

Web アプリケーション作成システム **Buddy**～開発ガイド

2020/8/5
インフォラボ

【01. 概要】

※「Buddy はじめに」を最初にお読みください。

- Buddy は、主に社内業務向けの Web アプリケーションを、開発・運用するシステムです。Buddy 自体も、Buddy で開発されたアプリケーションも、Web ブラウザ上で動作します。
- Buddy サーバーには、運用管理機能、開発機能、アプリ実行機能がひとつにパッケージされています。Buddy サーバーは利用契約ごとに一つ提供されます。
- 本文書では開発機能について解説します。開発機能は、開発の権限を持つユーザーのみが利用できます。
- モジュールや API の詳細については「**Buddy API リファレンス**」を、スクリプト（Javascript のプログラム）の書き方の注意点などは「**プログラミングガイド**」をご覧ください。

【02. Buddy と Web アプリケーション】

- Web アプリケーションは、Web ブラウザ上で動作するアプリケーションです。Buddy の運用管理機能や開発機能自体も、Buddy によって開発されたアプリも、Web アプリケーションです。
 - Web アプリケーションは、それを提供する Web サーバー（Buddy サーバー）と適宜通信しながら連携して動作します。したがって、Buddy サーバーと通信できないと利用できません。
- ※ 将来、Web ブラウザ内にアプリケーションやデータを保存して、サーバーと通信できない環境でも使えるアプリを開発できるようにする予定ですが、現在はサーバーと通信できないと Buddy 自体も開発されたアプリも利用できません。

【03. URL とログイン情報】

- Buddy のアドレス（URL）は契約時に連絡されます。
- ユーザーリストを管理し、ユーザーの権限（一般、開発、運用管理、運用管理・開

発)を設定する作業は運用管理者がおこないます。

【04. ブラウザ】

- Buddy と各アプリは、パソコンでは次のブラウザで動作します。開発にはパソコンを使用してください。

Firefox

Chrome

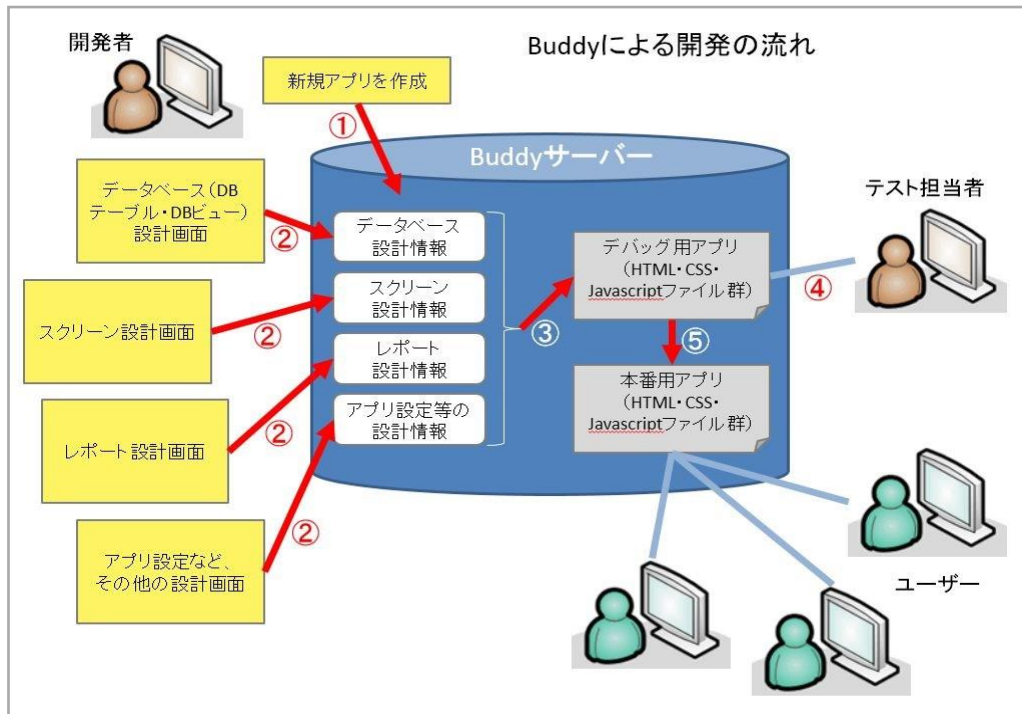
Edge

InternetExplorer11

- Buddy でのアプリ開発はパソコンのメモリを多く必要とします。4G バイトでも動きますが、8G バイト以上を推奨します。また、画面が広くないと効率よく開発が行えません。できるだけ解像度の大きなディスプレイをご利用ください。スマホやタブレットでの開発は想定していません。
- 開発されたアプリはスマホやタブレットでも動作しますが、ブラウザによって一部動かない部分がある場合があります。
- 不具合報告やご意見等をいただく際は、ブラウザとそのバージョンを書き添えていただくようお願いします。

【05. アプリ開発の概要】

- アプリケーションは利用契約のプランに応じた制限の範囲でいくつでも作成できます。(同時に稼働できるアプリ数はサーバーのメモリ容量の制約を受けます。また、ディスクの空き容量が少なくなるとアプリの作成ができなくなります。)
- アプリ作成は概ね次の手順でおこないます。



[設計]

- ① テンプレートを選び（もしくは白紙の状態）、名前を付けてアプリを作成する。
- ② データベース（DB テーブル、DB ビュー）設計、スクリーン設計、レポート設計などをおこなう。

[デバッグ用生成とテスト]

- ③ 設計情報に基づいてアプリをデバッグ環境用に生成する。
 - ④ 生成したデバッグ用アプリをテストする。
- ※ 必要に応じて②から繰り返す。

[本番用生成]

- ⑤ アプリを本番環境用に生成する（リリースと言う）。

- ・上記①と②ではアプリの設計情報が作成されるだけで、③と⑤のアプリの生成をおこなってはじめて実行できる状態のアプリができることに注意してください。
デバッグ用アプリと本番用アプリは URL が異なります。
- ・データベースもアプリの生成をおこなう際に設計情報にもとづいて、デバッグ用、本番用それぞれに作成・更新されます。
デバッグ用と本番用のデータベースは独立していますので、デバッグ環境でテストする時に入力したデータが本番環境に影響することはありません。
- ・スクリーン設計では、アプリを構成する各画面を設計します。
文字列、画像、入力欄、ボタンなど、様々なパーツ（モジュールと呼ぶ）が用意されていて、それを配置し、必要に応じてスクリプト（Javascript 言語によるプログラム）を書いて、様々な動作を実現します。

- ・レポート設計では、アプリから出力する Excel ファイルや PDF ファイルの内容を設計します。Excel レポートは、テンプレートとなる Excel ファイルを用意し、そこにデータを埋め込むことで出力します。

※生成の手順について

- ・データベースの構造は、デバッグ用生成の際にデータベース設計の内容がデバッグ用データベースに反映し、本番用生成の際にそれが本番用データベースに反映します。データベース設計を変更・保存したあと、いきなり本番用生成をおこなうと正しくデータベースに反映しません。必ずデバッグ用生成をおこない、動作確認をおこなって OK であれば本番用生成をおこなう、という手順でおこなってください。

※システムアップデートについて

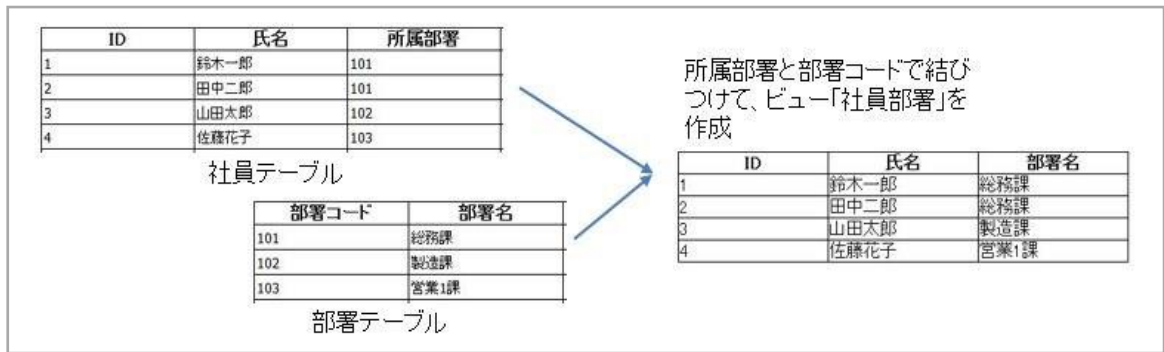
- ・改良のために Buddy のシステムをアップデートすることがあります。その際、アプリの構成要素となるもの（モジュールなど）についても機能の追加や変更がおこなわれることがあります。既存のアプリにそれを適用してアプリの挙動が変わったり不具合が生じると困るので、既存アプリには適用されません。既存アプリにアップデートを適用するには、「**既存アプリから作成**」という機能でアプリのコピーを作成する際にアップデート適用を指定します。詳しくは後述の【10. 開発】の「**既存アプリから作成**」の項をご覧ください。

【06. データベースの基本】

- ・データベースは、データを永続的に保存し取り出せる仕組みです。アプリで何らかのデータを保存する必要がある時にはデータベースを使用します。

※ Buddy では PostgreSQL というリレーショナルデータベースを使用しています。

- ・DB テーブルは「カラム」と呼ぶいくつかの項目を組み合わせたデータ（一件ずつを「レコード」と呼ぶ）がたくさん集まったものです。Excel になぞらえれば、各列がカラム、各行がレコードに相当する、と考えればよいでしょう。
- ・各カラムには名前を付けるとともに、そのカラムに格納するデータが、数値なのか、文字列なのか、といった型を与えます。型に合致しないデータを入れようとするとエラーになります。
- ・DB テーブルは実際にデータを格納できるものですが、DB ビューはそれ自体はデータを持たず、テーブルからカラムやレコードを限定したり、テーブル同士を結びつけたりして、仮想的なテーブルを作成するものです。（その他にも機能がありますが、あとで説明します。）



【07. 名前について】

- アプリ名、DB テーブル名、DB テーブルのカラム名、DB ビュー名、スクリーン名、レポート名など、アプリの設計では様々なものに名前を付ける必要があります。これらの名前には、半角英数字または「_」（アンダースコア）のみが使えます。大文字小文字を混ぜた名前を付けられますが、同じ名前かどうかをチェックする際には大文字小文字は区別されません。例えば「Test」という名前のスクリーンがすでにあれば、「test」という名前のスクリーンは作成できません。
- 多くのものについて、半角英数字の名前とは別に表示用の名前を付けられます。表示用の名前には制約はなく、日本語の名前を付けることができます。
- ユーザーIDについては、メールアドレスをそのまま使えるように、半角英数字と「_」「@」「-」「.」の記号が使えます。

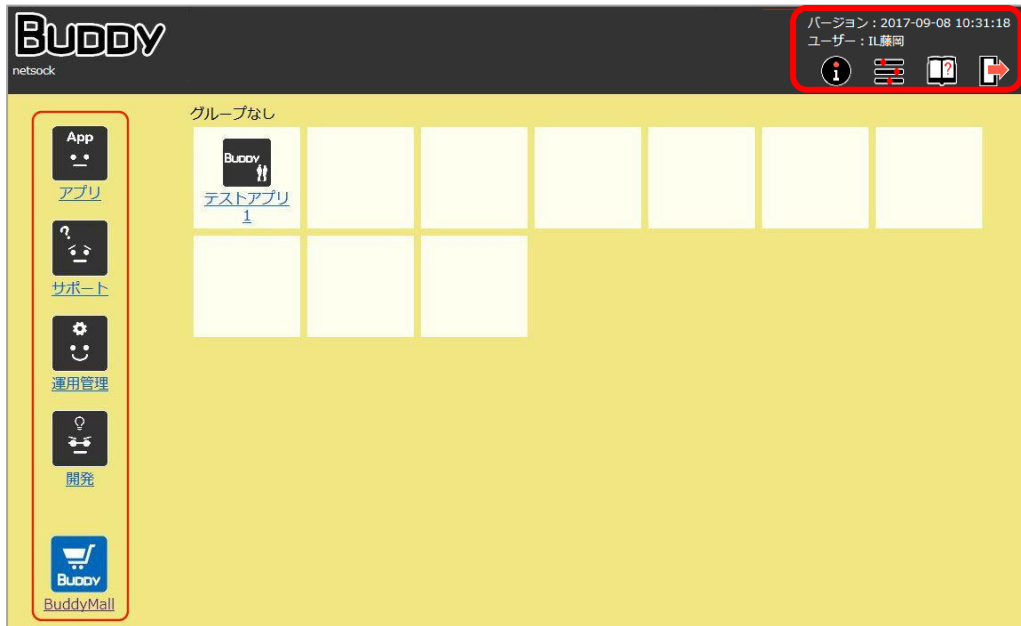
【08. ユーザーと権限について】

- Buddy の開発画面と各アプリの画面は、全てユーザー名とパスワードによるログインが必要です。

※ アプリについては、ログインを必要としないゲスト利用ができる設定が可能です。
 (【21. ユーザー権限管理】を参照)

- 原則として Buddy の開発画面と各アプリのユーザーは共通で、シングルサインオン（どれかのアプリでログイン済みであれば、他のアプリでもログイン済みの扱い）になります。特定のアプリについて、共通のユーザー管理と切り離して固有のユーザー管理をおこなうようにすることも可能です。
- 各アプリの開発や利用について、編集や閲覧をどのユーザーやグループがおこなえるか、の権限管理ができます。
- ユーザーやグループによってアプリの特定の画面に使用権限を与える（例えば、この画面は「経理部」グループのメンバーのみが利用できるというような）権限の設定ができます。

【09. トップページ】



- Buddy の画面は、大きく分けて「アプリ」「サポート」「運用管理」「開発」となり、画面左側にはそのメニューアイコンが並んでいて、クリックするとそれぞれの画面に移ります。
「BuddyMall」アイコンは、公開されているアプリを購入（無料の場合もあります）できる仕組みである「BuddyMall」の画面を開きます。
- ログインしたトップページは「アプリ」の画面になります。
実行可能なアプリのアイコンと名前がグループ分けされて一覧表示され、クリックすると実行できます。（ここから実行できるのは本番用にリリースされたアプリです）
- 画面右上の4つのアイコンは、左から「通知」「設定」「ヘルプ」「ログアウト」です。

アイコン	アイコン名	説明
	通知	メッセージ通知とお知らせを表示するダイアログを開く
	設定	自分自身のパスワードを変更できる
	ヘルプ	各種マニュアルを参照できる
	ログアウト	ログインのやり直しができる

- 「通知」アイコンをクリックすると次のダイアログが開きます。



- ・「メッセージ」と「お知らせ」のタブをクリックしてどちらを表示するか切り替えられます。
- ・「未読のみ表示」を「既読も表示」に切り替えると既読のものも表示されます。
- ・未読のものは番号が赤く表示されます。
- ・番号、日時、タイトルのみの一覧表示になっていますが、クリックすると内容を見ることができます。

【10. 開発】



- ・新規アプリの作成と既存アプリの修正ができます。

※ 通常はその Buddy サーバーの中で他のアプリと重ならない限り自由なアプリ名を付けることができますが、試用版ではアプリ名は自動的にユーザー名の後に日時の数字が付いたものになり、自由に付けることはできません。

- ・「開発中のアプリ」の一覧から既存のアプリのアイコンをクリックすると、そのアプリの開発画面が開いて修正できます。アプリ数が多くなった場合、次の機能で順序を変えたり絞り込んだりして探しやすくしてください。

機能	説明
ソート	「表示名」「アプリ名」「作成日」から選択して表示順序を変更
フィルタ	「すべて」では全部、「自分」では自分が作成したアプリのみを表示
検索	入力すると、アプリ名かアプリ表示名かタグにその文字列を含むアプリのみを表示

<白紙から作成>



中身のないアプリケーションが作成できます。

<サンプルから作成>



アプリケーションの雛形から作成できます。

作成後すぐに「生成」してもとりあえず動作するアプリができますが、これを元に修正・追加してアプリを作成していくことができます。



※「アドレス帳」「掲示板」「アンケート集計」「売上報告」「KitchenSink」などがあり、今後も様々なアプリの雛形を用意する予定です。「KitchenSink」はアプリとして実用的なものではなく、Buddyの様々なパーツの使い方のサンプルとなるものです。

<既存アプリから作成>



既存のアプリをコピーして新規アプリを作成できます。既存のアプリの一覧が表示されるので、選択すると次の画面になります。

- ・「バージョン」は通常は「(最新)」ですが、「アプリ設定」(後述 P12)で「現在のバージョンを保存」で保存した過去のバージョンがある場合はそこから選択して、過去のバージョンのコピーで作成することができます。
- ・「フレームワーク」は、アプリで使用するパーツ類(スクリーンやレポートのモジュール、登録・編集や検索などのスクリーンテンプレート、機能セットなど)のことで、改良のためにバージョンアップされることがあります。しかし、既存のアプリにそれを適用するとそのパーツに関係するスクリプトに互換性のない部分があると動かなくなる可能性があるため、既存アプリには適用されません。アプリのコピーとして新規アプリを作成する際に、「フレームワーク」を「アップデート」にすると最新バージョンが適用されます。その上で互換性のない部分のスクリプトを編集すること

で、最新バージョンのフレームワークを使用したアプリにアップデートすることができます。

- ・「新規アプリ名」は、選択した既存アプリの名前に「_copy」や「_copy2」などが付いたものになっていますが、変更できます。
- ・「アプリ作成」をクリックすると、コピーした新規アプリが作成されます。

<設計情報をアップロードして作成>



ダウンロードしておいたアプリの設計情報をアップロードすることで、同じ内容のアプリを作成します。その Buddy サーバー内の既存アプリについては上記の「**既存アプリから作成**」でコピーできますが、他の Buddy サーバーにアプリをコピーしたい場合にはこの機能を使用します。

※ 設計情報のダウンロードは、アプリ設定の画面でできます (P11 で後述)。

- ・「設計情報」の「参照」で設計情報のファイルを選択します。「名前」は作成するアプリの名前です。「作成」で、コピーした新規アプリが作成されます。

<Buddy モールから作成>



BuddyMall で購入したアプリから、同じ内容のアプリを作成します。

BuddyMall に公開されているアプリから直接作成できるのではなく、あらかじめ購入（無料のアプリもあります）が必要です。また、BuddyMall のアプリは編集不可のものもあり、編集不可のアプリは編集することができません。（データベースのテーブルへのインポートは可能です。）

- ・アプリのアイコンをクリックすると、詳細が確認できます。

Buddyモールから作成

面倒な○○○を効率化します
簡単○○○

作者 [ネットストック](#)

ウェブサイト なし

価格 無料

アプリの種類 編集可能

概要
面倒な○○○を効率化します。
いつも×××に悩んでいる方、使ってみてください。

公開日時: 2017/09/05 20:53:37
更新日時: 2017/09/07 22:22:32

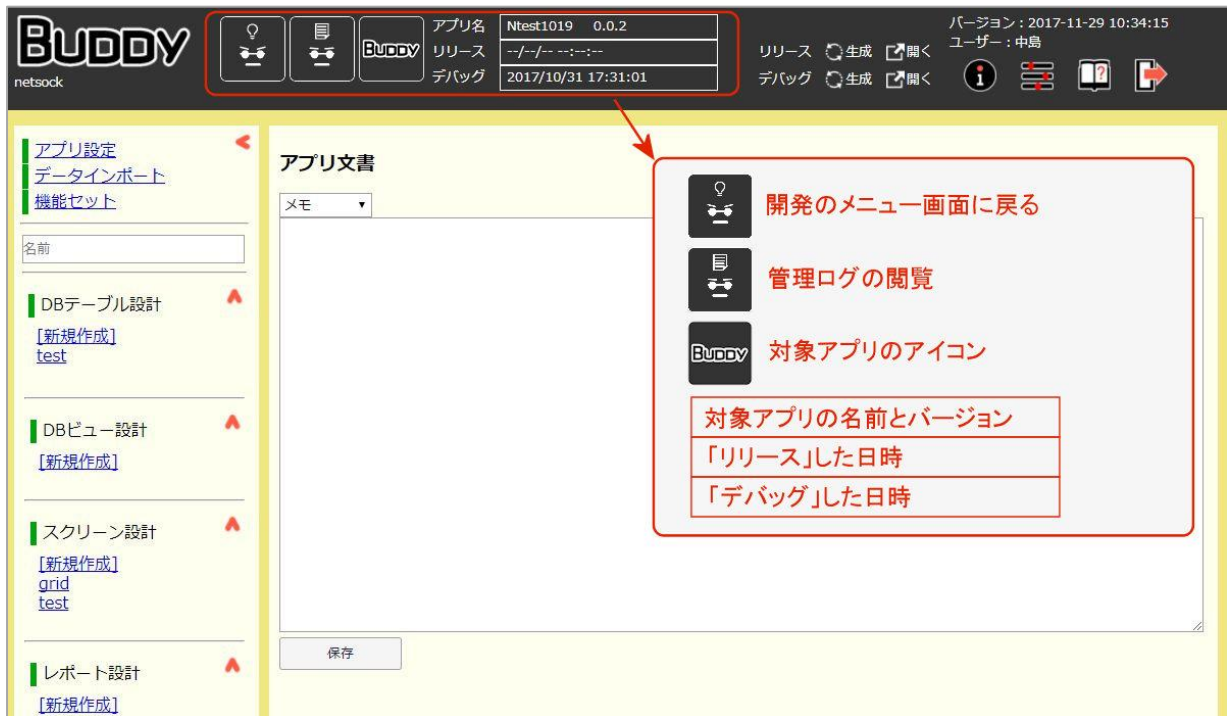
戻る 作成する

- ・「作成する」をクリックすると作成されます。

<アプリ作成時の制限>

- ・アプリを作成する際に、Buddy サーバーの空きディスク容量が 4G バイト（またはメモリが 4G バイト未満の場合はメモリ容量）よりも少ないと、作成できません。
- ・設計情報をアップロードして作成する場合は、上記の必要空きディスク容量に加えて、アップした dat ファイルのサイズの 12 倍の空きディスク容量が必要です。

【11. アプリ開発画面】



- ・上部の大きなアイコンは左から、「開発のメニュー画面に戻る」アイコン、「管理ログ閲覧」アイコン、「対象アプリのアイコン」です。
- ・その右の3行の表示は上から、現在の対象アプリの名前とバージョン、「本番用リリース」ボタンと最後に本番用リリースをおこなった日時、「デバッグ用生成」ボタンと最後にデバッグ用生成をおこなった日時です。
- ・その右の、「リリース」「 生成」をクリックすると本番用リリースがおこなえ、「 開く」をクリックするとリリースした本番用アプリを別タブで開くことができます。同様に、「デバッグ」「 生成」をクリックするとデバッグ用アプリの生成がおこなえ、「 開く」をクリックするとデバッグ用アプリを別タブで開くことができます。
- ・画面左のメニューは、右上隅にあるオレンジの「<」「>」をクリックすることで隠したり表示したりできます。メニューの中のオレンジの「^」「v」をクリックすると、そのメニューグループを隠したり表示したりができます。
- ・画面左のメニューは、次のものがあります。アプリ開発画面を開いたときは「アプリ文書」が選択された状態になっています。

メニュー	説明
アプリ設定	アプリ自体の設定をおこなう
データインポート	DB テーブルに他のアプリなどからまとめてデータをインポート

機能セット	機能セットの管理をおこなう
名前	入力すると以下の各種設定で、その文字列を含む名前のみが表示される
DB テーブル設計	データベーステーブルの設計をおこなう
DB ビュー設計	データベースビューの設計をおこなう
スクリーン設計	スクリーン（アプリの画面）の設計をおこなう
レポート設計	レポート（アプリの出力）の設計をおこなう
サーバー機能設計	アプリのサーバー側で動作するスクリプトの設計をおこなう
ユーザー権限管理	アプリの開発や利用の権限を設定する
スタイルシート管理	スタイルシート（CSS）ファイルを管理する
スクリーンテーマ編集	スクリーンに配置したモジュールの「クラス」や「部分クラス」で指定したクラス名に適用するスタイルをまとめて指定する
ファイル管理	アプリの動作に必要な各種のファイルを管理する
アプリログ一覧	アプリのログリストを閲覧する
アプリ文書	開発者や管理者向けのメモやマニュアルを記入

- ・「DB テーブル設計」「DB ビュー設計」「スクリーン設計」「レポート設計」には、「新規作成」の他に既存の DB テーブル、DB ビュー、スクリーン、レポートの名前が表示され、クリックすることで編集できます。

【12. アプリ設定】

アプリ設定

アプリ名	<input type="text" value="test0508"/>
アプリ表示名	<input type="text"/>
最初のスクリーン	(スクリーン一覧) ▼
スクリーン一覧	<input type="button" value="スクリーンの選択・順序変更"/>
WebAPI	<input type="button" value="WebAPI設定"/>
アプリタグ	<input type="text" value="#タグ"/>
アイコン	

アプリ使用量：4.57MB
開発 46.69KB
デバッグ 4.52MB
本番

現在のバージョン：0.0.1
次のバージョン： . .

保存されたバージョン：

キャッシュ：

設計情報ダウンロード
テーブルデータ： ▼

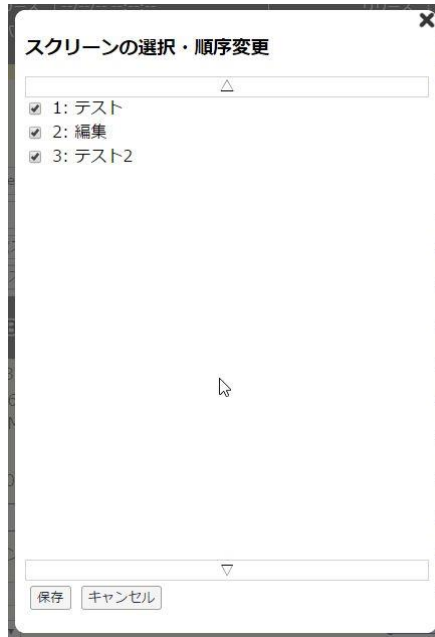
Buddyモジュール

アプリケーション削除

開発メモ

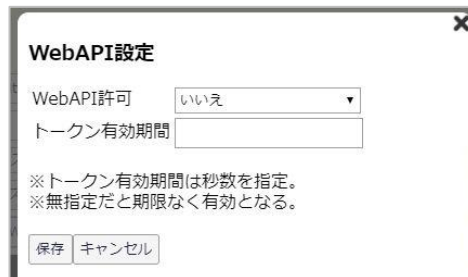
- [\[2019-05-08 11:03:55\] adminさんが生成に成功しました](#)

- ・「**アプリ名**」は新規作成時に指定した名前です、変更できません。
- ・「**アプリ表示名**」は自由に設定できます。
- ・「**最初のスクリーン**」は、アプリを開いた時にどのスクリーンを表示するかを指定します。「(スクリーン一覧)」にするとスクリーンの一覧がメニューとして表示される簡易画面となります。
「(スクリーン一覧)」を選択するとその下の「スクリーンの選択・順序変更」ボタンが有効になります。このボタンをクリックすると次のダイアログが開きます。このダイアログでチェックを入れたスクリーンがスクリーン一覧に表示されます。また、スクリーン名をドラッグすることで順序を変更できます。



- ・「**WebAPI**」は外部システムと Buddy アプリを連携させるための仕組みである **WebAPI** についての設定をおこないます。「**WebAPI 設定**」ボタンをクリックすると次のダイアログが開きます。「**WebAPI 許可**」を「はい」にすることでこのアプリで **WebAPI** が使えるようになります。「**トークン有効期間**」は **WebAPI** で認証に用いるトークンの有効期間を秒数で指定します。無指定だと期限なく有効となります。

※ **WebAPI** についてはマニュアル「**プログラミングガイド**」をご参照ください。



- ・「**アプリタグ**」は、開発中のアプリ一覧からアプリを検索するときの手がかりとなるような何らかの分類名を任意に指定します。アプリの名称と紛らわしくないように、「#」などの記号を先頭につけておくと良いでしょう。

- ・「アイコン」は、表示されているアイコンをクリックするとアイコンの一覧がダイアログで表示され、そこから選択して変更することができます。



- ・アイコン一覧の下にある「アップロード」では、「参照」で画像ファイル（JPEG または PNG）を選択すればその画像をアイコンとして使用できます。実際の表示で 64×64 ピクセル程度になりますので、正方形の適当なサイズの画像をご用意ください。

※選択したアイコンは、生成されたアプリの favicon としても使われます。アイコンを変更した場合は、ブラウザの表示を更新しないと favicon に変更が反映しない場合があるのでご注意ください。

- ・Buddy のアプリのバージョン番号は、0.0.0 のように 3 つの数字からなります。「現在のバージョン」に表示されています。
- ・「次のバージョン」には最後の数字を一つ進めたものが自動的にセットされていますが、それぞれ増やすことは可能です（減らすことはできません）。デバッグ用に生成すると「次のバージョン」で指定したバージョンになります。
- ・「保存されたバージョン」は、その下の「現在のバージョンを保存」で保存したバージョンの一覧が表示されます。
- ・「現在のバージョンを保存」は、現在の設計情報を現在のバージョン番号で保存します。保存しておくと、上記の「既存のアプリから作成」で過去のバージョンのコピーとして新規アプリを作成できます。

- ・「キャッシュ」を「クリア」にすると、アプリを生成する際に、一度生成すると変化のない部分はキャッシュしておいて生成時間を短くする機能をオフにし、再度全体を生成しなおします。通常は「**保持**」で構いません。
- ・「生成（デバッグ用）」はデバッグ用にアプリを生成します。「**デバッグ用アプリを開く**」は別タブでデバッグ用のアプリを開きます。
- ・「リリース」は本番用にリリースします。「**リリースアプリを開く**」は別タブで本番用アプリを開きます。
- ・以上のデバッグ用生成とリリースの操作は、画面上部のアイコンでおこなうこともできます。



- ・「**設計情報ダウンロード**」は、アプリの設計情報をひとまとめにしたものをダウンロードします。アプリを新規作成する時に「**設計情報をアップロードして作成**」を利用すれば、ダウンロードしたファイルを他の Buddy サーバーにアップロードしてアプリをコピーすることができます。

※ ブラウザによってはポップアップ抑止機能を解除しないと働かない場合があります。

- ・「**Buddy モール**」は BuddyMall へアプリを登録します。「**アプリ登録・更新**」をクリックすると MyBuddy の ID とパスワードの入力を求められます。入力して「OK」をクリックすると、MyBuddy への仮登録がおこなわれます。BuddyMall に登録・公開するには、MyBuddy 上での公開操作が必要です。

※ MyBuddy は、Buddy の契約に関する情報の管理コンソールです。MyBuddy のアカウントは Buddy のアカウントとは別なので注意してください。

- ・「**アプリケーション削除**」はアプリを削除します。取消はできないので慎重におこなってください。
- ・「**開発メモ**」は開発上のメモを残すためのものです。入力して「入力」をクリックすれば登録されます。その他、デバッグ用生成やリリースをおこなった場合は自動でそのメモが登録されます。

【13. データインポート】

- ・「データインポート」はDBテーブルの各テーブルに他のアプリのDBテーブルからデータをコピーしたり、そのアプリのデバッグ用と本番用のデータベース間でデータをコピーしたりできる機能です。



- ・「インポート先」では、デバッグ用と本番用のどちらのデータベースにインポートするかを選択します。
- ・アプリの一覧から、インポート元となるアプリをクリックします。



(そのアプリ自身を選ぶこともできます)

- ・①「インポート元」として、選択したアプリのデバッグ用か本番用かを選択します。

※データベースはアプリをデバッグ用に生成したりリリースしたりしてはじめて、設計情報を反映した状態になります。一度もデバッグ用生成やリリースをしていないとデータベース自体が存在しません。インポート先・インポート元を選択するときには注意してください。

- ・② 左側にインポート先のDBテーブル名のリストが表示されます。
■はインポート元のテーブルやカラムが指定されているもので、□は未指定のものです。▲はインポート元のテーブルやカラムが指定されているが、インポート元とインポート先で名前が異なるなどの相違がある状態を表します。テーブル名をクリックすると、右側にそのテーブルへのインポート設定が右側に表示されます。
- ・③ インポート設定では、インポート元のテーブルと、各カラムにどのカラムからインポートするかを設定します。
インポート元に同名のテーブルに同名のカラムがあれば、自動的にそれがセットされますが、変更することもできます。
- ・④ 画面下部ではインポート時の処理を設定します。「置換」は、「総入れ替え」「部分置換・追加」から選択します。
「総入れ替え」ではインポート先のテーブルのデータは削除され、インポートされたデータに全体が置き換えられます。
「部分置換・追加」ではプライマリーキーが同じデータがあれば更新され、なければ追加されます。
「エラー処理」は、インポート中にエラーが起きたときの処理を「エラーが起きた行を無視して続行」「全体を停止」から選択します。「全体を停止」にした場合、エラーがあるとインポート処理全体がなかったことになります。
「ユーザーファイルのコピー」は files の内容をコピーするかどうかを指定します。
- ・「インポート」 ボタンをクリックするとインポートが実行され、結果が表示されます。

【14. 機能セット】

- ・「機能セット」は、アプリの中でよく使われる機能で、DB テーブル、DB ビュー、スクリーン、レポートなどのいくつかのものを組み合わせて実現されるものに名前を付けたものです。

左のメニューの「機能セット」を選ぶと次のような画面が表示されます。



- ・「ワークフロー」は、何らかのデータ（例えば稟議書や旅費申請など）について、順次決裁処理をおこなう仕組みです。
- ・「コメント」は、アプリのユーザーが画面に表示されている内容に何らかのコメントを記入する機能です。

※ 機能セットは今後追加される可能性があります。

【14.1. ワークフロー】

- ・「ワークフロー」は、何らかのデータ（例えば稟議書や旅費申請など）について、順次決裁処理をおこなう仕組みです。その作成手順は次のようになります。

①	対象データの DB テーブル作成 →決裁対象となるデータの DB テーブルを作成する 単独のテーブルでも、それに明細テーブルが一対多で対応する形でも OK
②	ワークフロー追加 →関連する DB テーブル、DB ビュー、スクリーン、サーバー機能がまとめて作成される
③	担当者マスターに登録 →申請者と決裁者を登録する
④	経路マスターの登録 →決裁経路（どの決裁者から順に決裁するか）を設定する
⑤	申請と決裁のスクリーンを追加 →「追加・編集」で「申請/決裁画面」または「明細付き申請/決裁画面」を追加する

<①対象データの DB テーブル作成>

- ・ 決裁対象のデータはどのような内容でも構いません。一つの対象データに複数の明細が対応する場合は、明細テーブルを作成します。
- ・ 以下の説明の例として次のような DB テーブルを用意します。明細テーブル (kounyuusinseimeisai) の「申請 ID」が kounyuusinsei テーブルの「ID」と同じものが対応することになります。

	カラム名	表示名	型	型の詳細
	ID	ID	数値	自動連
	sinseibi	申請日	日付	日付
	sinseisha	申請者	文字列	フリー

	カラム名	表示名	型	型の詳細
	ID	ID	数値	自動連
	sinseiID	申請ID	数値	整数
	hinmei	品名	文字列	フリー
	suuryou	数量	数値	整数
	kingaku	金額	数値	整数



プライマリキー(後述 P29)

<②ワークフロー追加>

- ・ 「機能セット」の画面で「ワークフロー」アイコンの右の「作成」ボタンをクリックすると、次の画面になります。

機能セット

ワークフロー 追加

- ・ 「追加」ボタンをクリックすると、関連する DB テーブル、DB ビュー、スクリーン、サーバー機能が作成され、次の画面になります。

機能セット



ワークフロー

- ・ この画面で「削除」ボタンをクリックすると、関連する DB テーブル、DB ビュー、スクリーン、サーバー機能がまとめて削除されます。
- ・ ここで一度アプリを生成し、DB テーブルにデータが入力できる状態にします。

<③担当者マスターに登録>

- DB テーブルに追加された「担当者マスター」をクリックします。

DBテーブル設計 担当者マスター
(_assembly_workflow_tantou)

カラム名	表示名	型	型の詳細	
ID	ID	数値	自動連	✕
myoji	苗字	文字列	フリー	✕
namae	名前	文字列	フリー	✕
myoji_kana	苗字かな	文字列	フリー	✕
namae_kana	名前かな	文字列	フリー	✕
account	アカウントID	文字列	フリー	✕
(new)				

データの確認

デバッグ

↑インポート ↓エクスポート 🕒データ履歴 📄バージョン
✕データ削除

<< < 1 /0 > >>

ID	苗字	名前	苗字かな	名前かな	アカウントID	
						✓

- 「担当者マスター」には、申請者と決裁者を登録しておく必要があります。インポートまたは直接入力に登録してください。
- 自動連番の ID 以外では、myoji（苗字）と account（アカウント ID）が必須です。アカウント ID はログイン時のユーザー ID です。（つまり、Buddy のユーザーでない人を申請者や決裁者にはできません。）
- 必要に応じて、例えば部署や電話番号などのカラムを追加しても構いません。（生成されたスクリーンなどに手を加えてそれらのカラムを生かすようにするのはユーザーに委ねられます。）

<④経路マスターの登録>

- 次に DB テーブルの「経路マスター」をクリックします。

DBテーブル設計 経路マスター
(_assembly_workflow_keiro)

カラム名	表示名	型	型の詳細
ID	ID	数値	数値
next_id	次のID	数値	数値
prev_id	前のID	数値	数値
tantou_id	担当者ID	数値	整数

経路編集

デバッグ

保存

経路

1	1 中島靖		
2	2 兵頭尚志	3 中島靖	

- ・「**経路マスター**」では、決裁の経路を設定します。
- ・このテーブルは特殊なテーブルで、通常は「**データの確認**」と表示されるところが、「**経路編集**」となっています。③で設定した担当者の中からプルダウンリストで選択して設定します。設定が終わったら「**保存**」ボタンをクリックするのを忘れないようにして下さい。
- ・左端の番号が**経路 ID** です。上の例では、**経路 ID1** が「中島靖」さんが決裁したら終了、という経路。**経路 ID2** が、まず「兵頭尚志」さんが決裁したら次に「中島靖」さんが決裁して終了、という経路です。

<⑤申請と決裁のスクリーンを追加>

- ・この状態ではまだワークフローの申請と決裁を行うスクリーンが作成されていません。「**機能セット**」でワークフローの「**追加・編集**」ボタンをクリックすると次の画面になります。



- ・申請データが単独のテーブルに格納される場合は「**申請 / 決裁画面**」を、申請データに明細があって一対多の二つのテーブルに格納される場合は「**明細付き申請 / 決裁画面**」をクリックします。両者の違いは明細部分があるかどうかなので、「**明細付き申請 / 決裁画面**」の画面で説明します。

ワークフロー 編集

テンプレート: 明細付き申請画面

スクリーン名:

表示名:

決済画面表示名:

① テーブル: ▼

IDカラム: ▼

申請経路:

決済通知: ▼

名前	表示名	モジュール	カラム		
shinseiID	申請ID	データインプット ▼	申請ID ▼	+	-
shinseibi	申請日	データインプット ▼	申請日 ▼	+	-

② 明細テーブル: ▼

戻る 追加

- この画面で指定した内容で申請画面と決裁画面のスクリーンが作成されます。申請画面は指定した**スクリーン名**（上の例では kounyuusinsei）に、決裁画面はそれに「_kessai」が付加されたスクリーン名になります。「**表示名**」「**決裁画面表示名**」はそれぞれ適宜変更して構いません。
- ④「**テーブル**」は対象データのメインとなる DB テーブルを選択します。
- ④「**ID カラム**」はそのレコードを一意に識別するカラムを選択します。
- ④「**申請経路**」は④で設定した**経路 ID**のいずれかを指定します。（空白にしておいてあとで各スクリーンのスクリプトを編集することで指定することもできます。）
- ④「**決裁通知**」を「する」にすると、決裁の必要なデータが申請されると決裁者に通知メッセージが送信されます。その下の入力欄の指定は、最初は ID 以外の全てのカラムが表示されていますので、不要なものがあれば「-」ボタンで削除してください。
- ⑤「**明細テーブル**」で対象データの明細の DB テーブルを選択します。

ワークフロー 編集

テンプレート: 明細付き申請画面

スクリーン名:

表示名:

決裁画面表示名:

テーブル:

IDカラム:

申請経路:

決裁通知:

名前	表示名	モジュール	カラム		
<input type="text" value="sinseibi"/>	<input type="text" value="申請日"/>	<input type="text" value="データインプット"/>	<input type="text" value="申請日"/>	<input type="text" value="+"/>	<input type="text" value="-"/>
<input type="text" value="sinseisha"/>	<input type="text" value="申請者"/>	<input type="text" value="データインプット"/>	<input type="text" value="申請者"/>	<input type="text" value="+"/>	<input type="text" value="-"/>

⑤

明細1:

カラム:

名前	表示名	モジュール	カラム		
<input type="text" value="hinmei"/>	<input type="text" value="品名"/>	<input type="text" value="データインプット"/>	<input type="text" value="品名"/>	<input type="text" value="+"/>	<input type="text" value="-"/>
<input type="text" value="suuryou"/>	<input type="text" value="数量"/>	<input type="text" value="データインプット"/>	<input type="text" value="数量"/>	<input type="text" value="+"/>	<input type="text" value="-"/>
<input type="text" value="kingaku"/>	<input type="text" value="金額"/>	<input type="text" value="データインプット"/>	<input type="text" value="金額"/>	<input type="text" value="+"/>	<input type="text" value="-"/>

集計表示:

名前	表示名	カラム	種類		
<input type="text" value="kingakugoukei"/>	<input type="text" value="金額合計"/>	<input type="text" value="金額"/>	<input type="text" value="合計"/>	<input type="text" value="+"/>	<input type="text" value="-"/>

IDカラム:

明細テーブル:

- ・入力欄のカラム一覧が表示されますので、不要なものを「-」ボタンで削除します。
- ・㊸「IDカラム」はメインテーブルの ID カラムと対応する値を格納するカラムを選択します。ここに選択したカラムは入力欄からは削除してください。
- ・㊸「集計表示」は、合計などを表示したいカラムがあれば「+」ボタンをクリックして指定します。
- ・㊸「明細テーブル」を追加する場合は、DB を選択すれば「明細 2」が表示されます。
- ・「追加」ボタンをクリックすれば、申請画面と決裁画面のスクリーンが作成されます。

※ 「申請経路」を空白のまま申請画面と決裁画面のスクリーンを作成した場合は、それぞれのスクリプトの中で「// 経路 ID を選択する処理をここに記載する」というコメントがある箇所を適宜修正してください。

※ ワークフローを追加したときに作成されるスクリーンに、申請画面や決裁画面への移動ボタンがありますが、その時点ではまだ申請・決裁画面がないため、クリックしても機能しないボタンになっています。申請・決裁画面を作成後は、スクリーンのスクリプトの「//申請画面を作成した後、移動スクリプトを記述する」のようにコメントされている箇所を編集してボタンが機能するようにしてください。

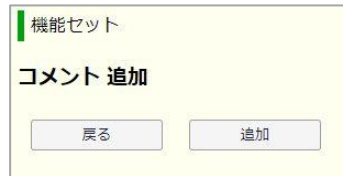
【14.2. コメント】

- ・「コメント」は、アプリのユーザーが画面に表示されている内容に何らかのコメントを記入する機能です。
- ・機能セット「コメント」を利用してスクリーンにコメント機能を付加するには、次の手順でおこないます。

①	コメント追加 →関連する DB テーブル、スクリーン、サーバー機能がまとめて作成される
②	対象のスクリーン作成 →コメント機能を付加するスクリーンを作成する
③	コメント付画面追加 →「追加・編集」で「コメント付画面」を追加する このとき、②で用意したスクリーンを対象に指定する

<①コメント追加>

- ・「機能セット」の画面で「コメント」アイコンの右の「作成」ボタンをクリックすると、次の画面になります。



- ・「追加」ボタンをクリックすると、関連する DB テーブル、DB ビュー、スクリーン、サーバー機能が作成され、次の画面になります。



- ・この画面で「削除」ボタンをクリックすると、関連する DB テーブル、DB ビュー、スクリーン、サーバー機能がまとめて削除されます。

<②対象のスクリーン作成>

- ・コメント機能を付加したいスクリーンを用意します。
どんなスクリーンでも構いません。

<③コメント付画面追加>

- ・「機能セット」のコメントで「追加・編集」ボタンをクリックすると次の画面になります。



- ・「コメント付画面」をクリックすると次の画面になります。

機能セット

コメント 編集

テンプレート: コメント付画面

組み込むスクリーン:

スクリーン名:

表示名:

コメント配置:

クエリ (id=XX) で指定されたデータ別にコメントする

投稿できるユーザー

ユーザーID: (カンマ区切り)

グループ名: (カンマ区切り)

オプション

投稿通知:

コメント閲覧:

- ・「組み込むスクリーン」は、②で用意したコメント機能を付加したいスクリーンを選択します。
- ・「スクリーン名」「表示名」は、これから作成するスクリーンの名前です。「組み込むスクリーン」で選択したスクリーンの名前を元に、「_withComment」「(コメント付)」が付加された名前が自動的に入りますが、変更できます。
- ・「コメント配置」は、組み込むスクリーンに対してコメント部分をどこに配置するかを指定します。「右 (モバイル対応)」ではコメント部分が右側に配置されますが、画面幅がスクリーン属性の「モバイル対応幅」より小さい時はコメント部分が下に配置されるように自動調整されます。「右」「下 (左寄せ)」「下 (中央)」「下 (右寄せ)」ではそれぞれ指定した位置にコメント部分が配置されます。
- ・「クエリ (id=XX) で指定されたデータ別にコメントする」は、組み込むスクリーンが「閲覧」のスクリーンテンプレート (後述) などで作成したもので、URL の末尾につく「?id=2」のようなクエリによって表示するデータを指定するようになっている場合にそのデータ別にコメントしたい時にチェックを入れます。
- ・「投稿できるユーザー」は、コメントを投稿できるユーザーを指定します。「ユーザーID」にはユーザーのログインIDをカンマ区切りで、「グループ名」にはユーザーグループ名 (ユーザーグループは運用管理のユーザー管理で作成できます) をカンマ区切りで入力します。両方指定することもできますが、その場合は、どちらかに該当すればコメント投稿ができます。

※ユーザーがどのグループに属するかは、ログイン時に判断されます。ログイン中のユーザーをグループのメンバーに追加・削除した場合、ログアウトしてログインし直さないと反映しない点に注意が必要です。

- ・「**クエリ (id=XX) で指定されたデータ別にコメントする**」のチェックが入っていると、「**投稿できるユーザー**」に次の入力欄が追加され、クエリで指定されたデータ id に基づいてデータベースからユーザーIDを取得することで投稿できるユーザーを決めることができます。

※投稿できるユーザーをクエリ (id=XX) を元にデータベースから取得する場合は以下を指定	
テーブル:	<input type="text"/>
クエリカラム:	<input type="text"/> (クエリidと対応)
ユーザーIDカラム:	<input type="text"/>

※「**テーブル**」は、対象の DB テーブルまたは DB ビューを選択します。選択したテーブルまたはビューには、クエリで指定された id と対応するカラムと、ユーザーIDの入ったカラムがなければなりません。それぞれを「**クエリカラム**」と「**ユーザーIDカラム**」で選択します。

- ・「**オプション**」の「**投稿通知**」を「**する**」にすると、コメント投稿があった時にコメント投稿できるユーザーに通知メッセージが送られます (アプリのヘッダの「**通知**」アイコンから確認できます)。「**コメント閲覧**」を「**全員**」にすると投稿されたコメントは誰でも閲覧できます。「**投稿できる人のみ**」にすると投稿できるユーザーのみが閲覧できます。
- ・「**追加**」ボタンをクリックすると、指定した内容でコメント機能を付加されたスクリーンが追加されます。このとき、「**組み込むスクリーン**」で指定した元のスクリーン自体には何も変更はないことに注意してください。元のスクリーンを遷移先に指定しているスクリーンがあった場合は、遷移先のスクリーン名を適宜変更してください。
- ・コメント機能が付加されたスクリーンでは、次のようなコメント部分が表示されま



- ・①のアイコンをクリックすると、コメント内容の入力欄と「コメントする」ボタンが現れて新しくコメントすることができます。
- ・②の「コメントを検索」欄に入力して Enter キーを押すと、表示されるコメントが入力した文字列を含むものに絞り込まれます。「クリア」をクリックすると全件の表示に戻ります。
- ・コメントの件数が 10 件以上あるときは③の「先頭へ」「前の 10 件」「次の 10 件」「末尾へ」のボタンで画面を切り替えて表示します。

【15. DB テーブル設計】

【15.01. DB テーブル新規作成】



- ・「白紙のテーブル」「CSV から作成」のいずれかをクリックします。
- ・「白紙のテーブル」の場合は空のテーブルの設計画面になります。

- ・「CSV から作成」は、先頭行に項目名が書かれている CSV を元に、それをインポートできるようなテーブルを作成する機能です。CSV ファイルの文字コードは、シフト JIS と UTF8 によるものが受け入れ可能です。

【15.02. DB テーブル新規作成 ～ CSV】



- ・「CSV から作成」を選択すると、CSV ファイルをアップロード・選択する画面になります。
- ・「CSV ファイルを選択してください」の下の領域（files/import）には、現在アップロードされている CSV ファイルの一覧が表示されます。
アップロードできるファイルは、「csv」「xlsx」「xls」です。
- ・「ファイルの種類」は、「すべて」と「CSV」から選択できます。
- ・新しく CSV ファイルをアップロードするには、CSV ファイル一覧の領域へ Windows のエクスプローラーから目的のファイルをドラッグアンドドロップします。
ブラウザのバージョンなど環境によってはドラッグアンドドロップでのアップロードができない場合もあります。その場合は「アップロード」の下の「参照」をクリックして開くファイル選択画面でファイルを選択してください。
- ・右上の「名前変更」で選択したファイルの名前を変更できます。
- ・「フォルダ作成」で新しくフォルダを作成できます。
フォルダのアイコンをクリックするとそのフォルダ内に入り、そこにファイルをアップロードすることができます。多くのファイルを扱うので分類・整理したいという時に使用してください。フォルダアイコン右上の「×」をクリックするとフォルダがその内容とともに削除されます。取消はできませんので、慎重におこなってください。
- ・一度アップロードしたファイルは、削除しない限り残り、何度でも利用することがで

きます。削除するには、ファイルアイコンの右上の「×」をクリックします。また、ファイルアイコンの右下の「↓」をクリックするとダウンロードすることもできます。

- DB テーブル作成に使用したい CSV ファイルをクリックして選択（背景がグレーになります）し、「次へ」をクリックします。
- CSV の先頭部分を示した表が表示されます。

DBテーブル作成

 CSV

テンプレート : CSV
テーブル名 :

カラム名を入力し、カラムの型を選択してください。
カラム名の上のアイコンをクリックしてプライマリキー（データを一意に特定する値）を指定します。

自動連番のIDカラムをプライマリキーとして追加する

				
ID	todoufuken	gurupu	seibetsu	nenrei
文字列	文字列	文字列	文字列	文字列
フリー	フリー	フリー	フリー	フリー
ID	都道府県	グループ	性別	年齢
1	東京都	A	男	34
2	東京都	B	男	44
3	東京都	B	男	42
4	大阪府	A	女	32
5	大阪府	A	女	35
6	京都府	A	女	45
7	京都府	B	男	35
8	京都府	C	男	35
9	京都府	C	男	25

エンコード :

- 「テーブル名」は CSV ファイル名からセットされていますが、適宜変更してください。テーブル名には半角英数字と「_（アンダースコア）」しか使えません。日本語のファイル名の場合、テーブル名はローマ字化してセットされ、元のファイル名は表示名にセットされます。
- 表の先頭行は、作成するテーブルの各カラムについて、「プライマリキー」「カラム名」「型」「型の詳細」を入力する欄です。
- 「プライマリキー」はデータを一意に特定する値です。原則としてテーブルにはプライマリキーを指定します。鍵のアイコンをクリックしてオレンジになっているカラムがプライマリキーです。複数のカラムをプライマリキーとして、その組み合わせによってデータを一意に特定することも可能です。「ID」という名前の項目があると自動的にプライマリキーにセットされますが、解除することもできます。また、「ID」という名前の項目がない場合は「自動連番の ID カラムをプライマリキーとして追加する」のチ

チェックを入れることができ、その場合はプライマリキーを指定する必要はありません。

- ・「**カラム名**」は CSV の先頭行からセットされています。日本語の場合はローマ字化してセットされます。変更可能ですので、適宜修正してください。
- ・カラムの「**型**」と「**型の詳細**」は、最初は「文字列」「フリー」になっていますが、その右の「▼」をクリックして選択できます。
- ・「**エンコード**」は Shift_JIS と UTF-8 が選べます。内容が文字化けして表示されている場合は、切り替えてみてください。
- ・テーブル名と各カラム名と型を指定したら、「**作成**」をクリックすると、テーブルが作成され、そのテーブルの設計画面になります。指定した内容は設計画面で変更することも可能です。

【15.03. DB テーブル設計】

DBテーブル設計 担当者マスター
(_assembly_workflow_tantou)
保存 モデル履歴 削除

(A)

カラム名	表示名	型	型の詳細
ID	ID	数値	自動連番
myoji	苗字	文字列	フリー
namae	名前	文字列	フリー
myoji_kana	苗字かな	文字列	フリー
namae_kana	名前かな	文字列	フリー
account	アカウントID	文字列	フリー
(new)			

(C)

テーブル属性	
属性	値
テーブル名	_assembly_workflow_tantou
表示名	担当者マスター
テーブルタイプ	FLOW
履歴管理	いいえ
バージョン管理	いいえ
権限管理	{...}
メモ	

カラム属性
カラムを選択してください

名前	表示名	重複可	式
(new)			

テーブルレール

名前	表示名	式
(new)		

テーブルトリガ

名前	表示名	いつ	DB操作	カラム	レベル	式
(new)						

(B)

データの確認

デバッグ

インポート エクスポート データ履歴 バージョン データ削除

<< < 1 / 5 > >>

ID	苗字	名前	苗字かな	名前かな	アカウントID
1	インフォ	一郎	いんぷお	いちろう	test001
2	インフォ	花子	いんぷお	はなこ	test002
3	インフォ	次郎	いんぷお	じろう	test003
4	インフォ	三郎	いんぷお	さぶろう	test004
5	インフォ	隆夫	NULL	NULL	infolab

- ・DB テーブル設計画面には、3つの部分があります。左側上部には、そのテーブルの「**カラム一覧**」(A)が表示され、カラムの追加・編集ができます。その下には「**データの確認**」領域(B)があり、デバッグ用や本番用のデータベースの当該テーブルに入っているデータ内容を確認・修正できます。

右側には、テーブル属性、カラム属性、テーブルインデックス、テーブルルール、テーブルトリガ、を設定する表 (C) があります。

- ・この画面の「データの確認」以外の部分で何らかの入力・修正をおこなった場合は、必ず「保存」して下さい。（下記の「メニューアイコン」を参照してください）
保存せずにブラウザを閉じたりリロードしたり他のアプリの開発画面に移ったりすると、変更内容は失われます。
- ・この画面で設計したテーブルの構造等は、アプリをデバッグ用に生成したときや本番用にリリースしたときに、初めて実際のデータベースに反映します。
従って、「データの確認」に表示されるテーブルの構造は、現在設計中のものとは異なる場合があることに注意してください。

<メニューアイコン>

- ・また、右上には次のメニュー (D) があります。

メニュー名	説明
保存	現在の DB テーブルと DB ビューの設計情報を保存する
モデル履歴	設計変更の履歴を閲覧し、未適用のものは取り消せる
削除	この DB テーブルを削除する

- ・「モデル履歴」をクリックすると次のような画面になります。「適用」が「済」はアプリをデバッグ用に生成することで実際のデータベースに反映したことを、「未」はそれがまだであることを表します。この画面では適用済みの履歴は表示されません。先頭の○にチェックを入れて「選択した時点まで戻す」をクリックすると、未適用の編集履歴を取り消すことができます。DB テーブル設計での不適切な指定のためにアプリの生成に失敗した時は、原因となった編集操作を取り消してください。その後正しい指定をしても、失敗の原因となる編集操作が履歴に残ったままだと、やはり生成に失敗します。

モデル編集履歴					
適用	更新番号	更新者	更新日時	変更点	
<input type="radio"/>	済	3	admin	2016-04-01 17:50:12	
<input type="radio"/>	未	4	admin	2016-04-01 17:51:38	テーブル: 社員(shain), カラム: 性別(sex), 選択肢の追加: '男'
<input checked="" type="radio"/>	未	5	admin	2016-04-01 17:51:45	テーブル: 社員(shain), カラム: 性別(sex), 選択肢の追加: '女'

キャンセル 選択した時点まで戻す

<カラム一覧>

- ・カラム一覧 (A) には次の項目があります。

	カラム名	表示名	型	型の詳細	
	ID		文字列	フリー	×
	name	名前	文字列	フリー	×
	busho	部署	文字列	フリー	×
	sex	性別	文字列	フリー	×
	(new)				

項目	説明
端の小さな枠	ピンアイコンがあるとプライマリーキーであることを示す → そのテーブルのレコードを一意に識別するキーとなるもの
カラム名	カラムの名前（半角英数字と「_」でなければならない）
表示名	わかりやすい日本語表示の名前
型	「数値」「真偽値」「文字列」「日付・時間」「ファイル」から、どの型のデータを格納するかを選択する
型の詳細	それぞれの型に合わせて詳細を選択する（数値の場合…「自動連番」「整数」「小数点」「金額」）
右端の×	そのカラムを削除する

※「型」に「ファイル」を指定した場合、そのカラムにファイルの内容そのもの（バイナリーデータ）を格納するのではなく、アップロードしてサーバー上の所定の場所に保存したファイルのファイル名をカラムの値とすることになります。そのために「データインプット」というモジュールが用意されています。（後述）

<データの確認>

- ・「データの確認」では、デバッグか本番かを選択すると、そのデータベースの当該テーブルの内容が表示されます。一度もアプリのデバッグ用生成や本番用リリースをしていない場合は、データベースが存在しないので、データは表示されません。データ内容は直接編集することができます。右端の「×」でそのレコードを削除できます。下端の行に入力して右端のチェックマークでレコードを追加できます。
- ・「データの確認」領域 (ⓑ) には、当該テーブルが生成されている時には次のメニューがあります。

ID	苗字	名前	苗字かな	名前かな	アカウントID	
1	インフォ	一郎	いんぷお	いちろう	test001	×
2	インフォ	花子	いんぷお	はなこ	test002	×
3	インフォ	次郎	いんぷお	じろう	test003	×
4	インフォ	三郎	いんぷお	さぶろう	test004	×
5	インフォ	陸夫	NULL	NULL	infolab	×
						✓

メニュー	説明
インポート	デバッグ用や本番用のデータベースの当該テーブルに CSV データやエクセルデータを投入する
エクスポート	デバッグ用や本番用のデータベースの当該テーブルの内容をファイルに出力する
データ履歴	履歴管理しているテーブルでデータ変更の履歴を確認できる
バージョン	バージョン管理しているテーブルでデータのバージョンを閲覧できる
データ削除	デバッグ用や本番用のデータベースの当該テーブルの内容を削除する (DB そのものは削除されない)

※ 「データの確認」で、カラム属性を「配列」にしたカラムに入力するときは、「{ 値 , ... }」と、カンマ区切りで値を並べて、「{」と「}」で囲みます。

<テーブル属性>

- ・「テーブル属性」(◎) は、このテーブル全体についての属性で、次の項目があります。

属性	値
テーブル名	shain
表示名	社員
テーブルタイプ	FLOW
履歴管理	いいえ
バージョン管理	いいえ
権限管理	{...}
メモ	

項目	説明
テーブル名	作成時に付けた半角英数字の名前
表示名	わかりやすい日本語表示の名前
テーブルタイプ	「FLOW」…追加されていくタイプ、「STOCK」…書き換えていくタイプ (今のところ機能上の違いはなし)
履歴管理	「はい」…データの変更履歴が内部的に保存される
バージョン管理	「はい」…コード化データの履歴問題(※)に対応するためのバージョン管理をおこなう
権限管理	権限設定が有効な状態では「{...}」と表示され、クリックすると権限設定がおこなえる (※後述)
メモ	覚え書きに使用

※ コード化データの履歴問題：コード化されたデータが変更されたとき、単純に変更すると過去のデータも新データで表示されてしまい、別コードとすると集計等の処理で別扱いになってしまう問題。これに対処するためのバージョン管理の仕組みを用意しています。

- ・「テーブル属性」の「権限管理」の値の部分をクリックすると、次の画面で権限設定が

できます。(※試用版では権限管理は使用できません。)

テーブル権限管理

権限管理をする

	許可ユーザー	拒否ユーザー	許可グループ	拒否グループ	優先
デバッグ用編集			all		許可 ▾
デバッグ用閲覧			all		許可 ▾
編集			all		許可 ▾
閲覧			all		許可 ▾

OK キャンセル

「権限管理をする」のチェックを入れると、その下の表で設定した内容にしたがって権限がチェックされるようになります。表で権限の内容を設定していても、「権限管理をする」のチェックを入れない限り権限チェックはされないの注意してください。

「デバッグ用編集」「デバッグ用閲覧」ではデバッグ用データベースでのこのテーブルの編集と閲覧の権限を設定します。「編集」「閲覧」では本番用データベースの権限を設定します。「許可ユーザー」「拒否ユーザー」の設定に該当するとそれによって許可・拒否が決まります。そうでない場合「許可グループ」「拒否グループ」の設定で決まります。

許可と拒否の両方に該当した場合は「優先」によってどちらになるかが決まります。

デフォルトでは上の図のように「許可グループ」が「all」（全ユーザーが所属するグループ）になっているので、誰でも編集・閲覧ともに可能です。

<カラム属性>

- ・「カラム属性」は、カラム一覧で選択しているカラムの属性で、次の項目があります。

カラム属性	
属性	値
デフォルト値	
選択肢	男 × 女 × (new)
範囲	min max interval
必須	いいえ ▾
読込専用	いいえ ▾
配列	いいえ ▾
複数選択可	いいえ ▾
全文検索対象 (日本語)	いいえ ▾
全文検索対象 (英語)	いいえ ▾
ルートディレクトリ	
メモ	

項目	説明
デフォルト値	レコードを追加する際に値を指定しなかった場合の値
選択肢	値をいくつかの選択肢のどれかに限定する
範囲	値を「開始後」「終了値」「刻み値」で指定した値のどれかに限定する
必須	省略できない項目を指定する
読込専用	レコードを追加する際に指定した値を後から変更できないようにする
配列	複数の値を格納する（{値, …}と、カンマ区切りで値を並べ、「{」と「}」で囲む）（※後述）
複数選択可	選択肢を指定した場合に、複数選択可を設定する
検索対象	検索の対象とするカラムを指定する
全文検索対象(日本語)	全文検索(日本語)の対象とするカラムを指定する（※後述）
全文検索対象(英語)	全文検索(英語)の対象とするカラムを指定する（※後述）
ルートディレクトリ	型が「ファイル」のカラムで、添付ファイルを格納するディレクトリ名を指定する
メモ	覚え書きに使用

※「デフォルト値」の欄をクリックするとダイアログが開きます。



「設定する」を「はい」にして「設定値」に入力すると（何も入力しないと空文字列として）デフォルト値が設定されます。「設定する」が「いいえ」だとデフォルト値は設定されないため、レコード追加時に値を指定しないと NULL になります。

日時、日付、時刻型のカラムに対しては、レコード追加をおこなった日時をセットする指定として「CURRENT_TIMESTAMP」「CURRENT_DATE」「CURRENT_TIME」を指定できます。

※「配列」を「はい」にしたカラムには複数の値を格納できます。ただし、後述する「検索」「単票」「登録・編集」などのスクリーンテンプレートやデータ入力フォームは、今のところ配列カラムには対応していません。

※「全文検索対象」とは、そのカラムについての全文検索を高速にするために内容を単語に分解した索引を作る機能ですが、今のところ機能しません。「全文検索対象」の指定はせず「いいえ」のままにしておいてください。

<テーブルインデックス/テーブルルール/テーブルトリガ>

テーブルインデックス						
名前	表示名	重複可	式			
(new)						

テーブルルール			
名前	表示名	式	
(new)			

テーブルトリガ						
名前	表示名	いつ	DB操作	カラム	レベル	式
(new)						

- ・「**テーブルインデックス**」は、検索を高速にしたいカラムや式についてインデックスを作成します。「**名前**」はこのアプリのデータベース内で重複しないように付ける必要があります。またテーブル名と同じ名前も許されません。例えば、**member** というテーブルの **age** というカラムについてインデックスを作る場合であれば、**member_age_index** というような名前にすることをお勧めします。「**表示名**」は自由に付けられます。「**重複可**」は「はい」にすると対象のカラムや式の値に重複が許されます。「いいえ」にすると重複は許されなくなります。「**式**」は対象のカラム名または PostgreSQL の式を指定します。
- ・「**テーブルルール**」は、そのテーブルの各レコードが満たすべきルールを作成します。「**名前**」はインデックスの場合と同様に重複しないような名前を、例えば **member_age_rule** のように付けます。「**表示名**」は自由に付けられます。「**式**」は PostgreSQL の式を指定し、この式が真になるデータしか受け入れられないようになります。例えば「**age < 100**」という式を指定すると、**age** カラムに 100 以上の値は入れられなくなります。
- ・「**テーブルトリガ**」は、そのテーブルの内容が変化（挿入、更新、削除）した時にそれをトリガ（引き金）としてイベントを発生する機能を設定します。

項目	説明
名前	テーブルトリガを識別する半角英数字の名前
表示名	好きな名前
いつ	「実行前」「実行後」で、内容の変化が実行される前か後かを指定する
DB 操作	「INSERT (挿入)」、「UPDATE (更新)」、「DELETE (削除)」、「TRUNCATE (空にする)」を指定する
カラム	指定のカラムの内容変化をトリガとしたい場合に、そのカラムを指定する
レベル	「文単位」…一度の DB 操作に対して一度、「行単位」…変化のあった各レコード毎に、を指定する
式	トリガの起動条件を式で指定する 式の中では「OLD. カラム名」で変更前の値、「NEW. カラム名」で変更後の値が参照できる

※例えば、テーブルに新しいレコードが挿入された時にイベントを発生させるなら、「DB操作」を「INSERT」にします。更新時に、カラム「count」の値が変化したらイベントを発生させるなら、「DB操作」を「UPDATE」、「カラム」を「count」にします。更新時にカラム「stock」と「used」の差が10未満の時にイベントを発生させるなら、「DB操作」を「UPDATE」、「式」を「NEW.stock - NEW.used < 10」とします。

※テーブルトリガによるイベントによってどういう動作をさせるかを設定する方法は、プログラミングガイドの【データベースのトリガイベント】で説明します。

【15.04. データ履歴】

テーブル属性	
属性	値
テーブル名	koutsuuhitantou
表示名	交通費(担当)
テーブルタイプ	FLOW
履歴管理	はい
バージョン管理	いいえ
権限管理	{...}
メモ	

- ・ テーブル属性で「履歴管理」を「はい」にした状態で生成・リリースした場合、そのテーブルではいつ誰がどのようなデータの変更（新規登録、編集、削除をおこなったかが記録されます。
- ・ メニューの「データ履歴」をクリックします。

データの確認					
デバッグ					
↑インポート		↓エクスポート		🔄データ履歴	☰バージョン
<input type="button" value="X"/> データ削除					
<input type="button" value="1"/> /6 > >>					
ID	名前	部署	性別		
1	鈴木一郎	101	男	✕	
2	田中二郎	101	男	✕	
3	山田太郎	102	男	✕	
4	佐藤花子	103	女	✕	
5	中島桃子	103	女	✕	
6	加藤真理	101	女	✕	
				✓	

- ・ クリックすると、下図のように「データ履歴」の閲覧できます。「処理」は「INSERT」が新規登録、「UPDATE」が編集、「DELETE」が削除です。データインポートにもなう場合は、「IMPORT:INSERT」や「IMPORT:UPDATE」などと表示されます。「詳細」ボタンをクリックすると変更内容が表示されます。

データ履歴

社員(shain)

デバッグ環境

<< < 1 /3 > >>

日時	ユーザ	処理	オプション	
2016/1/8 14:51:05	admin	INSERT	テーブル: shain; カラム: ID, name, busho	詳細
2016/1/8 14:51:40	admin	UPDATE	テーブル: shain; カラム: busho	詳細
2016/1/8 14:51:55	admin	DELETE	テーブル: shain	詳細

戻る

【15.05. バージョン情報】

- ・テーブル属性で「バージョン管理」を「はい」にした状態で生成・リリースした場合、そのテーブルではデータの変更があると直接書き換えられるのではなく別バージョンとしてレコードが保存されます。通常はバージョン番号は内部に隠れていて、閲覧されるのは最新バージョンになります。

※ この機能は上述した「コード化データの履歴問題」に対処するためのもので、他のテーブルなどから参照する際にバージョン番号を含めておくと、その後データの変更があったとしても、参照時の状態が保持されることとなります。

- ・メニューの「バージョン」で、各レコードのバージョン情報を見ることができます。例えば上の図の状態であれば、ID が 2 のレコードは氏名が変更されてバージョン 2 が、ID が 3 のレコードは busho が変更されてバージョン 2 が作られたことがわかります。

データの確認

デバッグ

↑インポート ↓エクスポート **データ履歴** ≡バージョン ✕データ削除

<< < 1 /6 > >>

ID	名前	部署	性別

バージョン情報

社員(shain)

デバッグ環境

<< < 1 /7 > >>

バージョン	更新日時	ID	氏名	busho
1	2015/11/19 11:11:10	1	鈴木一郎	101
1	2016/1/19 18:19:44	2	テスト三郎	101
2	2016/1/21 10:09:54	2	テスト次郎	101
1	2016/1/19 18:19:44	3	テスト三郎	102
2	2016/1/21 10:10:42	3	テスト三郎	103
1	2015/11/19 11:11:10	4	佐藤花子	103
1	2015/12/8 19:31:49	5	テスト太郎	101

戻る

【15.06. データインポート】

- ・「CSV から作成」で新規 DB テーブルを作成するときと同様に、必要に応じて CSV ファイルをアップロードします。
- ・アップロードできるファイルは、「csv」「xlsx」「xls」です。

データの確認

デバッグ

↑インポート ↓エクスポート データ履歴 バージョン データ削除

<< < 1 /6 > >>

ID	名前	部署	性別
1001	鈴木 一郎	1001	男

データインポート

インポートするファイルを選択してください

files/import

納品.csv 納品明細.csv

ファイルの種類: csv

アップロード:

参照... ファイルが選択されていません。

戻る 次へ

- ・「ファイルの種類」は、「すべて」と「CSV」から選択できます。
- ・目的の CSV ファイルを選択して、「次へ」をクリックします。

データインポート

対応するカラムを選択してください

ID	名字	名前	苗字かな	名前かな	アカウントID
1	インフォ	一郎	いんぷお	いちろう	test001
2	インフォ	花子	いんぷお	はなこ	test002
3	インフォ	次郎	いんぷお	じろう	test003
4	インフォ	三郎	いんぷお	さぶろう	test004

オプション

対象環境 デバッグ

文字コード Shift-JIS

先頭行 1行目として扱う

空欄 NULLとして扱う

置換 挿入し替え

エラー処理 エラーが起きた行を無視して続行

戻る インポート実行

- ・CSV の先頭部分が表示されますので、各列の「V」をクリックして、どのカラムに対応させるかを選択します。

- ・「対象環境」は「デバッグ」と「本番」のどちらのデータベースのテーブルにインポートするかを選択します。
- ・「文字コード」は「Shift-JIS」と「UTF-8」から選択します。表示された内容が文字化けしている場合は切り替えてみてください。
- ・「先頭行」は、CSV の先頭行の扱いを選択します。

項目	説明
データ行として扱う	CSVが1行目からデータ行である場合に指定する
ヘッダ行として扱う	CSVの1行目が項目を表すヘッダ行である場合に指定する カラム名やカラムの表示名と一致していれば自動的にそのカラムと対応づけられる
無視する	CSVの1行目が項目を表すヘッダ行である場合に指定するが、カラムとの対応づけを自動で行わず手動で行う時に指定する

- ・「空欄」は「NULLとして扱う」と「デフォルト値を入れる」から選択します。
- ・「置換」は「総入れ替え」と「部分置換・追加」から選択します。
- ・「エラー処理」は「エラーが起きた行を無視して続行」と「全体を停止」から選択します。
- ・「インポート実行」で実行されます。

【15.07. データエクスポート】



データエクスポート

出力対象: デバッグ環境 ▼

出力形式: CSV ▼

先頭行: カラム名 ▼

- ・「出力対象」は「デバッグ」と「本番」のどちらのデータベースのテーブルをエクスポートするかを選択します。
- ・「出力形式」は（現在は）「CSV」と「Excel (xlsx)」から選択します。
- ・「先頭行」は「カラム名」「カラム表示名」「なし」から選択します。
- ・「作成」でファイルが作成され、ダウンロードできます。

【16. DB ビュー設計】

【16.01. DB ビュー新規作成】



- ・「白紙のビュー」「クロス集計」「リンクビュー」から選択します。
- ・まず「白紙のビュー」を選択した場合を説明します。

DBビュー作成

白紙のビュー

クロス集計

リンクビュー

テンプレート: 白紙のビュー

ビュー名:

表示名:

ベース: test ▼

カラム:


表示	ソート
<input checked="" type="checkbox"/> ID	(ソートしない) ▼
<input checked="" type="checkbox"/> 氏名	(ソートしない) ▼
<input checked="" type="checkbox"/> 年齢	(ソートしない) ▼
<input checked="" type="checkbox"/> 性別	(ソートしない) ▼

戻る 作成

- ・「ビュー名」と「表示名」を入力します。
- ・「ベース」は基本となるテーブルを選択します。
- ・「表示」のチェックボックスは、どのカラムをビューに含めるか選択します。
- ・「ソート」はカラムの値によって並べ替えをする場合に指定します。
- ・「作成」で作成されます。

- 一に「一覧」と表示されています。(図は「一覧」です)
- ・「一覧」の場合、カラム一覧には次の項目があります (A)。


項目	説明
カラム名	DB ビューでのカラム名で、DB テーブルのカラム名とは別の名前を付けることができる
表示名	わかりやすい日本語での表示名
式	そのカラムの値となる PostgreSQL の式を指定する
メモ	覚え書きに使用

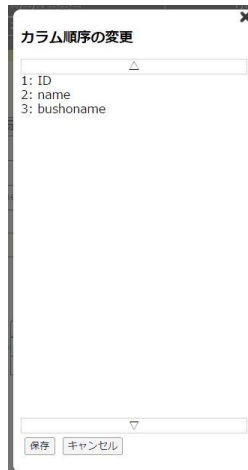
- ・式領域にある、 のアイコンをクリックすると、次の式ダイアログが開いて式を編集できます。編集領域で直接入力できるほか、「名前追加」「関数」のリストをクリックするとカーソル位置に挿入されます。「保存」ボタンをクリックするとカラム一覧の式領域に入ります。



※「式」に文字列を指定するときは「'文字列'::text」のようにシングルクォーテーションで囲んで「::text」のキャストを付けてテキスト型であることを明示する必要があります。

※ AGE関数は、PostgreSQLのインターバル型（日時の間隔を表す）の値を返しますが、今のところBuddyではインターバル型の値を正しく表示できません。例えば誕生日の入ったカラム「birthday」から年齢を得たい場合は「DATE_PART('year',AGE(birthday))」のようにしてAGEの結果から年部分を取り出してください。

- ・カラムの削除と順序変更は、 のアイコンをクリックすることでおこなえます。また、「順序変更」ボタンをクリックすると次のダイアログが開いて、カラム名をドラッグすることで順序を変更できます。



- ・「集計」は、指定のカラムの値が同じレコードをまとめて、件数や指定のカラムの合計値などを得るためのものです。右上のメニューに「集計」と表示されています。

「集計」では、カラム一覧に「集計タイプ」の列が現れます。

DBビュー設計 v報告日集計(vhokokubi)

集計 ▼ 保存 モデル履歴 削除

カラム名	表示名	式	集計タイプ	メモ	
houkokubi	報告日	vuriage.houkokubi	グループ項目		X↑↓
gokei	合計	vuriage.shokei	合計		X↑↓
ID	件数	vuriage.ID	カウント		X↑↓
(new)					

オプション

属性	値
ビュー名	vhokokubi
表示名	v報告日集計
権限管理	{...}
メモ	

テーブル

vuriage

テーブルを追加

- ・「集計タイプ」には、次の項目があります。

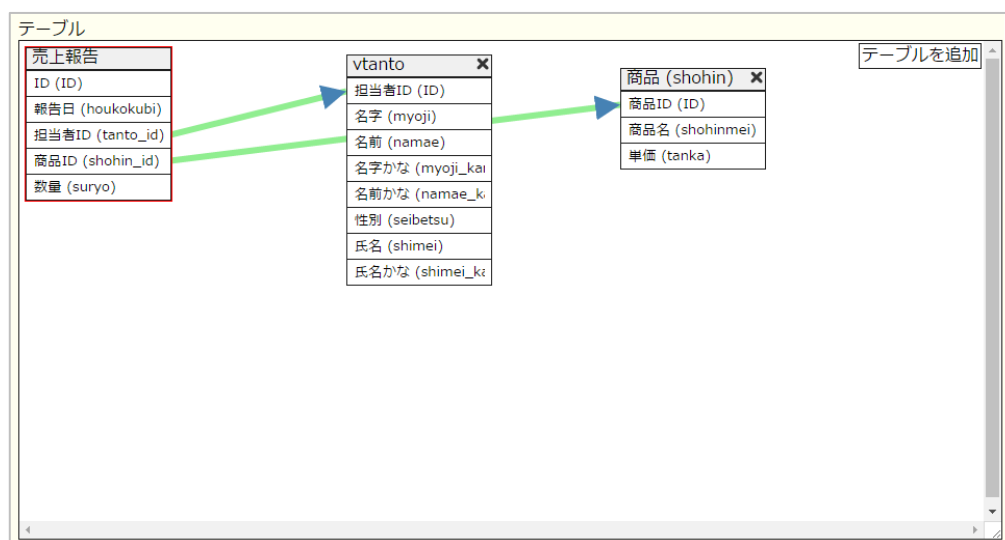
項目	説明
グループ項目	そのカラム値が同じレコードをグループ化する
合計	指定したカラムの値を合計する
カウント	件数を数える
最大	指定したカラムの値の最大値を得る
最小	指定したカラムの値の最小値を得る
平均	指定したカラムの値の平均値を得る
集計と無関係の式	上記とは無関係の式

- ・「集計」の「式」には、そのカラムの値となる式を指定します。PostgreSQLにおける式となります。

(どういう関数や演算子ができるかは、PostgreSQLのドキュメントを参照ください。) テーブルのカラムの値をそのまま表示する時は、テーブル名.カラム名とします。

- ・カラム一覧にテーブルのカラムを追加する時は、カラム一覧の下の「テーブル」に表示されているテーブルのカラム一覧からドラッグして下さい。

- ・「テーブル」の領域 (㊸) には、ビューを構成するテーブルやビューとそのカラム一覧が表示されます。最初は、ビューを新規作成する際にベースとして指定したテーブルが表示されています。別のテーブルを組み合わせたい場合は、「テーブル追加」をクリックして選択するか、画面左側の「DB テーブル」の下にあるテーブル名をドラッグしてテーブルを追加してください。不要なテーブルはテーブル名の右の「×」で削除できます (テーブル自体の削除ではなく、この DB ビューに含まなくなるという意味です)。
- ・テーブルやビューを追加したら、ベースのテーブルやビューとどう結合するかを指定します。



例えば「社員名簿」テーブルの「所属部署」カラムと、「部署」テーブルの「部署コード」カラムの値が同じレコードを結びつけるとすると、「所属部署」を「部署コード」へドラッグします。すると、線で結ばれます。

各テーブルの位置はテーブル名をドラッグして動かすことができるので、線が見やすいように調整して作業してください。

結合は多くの場合カラムの値が等しいレコードを結びつけますが、それだと対応する値がないレコードはビューに含まれなくなります。例えば「所属部署」カラムが空のレコードがあるような場合です。そのようなレコードもビューに含めたい場合は、次の結合の仕方の調整を行い、「left outer join」を使います。

線をクリックするとダイアログが開いて、結合の仕方を調整できます。



項目	説明
結合条件	「等しい」… 緑の線
結合の種類	「join」… 値が結合条件に合致するレコードだけが結果に含まれる 「left outer join」… 左側のテーブルのレコードは全て結果に含まれ、線が矢印になる（IEの場合は矢印にはならない）

- ・「データの確認」領域 (C) には次のメニューがあります。

メニュー	説明
エクスポート	デバッグ用や本番用のデータベースに当該ビューの内容をファイルに出力する

- ・この画面で何らかの入力・修正をおこなったら、必ず「保存」して下さい。
保存せずにブラウザを閉じたりリロードしたり他のアプリの開発画面に移るなどをおこなうと、変更内容は失われます。
- ・この画面で設計したビューの構造等は、アプリをデバッグ用に生成したときや本番用にリリースしたときに、初めて実際のデータベースに反映します。
従って、「データの確認」に表示されるビューの構造は、現在設計中のものとは異なる場合があることに注意してください。

※「データの確認」領域 (C) の下には何らかのエラーがあるとエラー内容が表示されます。例えば0での除算を行うような式を指定するとその旨表示されます。また、生成されたデータベースと、現在編集中的の内容が異なる場合、例えばカラムを追加した場合には「列”…”は存在しません」というエラーメッセージが表示されますが、保存して生成するとこのエラーメッセージは出なくなります。

- ・「オプション」は、このビュー全体についての属性で、次の項目があります。

オプション ▲

属性	値
ビュー名	shainbusho
表示名	社員部署
権限管理	{...}
メモ	

条件 ▲

並び順 ▲

new

▲

▼

項目	説明
ビュー名	作成時に付けた半角英数字の名前
表示名	わかりやすい日本語表示の名前
権限管理	権限設定が有効な状態では「{…}」と表示され、クリックすると権限設定がおこなえる
メモ	覚え書きに使用

- ・「条件」の下の領域には、表示するレコードを絞り込むための条件が表示されます。クリックすると入力用のダイアログが開きます。PostgreSQL における式で条件を指定します。
- ・「並び順」は、このビューを表示する時にレコードがどういう順序で並べるかを指定します。並び順の基準となるカラム名を入力します。するとその右に「昇順」「降順」の選択が現れるので設定します。複数のカラムを指定することもでき、その場合は上のカラムの値でまず順序づけされ、それが同じ場合に下のカラムの値で順序づけされる、というようになります。

【16.03. クロス集計ビュー】

- ・「クロス集計」ビューとは、元となる DB テーブルまたは DB ビューの、複数のカラムの値の組み合わせによってデータを分類し、集計するものです。
- ・例えば次の表のような名簿があるとして、部署と性別の組み合わせが何件あるか（つまり部署ごとの男女の人数）を集計すると、下の表のようになります。

ID	氏名	部署名	性別
1	鈴木一郎	総務課	男
2	田中二郎	総務課	男
3	山田太郎	製造課	男
4	佐藤花子	営業1課	女
5	中島桃子	営業1課	女
6	加藤真理	総務課	女

部署名	男	女
営業1課	0	2
総務課	2	1
製造課	1	0

- ・この場合の、部署名の「営業1課」「総務課」などを行ラベル、性別の「男」「女」を列ラベルと呼びます。
- ・DB ビューの新規作成で「クロス集計」を選ぶと次の画面になります。

DBビュー作成

テンプレート: クロス集計

ビュー名:

ベース: test

	<p>列ラベル</p> <p>指定方法: <input type="text" value="現在のカラム値を使う"/></p> <p>カラム: <input type="text" value="カラム選択..."/></p> <p><input type="checkbox"/> "その他"を設ける</p>
<p>行ラベル</p> <p>カラム: <input type="text" value="カラム選択..."/></p>	<p>集計値</p> <p>集計タイプ: <input type="text" value="合計"/></p> <p>カラム: <input type="text" value="カラム選択..."/></p>

戻る 作成

- ・「ビュー名」は既存の DB テーブルや DB ビューと重ならない半角英数字の名前を入力します。
- ・「ベース」は元になる DB テーブルまたは DB ビューの名前です。
- ・「行ラベル」の「カラム選択」で、行ラベルの元となるカラムを選択します。必要に応じて複数選択することができます。指定を取り消す時は「×」をクリックします。
- ・「列ラベル」は、まず「指定方法」を選択します。

項目	説明
現在の値	指定したカラムの現在の値から列ラベルを作成する
値の列挙	列挙した値を列ラベルとする
任意の式	SQLの式を指定してその値を列ラベルとする

- ・「列ラベル」の「カラム選択」で、列ラベルの元となるカラムを選択します。列ラベルは複数選択はできません。指定を取り消す時は「×」をクリックします。
- ・「"その他"を設ける」にチェックを入れると、「その他」という列ラベルを追加して、指定した列ラベルにはない値が現れた場合にそこに分類されるようにします。このチェックを入れない場合は列ラベルにない値が現れた場合は無視されます。
- ・「集計値」の「集計タイプ」は集計の方法を選択します。

項目	説明
合計	指定したカラムの値を合計する
カウント	指定したカラムの値の最大値を得る
最大	指定したカラムの値の最大値を得る
最小	指定したカラムの値の最小値を得る
平均	指定したカラムの値の平均値を得る

- ・「集計値」の「カラム選択」で、集計対象となるカラムを選択します。指定を取り消す時は「×」をクリックします。

例として、上記の表の例の場合であれば、次のように指定することになります。

- ・「作成」ボタンをクリックすると、作成されて、そのクロス集計ビューの設計画面になります。

カラム名	表示名	式	集計タイプ	メモ
bushoname	部署名	bushoname		
cat2	男	sex = '男'	カウント	
cat1	女	sex = '女'	カウント	
(new)				

部署名	男	女
営業1課	0	2
総務課	2	1
製造課	1	0

※ 上の画面例では集計結果の表が表示されていますが、クロス集計ビューを作成しただけではこの表示はされません。保存して生成をおこなって始めて結果の表示がされることに注意してください。

- ・列ラベルのカラム名は「cat1」「cat2」…のような連番の名前が自動的に付きます。適宜変更して構いません。

- ・各カラムの右端の↑↓をクリックして、列の順序を変更することができます。

【16.04. リンクビュー】

- ・「リンクビュー」とは、他のアプリの DB テーブルや DB ビューを参照するビューです。

DBビュー作成



テンプレート: リンクビュー

ビュー名:

表示名:

リンク先のアプリ: test1207

ベース: ▼

カラム:

<input checked="" type="checkbox"/> 表示	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/> ID	<input type="text" value="(ソートしない)"/>	▼
<input checked="" type="checkbox"/> name	<input type="text" value="(ソートしない)"/>	▼

- ・「ビュー名」は半角英数字の名前を指定します。「表示名」は日本語の名前を指定できます。
- ・「リンク先のアプリ」は「選択」ボタンをクリックするとアプリの一覧がダイアログで表示されて選択できます。「ベース」は選択したアプリの DB テーブルと DB ビューが選択肢となります。「カラム」は選択した DB テーブルまたは DB ビューのカラムが表示されます。「表示」のチェックボックスのチェックを外すとリンクビューに含まれなくなります。「ソート」は並べ替えの基準となるカラムを指定します。
- ・「作成」をクリックするとリンクビューが作成されます。

DBビュー設計 lvtest

名前	表示名
ID	ID
namae	名前
yuubinbangou	郵便番号
juusho	住所
denwabangou	電話番号
meruadoresu	メールアドレス

デバッグ ▼

[↑エクスポート](#)

|< << < 1 / 9 > >> >|

ID	名前	郵便番号	住所	電話番
1	山田太郎	606-8083	京都府京都市左京区修学院犬塚町8-75	075-123-4567
2	佐藤次郎	606-8084	京都府京都市左京区修学院薬師堂町1-23	075-234-5678

オプション ▲

属性	値
ビュー名	lvtest
表示名	
リンク先アプリ	test1116
リンク先テーブル	addr
メモ	

- ・画面下のデータ確認部分はリンクビューを作成しただけでは表示されず、アプリを生成してデータベースに反映してはじめて表示されることに注意してください。
- ・リンクビュー自体には権限設定はなく、リンク先の DB テーブルまたは DB ビューの権限設定に従います。

※ 今のところは、リンクビューを含んだビューを作成することはできません。

【17. スクリーン設計】

【17.01. スクリーン新規作成】



- ・スクリーンのテンプレートが表示されるので、目的のものをクリックすると、スクリーンが作成されてその設計画面に移ります。

テンプレート名	説明
白紙の画面	通常形式の白紙の画面が用意される
グリッド画面	グリッド形式の白紙の画面が用意される

- ・Buddy で作成するスクリーンには、通常形式とグリッド形式があります。通常形式は、横配置 BOX（要素が左から右へ配置されて幅を超えると改行）縦配置 BOX（要素が上から下へ配置）自由配置 BOX（要素を BOX 内の自由な位置に配置）を組み合わせることで画面を構成します。グリッド形式は、Excel のようにあらかじめ縦横に区切られた中に要素を配置します。

※ 以下のテンプレートはデータベースを操作する画面の雛形で、ベースとなる DB テーブル名または DB ビュー名を選択して、使用するカラムを選択する必要があります。これらのテンプレートにはそれぞれ通常形式とグリッド形式が用意されています。

テンプレート名	説明
閲覧画面	ーレコードずつ表示する
明細付き閲覧画面	ーレコードとそれに対応する別テーブルの複数レコードを表示する
検索画面	検索条件を入力して、該当するレコードを一覧表示する

登録・編集画面	レコードを追加登録・編集する
明細付き登録・編集画面	レコードとそれに対応した別テーブルの複数レコードを追加登録・編集する
クロス集計画面	クロス集計をおこなって、その結果の表やグラフを表示する

※ 次のテンプレートは、各種の画面を用意した後で、それらへのメニュー画面を作成するために使用します。

テンプレート名	説明
メニュー画面	他の画面へのリンクがメニューとして表示される

※ 次のテンプレートは、既存の画面から画面を作成する際に使用します。

テンプレート名	説明
既存のコピー	既存のスクリーンを選択するとそのコピーが作成される

※ 単票、検索、登録・編集、明細付き登録・編集、クロス集計、のスクリーンテンプレートでは、今のところ配列カラムを扱えません

<白紙の画面>

- ・「白紙の画面」を選択すると、次の画面になります。

スクリーン作成



テンプレート: 白紙のスクリーン

スクリーン名:

表示名:

- ・「白紙の画面」は画面のパーツ（モジュール）を横に並べる横配置 BOX、縦に並べる縦配置 BOX、自由な位置に置く自由配置 BOX を組み合わせて画面を作成します。
- ・「スクリーン名」（半角英数字）と「表示名」（好きな名前）を入力して「作成」をクリックするとスクリーンが作成されてその設計画面に移ります。表示名は省略することもできます。

<グリッド画面>

- ・「グリッド画面」を選択すると、次の画面になります。

- ・「**グリッド画面**」は Excel シートのような縦横のグリッドに区切られた中にモジュールを配置して画面を作成します。閲覧画面のテンプレートにもグリッド版が用意されているものもあります。

スクリーン作成



テンプレート: グリッドスクリーン

スクリーン名:

表示名:

大きさ: 行 列

- ・「**スクリーン名**」と「**表示名**」に加えて、「**大きさ**」でグリッドの行数と列数を指定できます。グリッドの行や列は設計画面で増減できますので、ここでの指定は適当で構いません。

< 閲覧画面 / 登録・編集画面 >

- ・「**閲覧画面**」および「**登録・編集画面**」を選択すると、次のような画面になります。

〔 閲覧画面 〕

スクリーン作成



テンプレート: ▼

スクリーン名:

表示名:

ベース: ▼

横幅の自動調整 ▼

表示項目:
名前 表示名 モジュールカラム

IDカラム: ▼

検索画面: ▼

登録・編集画面: ▼

〔登録・編集画面〕

スクリーン作成



テンプレート:

スクリーン名:

表示名:

ベース:

横幅の自動調整

表示項目:
名前 表示名 モジュールカラム +

IDカラム:

検索画面:

閲覧画面:

- ・テンプレートには、「… (縦並び)」と「… (横並び)」と「… (グリッド)」があります。「… (縦並び)」と「… (横並び)」は、入力欄が縦に並ぶか横に並ぶかを指定します。

※「登録・編集画面 (縦並び)」で説明します。

スクリーン作成



テンプレート:

スクリーン名:

表示名:

ベース:

横幅の自動調整

表示項目:

名前	表示名	モジュール	カラム	
ID	ID	データインプット	ID	+
myoji	苗字	テキストボックス 文字列	苗字	+
namae	名前	テキストボックス フルダウンス	名前	+
myoji_kana	苗字かな	データインプット	苗字かな	+
namae_kana	名前かな	データインプット	名前かな	+

IDカラム:

検索画面:

閲覧画面:

- ・「スクリーン名」と「表示名」を入力し、「ベース」で対象の DB テーブルを選択します。
- ・「横幅の自動調整」を「する」にすると、生成されるスクリーンのスクリプトに、画面の横幅が変わった時に画面上の各項目の幅を自動調整するプログラムコードが含まれるようになります。

- ・「表示項目」に、選択した DB テーブルのカラムに対応した入力欄の設定が表示されます。表示項目は「+」と「-」のボタンで増減できます。
- ・「名前」はスクリーンに配置されるモジュールを識別する名前に使われます（ここで指定した名前の前に DATA が付いたものになります）。
- ・「表示名」は入力欄に対応したラベルに表示される文字列です。
- ・「モジュール」は表示欄や入力欄として使われるモジュールの種類です。
 - ・ 文字列 … 文字列を表示
 - ・ 画像 … 画像を表示
 - ・ テキストボックス … 文字列の入力欄
 - ・ プルダウンリスト … 択肢からの選択
 - ・ データインプット … 指定したカラムの型に応じた入力欄
- ・「カラム」は対応するカラムの名前で、カラムを指定せず表示するためだけの項目を設けることもできます。
- ・「ID カラム」は、「ベース」で指定した DB テーブルのレコードを一意に識別できるカラムを指定します。検索画面など他の画面からこの画面に遷移した時に、どのレコードのデータを表示するかが、このカラムの値にもとづいて決められます。
- ・「検索画面」「閲覧画面」（閲覧画面の場合は「検索画面」「登録・編集画面」になります）は、他の画面に遷移するボタンを表示するかどうかを指定します。
スクリーン名を指定するとそのスクリーンに遷移するボタンのあるスクリーンが作成されます。
「後から指定する」だとボタンはありますが遷移先のスクリーン名が無指定の状態ですクリーンが作成されます（作成されたスクリーンのスクリプトを後で編集して遷移先スクリーン名を設定する必要があります）。
「スクリーン名を入力する」を選択するとその右のテキストボックスに入力できます。ここにはまだ存在しないスクリーン名も入力できます。後でその名前のスクリーンを作成するのを忘れないようにしてください。
- ・「作成」をクリックするとスクリーンが作成されてその設計画面に移ります。

<明細付き閲覧画面／明細付き登録・編集画面>

- ・「明細付き閲覧画面」を選択すると、次のような画面になります。

スクリーン作成



テンプレート： 明細付き閲覧画面(非グリッド) ▼

スクリーン名：

表示名：

ベース： 担当者マスター ▼

表示項目：

名前	表示名	モジュール	カラム	
ID	ID	文字列 ▼	ID ▼	+ -
myoji	苗字	文字列 ▼	苗字 ▼	+ -
namae	名前	文字列 ▼	名前 ▼	+ -
myoji_kana	苗字かな	文字列 ▼	苗字かな ▼	+ -
namae_kana	名前かな	文字列 ▼	名前かな ▼	+ -

IDカラム： ID ▼

検索画面： 後から指定する ▼

登録・編集画面： 後から指定する ▼

明細： kyuuka ▼

※ テンプレートは「非グリッド」「グリッド」から選べます。

- ・「明細付き登録・編集画面」を選択すると、次のような画面になります。

スクリーン作成



テンプレート： 明細付き登録画面(非グリッド) ▼
明細付き登録画面(グリッド) ▼

スクリーン名：

表示名：

① ベース： 員種 ▼

表示項目：

名前	表示名	モジュール	カラム	
ID	ID	データインプット ▼	ID ▼	+ -
hizuke	日付	データインプット ▼	日付 ▼	+ -
tokuisaki	得意先	データインプット ▼	得意先 ▼	+ -
tantou	担当	データインプット ▼	担当 ▼	+ -

IDカラム： ID ▼

検索画面： 後から指定する ▼

閲覧画面： 後から指定する ▼

② 明細： 員種 ▼

※ テンプレートは「非グリッド」「グリッド」から選べます。

- ・「明細付き閲覧画面」「明細付き登録・編集画面」によるスクリーンを作成する場合は「ベース」(A)となるテーブルと「明細」(B)となるテーブルが必要です。
例えば見積データであれば、一件の見積データの日付や得意先客を入れる「見積」テーブルと、それに一对多で対応して商品、単価、数量を入れる「見積明細」テーブルを用意しておきます。
上記の画面では「ベース」にその「見積」テーブルが選択されています。
「明細」に「見積明細」を選択して「追加」ボタンをクリックすると次のような画面になります。

スクリーン作成

テンプレート: 明細付き登録画面(非グラフ)

スクリーン名:

表示名:

① ベース: 見積

表示項目	名前	表示名	モジュール	カラム	
	ID	ID	データインプット	ID	+
	hizuke	日付	データインプット	日付	+
	bokusaki	得意先	データインプット	得意先	+
	tantou	担当	データインプット	担当	+

IDカラム: ID

検索画面: 後から指定する

閲覧画面: 後から指定する

② 明細#1: 詳細 (shousai)

表示項目	名前	表示名	モジュール	カラム	
	shousaiID	詳細ID	データインプット	詳細ID	+
	mitsumori_id	見積ID	データインプット	見積ID	+
	shouhin	商品	データインプット	商品	+
	kataban	形番	データインプット	形番	+
	tanka	単価	データインプット	単価	+
	suryou	数量	データインプット	数量	+
	shokei	小計	データインプット	小計	+

集計項目: 名前 表示名 カラム 種類 +

参照関係: 明細: 詳細ID - ベース: ID

削除

明細: 詳細

追加

戻る 作成

- ・①「IDカラム」は、「ベース」で指定したDBテーブルのレコードを一意に識別できるカラムを指定します。検索画面など他の画面からこの画面に遷移した時に、どのレコードのデータを表示するかが、このカラムの値にもとづいて決められます。
- ・②「検索画面」「閲覧画面」(閲覧画面の場合は「検索画面」「登録・編集画面」になります)は、他の画面に遷移するボタンを表示するかどうかを指定します。

スクリーン名を指定するとそのスクリーンに遷移するボタンのあるスクリーンが作成されます。

「後から指定する」だとボタンはありますが遷移先のスクリーン名が無指定の状態ですクリンが作成されます（作成されたスクリーンのスクリプトを後で編集して遷移先スクリーン名を設定する必要があります）。

「スクリーン名を入力する」を選択するとその右のテキストボックスに入力できます。ここにはまだ存在しないスクリーン名も入力できます。後でその名前のスクリーンを作成するのを忘れないようにしてください。

- ・⑧「集計項目」で「+」ボタンをクリックすると集計項目を設定できます。
- ・⑧「参照関係」で、明細のテーブルとベースのテーブルがどのカラムで結びつくかを設定します。上記の例では「詳細」テーブルの「見積 ID」が「見積」テーブル「ID」を参照するので、「参照関係」の「詳細」のところは「見積 ID」にすればよいわけです。

※ 上記の例では、「ベース」の「ID」が自動連番項目であれば入力する必要は無く、「詳細」の「見積 ID」も生成されたスクリーンで自動的に管理されるので、右端の「-」ボタンをクリックして表示項目から外すと良いでしょう。

- ・「作成」ボタンをクリックすると、スクリーンが作成されてその編集画面になります。この時、明細部分については別のスクリーンが作成されてスクリーンコンテナで埋め込まれる形となります。

<検索画面>

- ・「検索画面」を選択すると、次の画面になります。

スクリーン作成

🔍

テンプレート: 検索画面(特ダリ) ▼

スクリーン名:

表示名:

ベース: 見積 ▼

表示項目:

	表示	検索	ソート	全体検索	画像表示
☑ ID	部分一致 (LIKE) ▼	(ソートしない) ▼	する ▼	しない ▼	
☑ 日付	部分一致 (LIKE) ▼	(ソートしない) ▼	する ▼	しない ▼	
☑ 得意先	部分一致 (LIKE) ▼	(ソートしない) ▼	する ▼	しない ▼	
☑ 担当	部分一致 (LIKE) ▼	(ソートしない) ▼	する ▼	しない ▼	

IDカラム: ID ▼

キャッシュ: 有効にする

行クリック時の遷移先: 後から指定する ▼

閲覧画面: 後から指定する ▼

登録・編集画面: スクリーン名を入力する ▼

戻る 作成

- ・「**スクリーン名**」と「**表示名**」を入力し、「**ベース**」で対象の DB テーブルを選択し、「**表示**」「**検索**」「**ソート**」「**全体検索**」「**画像表示**」でその DB テーブルの表示するカラム、検索の方法、ソート順、画像表示について指定します。
- ・「**検索**」で「(検索しない)」を選択すると、そのカラムについての検索条件入力欄は作られません。「**範囲** (≧開始値、≦終了値)」を選択すると、検索条件入力欄が二つ (開始値と終了値) が作られます。
- ・「**ソート**」ではソート (並べ替え) をするカラムについて、上位から順に「**ソート 1**」「**ソート 2**」…と指定します。「(昇順)」「(降順)」は並べ替えの順序です。
- ・「**全体検索**」が「**する**」のカラムは、全体検索の対象となります。全体検索というのは、一つの入力欄に入力した検索キーワードが指定した複数のカラムのどれかに含まれるものを検索するものです。
- ・「**画像表示**」は、対象カラムの型が「**ファイル**」「**画像**」の場合に「**する**」にすると、検索結果の一覧画面で画像サムネイルが表示されます。「**しない**」だと画像ファイル名の表示となります。
- ・「**ID カラム**」は、「**ベース**」で指定した DB テーブルのレコードを一意に識別できるカラムを指定します。閲覧画面や登録・編集画面など他の画面へ遷移する時に、どのレコードのデータを表示するかが、このカラムの値にもとづいて決められます。

※以前の Buddy では「**キャッシュ**」という設定項目がありましたが、これは廃止されました。

- ・「**行クリック時の遷移先**」は、検索結果の表の行 (どれかのレコード) をクリックした時に、どのスクリーンに遷移するかを指定します。「**使用しない**」だと遷移しません。「**後から指定する**」だと遷移先のスクリーン名が無指定の状態で作成されます (作成されたスクリーンのスクリプトを後で編集して遷移先スクリーン名を設定する必要があります)。
- ・「**閲覧画面**」「**登録・編集画面**」は、他の画面に遷移するボタンを表示するかどうかを指定します。
スクリーン名を指定するとそのスクリーンに遷移するボタンのあるスクリーンが作成されます。
「**後から指定する**」だとボタンはありますが遷移先のスクリーン名が無指定の状態で作成されます (作成されたスクリーンのスクリプトを後で編集して遷移先スクリーン名を設定する必要があります)。
「**スクリーン名を入力する**」を選択するとその右のテキストボックスに入力できます。ここにはまだ存在しないスクリーン名も入力できます。後でその名前のスクリーンを作成するのを忘れないようにしてください。

※ 上記の画面遷移を使用する時は、検索画面での「ID カラム」と、遷移先画面での「ID カラム」に、同じカラムが指定されている必要があります。異なるカラムが指定されていると正しく動作しないので注意してください。

- ・「作成」をクリックするとスクリーンが作成されてその設計画面に移ります。

※ 作成された検索画面には、上記で指定した各カラムについての検索条件の入力欄とそれにもとづいて検索する「絞り込み検索」のボタンと、「全体検索」が指定されていれば検索キーワード入力欄とそれにもとづいて検索する「全体検索」のボタンが配置されます。

<クロス集計画面>

- ・「クロス集計画面」を選択すると、次の画面になります。

スクリーン作成

テンプレート: クロス集計画面

スクリーン名:

表示名:

戻る 作成

- ・「スクリーン名」と「表示名」を入力し、「作成」をクリックすると作成されます。
- ・作成される画面には、クロス集計モジュール、グラフモジュール、DB テーブルモジュールが配置されており、アプリ実行時には次のような画面となります。

テーブル: member

列ラベル

指定方法: 現在のカラム値を使う

カラム: グループ

カラム追加... 削除

指定方法: 現在のカラム値を使う

カラム: 性別

カラム追加... 削除

追加

"その他"を設ける

行ラベル

カラム: 都道府県

カラム追加...

集計値

集計タイプ: カウント

集計値カラム: ID

削除

追加

更新 行と列の入れ替え

棒グラフ

■ 京都府 ■ 大阪府 ■ 東京都

2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0

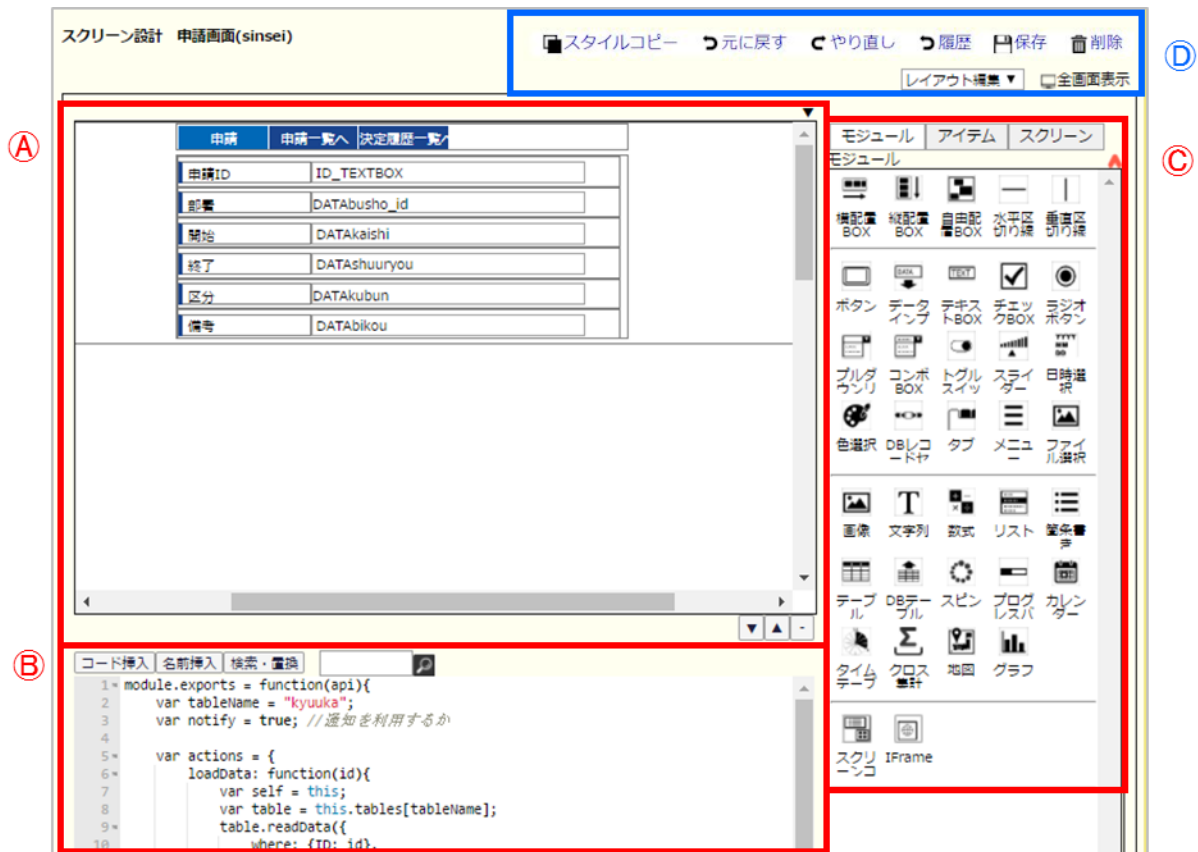
A 女 A 男 B 女 B 男 C 女 C 男

1 / 3

検索 画像出力 Excel出力

都道府県	A		B		C	
	女	男	女	男	女	男
京都府	1	0	0	1	0	2
大阪府	2	0	2	0	1	0
東京都	2	2	0	2	0	0

【17.02. スクリーン設計】



・スクリーン設計画面には、三つの部分があります。

左上には作成されるスクリーンに対応した領域 (A) があり、ここに画面を構成する要素 (モジュール) を配置していきます。

その下には、このスクリーンの動作を記述するスクリプト (Javascript のプログラム) を編集する領域 (B) があります。

右側には「モジュール」「アイテム」「スクリーン」のタブで切り替える領域 (C) があります。

※ 領域Aの高さはその右下のボタン で調節できます。「-」ボタンでは最小化され、「+」ボタンでは元に戻ります。

※ この領域 (C) は「レイアウト編集/機能編集」の切替で状態が変わります。(後述)



・右上 (D) には次のメニューがあります。

メニュー	説明
スタイルコピー	他のスクリーンからアイテムのスタイルをコピーする
元に戻す	変更した操作を元（最後に保存したところ）に戻す
やり直し	元に戻した操作をやり直す
履歴	保存の履歴にもとづいて過去の状態に戻す
保存	スクリーン設計を保存する
削除	このスクリーンを削除する
レイアウト編集/機能編集	編集モードを切り替える（後述）
全面表紙	スクリーン領域（A）のみを大きく表示する

- ・スクリーン設計画面で何かを変更をした場合は、**保存するのを忘れない**ようにしてください。
- ・「履歴」を選ぶと、保存の履歴が表示されます。履歴のリストからクリックして選択し、「履歴から復元する」をクリックすると、その時点の状態になります。



- ・「スタイルコピー」を選ぶとコピー元となるスクリーンを選択するダイアログが開きます。スクリーンを選択して「コピー」ボタンをクリックすると、コピー元と現在のスクリーンで、名前とタイプが同じアイテム（例えばどちらのスクリーンにも名前が「TourokuButton」のボタンがあるなど）について、スタイルがコピーされます。似た構成のスクリーンでスタイルを揃えたいときに使用できます。



- ・右上のメニューに「レイアウト編集」とある時は、「モジュール アイテム スクリーン」のタブとなります。

「モジュール」ではスクリーン領域にモジュールを配置します。

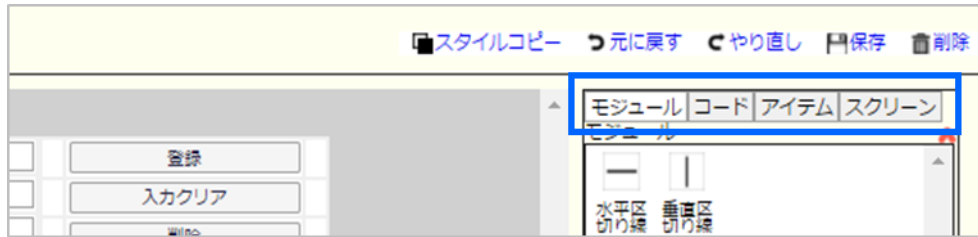
「アイテム」では配置したモジュールの位置や大きさなどの属性を調節します。

「スクリーン」ではこのスクリーン全体の属性を調節します。

The screenshot displays three panels from a software development tool:

- モジュール (Modules):** A grid of icons for various UI components such as buttons, text boxes, tables, and charts.
- アイテム (Items):** A table showing the properties of a selected item. The table has two columns: '属性' (Property) and '値' (Value).
- スクリーン (Screen):** A table showing the options for the current screen. The table has two columns: '属性' (Property) and '値' (Value).

- ※ スクリーンに配置された「モジュール」の一つ一つを「アイテム」と呼んでいます。
- ※ グリッド画面では、横配置 BOX、縦配置 BOX、自由配置 BOX は使えません。
- ※ 「スクリーン」タブの内容はグリッドでない場合とグリッドの場合で異なります。
- ※ グリッド画面では、「レイアウト編集/機能編集」の切替はなく、「モジュール コード アイテム スクリーン」のタブ構成となります。



【17.02.01. スクリーンへのモジュール配置】

- ・「モジュール」タブでは、スクリーンに配置できるモジュールのアイコンが一覧表示され、スクリーンの領域へドラッグアンドドロップすることで配置できます。また、その下の「モデル」には DB テーブルや DB ビューの名前が表示され、その名前をクリックするとさらにその中のカラム名が表示されます。そのカラム名をスクリーンの領域へドラッグアンドドロップすると、そのテーブルとカラムにデータを入力するためのデータ入力モジュールをスクリーンに配置することができます。
- ・スクリーン設計では、構成要素の配置方法に次のものがあります。

配置方法	説明
横配置	右へ右へと配置されていき、領域の幅に納まらない場合は折り返される
縦配置	下へ下へと配置されていく
自由配置	自由な位置に配置できる






※ 新規作成で「白紙の画面」を選択した時はスクリーン全体が横配置になっています。

※ スクリーンに直接モジュールを配置しても構いませんが、下記の「横配置 BOX」「縦配置 BOX」「自由配置 BOX」を配置してその中にモジュールを配置することで、画面全体を適切にレイアウトすることができます。ボックスを入れ子にすることも可能です。





※ 「グリッド画面」の場合は、縦横に仕切られたグリッドの各セルにモジュールを配置していきます。配置したモジュールの大きさはそのセルの大きさになります。行の高さや列の幅の調節、行や列の増減、セルの結合や分割ができます。（その操作方法は後述します）












※ スマホやタブレットなど画面の大きさによって自動的に配置などを変更して見やすく表示する「レスポンシブデザイン」に対応した配置ができるようにする予定です。

レイアウト用












アイコン	名前	説明
	横配置 BOX	その内部が横配置になる
	縦配置 BOX	その内部が縦配置になる
	自由配置 BOX	その内部が自由配置になる
	水平区切り線	水平の区切り線
	垂直区切り線	垂直の区切り線

入力・選択用





アイコン	名前	説明
	ボタン	クリックすることで何らかの操作をさせる
	データインプット	<p>DB テーブルのカラムを指定して、そのカラムの属性に適した入力をおこなう</p> <p>※アイテムの属性</p> <ul style="list-style-type: none"> ・ テーブル名 …対象テーブル名 ・ カラム名 …対象カラム名 ・ サムネイル表示…対象カラムが画像の場合に「はい」を選択するとサムネイルを表示 <p>※DB テーブル設計で設定した「型」と「型の詳細」に応じてそれに合致した入力のみを受け付ける</p> <ul style="list-style-type: none"> ・ 真偽値型 …チェックボックス (caption で指定した文字列がラベルになる) ・ 選択肢を設定…プルダウンリストでの入力 ・ ファイル型 …テキストボックスとして表示。クリックするとファイル選択ダイアログからファイルのアップロードが可能。 (サーバーに保存されファイル名が値となり、配列属性の場合は複数のファイルがアップロードできファイル名の配列が値となる) <p>※データインプットモジュール</p> <ul style="list-style-type: none"> ・ ユーザーの入力を受け付けるためのモジュール ・ データベースに値をセットする機能は持たない ・ データベース操作はスクリプトのデータベース操作機能を使用しておこなう <p>※カラムに範囲を指定しておけば、入力値がその範囲内にあるかどうかのチェックをするようになる予定</p>
	テキスト BOX	<p>テキストで入力できる</p> <ul style="list-style-type: none"> ・ placeholder…入力欄が空の場合に表示される説明文を指定する。あらかじめ入力内容をセットするには「caption 属性」で指定する。
	チェック BOX	<p>オン/オフを切り替えられる</p> <ul style="list-style-type: none"> ・ label…チェック BOX の横に表示される文字列

	ラジオボタン	<p>いくつかの中のひとつだけをオンにする</p> <ul style="list-style-type: none"> ・ label …ラジオボタンの横に表示される文字列 ・ radioname…ラジオボタンをグループ化する名前（この名前が同じラジオボタンはその中のひとつしかオンにならない）
	プルダウンリスト	<p>選択肢(リスト)の中から選ぶことができる</p> <p>※アイテムの属性</p> <ul style="list-style-type: none"> ・ options …値をクリックすると設定用ダイアログが開く
	コンボ BOX	<p>テキスト BOX とプルダウンリストを兼ねたもの</p> <ul style="list-style-type: none"> ・ listItems…リストを JSON 形式で指定する
	トグルスイッチ	オン/オフを切り替えられる
	スライダー	<p>左右にスライドすることで最小値から最大値の間を指定できる</p> <ul style="list-style-type: none"> ・ min …最小値 ・ max …最大値 ・ step …刻み値
	日付選択	<p>日付または日時の選択ができる</p> <p>※アイテムの属性</p> <ul style="list-style-type: none"> ・ 範囲先頭日 …選択範囲の先頭日 ・ 範囲末尾日 …選択範囲の末尾日 ・ 時刻使用 …「時分秒」または「時分」で時刻も選択 ・ 時選択肢 …時の選択肢をカンマ区切りで ・ 分選択肢 …分の選択肢をカンマ区切りで
	色選択	色が選択できる
	DB レコードセレクト	DB テーブルのレコードを選択できる
	タブ	<p>スクリーンコンテナで入れ子にした画面を切り替える時などに使用する</p> <p>※アイテムの属性</p> <ul style="list-style-type: none"> ・ menuItems …タブの項目を指定する <p>クリックすると項目編集用ダイアログが開く</p>
	メニュー	<p>メニューを入れる</p> <p>※アイテムの属性</p> <ul style="list-style-type: none"> ・ menuItems …メニューの項目を指定する <p>クリックすると項目編集用ダイアログが開く</p> <p>※非グリッドでは第一階層のメニュー項目が横に表示されるが、グリッドではクリックすると第一階層のメニュー項目が縦に開く動作となる</p>
	ファイル選択	<p>パソコンの中のファイルを選択し、内容を取得したり、サーバーにアップロードしたりする</p> <p>※アイテムの属性</p> <ul style="list-style-type: none"> ・ ファイルタイプ…選択できるファイルの種類 <p>※ファイルタイプを画像にすると、スマホなどではファイル選択時にカメラを起動して撮影した画像を使うことができる</p>



表示用

アイコン	名前	説明
	画像	画像を表示する ※アイテムの属性 ・ src …表示する画像ファイルを指定する
	文字列	文字列を表示する ※アイテムの属性 ・ リンク表示 …「はい」にすると URL をリンク表示 ・ フィルタ…適用するフィルタ名をカンマ区切りで指定
	数式	他のモジュールの値に基づいた計算式の値を表示する
	リスト	一行ずつの文字列をリストとして表示する ※アイテムの属性 ・ 複数選択…「はい」にすると複数の項目が選択できる ・ options …値をクリックすると設定用ダイアログが開く
	箇条書き	先頭にマークがついたリスト ・ listItems …リストを JSON 形式で指定する 配置したモジュールの中でひとつずつ入力・編集ができる
	テーブル	文字列を表形式で表示する
	DB テーブル	DB テーブルの内容を一覧表の形で表示する ※word-break : 改行方法を指定する ・ normal : 日本語は表示範囲を超えると改行 英語等は単語区切りで改行 ・ break-all : すべて表示範囲を超えると改行 ・ keep-all : 単語の切れ目のみで改行 ・ 無指定 : normal と同様 ※white-space : 空白文字 (半角スペース、タブ、改行) の表示方法を指定する ・ normal : 連続する空白文字はひとつの半角スペースとして表示 ・ pre : そのまま表示 ・ nowrap : 連続する空白文字はひとつの半角スペースとして表示し、改行なし ・ 無指定 : normal と同様
	スピンドル	動作中であることを回転で示す ※アイテムの属性 ・ visibility…初期状態で表示するかどうかを指定する
	プログレスバー	進行状況を表示する
	カレンダー	カレンダーを表示する ※年月を省略すると、当月のカレンダーが表示される
	タイムテーブル	タイムテーブルを表示する ・ 開始時…タイムテーブルの左端時刻を時の数値で指定 (省略=9)

→次ページへ

		<ul style="list-style-type: none"> ・ 終了時…タイムテーブルの左端時刻を時の数値で指定 (省略=18) ・ 開始日…タイムテーブルの先頭の日を指定 (省略=当日の週の日曜) ・ 日数…タイムテーブルに表示する日数を指定 (省略=7)
	クロス集計	クロス集計をおこなう
	地図	地図を表示する <ul style="list-style-type: none"> ・ 緯度 …地図の中心の緯度 ・ 経度 …地図の中心の経度 ・ ズーム…地図の倍率
	グラフ	グラフを表示する
	キャンバス	HTML5 の CANVAS 要素を埋め込む ※スクリプトによって図形の描画が可能

埋め込み・連携

アイコン	名前	説明
	スクリーンコンテナ	他のコンテナタイプのスクリーンを入れ子にする <ul style="list-style-type: none"> ・ screen …その中に表示するスクリーンを指定する
	IFrame	他の WEB ページを埋め込む <ul style="list-style-type: none"> ・ src …その中に表示する WEB ページの URL

スクリーン設計 register(register)

会社名	DATACompany
部署	DATASection
業務名	DATAgyoumumei
種類	DATAshurui
開始日	DATAkaisibi
終了日	DATAsyuryobi
分類	DATAbunrui
担当者	DATAtantousha
PDF	DATAPDF <input checked="" type="checkbox"/>

- ・ スクリーンに配置したモジュールは、クリックすると選択されて枠線が赤く表示されます。
- 右上の「×」で削除することができます。
- 右や下の枠線をドラッグすると大きさを変更できます。
- 自由配置 BOX 内のモジュールはそれ自体をドラッグすることで位置を変更できます。

「テーブル名」属性と「カラム名」属性





- ・「データインプット」以外にも、「テキストBOX」「文字列」などにも「テーブル名」「カラム名」の属性があります。これは、そのアイテムがどのDBテーブルのどのカラムを編集・表示するかを指定するものですが、この指定をすることによってそれぞれのモジュールが自動的にデータベースの読み書きをおこなうわけではありません。データベースの読み書きはスクリーンのスクリプトでおこないます。

そのための機能として `setItemDataFromDB()` と `getItemDataForDB()` という関数がスクリーンのスクリプトに用意されています。`setItemDataFromDB()` によってデータベースから読み出したデータが各アイテムの「テーブル名」と「カラム名」に従ってセットされ、`getItemDataForDB()` によって各アイテムのデータが収集されてデータベースに投入できる形にまとめられます。詳しくは【17.07. スクリプト ～データベース操作】を参照してください。


※「閲覧画面」「登録・編集画面」などのスクリーンテンプレートを用いてスクリーンを作成すると、データベースの読み書きに必要なスクリプトは自動的に生成されます。

【17.02.02. グリッドの操作】


- ・グリッドのセルをクリックすると選択されて赤い枠線が表示されます。右や下の赤い枠線をドラッグすると列の幅や行の高さを変更できます。また操作のためのアイコンがそのセルの右上に表示されます。
- ・セルの右上に表示されるアイコンをクリックすると次の操作ができます。

	そのセルの下に一行挿入します
	そのセルの右に一列挿入します
	そのセルのある行を削除します
	そのセルのある列を削除します

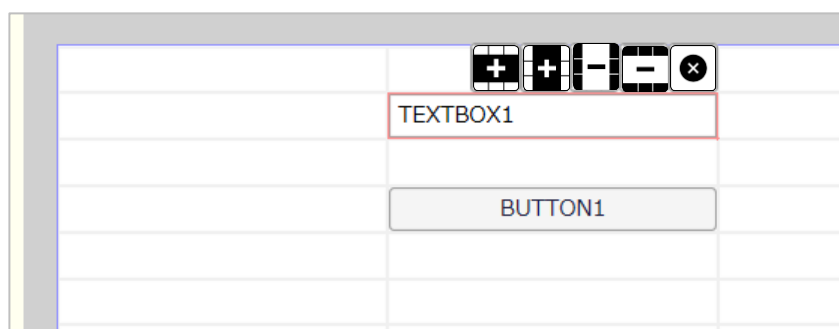
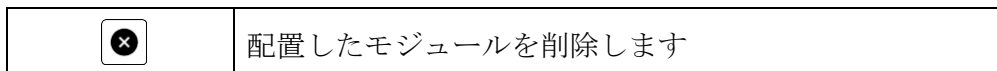
- ・Ctrl キーを押しながらセルをクリックすると、複数のセルを選択することができます。その時は次のアイコンでセルの結合ができます。

	選択されたセル群を囲む矩形内のセルを結合します
---	-------------------------

- ・結合されたセルをクリックすると、次のアイコンでセルの分割ができます。

	結合されたセルを分割して元に戻します
---	--------------------

- モジュール一覧からモジュールをセルにドラッグアンドドロップすると、セル内にモジュールを配置することができます。すでにセル内にモジュールが配置されている場合はそのセルの上に行追加した上でその追加したセルに配置します。
- モジュールの配置されたセルをクリックして選択すると、別のセルへドラッグアンドドロップして移動することができます。
- 次のアイコンで配置したモジュールを削除できます。（セルが削除されるわけではありません。）



【17.02.03. アイテム（配置したモジュール）の調整】

- 「アイテム」タブでは、スクリーンに配置したモジュールをクリックして選択すると、そのアイテムの属性が表示され、調節できます。



- ・「オプション」の下に選択されているアイテムの名前がプルダウンリストで表示されており、このプルダウンリストで別のアイテムを選択することもできます。プルダウンリストでの順序は、横配置 BOX や縦配置 BOX 中のアイテムでは配置の順序、自由配置 BOX 中のアイテムでは奥から手前の順序になります。順序は「前へ移動」「後へ移動」ボタンで変更できます（後述）。
- ・モジュールの属性で、共通するものは次の通りです。
（モジュールによって使われる属性は異なります。モジュール固有の属性は後述）

属性	説明
名前	個々の配置されたモジュールを識別する半角英数字の名前
タイプ	どのモジュールかを示す名前
クラス	スタイルをまとめて指定するためのクラス名
部分クラス	複合的なモジュールで、個々のパーツごとのクラス名を指定
caption	画面に表示される文字列（※仕様変更の可能性あり）
left	画面上での位置を指定する（自由配置の場合）
top	
width	
height	画面上での大きさを指定する
padding	内部の余白を指定する
margin	外部の余白を指定する
border	枠線を指定する

border-top	枠線を上、左、右、下について個別に指定する (入力は上の「border」属性の設定ダイアログでおこなう)
border-left	
border-right	
border-bottom	
font-color	文字色を指定する
font-size	文字サイズを指定する
改行	「はい」にすると横配置ボックス内で強制的に折り返す
text-align	そのモジュール内の横の配置
vertical-align	そのモジュール内の縦の配置
line-height	一行の高さ
box-sizing	「width」や「height」が「padding」と「margin」を含むかどうか (border-box=含む、content-box=含まない、無指定は border-box と同様)
bg-color	背景色を指定する
bg-image	背景を画像とする場合に指定する
readOnly	「はい」にすると、入力できず表示専用となる
disabled	「はい」にすると、表示はされるが使用できない状態になる

- ・表示スタイルに関する属性 (left、top、width、height、padding、margin、border、border-top、border-left、border-right、border-bottom、font-color、font-size、text-align、vertical-align、box-sizing、bg-color) は、CSS に準じます。

(CSS とは、WEB ページのレイアウトを指定する規格です)

※ 「padding」、「margin」は、CSS での指定と同様に単独の値を指定すると上下左右を一括指定したことになり、半角スペース区切りで複数指定すれば、2 つの場合「上下 左右」3 つの場合「上左右下」4 つの場合「上右下左」の指定となります。

- ・「width」や「height」に % 指定をすると、そのモジュールの親のボックスのサイズに対する割合となります。
- ・グリッドの場合の列幅は「width」の値の比率で決まり、ウィンドウサイズに応じて自動的に調節されます。
- ・グリッドの場合は、Ctrl キーを押しながらクリックすることで、複数のモジュールを選択することができます。その状態では「アイテム」タブには選択されたモジュールに共通する属性が表示され、属性の値を入力すると選択されたモジュールすべてに適用されます。
- ・DB テーブルの「width」属性は、セミコロン「;」区切りで各列の幅を指定できます。単独の数値を指定するとその中で各列の幅は均等になります。「auto」にすると各列の幅はブラウザによる自動調整になります。

- ・文字列やボタンなどのモジュールでは、配置したモジュール自体をクリック（IE ではダブルクリック）すると文字カーソルが表示され、表示されている「caption」をその場で編集することができます。
- ・「クラス」と「部分クラス」で指定した名前に対してどのようなスタイルを与えるかは「スクリーンテーマ編集」で指定することができます。

※今のところ、Firefox ではスライダーの readOnly 属性は機能しません。

【17.02.04. 属性コピー】

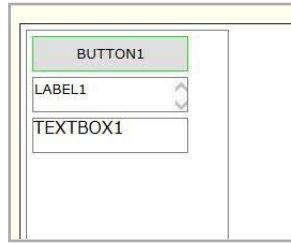
- ・モジュールの表示スタイルに関する属性のよく使うものは、モジュール間で簡単にコピーできる方法が用意されています。
コピーの元になるモジュールをクリックして選択し、赤枠で表示された状態にして、「属性コピー」をクリックします。
対象モジュールの枠線と「属性コピー」が緑で表示されて属性コピーモードになっていることを示します。



- ・コピー可能な属性名の前にチェックボックスが表示されるので、コピーしたい属性にチェックを入れます。



- ・その状態で、コピー先となるモジュールをクリックすると、選択した属性の値がコピーされます。



- ・コピーが終わったら「属性コピー」を再度クリックして、属性コピーモードを解除します。（緑色が消えます）

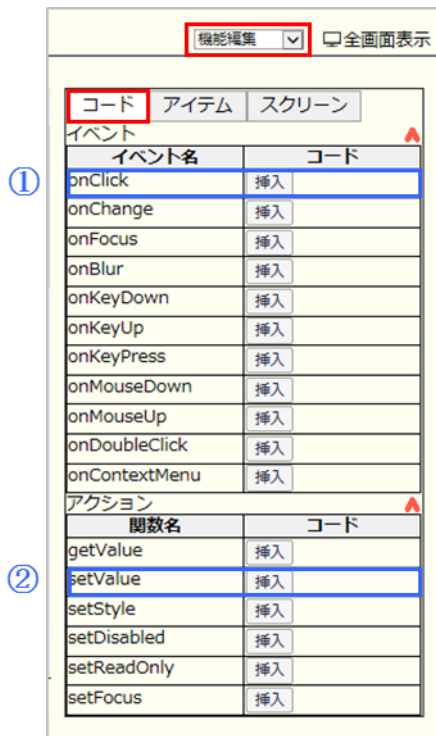
【17.02.05. 順序変更と削除】



- ・横配置ボックスや縦配置ボックス内に入れたモジュールは、順序を入れ替えることができます。
- ・順序を変えたいモジュールをクリックして選択し、画面右側の「オプション」の下の「前へ移動」「後へ移動」をクリックすると、順序を変更できます。上の図の状態であれば、「前へ移動」をクリックすれば BUTTON1 と BUTTON2 が入れ替わり、「後へ移動」をクリックすれば BUTTON2 と BUTTON3 が入れ替わることになります。
- ・「削除」ボタンをクリックすると、選択されているアイテムを削除できます。「×」をクリックすることと同じです。

【17.02.06. 機能編集のコード挿入】

- ・右上のメニューに「機能編集」とある時は、「コードアイテムスクリーン」のタブとなり、「コード」タブでは、配置したモジュールをクリックして選択すると、次のような画面となり「イベント」としてそのモジュールで可能なイベントの一覧が「アクション」としてそのモジュールで可能な操作の一覧が表示されます。



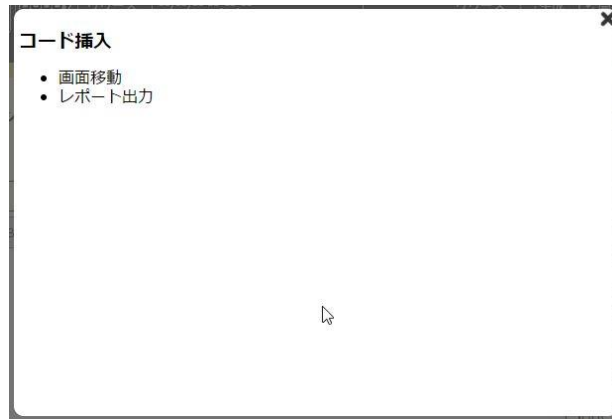
- ・「イベント」の「挿入」ボタンをクリックすると、スクリプトにそのイベント発生時に実行される関数（イベントハンドラ）の枠組みが挿入されます。
例えば「BUTTON1」という名前のボタンで、「onClick」イベントの「挿入」をおこなう（①）と、スクリプトの `events` の先頭に次の内容が挿入されます。挿入をおこなうとボタンが「挿入済」となり、「挿入済」ボタンをクリックした場合はスクリプトのそのイベントハンドラの部分にカーソルが移動します。

```
BUTTON1_onClick: function(evt){
},
```

- ・「アクション」の「挿入」ボタンをクリックすると、スクリプトの現在カーソルがある位置に、その操作をおこなうコードが挿入されます。
例えば「TEXTBOX1」という名前のテキストボックスで、「setValue」アクションの「挿入」をおこなう（②）と、カーソル位置に次の内容が挿入されます。（挿入された内容は選択された状態で青く表示されます。もし誤って挿入したときはキーボードの「Delete」キーを押せば挿入内容を削除できます。）

```
this.items.TEXTBOX1.setValue( value //
    セットする文字列
)
```

- ・ボタンモジュールについては、「機能編集」状態でスクリーンに配置されているアイテムをクリックすると、次のダイアログが開きます。「画面移動」などの機能リストから選んでクリックすると、onClick イベントハンドラとともにその内容を挿入することができます。



【17.02.07. スクリーン属性】

- ・「スクリーン」タブでは、スクリーンオプション（スクリーン全体の属性）を調節できます。グリッドでない場合とグリッドの場合で属性項目が異なります。

<グリッド画面以外>

モジュール	アイテム	スクリーン
スクリーンオプション		
属性	値	
スクリーン名	test	
表示名	test	
タイプ	ページ ▼	
クラス		
画面幅	1200	
画面高	1600	
モバイル対応幅	800	
モバイル用画面		
権限設定	...	
メモ		

<グリッド画面>

モジュール	コード	アイテム	スクリーン
属性	値		
スクリーン名	gtest		
表示名	gtest		
タイプ	ページ ▼		
クラス			
画面幅	840		
表示領域幅上限	800		
モバイル対応幅	480		
モバイル用画面	test		
権限設定	..		
メモ			
<input type="text" value="16行"/> <input type="text" value="1列"/> <input type="button" value="グリッド追加"/>			
グリッド 1 <input type="button" value="削除"/> <input type="button" value="前へ"/> <input type="button" value="後へ"/>			
属性	値		
幅	100%		
モバイル対応	しない ▼		
並び順(モバイル時)			

- ・スクリーンオプションでは次の設定ができます。

属性	説明
スクリーン名	新規作成時に指定したスクリーン名
表示名	自由に名前をつけることができる
タイプ	「ページ」と「コンテナ」から選択する ※ページ …独立した画面として表示されるもの ※コンテナ…スクリーンコンテナモジュール（後述）の中に配置することで他のスクリーンの中に入れ子にして表示するためのもの
クラス	クラス名を指定する（※この設定はまだ機能しません）
画面幅	想定される画面の幅（ピクセル）を指定する。設計画面のスクリーン領域の表示はこの幅に合わせたものになる
画面高	<非グリッド>スクリーン領域の高さ（ピクセル）を指定す
表示領域幅上限	<グリッド>グリッドの表示領域の幅（ピクセル）の上限（画面幅がこれより大きくなると、両側に余白が取られる）
モバイル対応幅	アプリ実行時の画面幅がこれより小さくなると次のようになる <非グリッド>「モバイル用画面」が指定されていればそのスクリーンが代わりに表示される <グリッド>「モバイル用画面」が指定されていればそのスクリーンが代わりに表示される。「モバイル用画面が指定されていない場合、各グリッドの幅指定はすべて100%とみなされて縦に並ぶようになる。また、「モバイル対応」が「する」のグリッドでは行のセルが横でなく縦に並ぶようになる
モバイル用画面	アプリ実行時の画面幅が「モバイル対応幅」未満の場合、代わりに表示するスクリーン名を指定する
権限設定	このスクリーンを誰が閲覧できるかの権限を設定する
メモ	開発上で記録しておくためのメモ

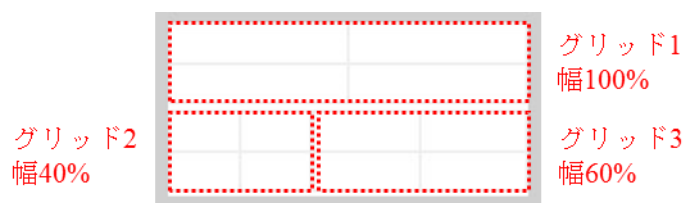
- ・「権限設定」の「…」の部分をクリックすると次のダイアログが開きます。



- ・「許可ユーザー」「拒否ユーザー」の設定に該当するとそれによって許可・拒否が決まります。
そうでない場合は「許可グループ」「拒否グループ」の設定で決まります。
許可と拒否の両方に該当した場合は「優先」によってどちらになるかが決まります。
- ・グリッド画面のグリッドの表示領域の幅は、画面幅に応じて伸び縮みします。ただし「表示領域幅上限」よりは広くならず、画面幅がそれより大きいと両側に余白が取られます。

【17.02.08. 複数グリッドの配置とモバイル対応】

- グリッド画面は通常は一つのグリッドで構成されますが、「グリッド追加」ボタンをクリックすると、その左で指定した行数と列数のグリッドが追加されます。順に「グリッド1」「グリッド2」…と番号が付き、それぞれについて「幅」「モバイル対応」「並び順 (モバイル時)」の属性を指定できます。「幅」はグリッド全体の幅に対する割合を「%」で指定します。各グリッドは1から順に左から右へ配置され、全体の幅を超えると改行されます。グリッドの順序は「前へ」「後へ」のボタンで変更できます。例えば2行2列のグリッド1、2、3を、それぞれ幅100%、40%、60%とすると次のように配置されることになります。



- アプリ実行時の画面幅が「モバイル対応幅」より小さいと上記の配置が変わり、各グリッドの幅がすべて100%とみなされて縦に並ぶようになります。このとき、「並び順 (モバイル時)」に番号が指定されているとその順で並びます。



- また、グリッドの「モバイル対応」を「する」にした場合、アプリ実行時の画面幅が「モバイル対応幅」より小さいと、通常は横に並ぶ各行のセルが縦に並ぶようになります。

名前	TEXTBOX1
所属	TEXTBOX2

設計画面

名前	
所属	

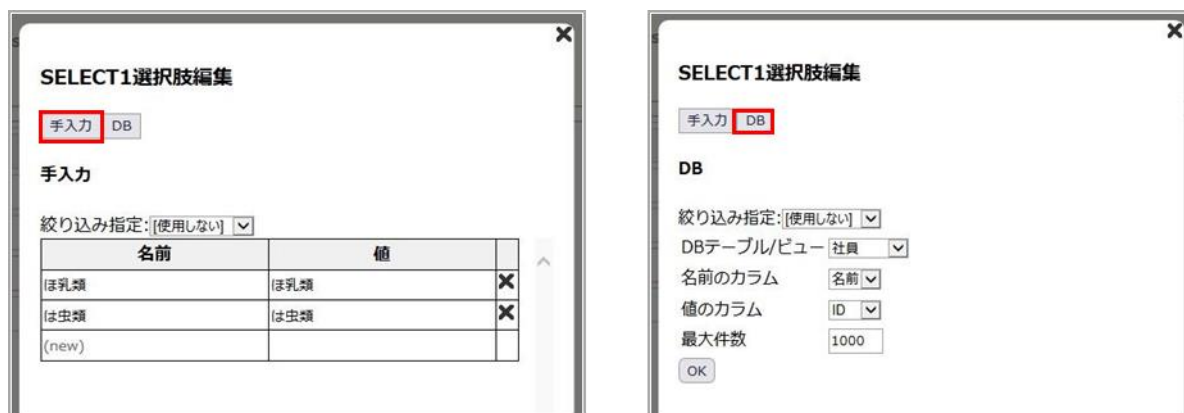
アプリ画面 (通常)

名前	
所属	

アプリ画面
(画面幅がモバイル対応幅より小さい場合)

【17.02.09. 選択肢】

- ・ options 属性の値の部分をクリックすると、選択肢を設定するダイアログが開きます。



- ・「手入力」では名前と値を入力します。「DB」では既存の DB テーブルや DB ビューの各レコードから名前と値を得て選択肢とするように指定することができます。
- ・「名前」や「名前のカラム」で指定したカラムの値は、選択肢として表示されます。「値」や「値のカラム」で指定したカラムの値は、その選択肢を選んだ時の値となります。
- ・「絞り込み指定」は他のモジュールの値にもとづいて選択肢を絞り込みたい場合に使用します。「絞り込み指定」を選択すると、「クラス」という欄が現れるので、絞り込み指定で指定したモジュールの値に応じた指定をします。
例えばプルダウンリストモジュール「SELECT2」で、「SELECT1」で選択されている値（「ほ乳類」か「は虫類」）によって選択肢を変えたい場合、次の図の例のようにします。すると、SELECT1 で「ほ乳類」が選ばれている時は SELECT2 の選択肢は「犬」「猫」となり、「は虫類」が選ばれている時は「蛇」「トカゲ」となります。



- 上記の例と同じことを、選択肢を DB テーブルから得ておこないたい場合は、次のようなテーブルを用意します。(名前を「ペットタイプ」「ペット」とします。)

ID	名前	
1	ほ乳類	✕
2	は虫類	✕

ID	タイプ	名前	
1	1	犬	✕
2	1	猫	✕
3	2	蛇	✕
4	2	トカゲ	✕

- そして、プルダウンリストモジュール「SELECT1」と「SELECT2」の options 属性を次のように設定します。

SELECT1 選択肢編集

手入力 DB

DB

絞り込み指定: [使用しない] ▼

DBテーブル/ビュー: ペットタイプ ▼

名前のカラム: 名前 ▼

値のカラム: ID ▼

最大件数: 1000

OK

SELECT2 選択肢編集

手入力 DB

DB

絞り込み指定: SELECT1 ▼

DBテーブル/ビュー: ペット ▼

名前のカラム: 名前 ▼

値のカラム: ID ▼

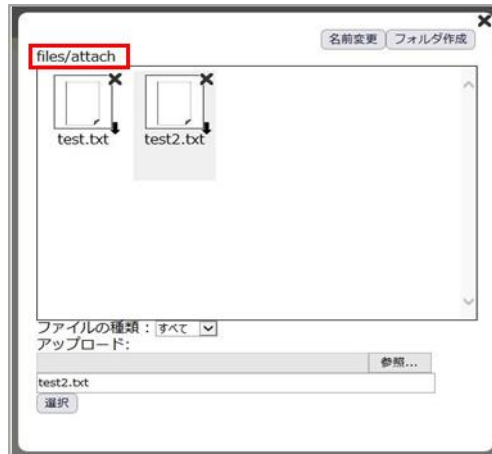
絞り込みのカラム: タイプ ▼

最大件数: 1000

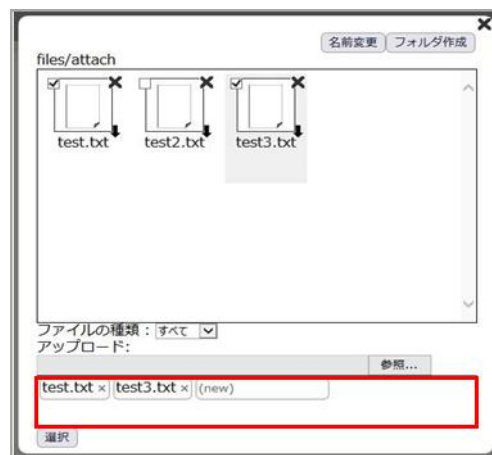
OK

【17.02.10. ファイル型カラムでのデータインプット】

- ファイル型カラムに対応したデータインプットモジュールをスクリーンに配置すると、ファイルをアップロードしてサーバーに保存することができます。アプリで 사용되는各種のファイルを納めるための大元のディレクトリとしてあらかじめ用意されている「files」というディレクトリがあり、その中にカラムの「ルートディレクトリ」属性で指定した名前のディレクトリが作られて、その中にアップロードされたファイルが保存されます。(後述の【23. ファイル管理】参照)
- ファイル型カラムに対応したデータインプットモジュールは、見かけはテキストボックスですが、クリックすると次のようなファイル選択ダイアログが開きます。



- 上の画面例では、対応するカラムの「ルートディレクトリ」に「attach」が指定されています。それにより、現在のディレクトリとして「files/attach」と表示されています。その下の領域にファイルをドラッグアンドドロップすることでファイルをアップロードできます。上の例では「test.txt」「test2.txt」の二つのファイルがアップロードされています。
- ファイルのアイコンの「x」で削除、「↓」でダウンロードができます。右上のメニューの「名前変更」でファイル名の変更、「フォルダ作成」でサブディレクトリの作成ができます。
- 「ファイルの種類」は、対応するカラムの型の詳細が「フリー」であれば「すべて」、「PDF」などの種類が指定されていればそれになります。「PDF」の場合であれば、拡張子が「.pdf」のファイルだけが表示されます。
- アップロードされているファイルのアイコンをクリックすると、そのファイル名がダイアログ下部のテキストボックスに入ります。「選択」ボタンをクリックすると、それがデータ入力モジュールの値としてセットされます。
- 対応するカラムに配列属性がセットされている場合は、複数のファイルを選択できます。



- ・複数のファイルが選択できる場合、各ファイルのアイコンにチェックボックスが表示され、選択されているファイルにチェックが入ります。上の画面例は「test.txt」「test3.txt」の二つのファイルが選択された状態です。

【17.03. スクリプト】

- ※ スクリーンのスクリプトについては、マニュアル「プログラミングガイド」をご参照ください。スクリプトから利用できる機能（api）についてはマニュアル「APIリファレンス」をあわせて参照ください。

【17.04. スクリプト ～コード挿入と名前挿入】

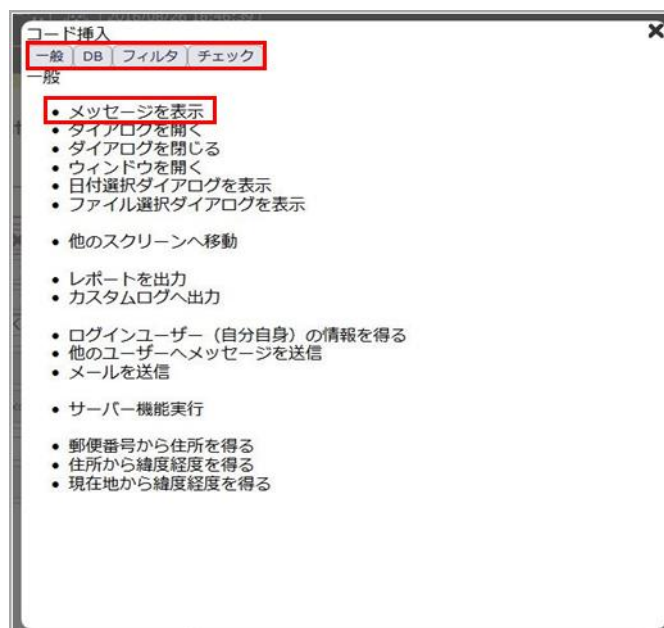


```

1
2 module.exports = function(api){
3   var actions = {
4   };
5   var events = {
6     onLoad: function(){
7
8     },
9     onUnload: function(){
10

```

- ・スクリプトの左上には「コード挿入」「名前挿入」「検索・置換」のボタンがあります。「コード挿入」「名前挿入」は、モジュールの「イベント」と「アクション」の「挿入」ボタンとともにスクリプトの入力を補助するためのものです。「検索・置換」は、このスクリーンのスクリプトを対象に、文字列を検索したり、検索して見つかったものを他の文字列に置換したりします。
- ・「コード挿入」では次のダイアログが開き、各種の操作をおこなうコードをカーソル位置に挿入できます。



- ・「フィルタ」はフィルタを適用するコードを挿入します。フィルタについては【19. フィルタ】を参照してください。
- ・「チェック」はある値が満たすべきパターン、例えばひらがなでなければならない時にそのチェックをおこなうコードを挿入します。
- ・例えば「メッセージを表示」をクリックすると、次のように挿入されます。挿入されたコードは選択された状態で、青く表示されます。もし誤って挿入した場合は、キーボードの「Delete」を押せば削除できます。

```

1
2 module.exports = function(api){
3   var actions = {
4
5   };
6   var events = {
7     BUTTON1_onClick: function(evt){
8       //メッセージを表示
9       var message = ""; //表示するメッセージ
10      api.dialog.message(message);
11    },
12  },
13

```

- ・「名前挿入」では次のダイアログが開き、DB テーブル、DB ビュー、それらのカラム名、スクリーン名、レポート名をカーソル位置に挿入できます。



【17.05. 式属性】

- ・文字列、テキストボックス、データインプットの各モジュールでは、「式」属性を設定することで、数式モジュールのように計算結果を表示させることができます。テキストボックス、データインプットに式属性を設定した場合は入力ができず表示専用になります。
- ・属性の「式」欄をクリックすると、数式入力のダイアログが開きます。

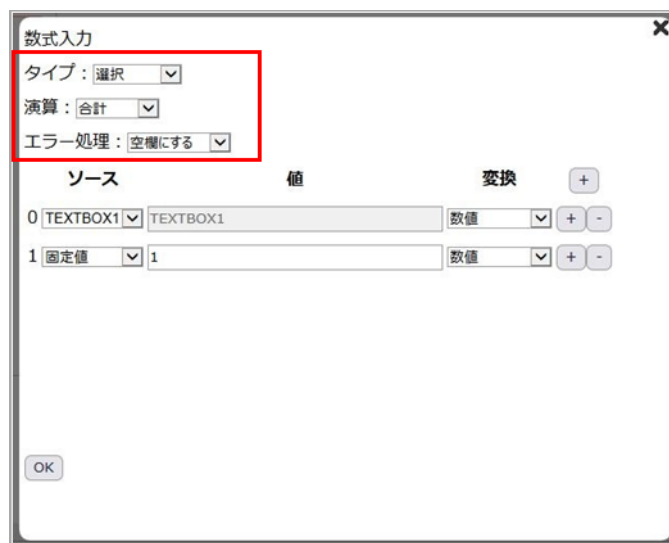


- ・「タイプ」は次から選択します。

タイプ	説明
使用しない	数式を使用しない。各モジュールの本来の機能となる
選択	ダイアログで演算やその引数を選択して数式を組み立てる
スクリプト	数式モジュールと同様にスクリプトの formulas の中に数式を記述する

- ・タイプが「**選択**」の場合、「**演算**」を選択し、その引数を指定します。
引数は最初は二つの入力欄がありますが、「+」「-」のボタンで増減できます。
引数の「ソース」は「**固定値**」か既存のモジュール名を選択します。
「**固定値**」を選んだ場合は「**値**」にその値を入力します。
「**変換**」は値をどう扱うかを指定します。

例えば TEXTBOX1 に入力された値に 1 を加えたものを表示したい場合であれば、下記の画面例のようにします。



- ・タイプが「スクリプト」の場合、「スクリプトのテンプレートを挿入」ボタンをクリックすると、スクリプトの formulas の中に数式を記述するための枠組みが挿入されます。



- ・タイプが「スクリプト」の場合、formulas の中に記述した関数は、そのスクリーンが表示される際に、スクリーンを構成する各モジュールの値が更新される毎に連鎖的に呼び出されます。(その関数の中で他のモジュールの値を参照している可能性があるため。) 従って、formulas 中の関数では他のモジュールの値を参照するのは OK ですが、モジュールの値を更新するような操作をすると無限ループに陥る危険があります。また、new Date() で現在の日時を返すなど、呼び出すたびに違う結果を返すような関数もやはり無限ループになりますので、注意してください。

【18. レポート設計】

【18.01. レポート新規作成】



- ・「Excel シート」「白紙の PDF」「PDF (単票)」「PDF (リスト)」から選択します。
- ・「Excel シート」は Excel ファイル (拡張子が .xlsx) でレポートを出力します。レポートの内容は、Excel ファイルとしてテンプレートファイルを作成し、そこにデータを埋め込むことで作成します。

- ・「白紙の PDF」「PDF (単票)」「PDF (リスト)」は PDF ファイルとしてレポートを出力します。A4 などの用紙サイズを選択した上で、そこに文字列や画像やデータを配置してレポートを作成します。「PDF (単票)」「PDF (リスト)」では、対象の DB テーブルや DB ビューと対象のカラムを選択すると、それを単票(1 ページに 1 レコード)やリスト (1 ページに複数レコード) として出力するレポートの雛形が作成されます。

※ Buddy で出力した PDF は、Chrome 内蔵の PDF ビューアーでは正しく表示できません。内蔵の PDF ビューアーを無効にするには、Chrome の右上にある、点が三つ縦に並んだアイコンをクリックして設定画面を開きます。画面下部の「詳細設定」をクリックして詳細設定を開きます。その中の「サイトの設定」をクリックします。下のほうにある「PDF ドキュメント」をクリックし、「PDF ファイルを Chrome で自動的に開く代わりにダウンロードする」をオンにします。そして Adobe Reader または Adobe Acrobat をインストールして下さい。

【18.02. レポート新規作成 ～ XLSX シート】



- ・あらかじめ作成した Excel のテンプレートファイルをアップロードします。方法は CSV の場合と同様です。目的のファイルを選択し、「次へ」をクリックします。

※ テンプレートとなる Excel ファイルをどのように作成するかは後述します。

※ 今のところ Excel ファイルは、拡張子が .xlsx の Excel2007 以降のファイル形式のもののみが扱えます。拡張子が .xls の古い形式のファイルは扱えません。

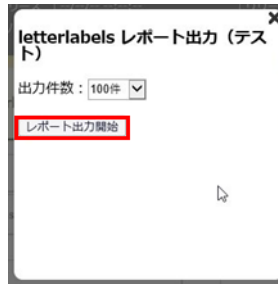
- ・画面右にある「…を一覧出力する Excel テンプレートの雛形を作成」は、DB テーブルや DB ビューを選択して「作成」ボタンをクリックすると、その DB テーブルや DB ビューを一覧出力する Excel テンプレートの雛形を作成します。

例えば「nouhinmeisai」というビューであれば、「nouhinmeisai_skelton_template.xlsx」というファイル名でテンプレートファイルが作られます。これをダウンロードして加工することで、Excel テンプレートを作成できます。

- ・「レポート名」と「出力ファイル名」を入力します。
- ・「どのシートを出力するか選択してください」では、テンプレートの Excel ファイルの中のシートの一覧が表示されますので、どのシートを出力するか選択します。
- ・「作成」で作成されて、そのレポート設計画面に移ります。
- ・「レポート表示名」は自由な名前を付けることができます。
- ・「メモ」は開発上のメモです。
- ・右上には次のメニューがあります。

メニュー	説明
保存	レポート設計を保存する
テスト出力	テスト出力をおこなう（データベースはデバッグ用が参照される）
削除	このレポートを削除する

- ・「テスト出力」では次のダイアログが開きます。出力件数を選択して「レポート出力開始」ボタンをクリックすると出力され、生成されたレポートをダウンロードできます。



【18.03. Excel テンプレートファイルの作成】

	A	B	C
1	社員名簿		
2			
3	番号	氏名	部署名
4	<code><repeat src="dbishainbusho" direction="down">[[ID]] [[name]] [[busho name]]</repeat></code>		
5			

※ 今のところエクセルファイルは、拡張子が .xlsx の Excel2007 以降のファイル形式のもののみが扱えます。拡張子が .xls の古い形式のファイルは扱えません。

- ・ レポートのテンプレートとなる Excel ファイルは、下記で示すデータ埋め込み命令の他は、通常の Excel ファイルのように作成すれば、文字やセルの書式（色、文字の修飾や配置、罫線など）はそのまま出力されます。

※ Excel の数式を含めるには特殊な書き方が必要です。（後述）

- ・ データベースのテーブルやビューのデータを埋め込むには、埋め込みをおこなう領域の左上隅のセルで、先頭に `<repeat src="dbi:テーブルまたはビュー名" direction="方向">` と書きます。方向は「down」が下への繰り返し、「right」が右への繰り返しです。そして埋め込みをおこなう領域の右下隅のセルで、末尾に `</repeat>` と書きます。repeat の src には、上述のように「dbi:テーブルまたはビュー名」と書くほか、「dbi:式」（式は、`{table: テーブルまたはビュー名, where: where 節内容, orderby: orderby 節内容}` というオブジェクトを返す Javascript の式）という指定も可能です。
- ・ 上記で作成した埋め込み領域の中で、`{{カラム名}}` と書けば、そのカラムの値が埋め込まれます。`{{式}}` と書いて Javascript の式の値を埋め込むこともできます。式の中ではカラムの値は「`this.data.カラム名`」として得ることができます。また、レポート出力実行時にパラメータを与えていれば、式の中でその値を「`this.data.パラメータ名`」として得ることができます。セルの中に`{{…}}`以外に文字列を書けばそれはそのまま出力されます。

例えばあるセルに、「`{{age}}`歳」と書けば、age カラムの値と「歳」が表示されます。

「`{{this.data.age < 6 ? '幼児': ''}}`」と書けば、age カラムの値が 6 未満の時に「幼児」と表示されます。

- ・ `{{カラム名}}` や `{{式}}` で、`{{フィルタ名:カラム名}}` や `{{フィルタ名:式}}` として、

表示する内容にフィルタを適用することができます。フィルタとは、文字列を加工する関数のことで、例えば数値に三桁ごとのカンマを挿入する **COMMA** などがあります。

{{COMMA:uriage}} とすれば、uriage カラムの値がカンマ区切りで表示されることになります。{{フィルタ 1:フィルタ 2:…}} のようにフィルタを複数適用することができます。

【入れ子の repeat ①～group による方法】

	A	B	C
1	社員名簿		
2			
3	部署名	番号	氏名
4	<repeat src="dbishainbusho" direction="down" group="this.data.bushoname">{{bushoname}} <repeat>{{ID}} {{name}}</repeat></repeat>		

	A	B	C
1	社員名簿		
2			
3	部署名	番号	氏名
4		1	鈴木一郎
5	総務課	2	田中二郎
6	製造課	3	山田太郎
7	営業1課	4	佐藤花子

- ・ <repeat …> に group="式" と書けば、その式の値が同じ連続するレコードがひとまとめになります。そして、その中に <repeat> を入れ子にすることができます。入れ子にする場合は内側の repeat では src の指定は不要です。direction も省略すると外側の repeat と同じになりますが、異なる方向を指定することも可能です。<repeat>は必ずその終了を示す </repeat> と対にならなければならないという点に注意してください。
- ・ <repeat group="…">の中では、<count> と書けばグループ化されたレコード数を、<sum exp=" 式 "> と書けば指定した式の値の合計を表示することができます。例えば「<repeat group="this.data.year">」の中で「<sum exp="this.data.uriage">」と書けば、year カラムの値が同じレコードがグループ化されて、その uriage カラムの値の合計が表示されます。
- ・ 全レコードの件数や合計を表示したい場合は、全体を <repeat group="1">…</repeat> で囲んでその中に <count> や <sum exp=" 式 "> を書きます。
- ・ <count> や <sum exp=" 式 "> にフィルタを適用したい場合は、<count filter="COMMA"> や <sum exp=" 式 " filter="COMMA"> のように書きます。複数のフィルタを適用するときはスペース区切りで指定します。

【入れ子の repeat ②～ src による方法】

- ・ 上述の group による repeat の入れ子では、一つのテーブルまたはビューに対して、同じ値が連続する部分をグループ化します (src 指定は一番外側の repeat のみ) が、異なる src 指定を持つ repeat を入れ子にすることもできます。例えば外側では部署テーブル、内側では社員テーブルを指定して、各部署ごとの社員リストを出力したいとすると、外側では「<repeat src="dbi:busho">」とし、内側では「<repeat src="dbi:{table: 'shain', where: {bushoID: this.data.ID}}">」とします。「where: {bushoID: this.data.ID}」という指定によって、shain テーブルの bushoID カラムの値が外側の部署テーブルの ID に等しいものに絞り込んでいます。

【Excel の数式の扱い】

- ・ セルに Excel の数式を書いた場合、出力される Excel では数式とはならず、数式の結果が表示されます。出力される Excel で数式となるようにしたい場合は先頭に「'」を入れて「'=…」と書きます。これによってテンプレートの Excel では文字列として扱われ、出力された Excel では数式と扱われます。
数式を書く場合、セルの参照には注意が必要です。「A1」のような絶対参照は<repeat>によってレコードが繰り返し埋め込まれることによってセル位置がずれることがないセルでないと正しく参照できません。<repeat>によるセル位置のずれが起こっても相対位置は変わらないセルであれば、Excel の indirect()関数を使って、例えば「'=indirect("RC[-3]",false)」と書けば三つ左のセルの値を参照できます。
数式の中に上述の {{…}} や<count>や<sum>を書くことができます。例えば「'=<sum exp="this.data.jisseki">/<sum exp="this.data.yosan">」と書いて jisseki の合計を yosan の合計で割った値を表示することができます。

【18.04. レポート新規作成 ～ PDF】

- ・ 白紙の PDF の場合、次のような画面になります。

レポート作成



テンプレート: 白紙のPDF

レポート名:

表示名:

出力ファイル名:

用紙: A4 向き: 縦向き → 横: 210mm 縦: 297mm

複数配置: 用紙1ページに複数のレポートページを配置する

横配置数 1 縦配置数 1 横間隔 0 縦間隔 0

上余白 0 下余白 0 左余白 0 右余白 0

レポートサイズ: 横 210mm 縦 297mm

- ・「レポート名」と「出力ファイル名」を入力します。
- ・「用紙」と「向き」を選択すると、「横」と「縦」のサイズが表示されます。選択肢のない用紙の場合は、横と縦のサイズを直接入力することができます。
- ・通常は用紙の 1 ページとレポートの 1 ページは同じサイズで一致しますが、「用紙 1 ページに複数のレポートページを配置する」のチェックを入れると、タックシールのような出力ができます。「横配置数」「縦配置数」「横間隔」「縦間隔」「上余白」「下余白」「左余白」「右余白」をそれぞれ指定すると、レポートサイズは自動的に計算されます。
- ・PDF（単票）と PDF（リスト）の場合、次のような画面になります。

レポート作成



テンプレート: PDF (単票)

レポート名:

表示名:

出力ファイル名:

データソース: shain

表示カラム: ID 名前 部署 性別

用紙: A4 向き: 縦向き → 横: 210mm 縦: 297mm

複数配置: 用紙1ページに複数のレポートページを配置する

- ・白紙の PDF の場合と違うのは、「データソース」で対象の DB テーブルか DB ビューを選択し、表示するカラムを選択することです。
- ・必要な入力をして「作成」ボタンをクリックすると作成され、レポート設計画面に移ります。

【18.05. PDF レポート設計】

レポート設計 テスト(test) 元に戻す やり直し 保存 テスト出力 削除

(A)

名前	namae	✕
部署	busho	
性別	seibetsu	

(B)

モジュール

T TEXT 画像 T 領域

文字列 データ 画像 式 領域

(C)

アイテムオプション

属性	値
名前	DATA1
タイプ	DATA
left	60mm
top	10mm
width	50mm
height	10mm
z-index	0
コンテンツ	ID
フィルタ	
配置	
余白	
フォント	
フォントサイズ	
フォント色	
ボーダー表示	<input type="checkbox"/> 上 <input type="checkbox"/> 下 <input type="checkbox"/> 左 <input type="checkbox"/> 右
ボーダースタイル	
ボーダー幅	
ボーダー色	
出力	<input type="checkbox"/> テストのみ

(D)

領域オプション

属性	値
領域名	ページ
領域タイプ	自由配置
データソース	dbi:shain
left	0mm
top	0mm
領域幅	210mm
領域高	297mm
繰り返し方向	DOWN
繰り返しの高さ	
繰り返しの間隔	0
1頁あたりの繰回数	1

(E)

レポートオプション

属性	値
レポート名	test
表示名	テスト
タイプ	PAGE
データソース	
出力ファイル名	test.pdf
用紙幅	210mm
用紙高	297mm
レポート幅	210mm
レポート高	297mm
フォント	mincho
フォントサイズ	12pt
フォント色	#000000
メモ	

- ・自由配置 PDF のレポート設計画面には、三つの部分があります。
左側にはレポートの用紙 1 ページに対応する領域 (A) があり、ここに構成要素 (モジュール) を配置していきます。
右側上には、モジュールの一覧があります (B)。その下には、アイテムオプション

(©)、領域オプション (D)、レポートオプション (E) の表があり、それぞれの設定ができます。

- ・レポートは、領域 (A) の中に文字列、データ、画像、式を配置していくことで作成します。

自由配置 PDF では、最初にページ全体を覆う自由配置領域が配置されており、右上のモジュールからドラッグアンドドロップすることで配置していきます。

- ・モジュールは次の機能を持っています。

モジュール	説明
文字列	文字列を表示
データ	そのモジュールを含む領域に指定したデータソース (DB テーブルや DB ビュー) の指定のカラムの値を表示
画像	画像を表示 (JPEG または PNG)
式	Javascript の値を表示し、式の中でカラムの値を参照できる
領域	領域を入れ子にする

※今のところ PDF のレポートに表示できる画像は JPEG と PNG のみです。また、インターレース PNG は表示できません。

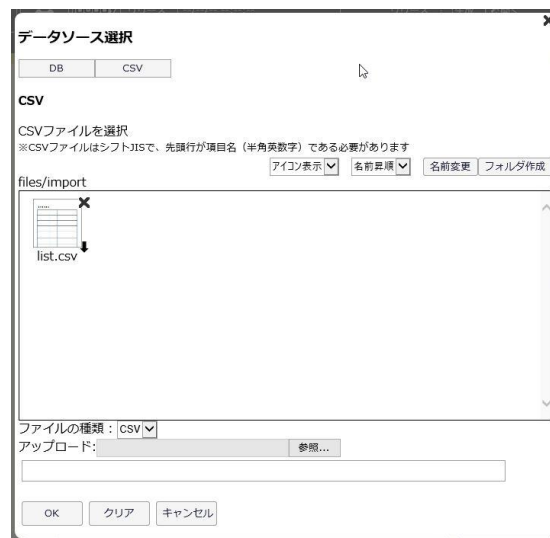
- ・領域には、次の領域タイプがあります。

タイプ	説明
自由配置	領域内の自由な位置にモジュールを配置できる
横配置	右へ右へとモジュールが配置される
縦配置	下へ下へとモジュールが配置される
テーブル	項目名などを示すためのヘッダ行と繰り返されるデータ行からなり、行内は横配置

※ 領域タイプは、領域を配置したあとの最初におこなってください。内部にモジュールを配置したあとで領域タイプを変更すると、不正常的な状態となる場合があります。(例外として、横配置と縦配置の相互の変更はモジュール配置後でも OK です。)

※ 今のところ、内部に別の領域を入れ子にできるのは、自由配置領域のみです。

- ・領域はデータソースのレコード毎に繰り返して表示することができます。このためには、領域オプションの「データソース」を指定します。データソースの欄をクリックすると次のダイアログが開きます。



データソースにはDB テーブルまたはDB ビュー、あるいはCSV ファイルを指定することができます。ダイアログ上部の「DB」「CSV」で切り替えます。「DB」の場合はテーブルまたはビューを選択します。「条件」欄は後述する入れ子の領域で使用します。「CSV」の場合はアップロードしたCSV ファイルを選択します。CSV ファイルは先頭行が半角英数字の項目名になっている必要があります。

- ・「データソース」を指定した領域の中では、データモジュールを配置したり、式モジュールの中でカラムの値を参照できます。データモジュールの場合は、アイテムオプションの「コンテンツ」にカラム名を指定します。式モジュールの場合、「コンテンツ」に指定した Javascript の式の中で「this.data. カラム名」とすることでカラムの値を参照できます。
データソースが CSV の場合は先頭行の項目名がカラム名の扱いになります。

- ・「データソース」を指定した領域を入れ子にすると、例えば外側の領域では納品テーブル (nouhin) の各レコードについて繰り返し、内側の領域ではそれぞれの納品データに対応した納品明細テーブル (nouhinmeisai) のレコードについて繰り返し、というようなことができます。これをおこなうには、内側の領域のデータソースで、「テーブル・ビュー」に「nouhinmeisai」を、「条件」に「'nouhinid =' + this.data.ID」のようにします。これは、nouhin テーブルの「ID」カラムと、nouhinmeisai テーブルの「nouhinid」カラムが、それぞれ納品データ番号を表しているため、外側の nouhin テーブルのレコー

ドの「ID」の値と等しい「nouhinid」を持つ nouhinmeisai テーブルのレコードを内側で繰り返す、ということになります。

「条件」に設定する内容は、結果が PostgreSQL での条件式を表す文字列となるような Javascript の式です。

- データソースを指定して繰り返す領域は、指定したデータソースのレコード数が多い場合は、領域オプションの「1 頁あたりの繰返回数」の指定があればその件数毎に区切られて、次のページに表示されます。「1 頁あたりの繰返回数」の指定がない場合は、領域の大きさに納まる範囲で区切ります。
- テーブルタイプの領域に配置できるのは、文字列モジュールとデータモジュールのみです。モジュールをドロップすると、それに対応した項目名表示のための文字列モジュールが自動的に配置されます。テーブルの一行目が項目名表示となり、二行目以降はデータソース指定に従ってレコード毎に繰り返されます。
- アイテムオプションには次の属性があります。

アイテム	説明
名前	半角英数と「_」による名前
タイプ	モジュールのタイプ
left	モジュールの領域左端からの位置
top	モジュールの領域上端からの位置
width	モジュールの幅
height	モジュールの高さ
z-index	領域内でのアイテムの重なり of 前後関係を決める数値を指定 (数値が大きいほど手前に表示、省略すると 0)
コンテンツ	表示する内容 (データの場合はカラム名、式の場合は式、画像の場合は画像ファイルパス名)
フィルタ	適用するフィルタ名をカンマ区切りで指定
配置	表示する文字列の配置を選択。省略すると両端寄せ (末尾行は左寄せ、それ以外は両端寄せ)
余白	周囲の余白
フォント	フォントを選択
フォントサイズ	フォントのサイズ (ポイント) フォント
色	文字色
ボーダー表示	枠表示指定 (上下左右の表示したいものにチェック)
ボーダースタイル	実線または破線 1~4 (破線は 1 が細かく、2、3、4 と粗くなる)
ボーダー幅	枠線幅
ボーダー色	枠線色
画像表示	データモジュールでカラムが画像の場合に画像として表示するか
伸縮	画像の場合に、width と height に画像をどうあわせるかを指定
出力	「テストのみ」のチェックを入れると、設計画面のテスト出力では表示されるが、アプリでの出力では表示されない

※ ボーダー関係の 4 つの属性は全て指定しないと枠線は表示されません。

※ ボーダースタイルの破線 1 ~ 4 は画面上では同じ破線として表示されます。実際の PDF 出力でどのような表示になるかは、テスト出力で確認してください。

- ・領域オプションには次の属性があります。

領域	説明
領域名	半角英数と「_」による名前
領域タイプ	タイプを選択
データソース	データソース
left	領域の外側領域左端からの位置
top	領域の外側領域上端からの位置
領域幅	領域の幅
領域高	領域の幅（高さ）
繰返し方向	データソースを指定した場合の繰返し方向 （現在は DOWN のみ）
繰返しの高さ	（将来の使用のための属性）
繰返しの間隔	（将来の使用のための属性）
1 頁あたりの繰返回数	繰返しを何件ずつで区切るか

- ・レポートオプションには次の属性があります。

レポート	説明
レポート名	新規作成時に指定したレポート名
表示名	自由な名前を付けることができる
タイプ	今のところ PAGE に固定
データソース	新規作成時に指定したデータソース
ページ幅	用紙の幅
ページ高	用紙の高さ
フォント	デフォルトのフォント
フォントサイズ	デフォルトのフォントサイズ（ポイント）
フォント色	デフォルトの文字色（#RRGGBB）
メモ	開発上のメモ

- ※ 「用紙 1 ページに複数のレポートページを配置する」のチェックを入れた場合、次の情報も表示されます

：レポート幅、レポート高、横配置数、縦配置数、横間隔、縦間隔、左余白、上余白

- ※ 位置や大きさの属性は、「10mm」のように「mm」を付けて指定してください。
フォントサイズ（ポイント）は数値のみで指定します。

【位置ガイドとしての背景画像】

- ・上記のアイテムオプションの「出力」の「テストのみ」と「z-index」を使うと、設計時の位置ガイドとして背景に画像を置くことができます。出力する用紙にあらかじめ罫線などが印刷されていてそれに合わせた位置に文字列やデータを表示したい場合、それぞれの位置（用紙左端から何 mm、上端から何 mm）を定規で測って left と top に設定すれば良いのですが、項目が多いと面倒です。その用紙のスキャン画像を作って、left と top は 0、width と height は用紙サイズに合わせ、z-index を -1、出力を「テストのみ」にして配置します。テスト出力で印刷してみて、実際の用紙とのずれがあれば、left や topなどを調節してください。その背景画像を位置ガイドとして、実際に出力する項目を配置すれば良いわけです。

【18.06. レポートの使用】

- 作成したレポートをアプリの中で使用するには、スクリーンのスクリプトの中で、`api.outputDialog.open(レポート名 , オプション)`としてレポート出力ダイアログを開き、その中でレポート出力をおこないます。オプションには必要に応じて次のプロパティを持つオブジェクトを指定します。

	説明
<code>where</code>	絞り込み条件
<code>order</code>	並べ替え指定
<code>offset</code>	何件目から出力するか（先頭が0）
<code>num</code>	何件出力するか

- 上記以外のプロパティも指定することができ、レポート中の式で「`this.param. プロパティ名`」で参照できます。
- `jsdata` プロパティでデータを与えると、データベースでなくそのデータ内容を出力することができます（次項）。

【18.07. スクリプトで与えたデータの出力】

- 通常は `src` やデータソースに「`dbi:テーブル名`」のように指定してデータベースから読み取ったデータをレポート出力しますが、次のようにしてスクリプトで組み立てたデータを与えて出力することができます。
- 上述の、レポート出力ダイアログを開くときのオプションで、`jsdata` プロパティとして例えば次のような形で出力したいデータを指定します。

```
jsdata: {
  shain: [
    { name: '田中', nenrei: 34, ... },
    { name: '鈴木', nenrei: 45, ... },
    ...
  ],
  kokyaku: [
    ...
  ]
}
```

- この例では、「shain」「kokyaku」がテーブル名にあたり、「name」「nenrei」がカラム名にあたります。

- そして、レポートの `src` やデータソースで「`jsdata:shain`」や「`jsdata:kokyaku`」を指定すると、上記で与えたデータを出力することができます。
- 「`jsdata:...`」の `src` やデータソースに対しては、`where`、`order`、`offset`、`num` のオプションは指定しても効果はありません。スクリプトで与えたデータの全体がそのままの順序で出力されます。

【19. フィルタ】

- フィルタは、引数を受け取って、それを加工した結果の文字列を一つ返す Javascript の関数です。上記のようにレポートで使用する他、スクリーンのスクリプトでも使用することができます。
- フィルタは下記のような名前指定しますが、ハイフンで区切ってオプションパラメータを付けることができます。(例：`YMD-h-1`)
- あらかじめ用意されているフィルタには次のものがあります。

<一般>

フィルタ	説明
TRIM	前後の半角スペースやタブを取り除く。 オプションパラメータで数字を指定するとその文字数で切る。

<数値>

フィルタ	説明
COMMA	桁区切りのカンマを挿入する。
YEN	¥表記する（桁区切りのカンマ挿入もおこなう）。

<日時関係>

フィルタ	説明
YMD	日付 (Date 型または Date 型に変換できる文字列) を yyyy/mm/dd にする。オプションパラメータ「h」を付けると区切りが「/」でなく「-」になる。オプションパラメータ「1」を付けると0を付けない。
JYMD	日付 (Date 型または Date 型に変換できる文字列) を yyyy 年 mm 月 dd 日 にする。オプションパラメータ「1」を付けると0を付けない。
GYMD	日付 (Date 型または Date 型に変換できる文字列) を 元号 y 年 m 月 d 日 にする。
YMDHMS	日時 (Date 型または Date 型に変換できる文字列) を yyyy/mm/ddhh:mm:ss にする。オプションパラメータ「h」を付けると区切りが「/」でなく「-」になる。オプションパラメータ「1」を付けると0を付けない。
JYMDHMS	日時 (Date 型または Date 型に変換できる文字列) を yyyy 年 mm 月 dd 日 hh 時 mm 分 ss 秒にする。オプションパラメータ「1」を付けると0を付けない。
HMS	時刻 (Date 型または文字列) を hh:mm:ss にする。オプションパラメータ「1」を付けると0を付けない。
AGE	日付 (Date 型または Date 型に変換できる文字列) を誕生日として現在の満年齢にする。オプションパラメータ「d」を付けると日単位計算 (誕生日前日に年を取る) となる。

※ 日付関係のフィルタでは、「元号 y 年 m 月 d 日」の形式の文字列も受け付けるようになっていきますので、元号表記と西暦表記の相互変換に使用することができます。

※ 日付関係のフィルタでは、ISO8601 の「2016-06-01T10:20:30.456Z」の形式 (タイムゾーンは Z の他「+0900」「+09:00」も OK) にも対応しています。

<文字種変換>

フィルタ	説明
HIRA2KATA	ひらがなをカタカナにする
KATA2HIRA	カタカナをひらがなにする
ZEN2HAN	全角英数記号を半角にする
HAN2ZEN	半角英数記号を全角にする
ZEN2HANKATA	全角カナを半角カナにする
HAN2ZENKATA	半角カナを全角カナにする

・ユーザーが独自のフィルタを追加できる仕組みを用意する予定です。

【20. サーバー機能設計】

・アプリのサーバー側で動作するスクリプトを設計します。スクリーンのスクリプトはアプリのクライアント側、つまりブラウザ上で動作しますが、サーバー側で動作するスクリプトが必要な時は「サーバー機能設計」でおこないます。

※ 体験版ではサーバー機能設計は使用できません。

- ・「新規作成」をクリックすると、作成するサーバー機能のテンプレートを選択する画面になります。



- ・「白紙のサーバー機能」をクリックすると、次の画面になります。

サーバー機能作成

白紙のサーバー機能

テンプレート： 白紙のサーバー機能

機能名：

表示名：

- ・半角英数字の「機能名」と「表示名」（省略可）を入力して「作成」をクリックすると作成されてその編集画面になります。

test (B) 保存 ↑ テスト実行 削除

検索・置換

オプション	
属性	値
機能名	test
表示名	
実行タイプ	同期実行 ▼
実行ユーザー	ユーザー ▼
権限設定	
タイムアウト	5000
テスト実行時のoptions	

(C)

- ・サーバー機能の編集画面には、左側にスクリプトの編集画面 (A)、右上にメニュー (B)、右側にオプション (C) があります。
- ・スクリプトの編集画面は、上部の「**検索・置換**」ボタン、その右の検索窓、スクリプト領域、下部の▼▲ボタンがあります。検索窓に入力してその右のボタンをクリックするとスクリプトを検索できるほか、「**検索・置換**」ボタンをクリックするとダイアログが開いて検索と置換が行えます。▼ボタンはスクリプト領域を広げ、▲ボタンは縮めます。
- ・サーバー機能編集画面のメニューは次のものがあります。

メニュー	説明
保存	編集内容を保存する
テスト実行	テストのために実行する
削除	このサーバー機能を削除する

- ・サーバー機能のオプションには次の属性項目があります。

項目	説明
機能名	作成時に指定した機能名
表示名	作成時に指定した表示名
実行タイプ	「同期実行」か「非同期実行」かを選択する
実行ユーザー	「ユーザー」か「system」かを選択する
権限設定	このサーバー機能を誰が実行できるかの権限設定をおこなう
タイムアウト	実行時のタイムアウトをミリ秒で指定する
テスト実行時の options	テスト実行の際にスクリプトに渡す options の内容を指定する

- ・実行タイプによってスクリプトの実行のされ方が異なります。
「同期実行」の時は、スクリプトの実行が終わるまで待って、その時点の変数 `result` の内容が下記のコールバック関数を通して返されます。
「非同期実行」の時は、スクリプトの中であらかじめ用意されている `callback` という関数を呼び出すとサーバー機能の実行は終わり、その引数（第一引数がエラー、第二引数が結果）によって結果が返ります。
いずれの場合もタイムアウトで指定した時間内に結果が返らないとエラーになります。
 - ・実行ユーザーは、サーバー上でこのスクリプトが実行される時に呼び出したユーザーの権限で実行されるのか、「system」というユーザーの権限で実行されるのか、を設定するものです。
- ※ サーバー機能のスクリプトの書き方、その中で使える機能、スクリーンスクリプトからのサーバー機能の呼び出し方などについては、マニュアル「**プログラミングガイド**」をご参照ください。

【21. ユーザー権限管理】

ユーザー権限管理

アプリケーションの開発権限

アプリケーションの作者：admin

	許可ユーザー	拒否ユーザー	許可グループ	拒否グループ	優先
運用管理	admin				許可 ▾
編集	admin				許可 ▾
削除	admin				許可 ▾
閲覧	admin				許可 ▾

アプリケーションの利用権限

ユーザー管理をBuddyから分離(個別のログインが必要)

	許可ユーザー	拒否ユーザー	許可グループ	拒否グループ	優先
デバッグ用編集			all		許可 ▾
デバッグ用閲覧			all		許可 ▾
編集			all		許可 ▾
閲覧			all		許可 ▾

- ・アプリケーションの運用・開発に関する権限と、利用に関する権限の設定ができます。アプリケーションの利用権限は、デバッグ用アプリと本番用アプリに分けて設定します。

※ 試用版ではユーザー権限管理は使用できません。

※ 権限を変更した場合、その権限設定をデータベースにも反映させるにはアプリを生成する必要がありますのでご注意ください。

- ・権限設定の各列の働きは次のようになります。

	説 明
許可ユーザー	指定されたユーザーは、その権限を許可される
拒否ユーザー	指定されたユーザーは、その権限を拒否される
許可グループ	指定されたグループに属するユーザーは、その権限を許可される
拒否グループ	指定されたグループに属するユーザーは、その権限を拒否される

※ 許可ユーザーか拒否ユーザーに該当した場合は、それによって（グループに関係なく）権限が決まります。

※ 許可ユーザーと拒否ユーザーの両方に該当した場合や、許可グループと拒否グループの両方に該当した場合は、「優先」の指定によって権限が決まります

- ・「ユーザー管理を Buddy から分離（個別のログインが必要）」のチェックを入れると全体のユーザー管理にしたがったシングルサインオンではなく、このアプリ独自のユーザー管理にもとづいてログインすることになります。

ユーザー管理をBuddyから分離(個別のログインが必要)

ゲストアカウント:

ユーザーリスト

ユーザーID	表示名	パスワード	状態	メール	メモ	
<input type="text"/>	<input type="text"/>	<input type="text"/>	利用可能	<input type="text"/>	<input type="text"/>	追加

名前	メンバー	
all		
<input type="text"/>		追加

- ・「ユーザー管理を Buddy から分離（個別のログインが必要）」のチェックを入れるとユーザーとグループのリストが表示されますので、入力します。方法は全体のユーザー管理と同様です。
- ・ユーザーリストにユーザーを追加した上で、「ゲストアカウント」を「(無効)」でなくどれかのユーザーを選択すると、アプリの URL を開いた時にログイン不要となり、そのゲストアカウントのユーザーでログインした扱いとなります。これにより、ログインの不要なオープンなアプリを作成できます。

【22. スクリーンテーマ編集】

- ・スクリーンテーマは、スクリーンの要素にあらかじめ設定されたクラス名や、スクリーンに配置したモジュールの「クラス」や「部分クラス」で指定した名前に適用するスタイルをまとめて指定する機能です。アプリ全体のスタイルを統一するのに利用できます。
- ・右上 (㊦) にはスクリーンテーマ全体を操作するための次のアイコンがあります。

メニュー	説明
読み込み	「ダウンロード」で保存しておいたスクリーンテーマを読み込む
ダウンロード	現在のスクリーンテーマをファイルとして保存する
保存	編集中のスクリーンテーマを保存する（編集後は必ず保存してください）
アイテムのクラス属性での指定をItemXXXより優先する	<p><ItemLabelなどのItemXXXクラスはモジュールの種類に応じて自動的に割り当てられるクラス></p> <p>アイテムのクラス属性で指定したDATA_LABELなどのクラスと両方を指定した場合、クラス属性の方を優先して指定する</p> <p>チェックを入れないとItemXXXの方が優先される</p>

スクリーンテーマ編集

↑読み込み ↓ダウンロード 保存

アイテムのクラス属性での指定をItemXXXより優先する

クラス定義 (A)

DATA_LABEL ✕
DATA_SELECTOR ✕
DATA_VALUE ✕
ItemLabel ✕
TABLE_DATA ✕
TABLE_HEADER ✕
new

DATA_LABEL の編集 (B)

文字 ▼ 日本語 ▼

属性	値
文字色	
文字のサイズ	12
文字の太さ	
フォント	
文字スタイル	▼
行揃え	▼
文字装飾	▼
文字間隔	
行の高さ	20px
折り返し方法	▼
改行の方法	▼
空白の表示	▼
文字の影	

クラスサンプル (C)

labelサンプル

buttonサンプル

textboxサンプル

- スクリーンテーマ編集の画面には三つの部分があります。
左の「**クラス定義**」(A)はこのスクリーンテーマでスタイルを指定するクラス名のリストです。
真ん中の「**○○○の編集**」(B)は「**クラス定義**」のリストで選択したクラス名に適用されるスタイルの内容で、編集できます。
右の「**クラスサンプル**」または「**スクリーンサンプル**」(C)は編集中のスタイルが実際にどのような表示になるかを確認するためのものです。
- 「**クラス定義**」(A)のクラス名リストは、クラス名をクリックするとそのスタイル編集ができます。削除する場合は「✕」をクリックします。「new」をクリックすると直接入力かプルダウンリストからの選択で追加ができます。
次のクラス名があらかじめ用意されていますが、直接入力で独自のクラス名を追加することもできます。

クラス名	説明
APP_で始まる名前	アプリ全体のパーツに対するクラス名(後述)
DATA_LABEL などの大文字の名前	登録・編集や検索などのテンプレートで作成したスクリーンで設定されているクラス名
Item で始まる名前	モジュールの種類に応じて内部的に付けられているクラス名

※ 今のところ、テーマに設定されているクラス名とスクリーンのモジュールで設定されているクラス名が対応しているかのチェックなどはされません。ユーザーの責任で管理する必要があります。

- アプリ全体のパーツには次のクラス名が付けられています。

クラス名	説明
APP_BODY	アプリのHTMLでのBODY ※デフォルトのカーキ色部分の色はこの背景色で指定可
APP_BACKGROUND	各スクリーンの背景
APP_HEDER	ヘッダ全体
APP_HEDER_TITLE	ヘッダのアプリアイコンとアプリ名部分
APP_HEDER_NAME	ヘッダのアプリ名部分
APP_HEDER_MENU_BUTTONS	ヘッダの右側のアイコン部分

※ 例えば APP_HEADER の背景色を設定すると、現在は黒になっているヘッダの色を変えられます。

The screenshot shows the default interface of the 'BUDDY' app. The header is black with the 'BUDDY' logo and '家計簿' (Home Ledger) text. The main content area has a yellow background. It includes a date selection form (Year: 2017, Month: 1, Day: 19) and a user login field (infolab). Below are two tables: '明細#1: 収入明細' (Income Details) and '明細#2: 支出明細' (Expense Details).

<デフォルト画面>



The screenshot shows the customized interface of the 'BUDDY' app. The header is purple with the 'BUDDY' logo and '家計簿' (Home Ledger) text. The main content area has a light purple background. The layout is identical to the default screen, but the color scheme is changed to purple.

<カスタマイズ画面>

設定箇所

- APP_BODY
- APP_BACKGROUND
- APP_HEADER

- ・「〇〇〇の編集」(ⓑ)の部分では選択したクラス名に対するスタイルを編集します。
「すべて」「文字」などのプルダウンリストは編集対象のスタイル項目を絞り込みます。「日本語」「英語」のプルダウンリストはスタイル項目名の表記を変更します。
英語表記にすると CSS との対応がわかりやすくなります。
- ・「クラスサンプル」「スクリーンサンプル」(ⓒ)のプルダウンリストは、その下のサンプル表示を切り替えます。
「クラスサンプル」は現在編集中のスタイルを、ラベル、ボタン、テキストボックスにそれぞれ適用したらどう表示されるかを示します。(実際にスクリーンでどのモジュールに適用されているか、とは無関係ですので注意してください)
「スクリーンサンプル」では、実際のスクリーンに近いイメージで現在編集中のテーマがどう表示されるかを示します。(すべてのクラス名に対応するわけではありません)
サンプルで確認できないスタイルについては、スクリーンテーマを保存してからアプリを生成し、そのアプリを開いて確認する必要があります。
- ・「ダウンロード」と「読み込み」を使うと、スクリーンテーマをファイルとして保存し、他のアプリに適用するなど、いくつかのテーマを簡単に切り替えて利用することができます。

【23. スタイルシート管理】



- ・そのアプリで使用するスタイルシート (CSS ファイル) の管理をおこないます。
 - ・最初は「./stylesheets/default.css」のみが表示されています。これは Buddy のシステムで用意されているものです。
 - ・「追加」でユーザーが用意した CSS ファイルをアップロードすることができます。
 - ・「X」をクリックすると、削除することができます。(そのアプリで使われなくなるだけで、css ファイルそのものが削除されるわけではありません。)
- ※ 画面を構成する各要素にどのようにクラス名などが当てられているか、の資料は別途用意する予定です。

【24. ファイル管理】



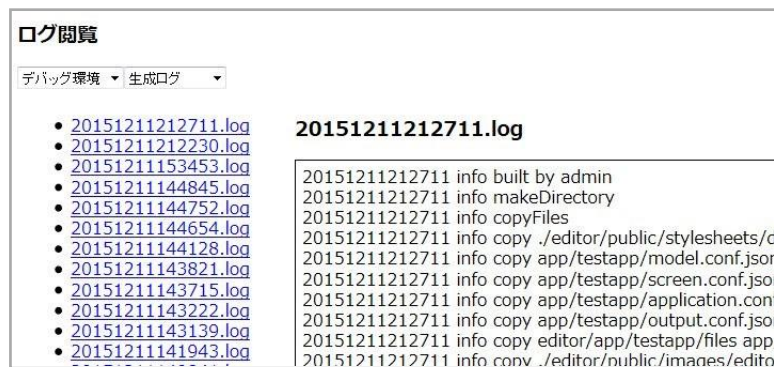
- そのアプリで使用するファイルを管理します。
- 次のフォルダはあらかじめ用意されています。(①)

フォルダ名	説明
images	スクリーンやレポートのモジュールで使用する画像を格納
import	DB テーブルの作成やデータインポートに使用する CSV ファイルを格納
javascripts	アプリで利用する Javascript のライブラリファイルを格納
outputs	レポートの出力で作成されるファイルを格納
stylesheet	スタイルシート (CSS) ファイルを格納
templates	XLSX レポートの Excel テンプレートファイルを格納
theme	スクリーンテーマファイルを格納
users	ユーザー毎にフォルダを用意する場合のためのフォルダ

- ファイル管理で操作するこれらのフォルダやファイルは、当該アプリの files というフォルダ内にあります。files フォルダは、設計情報、デバッグ環境、本番環境、のそれぞれに存在します。「操作対象」(②) で、どの files の中を閲覧・操作するか選択します。
- アプリをデバッグ用に生成すると、設計情報の files フォルダは、デバッグ環境の files フォルダにコピーされます。デバッグ環境の files フォルダは、/debug/ アプリ名 /files という URL に相当します。例えば images フォルダ内にある sample.jpg というファイルは、/debug/ アプリ名 /files/images/sample.jpg という URL になります。
- アプリを本番用にリリースすると、設計情報の files フォルダは、本番用アプリの files フォルダにコピーされます。本番用アプリでの URL は、/app/ アプリ名/files/… となります。
- データベースのファイル型カラムに入れたファイルは、デフォルトでは attachments というフォルダに入ります。この attachments フォルダはデータベースと対応するものなので、デバッグ環境と本番環境にのみ作られ、設計情報の files にはありません。

- ・設計画面でアップロードする各種のファイル以外にも、そのアプリで必要となるファイルがあれば、独自のフォルダを作成するなどしてアップロードすることができます。例えばヘルプを PDF ファイルで用意したら、「help」フォルダを作ってその中にに入れておき、その URL を開くボタンをスクリーンに配置する、といった使い方ができます。
- ・フォルダアイコンをクリックするとフォルダの中に入ったり、名前が「..」のフォルダアイコンをクリックすると親のフォルダへ戻ることができます。
- ・ファイルのアップロードは、フォルダとファイルのアイコンが並んでいる領域へファイルをドラッグアンドドロップするか、「参照」ボタンをクリックしてファイルを選択しておこないます。
- ・各フォルダやファイルのアイコン右上の「×」で削除できます。ファイルアイコンの右下の「↓」でダウンロードできます。
- ・対象のフォルダやファイルをクリックして選択した上で、右上の「名前変更」で名前を変更できます。
- ・右上の「フォルダ作成」で新しいフォルダを作成できます。

【25. アプリログ閲覧】



- ・アプリの動作に関するログを閲覧します。
- ・「デバッグ環境」「本番環境」を選び、「アクセスログ」「エラーログ」「API ログ」「生成ログ」「カスタムログ」を選ぶと、ログファイルの一覧が表示されます。閲覧したいログファイルをクリックすると、右側に内容が表示されます。

【26. アプリ文書】



アプリ文書

メモ

本アプリを使用するには...

保存

- ・「メモ」「マニュアル」を選んで入力し、「保存」ボタンをクリックすると保存されます。
- ・「メモ」は開発者や管理者向けの説明のために使用してください。
- ・「マニュアル」は、将来はユーザー向けのマニュアルとしてアプリ画面から表示できるようにする予定ですが、現在はまだ機能しません。