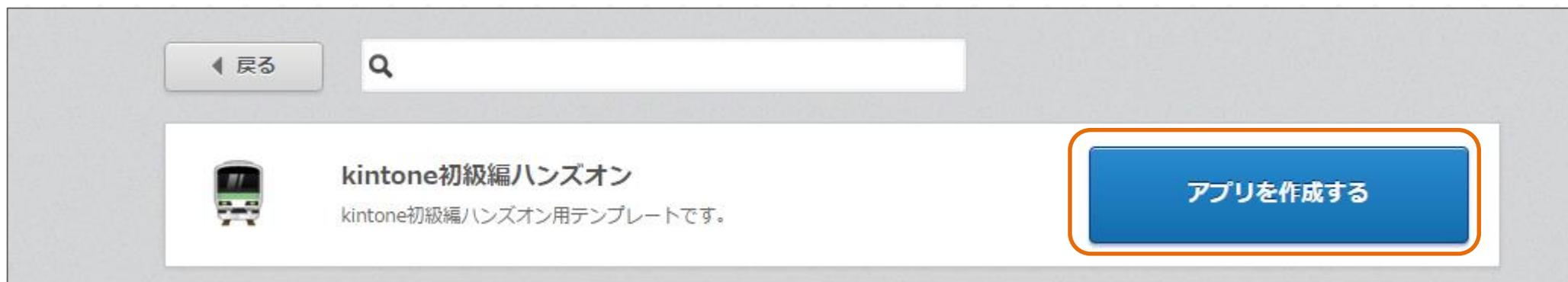
読み込む キャンセル

The background shows several application cards, including "DM送付先リスト作成アプリ", "IgniteUIハンズオン", "顧客マスタ (Restハンズオン)", and "kintone_devCamp". Each card displays the application name, a brief description, and the creator information.

■ アプリ横の+アイコンをクリック



■ 「kintone初級編ハンズオン」横のアプリを作成するをクリック



■ アプリが2つ追加されたことを確認



出張申請(未完成)で出張申請をしてみましょう

出張することになったとき・出張で交通費等の立替金が発生したときに申請するアプリです。
社外にいても確認・処理ができるので、申請から決裁までのスピードアップが図れます。

キャンセル **保存**

※データを保存した段階では、下書きの状態になっています。
上長に申請するときには「申請する」ボタンを押してください。

タイトル 日付 *

札幌出張 2016/11/11 2016-11-01 16:32

申請者 * 承認者

Administrator 人事部のベテラン

所属部署 人事部

出張日

2016-11-01

申請する 現在の作業者を変更

ステータス: 未申請 (下書き) ステータスの履歴

※データを保存した段階では、下書きの状態になっています。
上長に申請するときには「申請する」ボタンを押してください。

タイトル 日付

札幌出張 2016/11/11 2016-11-01 16:32

申請者 所属部署 承認者

Administrator 人事部 人事部のベテラン

出張日

2016-11-01

出張先 地域 日当

0円

**所属部署と承認者の入力が毎回めんどくさい・・・
(;´・ω・)**

社員マスタからデータを参照できれば・・・

解決策：ルックアップ機能

cybozu.com ヘルプ
kintone ユーザーヘルプ

何をお探しですか？ キーワードで検索できます。

kintoneを新デザインで使用中心 ?

kintone ユーザーヘルプ > アプリ > アプリのカスタマイズ > フォームの設定 > 他のアプリからデータを取得できるようにする > ルックアップとは

ルックアップとは

ルックアップとは、アプリに入力するデータを他のアプリから取得できるようにする機能です。データ入力の手間を省き、入力ミスも防げます。

アプリB

顧客ID	会社名	住所	電話番号
B-1327	サイボウズ	東京都...	03-580...
B-1538	安井電機	大阪府...	06-430...
C-1328	岡崎商会	東京都...	03-232...

アプリA

会社名 住所

担当者名 電話番号 メールアドレス

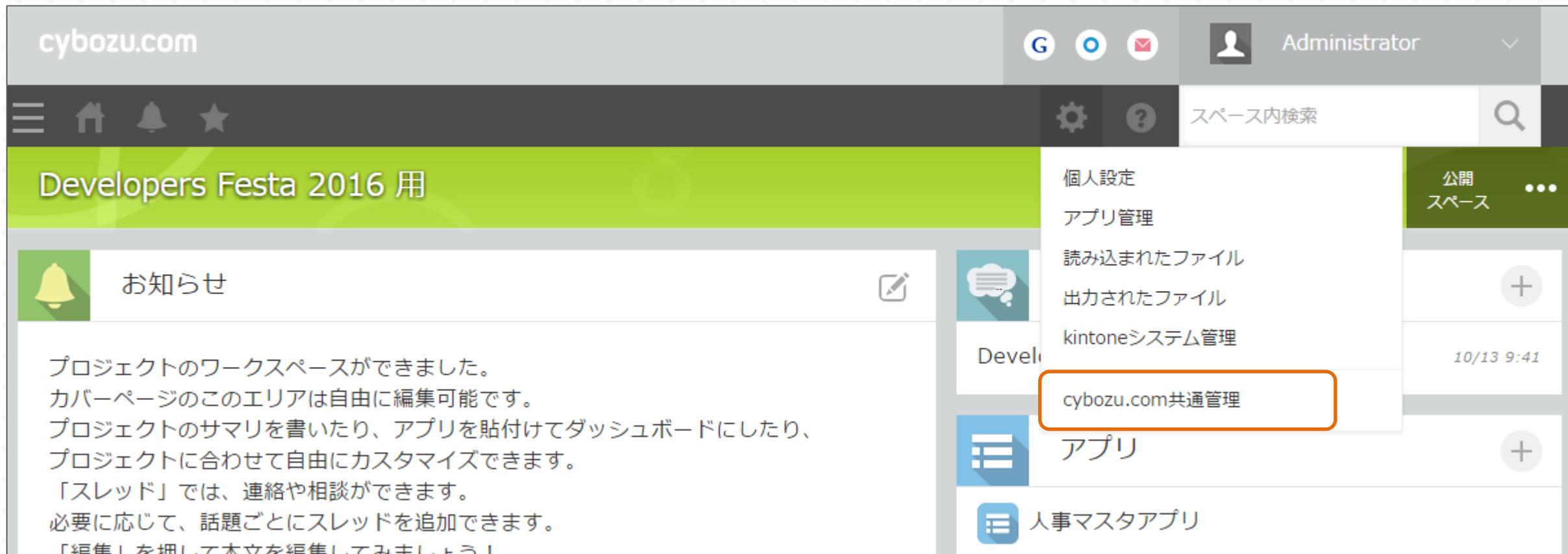
販売商品 単価 販売数 小計

案件担当者 確度 状況

たとえば、営業案件を管理するアプリに、顧客情報を管理するアプリを参照するルックアップを設定したとします。すると、次のように、顧客情報を取得して営業案件を管理するアプリにコピーできるようになります。

https://help.cybozu.com/ja/k/user/whats_lookup.html

準備（組織とユーザーの追加）



The screenshot shows the cybozu.com web interface. At the top left, the URL "cybozu.com" is displayed. In the top right corner, there are navigation icons (Google, Home, Mail) and a user profile labeled "Administrator". Below the navigation bar, there is a green header for "Developers Festa 2016 用" and a search bar labeled "スペース内検索".

A notification banner is visible, titled "お知らせ" (Notice), with a bell icon. The notification text reads:

プロジェクトのワークスペースができました。
 カバーページのこのエリアは自由に編集可能です。
 プロジェクトのサマリを書いたり、アプリを貼付けてダッシュボードにしたり、
 プロジェクトに合わせて自由にカスタマイズできます。
 「スレッド」では、連絡や相談ができます。
 必要に応じて、話題ごとにスレッドを追加できます。
 「編集」を押して本文を編集してみましょう！

- 個人設定
- アプリ管理
- 読み込まれたファイル
- 出力されたファイル
- kintoneシステム管理
- cybozu.com共通管理**

アプリ

人事マスタアプリ

公開
スペース

10/13 9:41

準備（組織とユーザーの追加） 営業部を追加

組織とユーザーの設定

組織
組織の追加



組織の追加

組織名 表示名 *

別言語での表示名

日本語 ▾

指定した言語のユーザーに表示される組織名を入力します。

組織コード ✔ 使用できます。

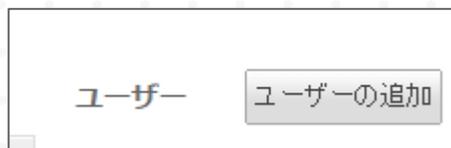
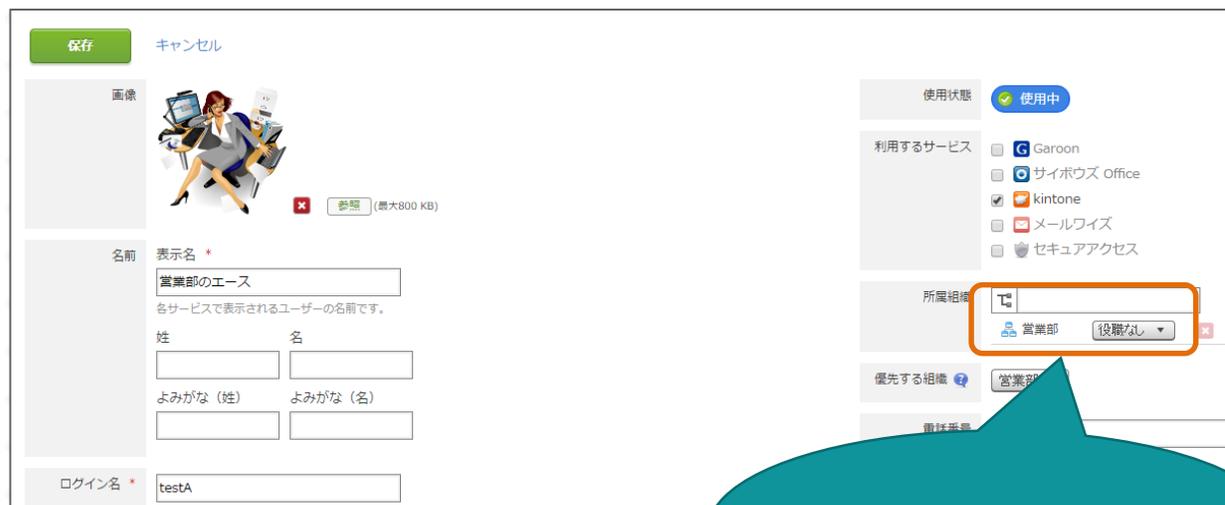
ほかの組織と重複しない任意の文字列を入力します。空にすると自動で設定されます。

親組織

説明

保存
キャンセル

準備（組織とユーザーの追加） 営業部所属ユーザーの追加

保存 キャンセル

画像  参照 (最大800 KB)

名前 表示名 *
営業部のエース
各サービスで表示されるユーザーの名前です。

姓 名
よみがな (姓) よみがな (名)

ログイン名 * testA

使用状態 使用中

利用するサービス
 Garoon
 サイボウズ Office
 kintone
 メールワイズ
 セキュアアクセス

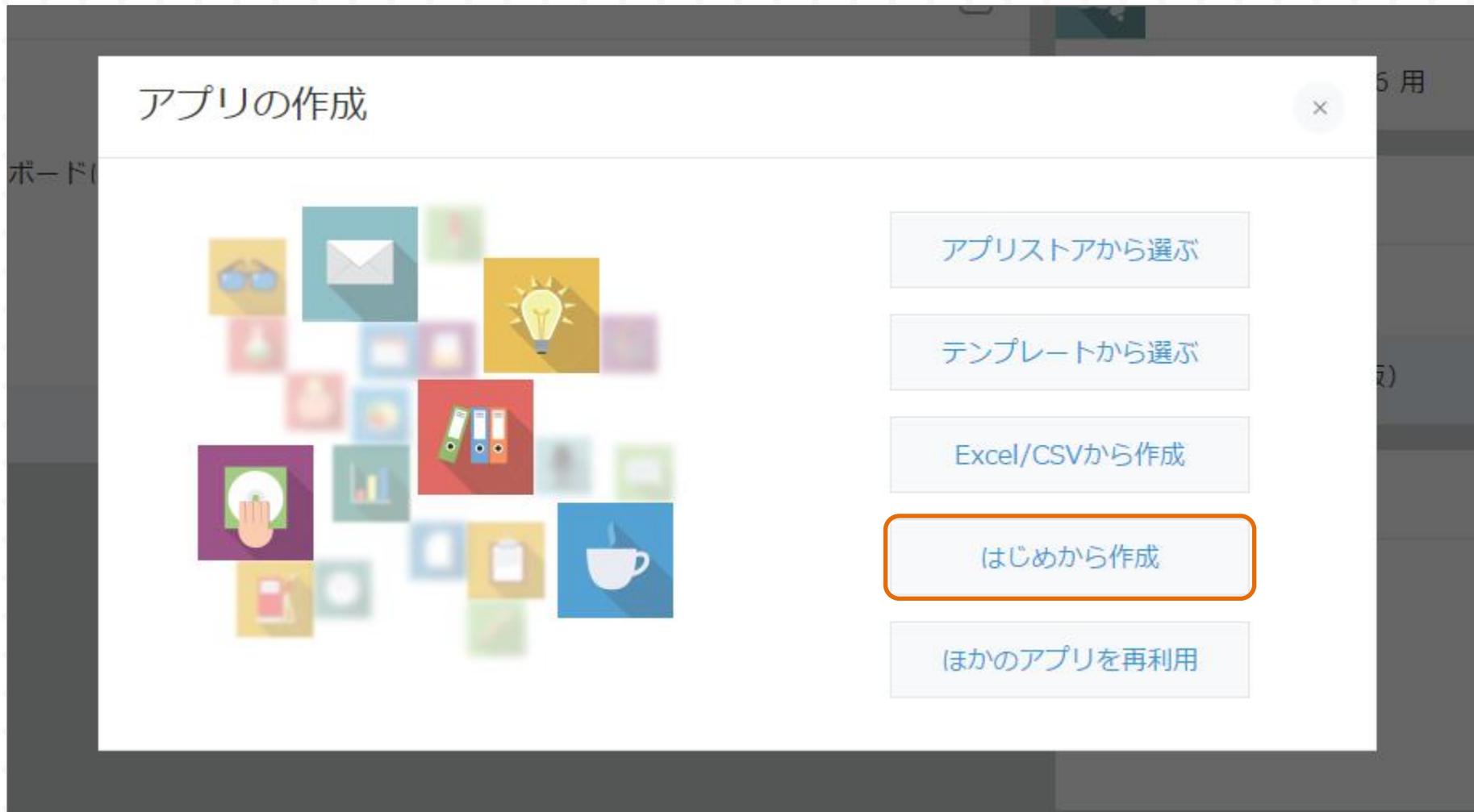
所属組織 [役職なし]

優先する組織

電話番号

所属を営業部に

社員マスタアプリをはじめから作成より作成



社員を登録

人事マスタアプリ > (すべて) > 1

社員番号	氏名
<input type="text" value="1234"/>	<input type="text" value="中田 羽平"/>
所属部署	承認者
<input type="text" value=""/>  人事部 <input type="button" value="×"/>	<input type="text" value=""/>  人事部のベテラン <input type="button" value="×"/>

出張申請(未完成)アプリで氏名フィールドを追加

フィールド名	フィールドの種類
文字列(1行)	氏名



出張申請 (未完成版)

フォーム

一覧

グラフ

設定

フォームを保存

📍 ラベル

ABC 文字列 (1行)

A リッチエディター

☰ 文字列 (複数行)

123 数値

📊 計算

📻 ラジオボタン

☑️ チェックボックス

☰ 複数選択

☑️ ドロップダウン

📅 日付

🕒 時刻

📅 日時

📎 添付ファイル

🌐 リンク

👤 ユーザー選択

👥 組織選択

👥 グループ選択

🔗 関連レコード一覧

🔍 ルックアップ

※データを保存した段階では、下書きの状態になって
上長に申請するときには「申請する」ボタンを押し

タイトル

申請者 (自動入力)

氏名

出張日

2016-10-13

出張先

出張申請(未完成)アプリでルックアップを設定

設定項目	設定値
フィールド名	社員番号
ほかのフィールドのコピー	所属部署 < [人事マスタアプリ]所属部署 承認者 < [人事マスタアプリ]承認者 氏名 < [人事マスタアプリ]氏名
コピー元のレコード選択時に表示するフィールド	社員番号、氏名

フィールドの設定

必須項目にする

関連付けるアプリ * コピー元のフィールド *

人事マスタアプリ 社員番号

フォームの保存後は、上記2つの設定は変更できません。

ほかのフィールドのコピー

承認者 < [人事マスタアプリ]承認者

所属部署 < [人事マスタアプリ]所属部署

氏名 < [人事マスタアプリ]氏名

コピー元のレコードの選択時に表示するフィールド

社員番号

氏名

絞り込みの初期設定

すべてのレコード

すべてクリア

ソートの初期設定

レコード番号 降順

フィールドコード *

ルックアップ

キャンセル 保存

動作確認

Developers Festa 2016 用 > 出張申請 (未完成版)

出張することになったとき・出張で交通費等の立替金が発生したときに申請するアプリです。
社外にいても確認・処理ができるので、申請から決裁までのスピードアップが図れます。

※データを保存した段階では、下書きの状態になっています。
上長に申請するときには「申請する」ボタンを押してください。

タイトル
 日付 *

申請者 *
 社員番号

 氏名
 所属部署
 承認者

出張日

豆知識 : kintoneのショートカットキーを紹介

一覧画面で裏コマンド「？」を入力

Developers Festa 2016 用

Developers Festa 2016 用 > 出張申請 (未完成版)

出張することになったとき・出張で交通費等の立替金が発生したときに申請するアプリです。
社外においても確認・処理ができるので、申請から決裁までのスピードアップが図れます。

申請一覧

申請者	タイトル	出
Administrator	札幌出張	札幌

1 - 1 (1件中)

1 - 1 (1件中)

ショートカットキーの一覧

- enter** : 選択したレコードの詳細を表示する
- /** : 検索キーワードを入力する
- ?** : このウインドウの表示/非表示を切り換える
- c** : レコードを追加する
- e** : 選択したレコードを編集する
- j** : 次のレコードを選択する
- k** : 前のレコードを選択する
- n** : 次のページを表示する
- o** : 選択したレコードの詳細を表示する
- p** : 前のページを表示する

レコード選択「j」と「k」

Developers Festa 2016 用 > 出張申請 (未完成版)

出張することになったとき・出張で交通費等の立替金が発生したときに申請するアプリです。
社外にいても確認・処理ができるので、申請から決裁までのスピードアップが図れます。

申請一覧   

申請者	タイトル	出張先
 Administrator	札幌出張	札幌コンベンションセンター

選択された行が反転 (Enterで詳細へ)

レコード追加「c」

Developers Festa 2016 用 > 出張申請 (未完成版)

出張することになったとき・出張で交通費等の立替金が発生したときに申請するアプリです。
社外にいても確認・処理ができるので、申請から決裁までのスピードアップが図れます。

キャンセル 保存

※データを保存した段階では、下書きの状態になっています。
上長に申請するときには「申請する」ボタンを押してください。

タイトル 日付 * (自動入力)

申請者 * (自動入力) 社員番号 取得 クリア 氏名 所属部署 承

出張日

出張先 地域 日当 円

出張目的

追加画面をサクッと開く

レコード編集「e」

Developers Festa 2016 用 > 出張申請 (未完成版)

出張することになったとき・出張で交通費等の立替金が発生したときに申請するアプリです。社外においても確認・処理ができるので、申請から決裁までのスピードアップが図れます。

申請一覧

申請者	タイトル	出張先
Administrator	札幌出張	札幌コンベンションセンター



Developers Festa 2016 用 > 出張申請 (未完成版) > 申請一覧 > 1

出張することになったとき・出張で交通費等の立替金が発生したときに申請するアプリです。社外においても確認・処理ができるので、申請から決裁までのスピードアップが図れます。

キャンセル 保存

※データを保存した段階では、下書きの状態になっています。
上長に申請するときには「申請する」ボタンを押してください。

タイトル 日付*

申請者* 社員番号 取得 クリア 氏名 所属部署 承認者  K.Kitagawa

出張日

出張先 地域 日当 円

出張目的

kintoneをつかっているうちに更なる欲望が・・・

- **日当を編集不可にしたい**
- **日当を出張先によって、変更したい**

kintone をJavaScriptでカスタマイズしましょう！
ω ('ω') ω

■ 追加された出張申請（完成版）をクリック



カスタマイズ①

レコードの登録画面を表示

キャンセル
保存

※データを保存した段階では、下書きの状態になっています。
上長に申請するときには「申請する」ボタンを押してください。

タイトル		申請日 *	(自動入力)
	<input style="width: 100%;" type="text"/>		
申請者 *	社員番号	所属部署	承認者
(自動入力)	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text" value="-----"/> ▾	<input style="width: 100%;" type="text"/> 🔍
出張日			
<input style="width: 100%;" type="text" value="2016-01-15"/>			
出張先	地域	日当	
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text" value="-----"/> ▾	<input style="width: 100%;" type="text" value="0"/>	円

日当が編集不可になっている

カスタマイズ②

■ 地域の選択内容を変更すると・・・

キャンセル
保存

※データを保存した段階では、下書きの状態になっています。
上長に申請するときには「申請する」ボタンを押してください。

タイトル		申請日 *	(自動入力)
申請者 *	社員番号	所属部署	承認者
(自動入力)	<input type="text"/>	<input type="text" value="-----"/>	<input type="text"/>
出張日	2016-01-15		
出張先	地域	日当	
<input type="text"/>	<input type="text" value="関東 (首都圏以..."/>	<input type="text" value="1000"/>	円

地域を変更すると日当の値が変わる

カスタマイズの適用手順

カスタマイズの適用手順

1. JavaScriptファイルの作成（修正）
 2. kintoneアプリに適用
 3. 動作確認
- 

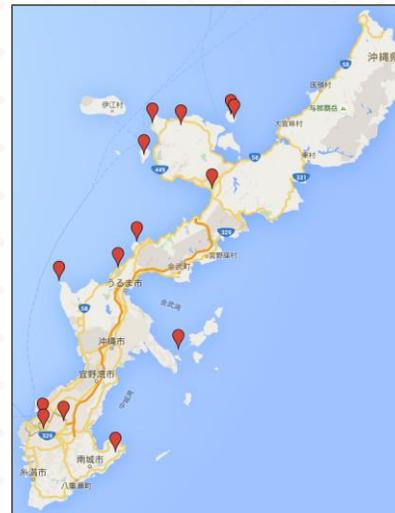
JavaScript

JavaScriptとは

- JavaScriptはプロトタイプベースのオブジェクト指向プログラミング言語である
- 実行環境が主にウェブブラウザに実装され、動的なウェブサイト構築や、リッチインターネットアプリケーションなど高度なユーザインタフェースの開発に用いられる

※引用) Wikipedia

- 1995年誕生
- 2005年のGoogleMapで有名に



GoogleMapでどう使われているか

- | | | |
|-----------|-----|--------------|
| ■ ドラッグ | したら | 移動 |
| ■ クリック | したら | ピンを立てる |
| ■ ホイールが回転 | したら | ズームアウト/ズームイン |

GoogleMapでどう使われているか

- ドラッグ したら
- クリック したら
- ホイールが回転 したら

イベント

- 移動
- ピンを立てる
- ズームアウト/ズームイン

操作

kintoneにもイベントと操作がある

例えば・・・

- | | | |
|-----------------|-----|--------------------|
| • レコード一覧画面を表示 | したら | フィールドの値を書き換える |
| • レコード詳細画面を表示 | したら | フィールドの表示/非表示を切り替える |
| • レコード追加画面で保存 | したら | フィールドにエラーを表示する |
| • レコード編集画面で値を変更 | したら | フィールドの表示/非表示を切り替える |

kintoneのイベントと操作

イベントと操作の確認

- <https://cybozudev.zendesk.com/>に移動し、ブックマーク
- API関連にマウスオン
- kintone JavaScript APIへ



■ kintoneにはレコード一覧やレコード表示などのイベントがある



The screenshot shows the Kintone API documentation page for 'Event Handling' (イベント処理). The left sidebar contains a navigation menu with categories like 'Kintone API', 'REST API', and 'JavaScript API'. Under 'JavaScript API', the 'Event' (イベント) category is expanded, and 'Event Handling Method' (イベント処理の記述方法) is selected and highlighted with an orange circle. The main content area is titled 'イベント処理の記述方法' and includes an 'Index' section with links to 'Event Handling Overview', 'Register Event Handler', 'Delete Specific Event Handler', 'Delete All Event Handlers', and 'Notes'. Below this is the 'イベント処理の概要' (Event Handling Overview) section, which states that Kintone supports various events and handlers. The 'イベントハンドラーを登録する' (Register Event Handler) section includes a list of supported devices (PC and smartphone) and a note that it is not supported in mobile apps. The '関数' (Function) section shows the `kintone.events.on(event, handler(event))` syntax. The 'プロパティ' (Property) section has a table with columns for parameter name, value, required status, and description.

■ 詳細ページにイベントと操作の一覧がある

JavaScript API

イベント

- イベント処理の記述方法
- レコード一覧イベント
- レコード表示イベント
- レコード追加イベント
- レコード編集イベント
- レコード印刷画面表示イベント
- グラフ表示イベント

レコード一覧イベント

Index

イベント

- レコード一覧画面の表示後イベント
- レコード一覧画面のインライン編集保存実行前イベント
- レコード一覧画面のインライン編集保存成功後イベント
- レコード一覧画面のインライン編集開始時イベント
- レコード一覧画面のフィールド値変更時イベント
- レコード一覧画面のレコード削除前イベント

イベント発生時に実行できる操作

- フィールドの値を書き換える
- フィールドの編集可/不可を設定する
- フィールドにエラーを表示する
- レコードにエラーを表示する
- 変更されたフィールド、テーブル内の行オブジェクトを取得する

JavaScript API

イベント

- イベント処理の記述方法
- レコード一覧イベント
- レコード表示イベント
- レコード追加イベント
- レコード編集イベント
- レコード印刷画面表示イベント
- グラフ表示イベント

レコード表示イベント

Index

イベント

- レコード詳細画面が表示された時のイベント
- レコード詳細画面の削除前イベント
- プロセス管理のアクションイベント

イベント発生時に実行できる操作

- フィールドの表示非表示を切り替える

- JavaScript APIは各画面で、可能なイベントと操作が決まっている
- 以下は各画面におけるイベントと操作の例（一部）

画面	イベント	操作
一覧	編集開始時	フィールドの編集可/不可を設定する
一覧	保存実行前 (保存ボタンクリック後)	フィールドの値を書き換える フィールドにエラーを表示する レコードにエラーを表示する
詳細	表示後	フィールドの表示/非表示を切り替える
追加/編集	表示後	フィールドの値を書き換える フィールドの表示/非表示を切り替える フィールドの編集可/不可を設定する
追加/編集	保存実行前/保存成功後 (保存ボタンクリック後)	フィールドの値を書き換える フィールドにエラーを表示する レコードにエラーを表示する

例) 可能 …… 追加画面で「保存実行前」に「フィールドの値を書き換える」

不可能 …… 詳細画面で「表示後」に「フィールドの値を書き換える」

※APIを利用せずに実装するとバージョンアップで動作しなくなる可能性

例1

■ 一覧画面を編集開始したらフィールドの編集可/不可を設定する

レコード番号	見積番号	見積日	タイトル	見積金額
1	100	2015-02-15	GR_WFカスタマイズ	¥ 1000

レコード番号	見積番号	見積日	タイトル	見積金額
1	100	2015-02-15	GR_WF	¥ 1000

編集不可

例2

■ 詳細画面を表示したらフィールドの表示/非表示を切り替える

1, 今までにかかったことのある病気はありますか？

インフルエンザ
おたふくかぜ
じんま疹
その他

2, 今までに受けた予防接種はありますか？ *

ない ある

例2

■ 詳細画面を表示したらフィールドの表示/非表示を切り替える

1, 今までにかかったことのある病気はありますか？ その他

インフルエンザ

おたふくかぜ

じんま疹

✓ その他

その他を選択すると

2, 今までに受けた予防接種はありますか？ *

ない ある

フィールドを表示

例2

■ 詳細画面を表示したらフィールドの表示/非表示を切り替える

1, 今までにかかったことのある病気はありますか？ その他

インフルエンザ

おたふくかぜ

じんま疹

✓ その他

2, 今までに受けた予防接種はありますか？ * 予防接種名

ない
 ある

あるを選択すると

フィールドを表示

カスタマイズ① フィールドの編集不可

カスタマイズ①でやりたいこと

■ やりたいこと

利用者が日当を自由に入力できないようにしたい

キャンセル
保存

※データを保存した段階では、下書きの状態になっています。
上長に申請するときには「申請する」ボタンを押してください。

タイトル		申請日 *	(自動入力)
申請者 *	社員番号	所属部署	承認者
(自動入力)	<input type="text"/>	-----	<input type="text"/>
出張日			
<input type="text" value="2016-01-15"/>			
出張先	地域	日当	
<input type="text"/>	-----	<input style="background-color: #ccc;" type="text" value="0"/>	円

日当を編集不可にする

■ やりたいこと

利用者が日当を自由に入力できないようにしたい

 レコード追加画面で日当を入力不可にすれば実現できる

 ・ イベント
レコード追加画面が表示されたら

・ 操作
フィールドを編集不可にする

★TRY★

以下をdeveloper networkで確認しよう

- ・ イベント「レコード追加画面が表示されたら」が存在するか
- ・ 操作「フィールドを編集不可にする」が存在するか

イベントも操作もあるので、カスタマイズ可能！

レコード追加イベント

Index

イベント

- レコード追加画面が表示された時のイベント
- レコード追加画面の保存実行前イベント
- レコード追加画面の保存成功後イベント
- レコード追加画面のフィールド値変更時イベント

イベント発生時に実行できる操作

- フィールドの値を書き換える
- フィールドの編集可/不可を設定する
- フィールドにエラーを表示する
- レコードにエラーを表示する
- ルックアップの取得を自動で行う
- フィールドの表示非表示を切り替える
- 変更されたフィールド、テーブル内の行オブジェクトを取得する

レコード追加画面が表示された時のイベント

- PC用とスマートフォン用で利用できるイベントです。
- スペースに貼りつけたアプリにはこのイベントは発生しません。

関数

環境	関数
PC用	app.record.create.show
スマートフォン用	mobile.app.record.create.show

eventオブジェクトのプロパティ

PC用、スマートフォン用で同一の内容となります。

プロパティ名	型	説明
appId	数値	アプリID
reuse	真偽値	再利用の場合は true。通常の追加画面は false。
record	オブジェクト	レコード登録画面表示時のデータ（フォー

フィールドの値に応じた制御

- フィールドの値の書き換える をご参照ください。
- フィールドの編集可/不可を設定する をご参照ください。
- フィールドにエラーを表示する をご参照ください。
- 画面の上部にエラーを表示する をご参照ください。
- ルックアップの取得を自動で行う をご参照ください。
- フィールドの表示/非表示を切り替えるをご参照ください。

コーディング

コーディングルール1

- ファイルの拡張子はjsで保存（例：sample1.js）

今後はこのファイルを編集していきます。

- ファイルは文字コードUTF-8で保存
正確にはUTF-8N（BOMなし）で保存

コーディングルール2

■ 即時関数で記述

```
(function() {  
})();
```



- 変数の影響範囲を即時関数内だけに制限する
(即時関数外の変数を上書きしない)
- 関数の定義と呼び出しを同時に実行する

コーディングルール3

■ strict (厳格) モードを使用

```
(function() {  
    'use strict';  
})();
```



- 的確なエラーチェックが行われる。

例) 宣言していない変数の使用

イベントの記述

■ イベントと操作は以下のように記述

```
kintone.events.on('イベント', function(event) {
```

```
  // 操作
```

```
});
```

//で始まる行はコメント行

```
(function() {
  'use strict';

  // レコード追加画面が表示されたら
  kintone.events.on('イベント', function(event) {
    // 操作
  });
})();
```

★TRY★

'イベント'をレコード追加画面が表示されたときのイベント名に書き換えよう

■ ヒント

レコード追加画面が表示された時のイベント

- PC用とスマートフォン用で利用できるイベントです。
- スペースに貼りつけたアプリにはこのイベントは発生しません。

関数

環境	関数	イベントが発生するタイミング
PC用	app.record._____	<ul style="list-style-type: none"> • レコード追加画面または再利用画面が表示された時
スマートフォン用	mobile.app.record.create	<ul style="list-style-type: none"> • レコード追加画面または再利用画面が表示された時

これがイベント

<https://cybozudev.zendesk.com/hc/ja/articles/201941984>

■ 正解

```
(function() {  
  'use strict';  
  
  // レコード追加画面が表示されたら  
  kintone.events.on('app.record.create.show', function(event) {  
    // 操作  
  });  
})();
```

操作の記述

■ フィールドの編集可／不可を設定するをクリック

レコード追加イベント
Index
イベント
レコード追加画面が表示された時のイベント
レコード追加画面の保存実行前イベント
レコード追加画面の保存成功後イベント
レコード追加画面のフィールド値変更時イベント
イベント発生時に実行できる操作
フィールドの値を書き換える
フィールドの編集可／不可を設定する
フィールドにエラーを表示する
レコードにエラーを表示する
ロックアップの取得を自動で行う
フィールドの表示非表示を切り替える
変更されたフィールド、テーブル内の行オブジェクトを取得する
その他
関連Tips

1行目 `var record = event.record;`
2行目 `record['文字列_0']['disabled'] = true;`
3行目 `return event;`

フィールドの編集可/不可を設定する

ハンドラーが record オブジェクトのフィールドの disabled に true/false を代入して event オブジェクトを return し

- PC用とスマートフォン用で利用できます。
- スペースに貼りつけたアプリには利用できません。
- 編集権限のないフィールドを編集可能 (disabled を false) にしても反映されません。

サンプル

「ド롭ダウン_0」フィールドの値が変更になった時に、「文字列_0」フィールドを編集不可にします。

```
1 kintone.events.on('app.record.create.change.ド롭ダウン_0', function(event) {  
2   var record = event.record;  
3   record['文字列_0']['disabled'] = true;  
4   return event;  
5 });
```

編集可/不可にできないフィールド

以下のフィールドの値を編集可/不可を設定して return しても、フォームには反映されません。

- 計算
- ルックアップコピー先フィールド (旧デザインのみ)

サンプルの1行目の解説

```
kintone.events.on('app.record.create.show',  
function(event) {  
  var record = event.record;  
  record['文字列_0']['disabled'] = true;  
  return event;  
}
```

変数event配下のrecordを変数recordに代入

変数eventって？

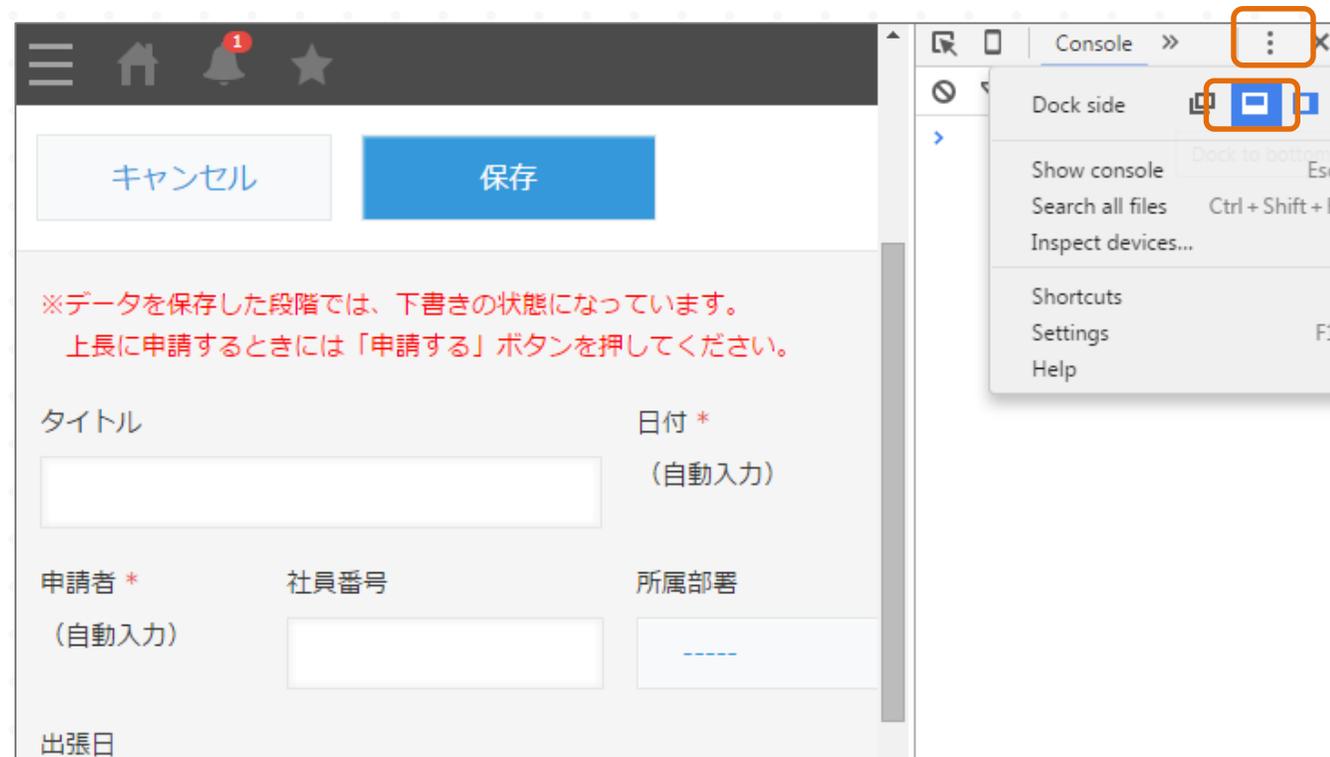
開発者ツール

開発者ツールとは

- 各ブラウザには開発者ツールが用意されている
- これを使う事で効率的に開発できる



■ 開発者ツールが右にある人は下記枠のアイコンをクリックして開発者ツールを下に移動



変数eventの内容を確認

- console.logで変数の内容を開発者ツールに出力
- console.log(event);は変数eventの内容を開発者ツールに出力

```
(function() {  
  'use strict';  
  
  // レコード追加画面が表示されたら  
  kintone.events.on('app.record.create.show', function(event) {  
    console.log(event);  
    var record = event.record;  
    console.log(record);  
  });  
})();
```

変数eventと変数recordの内容を開発者ツールに出力

作成したJavaScriptファイルをアプリに適用

- 追加された出張申請（未完成版）をクリック



・・・をクリック

アプリの設定を変更をクリック



The screenshot shows the cybozu.com web interface. The user is logged in as 'Administrator'. The main content area displays the '出張申請 (未完成版)' app. Below the app title, there is a description: '出張することになったとき・出張で交通費等の立替金が発生したときに申請するアプリです。社外にいても確認・処理ができるので、申請から決裁までのスピードアップが図れます。' Below the description, there are several icons for filtering and sorting, including a dropdown menu for '自分が作成したもの'. A three-dot menu icon is highlighted with a red box. The menu is open, showing options: '表示件数', '先頭行を固定表示' (checked), 'ファイルに書き出す', 'ファイルから読み込む', and 'アプリの設定を変更' (highlighted with a red box).

■ 詳細設定をクリック

■ JavaScript / CSSでカスタマイズをクリック

Developers Festa 2016 用 > 出張申請 (未完成版) > アプリの設定

 出張申請 (未完成版)

フォーム 一覧 グラフ 設定

 一般設定	 カスタマイズ	 その他の設定
アイコンと説明	プラグイン	カテゴリー
プロセス管理	JavaScript / CSSでカスタマイズ	言語ごとの名称
 通知	APIトークン	レコードのタイトル
アプリの条件通知	 アクセス権	高度な設定
レコードの条件通知	アプリ	アクション
リマインダーの条件通知	レコード	アプリの動作テスト
	フィールド	アプリを削除

- アップロードして追加をクリックし、先ほど作成したJavaScriptファイルを選択
- 保存をクリック

※console.logを使って変数event、recordの内容を確認



- アプリを更新をクリック
- OKをクリック



- 開発者ツールを表示し、Consoleをクリック
(WindowsはF12、MacはCommand+Option+I)
- +をクリックして追加画面を表示



■ 2行データが表示されていることを確認

キャンセル
保存

タイトル

申請日 *

(自動入力)

申請者 *

(自動入力)

社員番号

所属部署

承認者

Elements Console Sources Network Timeline Profiles Resources Audits

<top frame> Preserve log

- ▶ Object {type: "app.record.create.show", appId: 275, record: Object, reuse: false}
- ▶ Object {文字列_1行_: Object, 数値: Object, ドロップダウン: Object, ユーザー選択: Object, 日付_0: Object...}

eventの内容

recordの内容

■ 三角アイコンをクリックして内容を展開

キャンセル
保存

タイトル

申請日 *

(自動入力)

申請者 *

(自動入力)

社員番号

所属部署

承認者

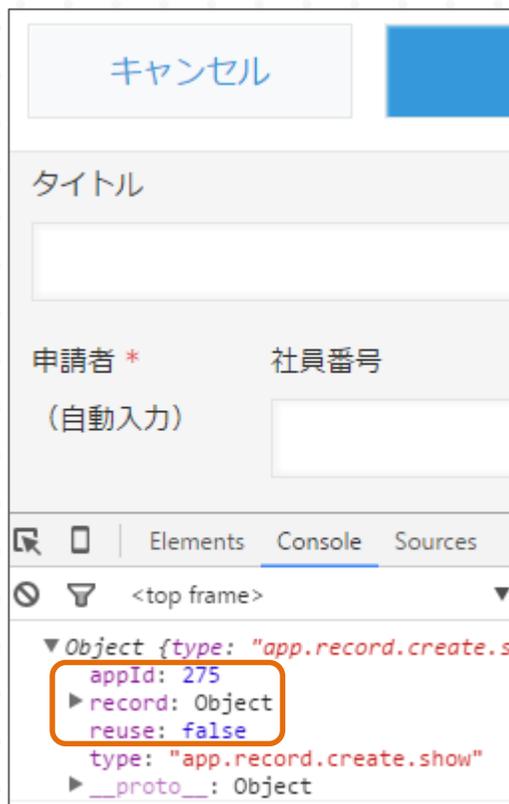
Elements
Console
Sources
Network
Timeline
Profiles
Resources
Audits

< top frame >
▼ Preserve log

▶ Object {type: "app.record.create.show", appId: 275, record: Object, reuse: false}

▶ Object {文字列_1行_: Object, 数値: Object, ドロップダウン: Object, ユーザー選択: Object, 日付_0: Object...}

1行目の内容



キャンセル

タイトル

申請者 * 社員番号

(自動入力)

▼ Object {type: "app.record.create.s
 appId: 275
 ▶ record: Object
 reuse: false
 type: "app.record.create.show"
 ▶ __proto__: Object

レコード追加画面が表示された時のイベント

- PC用とスマートフォン用で利用できるイベントです。
- スペースに貼りつけたアプリにはこのイベントは発生しません。

関数

環境	関数	イベントが発生するタイミング
PC用	app.record.create.show	<ul style="list-style-type: none"> • レコード追加画面または再利用画面が表示された時
スマートフォン用	mobile.app.record.create.show	<ul style="list-style-type: none"> • レコード追加画面または再利用画面が表示された時

eventオブジェクトのプロパティ

PC用、スマートフォン用で同一の内容となります。

プロパティ名	型	説明
appId	数値	アプリID
reuse	真偽値	再利用の場合は true。通常の追加画面は false。
record	オブジェクト	レコード登録画面表示時のデータ（フォームの初期値）を保持したレコードオブジェクト。

■ 2行目の内容

キャンセル
保存

タイトル

申請者 * 社員番号

(自動入力)

Elements Console Sources Network

<top frame>

- ▼ Object {type: "app.record.create.show", appId: 275, record: Object, reuse: false, type: "app.record.create.show", __proto__: Object}
- ▼ Object {文字列_1行: Object, 数値: Object, table: Object, area: Object, daily: Object, ドロップダウン: Object, ユーザー選択: Object, 数値: Object, 数値_1: Object, 文字列_1行_: Object, 文字列_1行__0: Object, 文字列_複数行_: Object, 文字列_複数行__0: Object, 日付_0: Object, __proto__: Object}

2行目の内容

```
▼ Object {タイトル: Ob  
  ▶ area: Object  
  ▶ daily: Object  
  ▶ タイトル: Obj  
  ▶ 交通手段: Object  
  ▶ 備考: Obj  
  ▶ 出張先: O  
  ▶ 出張日: O  
  ▶ 出張目的: Object  
  ▶ 合計金額: Object  
  ▶ 所属部署: Object  
  ▶ 承認者: Object  
  ▶ 社員番号: Object  
  ▶ 金額: Object
```

areaやdaily、タイトルなどがフィールドコード

areaとdailyの内容を展開

```
▼ Object {タイトル: Object,  
  ▼ area: Object  
    type: "DROP_DOWN"  
    value: undefined  
    ▶ __proto__: Object  
  ▼ daily: Object  
    type: "NUMBER"  
    value: "0"
```

フィールドコード

…をクリック

アプリの設定を変更をクリック



アプリ内検索

出張申請 (未完成版)

出張申請 (未完成版)

出張することになったとき・出張で交通費等の立替金が発生したときに申請するアプリです。
社外にいても確認・処理ができるので、申請から決裁までのスピードアップが図れます。

自分が作成したもの

申請者 タイトル

データがありません。

- 表示件数 >
- 先頭行を固定表示
- ファイルに書き出す
- ファイルから読み込む
- アプリの設定を変更**

フォーム

Developers Festa 2016 用 > 出張申請 (未完成版) > アプリの設定

出張申請 (未完成版)

フォーム 一覧 グラフ 設定

フォームを保存

- ラベル
- リッチエディター
- 数値
- ラジオボタン
- 複数選択
- 日付
- 日時
- リンク
- 組織選択
- 関連レコード一覧
- スペース
- グループ
- レコード番号
- 作成日時
- 更新日時

- 文字列 (1行)
- 文字列 (複数行)
- 計算
- チェックボックス
- ドロップダウン
- 時刻
- 添付ファイル
- ユーザー選択
- グループ選択
- ルックアップ
- 罫線
- 作成者
- 更新者

※データを保存した段階では、下書きの状態になっています。
 上長に申請するときには「申請する」ボタンを押してください。

タイトル 日付 (自動入力)

申請者 (自動入力) 社員番号 取得 クリア 氏名 所属部署 承認者

出張日

出張先 地域 日当 円

出張目的

- 地域にフォーカスをあわせて歯車をクリック
- 設定をクリック



The screenshot shows a web application interface with a dark header bar containing navigation icons (hamburger menu, home, bell, star). Below the header, there are tabs for 'フォーム' (Form), '一覧' (List), 'グラフ' (Graph), and '設定' (Settings). The 'フォーム' tab is active, displaying a form with a green 'フォームを保存' (Save Form) button at the top left. A red warning message is displayed: '※データを保存した段階では、下書きの状態になっています。上長に申請するときには「申請する」ボタンを押してください。' (Note: At the stage of saving data, it is in draft status. When applying to your supervisor, please click the 'Apply' button.)

The form contains several input fields: 'タイトル' (Title), '日付' (Date) with '(自動入力)' (Auto-input) below it, '申請者' (Applicant) with '(自動入力)' (Auto-input) below it, '氏名' (Name), '所属部署' (Department), and '出張日' (Business Trip Date) with the value '2016-11-02'. Below these is a table with columns for '出張先' (Business Trip Destination), '地域' (Region), '日当' (Daily Allowance), and '出張目的' (Business Trip Purpose). The '地域' column has a gear icon (settings) and a '設定' (Settings) button. A context menu is open over the '設定' button, showing '複製' (Copy) and '削除' (Delete) options.

- 地域のフィールドコードがareaになっていることを確認
- キャンセルをクリック



フィールドの設定

フィールド名 *

地域

フィールド名を表示しない

必須項目にする

項目と順番

首都圏

関東 (首都圏以外)

その他

初期値

フィールドコード *

area

キャンセル

保存

■ 同じように日当のフィールドコードがdailyになっていることを確認

フィールドの設定

フィールド名 *

日当

フィールド名を表示しない

桁区切りを表示する

必須項目にする

値の重複を禁止する

値の制限 (整数で指定)

最小 最大

初期値

小数点以下の表示桁数

単位記号

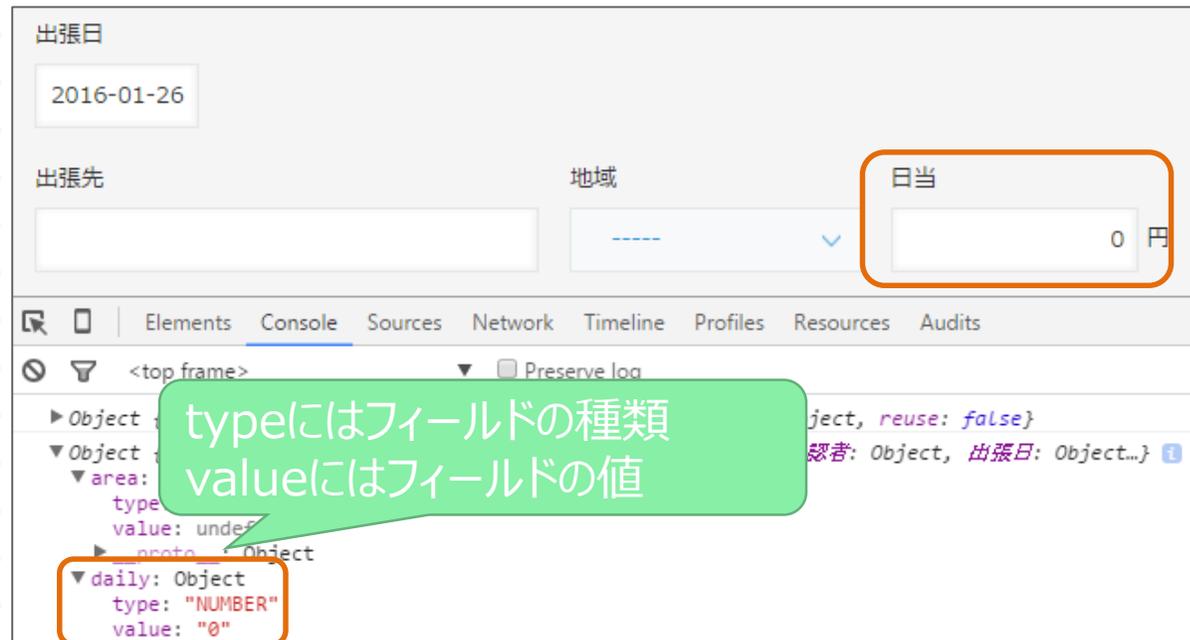
円 前に付ける (例: ¥100) 後ろに付ける (例: 100km)

フィールドコード *

daily

キャンセル 保存

- もう一度、開発者ツールを表示して、レコード追加画面を表示
- 表示直後の日当の値は0
- `record['daily']['value']`の値も0



出張日
2016-01-26

出張先 地域 日当
0 円

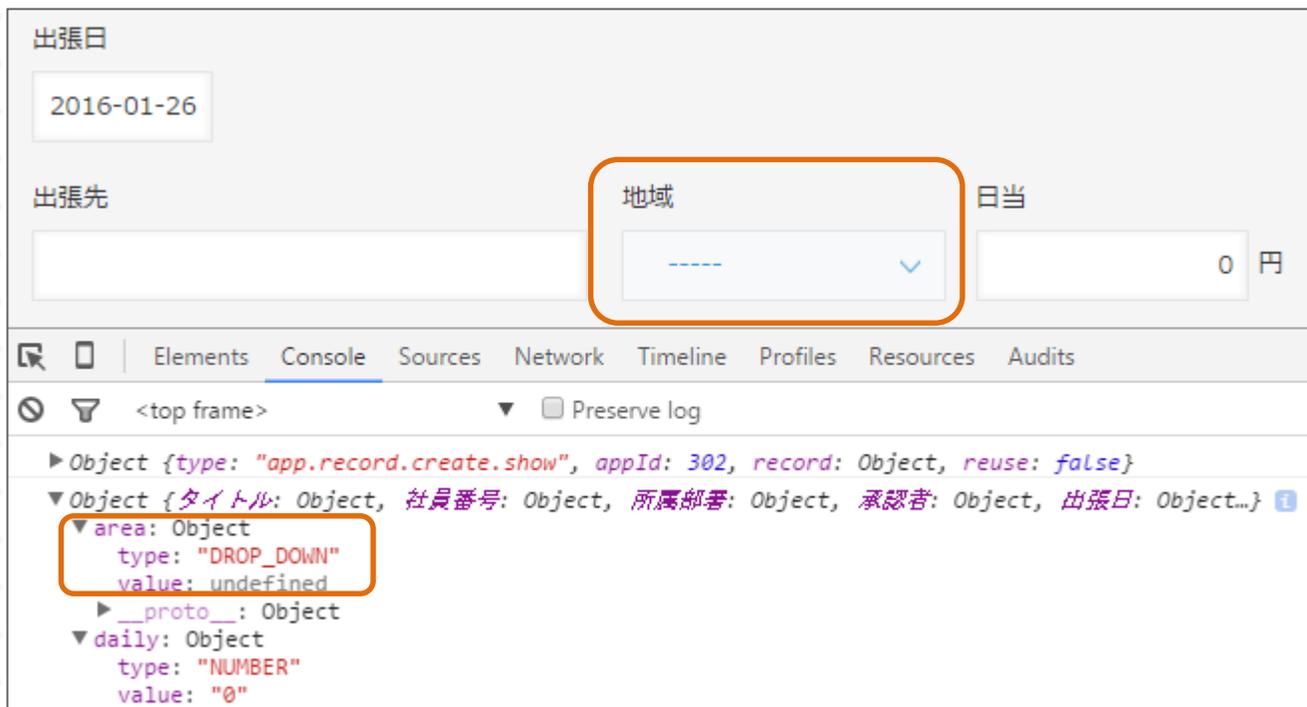
typeにはフィールドの種類
valueにはフィールドの値

```

Object {
  area: Object {
    type: "NUMBER"
    value: "0"
  }
  daily: Object {
    type: "NUMBER"
    value: "0"
  }
}

```

- 表示直後の地域は未選択
- record['area']['value']の値はundefined



出張日

2016-01-26

出張先

地域

日当

0 円

Elements Console Sources Network Timeline Profiles Resources Audits

<top frame> Preserve log

```

Object {type: "app.record.create.show", appId: 302, record: Object, reuse: false}
Object {タイトル: Object, 社員番号: Object, 所属部署: Object, 承認者: Object, 出張日: Object...}
  area: Object
    type: "DROP_DOWN"
    value: undefined
    __proto__: Object
  daily: Object
    type: "NUMBER"
    value: "0"
  
```

まとめ

- 変数eventにはイベント時のデータが入っている

イベントがapp.record.create.showの時はレコード追加画面表示時のデータ

参考)

- この内容を変更することでデータ（フォーム）の状態を変更できる

サンプルの2行目の解説

1行目 `var record = event.record;`

2行目 `record['文字列_0']['disabled'] = true;`

3行目 `return event;`

文字列_0はフィールドコード
フィールドコード配下のdisabledにtrue/falseに設定することで編集の可否を設定

例)
// フィールドコードがXXXフィールドの編集を不可にする
`record['XXX']['disabled'] = true;`

// フィールドコードがXXXフィールドの編集を可能にする
`record['XXX']['disabled'] = false;`

サンプルの3行目の解説

1行目 `var record = event.record;`

2行目 `record['文字列_0']['disabled'] = true;`

3行目 `return event;`

変更した内容を反映する
(これを忘れると2行目の変更が反映されない)

参考)

オブジェクトは参照渡しなので、2行目は以下と同じ意味
`event.record['文字列_0']['disabled'] = true;`

★TRY★

XXXを埋めて、日当を編集不可によう

```
(function() {  
  'use strict';  
  
  // レコード追加画面が表示されたら  
  kintone.events.on('app.record.create.show', function(event) {  
    var record = event.record;  
    record['XXX']['XXX'] = XXX;  
    return event;  
  });  
})();
```

正解

```
(function() {  
  'use strict';  
  
  // レコード追加画面が表示されたら  
  kintone.events.on('app.record.create.show', function(event) {  
    var record = event.record;  
    record['daily']['disabled'] = true;  
    return event;  
  });  
})();
```

動作確認

- アプリに適用
- レコード追加画面を表示
- 日当が編集できないことを確認

🚗 📌 出張申請 (未完成版)

出張申請 (未完成版)

出張することになったとき・出張で交通費等の立替金が発生したときに申請するアプリです。
社外にいても確認・処理ができるので、申請から決裁までのスピードアップが図れます。

キャンセル
保存

※データを保存した段階では、下書きの状態になっています。
上長に申請するときには「申請する」ボタンを押してください。

<p>タイトル</p> <input style="width: 90%; height: 25px;" type="text"/>	<p>日付 *</p> <p>(自動入力)</p>		
<p>申請者 *</p> <p>(自動入力)</p>	<p>社員番号</p> <input style="width: 80%; height: 25px;" type="text"/>	<p>所属部署</p> <div style="border: 1px solid #ccc; padding: 2px; text-align: center;">-----</div>	<p>承認者</p> <input style="width: 80%; height: 25px;" type="text"/> <div style="float: right; margin-top: -20px;"> 🔍 👤 </div>
<p>出張日</p> <div style="border: 1px solid #ccc; padding: 2px; text-align: center;">2016-07-27</div>			
<p>出張先</p> <input style="width: 90%; height: 25px;" type="text"/>	<p>地域</p> <div style="border: 1px solid #ccc; padding: 2px; text-align: center;">関東 (首都圏以…</div>	<p>日当</p> <div style="border: 1px solid #ccc; padding: 2px; text-align: center; background-color: #f0f0f0;">1000 円</div>	

カスタマイズ② フィールドの値書き換え

カスタマイズ②でやりたいこと

■ やりたいこと

地域が変更されたとき、日当の値を変更したい

キャンセル
保存

※データを保存した段階では、下書きの状態になっています。
上長に申請するときには「申請する」ボタンを押してください。

タイトル		申請日 *	
<input type="text"/>		(自動入力)	
申請者 *	社員番号	所属部署	承認者
(自動入力)	<input type="text"/>	-----	<input type="text"/>
出張日			
<input type="text" value="2016-01-15"/>			
出張先	地域	日当	
<input type="text"/>	関東 (首都圏以...)	1000 円	

地域を変更すると日当の値を変更する

■ やりたいこと

地域が変更されたとき、日当の値を変更したい



- イベント
レコード追加画面のフィールド値が変更されたら
- 操作
フィールドの値を書き換える

★TRY★

以下をdeveloper networkで確認しよう

- イベント「レコード追加画面のフィールド値変更」が存在するか
- 操作「フィールドの値を書き換える」が存在するか

イベントも操作もあるので、カスタマイズ可能！

レコード追加イベント

Index

イベント

レコード追加画面が表示された時のイベント

レコード追加画面の保存実行前イベント

レコード追加画面の保存成功後イベント

レコード追加画面のフィールド値変更時イベント

イベント発生時に実行できる操作

フィールドの値を書き換える

フィールドの編集可/不可を設定する

フィールドにエラーを表示する

レコードにエラーを表示する

ルックアップの取得を自動で行う

フィールドの表示非表示を切り替える

変更されたフィールド、テーブル内の行オブジェクトを取得する

その他

関連Tips

レコード追加画面が表示された時のイベント

- PC用とスマートフォン用で利用できるイベントです。
- スペースに貼りつけたアプリにはこのイベントは発生しません。

関数

環境	関数
PC用	app.record.create.show
スマートフォン用	mobile.app.record.create.show

eventオブジェクトのプロパティ

PC用、スマートフォン用で同一の内容となります。

プロパティ名	型	説明
appid	数値	アプリID
reuse	真偽値	再利用の場合は true。通常の追加画面は false。
record	オブジェクト	レコード登録画面表示時のデータ（フォームの初期値）

フィールドの値に応じた制御

- **フィールドの値の書き換える** をご参照ください。
- **フィールドの編集可/不可を設定する** をご参照ください。
- **フィールドにエラーを表示する** をご参照ください。
- **画面の上部にエラーを表示する** をご参照ください。
- **ルックアップの取得を自動で行う** をご参照ください。
- **フィールドの表示/非表示を切り替える** をご参照ください。

イベントの記述

```
(function() {  
  'use strict';  
  
  // レコード追加画面でフィールドの値が変更されたら  
  kintone.events.on('イベント', function(event) {  
    // 操作  
  });  
})();
```

★TRY★

‘イベント’をレコード追加画面で「地域」（フィールドコード：area）の値が変更されたときのイベント名に書き換えよう

■ 正解

```
(function() {  
  'use strict';  
  
  // レコード追加画面でフィールドの値が変更されたら  
  kintone.events.on('app.record.create.change.area', function(event) {  
    // 操作  
  });  
})();
```

操作の記述

■ フィールドの値を書き換えるをクリック

レコード追加イベント
Index
イベント
レコード追加画面が表示された時のイベント
レコード追加画面の保存実行前イベント
レコード追加画面の保存成功後イベント
レコード追加画面のフィールド値変更時イベント
イベント発生時に実行できる操作
フィールドの値を書き換える
フィールドの編集可/不可を設定する
フィールドにエラーを表示する
レコードにエラーを表示する
ルックアップの取得を自動で行う
フィールドの表示非表示を切り替える
変更されたフィールド、テーブル内の行オブジェクトを取得する
その他
関連Tips

1行目 `var record = event.record;`

2行目 `record['文字列_0']['value'] = 'この文字列で上書き';`

3行目 `return event;`

フィールドの値を書き換える

ハンドラーが record オブジェクトのフィールドの値を書き換えて event オブジェクトを return した場合、その値でフィールドの値を更新します。

- PC用とスマートフォン用で利用できます。
- スペースに貼りつけたアプリには利用できません。
- ラジオボタンフィールドで空文字列を指定した場合、初期値の選択肢となります。
- フィールドの編集を不可にしても、値を書き換えることができます。
- 編集権限のないフィールドの値を書き換えた場合はフォームには反映されません。
- 最後のハンドラーが return しない場合、フィールドの値を更新しません。
- 複数のハンドラーが登録されている場合、最後のハンドラーが return した戻り値を基準に反映されます。

サンプル

レコード追加画面を開いた時に「文字列_0」フィールド、テーブル「Table」の1行目の「文字列_1」フィールドに初期値をセットし、さらにテーブルの末尾に行を追加します。
※サブテーブルの行追加にはフィールドタイプの指定が必要です

```
1 kintone.events.on('app.record.create.show', function(event) {
2   var record = event.record;
3   record['文字列_0']['value'] = 'この文字列で上書き';
4   record['Table']['value'][0]['value']['文字列_1']['value'] = 'テーブルの1レコード目を上書き';
5
6   //サブテーブルの末尾に行を追加する
7   var newRow = {
8     value: {
9       文字列_1: {
10          type: 'SINGLE_LINE_TEXT',
11          value: 'サブテーブルに追加する行'
12        }
13      }
14    };
15   record['Table']['value'].push(newRow);
```

サンプルの2行目の解説

1行目 `var record = event.record;`

2行目 `record['文字列_0']['value'] = 'この文字列で上書き';`

3行目 `return event;`

文字列_0はフィールドコード
フィールドコード配下のvalueに値を設定することで書き換えられる

例)

```
// フィールドコードがXXXの値を「あいうえお」にする  
record['XXX']['value'] = 'あいうえお';
```

```
// フィールドコードがXXXの値を「1000」にする  
record['XXX']['value'] = '1000';
```

```
(function() {  
  'use strict';  
  
  // レコード追加画面でフィールドの値が変更されたら  
  kintone.events.on('app.record.create.change.area', function(event) {  
    var record = event.record;  
    var area = record.area.value;  
  
    if (area === '首都圏' || area === undefined) {  
      record['XXX']['XXX'] = 0;  
    } else if (area === '関東（首都圏以外）') {  
      record['XXX']['XXX'] = 1000;  
    } else if (area === 'その他') {  
      record['XXX']['XXX'] = 2000;  
    }  
  
    return event;  
  });  
})();
```

★TRY★

日当の値（フィールドコードはdaily）を書き換えよう

正解

```
(function() {
  'use strict';

  // レコード追加画面でフィールドの値が変更されたら
  kintone.events.on('app.record.create.change.area', function(event) {
    var record = event.record;
    var area = record.area.value;

    if (area === '首都圏' || area === undefined) {
      record['daily']['value'] = 0;
    } else if (area === '関東（首都圏以外）') {
      record['daily']['value'] = 1000;
    } else if (area === 'その他') {
      record['daily']['value'] = 2000;
    }

    return event;
  });
})();
```

動作確認

- アプリに適用
- レコード追加画面を表示
- 地域を変更すると日当の値が変更されることを確認

🏠 🔍 出張申請（未完成版）

出張申請（未完成版）

出張することになったとき・出張で交通費等の立替金が発生したときに申請するアプリです。
社外にいても確認・処理ができるので、申請から決裁までのスピードアップが図れます。

キャンセル
保存

※データを保存した段階では、下書きの状態になっています。
上長に申請するときには「申請する」ボタンを押してください。

タイトル	日付 *
<input style="width: 90%;" type="text"/>	(自動入力)
申請者 *	社員番号
(自動入力)	<input style="width: 80%;" type="text"/>
	所属部署
	<input style="width: 80%;" type="text" value="-----"/> ▼
	承認者
	<input style="width: 80%;" type="text"/> 🔍
出張日	
<input style="width: 100%;" type="text" value="2016-01-26"/>	
出張先	地域
<input style="width: 90%;" type="text"/>	<input style="width: 80%;" type="text" value="-----"/> ▼
	日当
	<input style="width: 80%;" type="text" value="0"/> 円

答えのファイルの用意しておきました。

<https://bit.ly/kin-hands-on>

残課題②



カスタマイズの適用（ファイルのアップロード）が面倒くさい

解決策1

- Tipsでノウハウを知ろうをクリック
- kintoneカスタマイズTipsをクリック



The screenshot shows the homepage of the cybozu developer network. The header is blue with the logo and a search bar. The navigation menu includes 'はじめに', 'チュートリアル', 'API関連', 'Tips' (highlighted with an orange border), 'コミュニティ', 'キャンペーン', and 'ログイン'. A yellow banner for a 'リニューアルキャンペーン' (Renewal Campaign) features prizes like 'モバイルバッテリー', 'ドライバーセット', and 'レギンス', with a total of 220 winners. Below the banner, the main content area is divided into four sections: 'Hello Developers!' with introductory text, 'はじめに' (Getting Started) with a clock icon, 'チュートリアル' (Tutorial) with a mouse cursor icon, and 'API' (partially visible) with a code icon.

JavaScriptカスタマイズのデバッグをかんたんにするウラワザを見れば解決

tips

カスタマイズ ^

kintone JavaScript API (kintone内の操作)

kintone REST API

画面の見た目をカスタマイズ

アプリのレコード操作

ファイルのアップロード、ダウンロード

Cybozu CDN活用

kintone × Excel

サイボウズ製品連携(Garoon等)

モバイル

IoT

PHP

プチ情報

外部サービス連携 v

サンプル

カスタマイズ v

kintone プラグイン v

データ移行 v

kintone JavaScript API (kintone内の操作)

このセクションをフォロー v

- アプリの「親子関係」を理解しよう！ 後編
- アプリの「親子関係」を理解しよう！ 前編
- 関連レコードの参照先アプリに単レコード登録
- 8/14のメンテナンスで実装された新機能を試す
- レコード保存後イベントを使って自動採番をシンプルに実装する方法
- ワンクリックで一括承認をしてみよう
- 自動ロックアップで名入れレコードを作る
- ユーザー選択パーツに一人だけ選べるように制御してみる
- 特定条件でレコード削除をキャンセル！
- 変更された行の情報を取得して合計を変更してみよう
- テーブルデータをCSVでアップデートしてみよう！
- 回答の条件によってグループフィールドの開閉を切り替える
- テーブルデータをCSVでインポートしてみよう！
- kintone API で Promise を使ってみよう！
- Promiseを利用したモダンなアプリの全レコード取得の書き方
- 必ず選んでもらえるラジオボタン
- ソースのモジュール化とビルドの自動化について
- テーブルに連携をつける

解決策2

■ kintone上でカスタムJSやCSSを編集できるプラグイン「JSEdit for kintone」を使ってみようを見れば解決



The screenshot shows a web page from the kintone developer network. The main heading is "kintone上でカスタムJSやCSSを編集できるプラグイン「JSEdit for kintone」を使ってみよう". The article content includes:

- Greeting: "こんにちは。クローバの門屋です。以前こんな記事を書かせていただきました。"
- Introduction: "kintoneのカスタマイズでは作成したファイルをkintone や公開サーバにアップロードする必要があります。デバッグのときにいちいちこれをやるのは非常に手間がた dropboxをインストールするのはどうも、という方がいらっしゃったり、完成したファイルをkintoneにアップロードするのを忘れてしまったりするケースもありまあという声をよく聞いていました。"
- Update: "そこでついに、kintone上でJavaScriptやCSSファイルの編集、更新ができるようになりました！"
- Section: "JSEdit for kintone のダウンロード"
- Download link: "jsEdit_v1.0.zip"
- Notes: "※ソースコードは、plugin-sdk (Github) の「js-edit」内にあります。 ※不具合や改善要望などはGithubのIssueかコメントからご連絡ください。"
- Section: "JSEdit for kintone のインストール"
- Installation instruction: "kintone のシステム管理から、プラグインの設定画面を開き、「読み込む」をクリックしてプラグインを読み込みます。"
- Image: A screenshot of the kintone system management interface showing a dialog box titled "プラグインの読み込み" (Load Plugin) with a warning message: "⚠ 信頼できるプラグインファイル以外は読み込まないでください。"

おわり