

スマートフォン開発ガイドライン

Ver.7.2

2013/4/5 第4版

<< 変更履歴 >>

変更年月日	変更内容
2011/08/12	初版
2011/11/01	第1版 「3.4.1 IM-Mobile Framework 画面共通IMARTタグ一覧」において、spDatePickerおよびspCalendarに関する記述を追記しました。 「3.4.4 日付と時刻の入力について」において、spDatePickerタグに関する記述を追記しました。 「4 各工程におけるポイント」を追加しました。
2012/10/31	第2版 バージョンを7.2.1に変更しました。 「3.3.1 IM-Mobile Framework 共通画面一覧」にユーザ共通検索画面を追記しました。 「3.3.3 ユーザ共通検索画面」を追加しました。 「3.4.1 IM-Mobile Framework 画面共通IMARTタグ一覧」において、spCollapsibleに関する記述を追加しました。 「3.5 IM-Mobile Framework 画面共通 JavaEE タグの使い方」を追加しました。 「3.6.3 data-rel="back"時の画面遷移エラーについて」を追加しました。
2013/2/8	第3版 コピーライトを修正しました。
2013/4/5	第4版 「3.3.3.2 初期状態・検索の絞り込み条件の設定」に指定可能なプロパティを追記しました。 「3.3.3.2.1 検索画面タブ Plugin 指定」にプロパティの使用方法を追記しました。

要件は予告なく変更される場合があります。

<< 目次 >>

1	はじめに.....	4
1.1	本書の目的.....	4
1.2	概要.....	5
1.2.1	スマートフォン向けWebサイト開発.....	5
1.2.2	IM-Mobile Frameworkについて.....	5
2	スマートフォン向けサイト制作方針.....	6
2.1	基本概念.....	6
2.1.1	想定する適用範囲.....	6
2.1.2	コンテンツ制作における基本的な考え方.....	6
2.1.3	Webアプリケーションの利用シーンについて考える.....	6
2.1.4	必要最低限、かつ最適な情報量でページを構成する.....	6
2.1.5	ディスプレイのサイズ、デバイスの向きについて.....	7
2.1.6	タッチ領域のサイズ.....	8
2.1.7	使用するフォント.....	8
2.1.8	一貫性のあるユーザインタフェースにする.....	9
2.1.9	コンテンツの装飾.....	9
2.1.10	必須項目の明示.....	10
2.1.11	トランザクション処理を行うときはユーザに確認を求める.....	10
2.2	画面設計方針.....	11
2.2.1	ページの構成.....	11
2.2.2	スタイルシートの利用.....	11
2.2.3	HTML5 への対応.....	11
2.2.4	複数の内部ページを持つHTMLについて.....	12
2.2.5	スマートフォン用のコンテンツとして相応しいデザインを考える.....	12
2.2.6	ページ切り替えのエフェクト.....	13
2.2.7	一覧形式のコンテンツを表示するページのデザイン方針.....	13
2.2.8	ボタンの配置.....	14
2.2.9	入力部品とラベル.....	15
2.2.10	キー入力の頻度に配慮する.....	15
2.2.11	画面のスクロール.....	16
2.2.12	ページ間の遷移は、どこからどこに行くのかわかるようにする.....	17
2.2.13	スマートフォン用コンテンツでは相応しくないと思われる表現.....	17
2.2.14	表示する情報の優先度を考える.....	18
2.2.15	『戻るキー』について.....	18
2.2.16	応答時間と画面の設計.....	18
2.2.17	サーチフィルターの適用方針.....	18
2.2.18	スライド遷移とボタン操作イメージ.....	19
2.2.19	トグルスイッチの適用方針.....	19
3	IM-Mobile Frameworkプログラミングガイド.....	20
3.1	基本的な記法.....	20
3.1.1	基本的な構成.....	20
3.1.2	HEADタグ内部の記述.....	21
3.1.3	BODYタグ内部の記述.....	21
3.2	業務スケルトンの使い方.....	22
3.2.1	業務スケルトンのインストール.....	23
3.2.2	一覧.....	24

3.2.3	参照.....	25
3.2.4	登録.....	26
3.2.5	編集.....	27
3.3	IM-Mobile Framework共通画面の使い方.....	28
3.3.1	IM-Mobile Framework共通画面一覧.....	28
3.3.2	ポップアップメニューについて.....	30
3.3.3	ユーザ共通検索画面.....	32
3.4	IM-Mobile Framework画面共通IMARTタグの使い方.....	42
3.4.1	IM-Mobile Framework画面共通IMARTタグ一覧.....	42
3.4.2	チェックボックスとラジオボタンの実装について.....	43
3.4.3	ファイルのダウンロード用部品の実装方針.....	43
3.4.4	日付と時刻の入力について.....	46
3.5	IM-Mobile Framework画面共通JavaEEタグの使い方.....	51
3.5.1	JSPタグディレクティブ.....	51
3.5.2	IM-Mobile Framework画面共通JavaEEタグ一覧.....	51
3.6	TIPS.....	52
3.6.1	ページ遷移について.....	52
3.6.2	Debug.Browseについて.....	53
3.6.3	data-rel="back"時の画面遷移エラーについて.....	53
4	各工程におけるポイント.....	54
4.1	開発着手前(要件定義まで).....	54
4.1.1	端末の選定.....	54
4.1.2	画面イメージ.....	54
4.2	設計工程.....	55
4.2.1	画面の設計.....	55
4.2.2	ビジネスロジックの設計.....	55
4.3	試験工程.....	56
4.3.1	試験端末の選定.....	56
4.3.2	試験観点について.....	56

1 はじめに

1.1 本書の目的

本書では、IM-Mobile Framework(以下『本フレームワーク』と表記)を使用したスマートフォン向け Web サイトの制作にあたり、画面のデザインの仕方、部品の効果的な利用についてのガイドラインについて説明しています。本書に記載されている内容を理解していただくことによって、統一的なユーザインタフェースをもつスマートフォン向け Web サイトが構築していただけることを目的としています。

本フレームワークでは、本書で説明するガイドラインに沿った Web サイトを構築していただくための一助として、業務スケルトンや共通画面、各種タグを提供しています。これらの詳細な使用方法については、各種ドキュメントを参照してください。

1.2 概要

1.2.1 スマートフォン向けWebサイト開発

スマートフォン向けの Web サイトでは、PC 向けやフィーチャーフォン向けの Web サイトとは異なる表現が必要となります。例えば、テキストボックスの端に虫眼鏡のマークがついた検索ボックスや、独特なトグルスイッチを使用したコンテンツの表現が挙げられます。また、画面サイズに応じたレイアウトを考える必要もありますし、デバイスが回転することによって縦長表示から横長表示に切り替わることも念頭に置いておかなければなりません。

本フレームワークを使用することで、スマートフォン向けに最適化された Web サイトを開発することができます。また、本書に示すガイドラインに沿ってコンテンツを制作することで、統一性のあるサイトを制作することができます。本フレームワークにて提供する各部品を使用する前に、まず本書をご一読ください。

1.2.2 IM-Mobile Frameworkについて

本フレームワークでは、入力部品やページデザイン等に jQuery Mobile を導入し、統一的なデザインでスマートフォン向け Web サイトを構築することができるようになっています。本フレームワークの提供機能を下表に示します。

また、jQuery Mobile で提供されている機能をそのまま使用することもできるようになっていますので、本書と併せて jQuery Mobile のリファレンスもご参照ください。

項番	機能	概要
1	業務スケルトン	ウィザードによる対話形式で、スマートフォン向け Web コンテンツのスケルトンを作成することができます。
2	共通画面	ログイン等の定型的な画面をスマートフォン向けに提供しています。
3	タグライブラリ	スマートフォン向け Web コンテンツを制作するためのタグライブラリです。

■jQuery Mobile

<http://jquerymobile.com/>

2 スマートフォン向けサイト制作方針

2.1 基本概念

2.1.1 想定する適用範囲

本フレームワーク、および本書では、生産性・操作性を重視した業務システムに対して適用することを想定しています。統一的なユーザインタフェース、画面構成、サイト構成によって、ユーザがドキュメントを見なくとも直観的にコンテンツを操作できることを目指します。グラフィック装飾やフォント装飾、アニメーションやエフェクトは過度になりすぎず、適度なレベルでサイト制作を行ってください。このため、本フレームワーク、および本書では、ゲームサイトやECサイトの構築については想定していません。

2.1.2 コンテンツ制作における基本的な考え方

スマートフォン向け Web コンテンツの制作において、どのようにすれば利便性や生産性が高まるのかは、PC 向けサイト制作の場合と大きな違いはありません。重要なのは、スマートフォン向け Web コンテンツでは表現上の制約が PC 向けサイトよりも多いことであり、コンテンツを構成する部品をどう配置し、どう構成するか、あるいは分割、組み合わせるのかについて考慮することです。

2.1.3 Webアプリケーションの利用シーンについて考える

コンテンツのデザインにあたっては、その Web アプリケーションがどのようなシーンで利用されるかについての検討を行ってください。例えば、太陽光がディスプレイに射しこむ外出先で利用されることや、暗い建物の中で利用されることもあるかもしれません。コンテンツのデザインを行う上では、このような利用シーンが Web アプリケーションの使い勝手にどのような影響を及ぼすかを念頭に置いてデザインを検討してください。特に、画面の配色やコントラストについてはよく検討を行ってください。

2.1.4 必要最低限、かつ最適な情報量でページを構成する

スマートフォンのユーザは電波状況が安定した場所でスマートフォンを操作するとは限りません。外出先で操作することも多く、また、電波状況が良好ではない環境で使用することも考えられます。このため、ページに掲載する情報は必要最低限に留め、かつ最適な情報量で構成するようにします。これにより、ユーザは最適なパフォーマンスと応答時間を得ることができます。また、画面遷移中は処理中メッセージ等のフィードバックを通知するようにし、システムとスマートフォンが通信を行っていることをアピールするようにしてください。

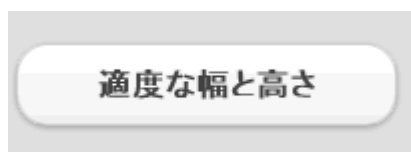
2.1.5 ディスプレイのサイズ、デバイスの向きについて

ページのサイズは、特定の解像度を意識しないようにします。jQuery Mobileは端末の解像度や縦横の向きを吸収して部品の配置を行うようになっていきます。また、端末の向きを変えたときの再配置処理についても jQuery Mobile が行います。ただし、スマートフォンは縦画面で使用する人が多いと思われるため、作成するページも基本的には縦表示で最適化するようにしてください。



2.1.6 タッチ領域のサイズ

ボタンやリンク等のタッチ領域のサイズは、最低 6mm 四方以上のタッチ領域を確保するようにしてください。また、他のボタンやリンクと密接に並べすぎずに、適度な間隔を空けて配置するようにしてください。本フレームワークを使用していればユーザが快適に感じるサイズの入力部品でページを構成することができます。カスタマイズにより独自の部品を使用する場合には注意してください。



本フレームワークを使用することで、ボタンのサイズはスマートフォン用に最適な大きさに自動で調整が行われます。小さすぎるボタンはユーザの指でうまくタップできなくなるおそれがあります。

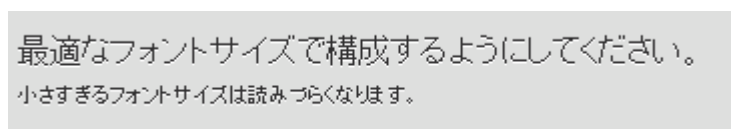


あまりよくないレイアウトの例。4 つのリンクが密接してしまっているので、ページ 1 を見たいと思っても他のリンクをタップしてしまうかもしれません。フォントサイズを小さくしてしまうとさらに扱いづらくなってしまいます。

2.1.7 使用するフォント

フォントのサイズは、なるべく 12 ポイント以上を使用するようにしてください。やむを得ず小さいフォントサイズを使用する場合は、ユーザ設定等でサイズの設定を変えられるようにする等の検討を行ってください。ただし、見出しと補足情報を持つリスト情報のような場合は、見出しは標準のサイズとし、補足情報についてはフォントサイズを下げてレイアウトを整え、次画面(詳細画面等)では標準サイズのフォントで表示して読みやすいようにする、といった柔軟な対応を行ってもよいでしょう。

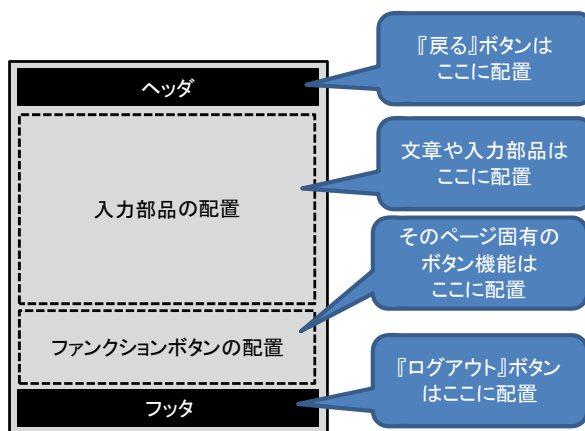
また、使用するフォントをローカルフォントにするかサーバフォントにするかについての検討を十分に行ってください。使用するフォントを何にするかの検討も必要ですが、ターゲットとするスマートフォンの機種をどれにするのかという検討も必要になります。本フレームワークを使用していれば、指定したフォントに最も近いフォントを使用してページを構成するようになっています。ローカルフォントを使用する場合は、スマートフォンの機種によってインストールされているフォントが異なります。また、同じフォント名でも機種によっては文字の大きさが異なることもあります。サーバフォントを使用する場合は、フォントによる機種依存問題を抑えることができますが、サーバからスマートフォンへフォントファイルの転送が発生するため、レスポンスへの影響があります。



2.1.8 一貫性のあるユーザインタフェースにする

作成するサイトは、画面のレイアウトやスタイル、ボタン・アイコンとそれらが持っている意味や配置場所を統一するようにしてください。また、画面もパターン化に留意して作成するようにします。このような一貫性のあるユーザインタフェースで構築されたサイトは、ユーザが少ない時間でシステムの使い方を学習することができるため、他の機能を使用する際もすぐに使いこなせるようになります。

なお、統一された意味合いを持つ、よく知られているアイコンを用いることにより、必ずしも単語を用いる必要性もなくなります。例えば、家のアイコンのボタンをタップすると、まず間違いなくホーム画面へページが切り替わるでしょう。



画面レイアウトの一例。画面レイアウトのパターン化に留意し(1 つではなくいくつかのパターンがあってもよい)、作成する画面はなるべくそれらのパターンから逸脱しないようにします。



ボタンアイコンの例。テキストのないボタンの例ですが、それらをタップすると何が起きるかは予想がつかます。

2.1.9 コンテンツの装飾

スマートフォン向けのコンテンツは、シンプルな装飾をおすすめします。スマートフォンのディスプレイが表示できる領域は限られています。派手なデザインやアニメーションは主たる情報からユーザの意識がそれてしまうこともあるかもしれません。業務システムの効率を落としてしまうため、結果的にユーザは『使い勝手がよくない』と感じてしまうかもしれません。

2.1.10 必須項目の明示

何か入力するための部品を設計する際、ラベルには必須入力であることがわかるような単語、または記号を用い、ユーザにアピールするようにしてください。本フレームワークにはラベルに必須かどうかの情報を付加するための属性も用意されています。

また、HTML5 では `placeholder` 属性を使用することができるようになりましたが、一部の入力部品ではこの属性がうまく機能しない場合がありますので、`placeholder` 属性の使用はあまりおすすめできません。

2.1.11 トランザクション処理を行うときはユーザに確認を求める

いったん登録すると元に戻せないような、トランザクション処理を行う前には、個別の確認画面を用意するほうがユーザにその重要度が伝わるでしょう。後からユーザが編集や削除ができるような場合は、一貫性の観点や、コーディング量の観点から、個別に確認画面を用意するよりもダイアログを使用してもよいでしょう。

2.2 画面設計方針

2.2.1 ページの構成

各ページを構成する HTML ファイルは、上部にヘッダ、下部にフッタを備えた構成にします。ヘッダ部にはそのページの内容を端的に表すタイトルを表示するようにします。フッタ部にはアプリケーションやシステムの名前を表示してもよいですが、ログアウトボタンやメニューへ復帰するためのボタン等、共通的な処理を起動するためのボタンを配置するほうがユーザは利便性が高いと感じるでしょう。ただし、ダイアログについては、フッタを省略、場合によってはヘッダを省略する等、ユーザのニーズや業務要件に則したデザインを検討してください。



2.2.2 スタイルシートの利用

コンテンツの制作時には、すべてのページで統一したデザインにするために、スタイルシートを利用することを検討してください。本フレームワークでは、既定のスタイルシートを読み込むようになっているため、ある程度統一したページデザインにすることが可能になっています。

2.2.3 HTML5 への対応

スマートフォン向けコンテンツの制作にあたっては、HTML5 への準拠を念頭に置いて開発を行ってください。この際、HTML5 で非推奨となっているタグを使用しないように配慮を行ってください。frame や font 等、これまで一般的に用いられてきたタグも非推奨になっているため、注意が必要です。

2.2.4 複数の内部ページを持つHTMLについて

jQuery Mobile では、ひとつの HTML ファイルの中に複数の内部ページを定義することができます。これを活用することは有用ですが、データベースの検索結果と HTML ファイルの単位を一致させて設計を行っている場合等、業務データの持ち方に関する粒度が煩雑になってしまうような場合は、HTML ファイルは 1 画面につき 1 つにしたほうがよいと考えられます。

2.2.5 スマートフォン用のコンテンツとして相応しいデザインを考える

スマートフォン用のコンテンツを制作するときは、PC 用サイトやフィーチャーフォン用サイトで用いられている入力部品は用いるべきではありません。例えば、検索キーワードを入力するためのテキストボックスはスマートフォン用サイトでは専用のテキストボックスを用いることが一般的です。通常のテキストボックスを使用した場合、ユーザーはそのテキストボックスが検索キーワードを入力するためのものと気付かない可能性が考えられるからです。



上は検索ボックスを使用した例。下は通常のテキストボックス。『検索キーワード』というラベルが付いていなかったとしても上の例は検索キーワードを入力するための領域であることが理解できます。

2.2.6 ページ切り替えのエフェクト

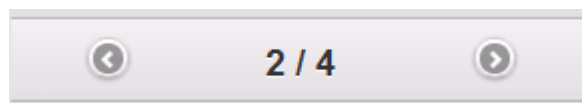
スマートフォン用サイトのページ間の切り替えは、スライドのアニメーション効果等を付加することができます。また、PC 用サイトのようにページを切り替えることもできます。これらのエフェクトは、できるだけサイト内で統一したものを使用するようにします。例えば、同一業務内での画面遷移をスライド効果で遷移させることにより、業務・機能が繋がっていることをユーザに示唆することができます。

また、同一業務内でスライド遷移を行い、最後に登録処理を行うような場合に通常のページ遷移を使用することで、ユーザは業務フローが完了したと感じるでしょう。通常のページ遷移は、他の業務へ遷移する場合にもユーザへ業務・機能が変わったことを伝えることができます。

また、ポップアップのエフェクトも効果的に使用してください。簡易検索や絞り込み検索等、メインタスクの画面で何らかの情報を参照入力するような子画面へ遷移させたい場合、子画面への遷移はポップアップ効果を使用するとよいでしょう。子画面で入力する情報量を必要最低限に絞ることで、ユーザの意識がメインタスクからそれてしまうことのないような配慮も必要です。キーワードによる絞り込みであれば、画面上部等に検索用のテキストボックスを配置するのも効果的です。

2.2.7 一覧形式のコンテンツを表示するページのデザイン方針

一覧形式のコンテンツにおいて、2 ページ目以降のデータをどのようにユーザに見せるかについては、おおまかに 2 通りの方法が考えられます。ひとつはページ遷移によって HTML を作成する方法です。この方法は従来の PC 用サイト等で一般的に用いられてきた方法です。もうひとつは Ajax を活用した方法です。一覧の最下部に配置された『もっと見る』ボタンをタップすることで、現在表示されている一覧に次の検索結果セットが順次追加されていく表示方法です(Apple Store のアプリ検索画面等で用いられています)。本フレームワークではどちらの表示方法も対応できるようになっていますので、コンテンツをデザインするときにどちらの方法が最適かを検討してください。



spPagingButton の例。『<』と『>』のボタンでページを移動することができます。

2.2.8 ボタンの配置

スマートフォンの画面のサイズには限りがあります。このため、ボタンの配置にも工夫が必要です。例えば、ひとつの画面内にたくさんのファンクションボタンを配置することは、スマートフォン向けコンテンツとしては効果的ではない場合が考えられます。狭い画面の中にたくさんのボタンが配置されることで、ユーザは困惑し、それらの機能を理解するために時間を要することになり、メインタスクから意識がそれてしまいます。直観的に使い方がユーザに伝わりにくくなるため、利便性を損なうことになるかもしれません。このため、ひとつの画面内に配置するボタンは必要最小限に絞ることを常に検討してください。どうしてもボタンの数を絞ることができない場合は、開閉式リストを活用して、優先順位の低いボタンをなるべく表示しないような検討を行ってください。

また、ボタンの並び方についても配慮が必要です。ある画面では『キャンセルボタンが左に、実行ボタンが右』に配置されていたとします。別の画面で、『実行ボタンが左に、キャンセルボタンが右』に配置されていたとしたら、誤って操作してしまう原因になるでしょう。さらに、実行ボタン等の『そのページでの操作を確定させるボタン』は色を変えて強調表示にする、といった工夫を加えることで、ユーザの操作ミスを少なくすることに貢献できます。

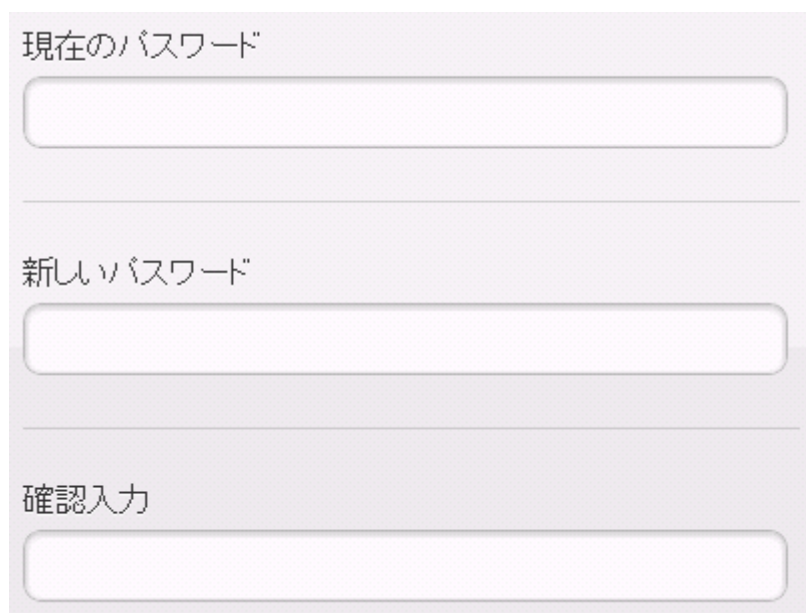


『確定』を強調表示している例。重要なボタンは色分けを行うことで、ユーザの注意をひきつけることができます。

2.2.9 入力部品とラベル

テキストボックス等の入力部品は、それが何のためのものであるかを明らかにするため、必ずラベルをつけるようにしてください。本フレームワークでは、入力部品とラベルを関連付けるためのタグもありますので、積極的に活用してください。ラベルと入力部品の表示位置も自動で調整されますので、機種依存を抑えることにもつながります。

また、ラベルに表示するタイトルは、複数の画面で統一されたタイトルを使用してください。例えば、ホーム画面に戻るためのボタンを定義する場合、ある画面では『HOME ボタン』で、別の画面で『MENU ボタン』としてしまうと、ユーザはそれらのボタンが同一の機能であることに気づくことができません。



The image shows a form with three text input fields. The first field is labeled '現在のパスワード' (Current Password), the second is '新しいパスワード' (New Password), and the third is '確認入力' (Confirmation Input). Each label is positioned to the left of its corresponding text box, and the text boxes are vertically aligned.

3つのテキストボックスは適切にラベリングされているため、その役割がユーザにも伝わります。

2.2.10 キー入力の頻度に配慮する

スマートフォンでの入力作業は煩雑になりがちです。ユーザがテキスト入力する頻度はできる限り少なくし、セレクトメニューや参照入力系の入力部品を活用するようにします。入力時間の短縮や生産性の向上のみでなく、ユーザのストレスを軽減することにも貢献できます。



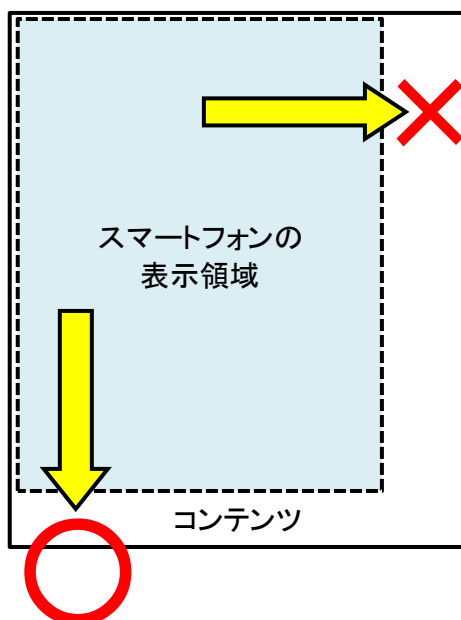
The image shows a text input field with the label '日付を入力してください' (Please enter the date) positioned above it. A large red 'X' is drawn over the input field, indicating that this is an incorrect or discouraged practice.

キー入力に依存しすぎる画面の場合、ユーザがわずらわしいと感じるだけでなく、入力チェックの手間も発生します。

2.2.11 画面のスクロール

1 ページの内容が、スマートフォンの画面に表示しきれない場合、ユーザはフリック操作によってページをスクロールします。このとき、スクロール可能な方向は縦方向のみとし、横方向へのスクロールはしないようにします。PC版サイトを閲覧するような場合は例外ですが、スマートフォン用コンテンツでは横方向へのフリックスクロールは一般的であるとは言えません。また、ユーザも現在表示されているページの横に情報が隠されているとは思っていないでしょう。

また、1 ページあたりの情報量が膨大になってしまうような場合は、開閉式コンテンツを導入し、スクロールする量を少なくするような検討も行ってください。

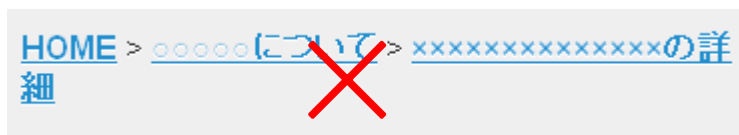


このような場合に横スクロールの必要性が発生してしまいます。ユーザは縦方向にコンテンツをスクロールすることには慣れていますが、横方向にコンテンツをスクロールすることには慣れていません。

2.2.12 ページ間の遷移は、どこからどこにいくのかわかるようにする

一般的なホームページの制作でも同様のことが言えますが、サイト内の構成がユーザーにとってイメージしやすいように、複雑になりすぎないように配慮してください。サイトを構成するファイルをどのような配置にするのか配慮するだけでなく、ユーザーを誘導するボタンやリンクのテキストを統一し、わかりやすい内容にしておくことも大切です。

また、フットパス(パンくず)はスマートフォン用サイトにおいては必要とされないかもしれません。限られた表示領域をフットパスで埋めてしまうことと、そもそもスマートフォン用サイトの構成がシンプルであるならば、迷子になることを防ぐためのフットパスは必要となりません。フットパスを使用しない場合は、ユーザーが来た道を戻れるように『戻るボタン』を各ページに配置するようにしてください。



フットパスが 2 行にわたって表示されてしまっている例。リンクテキストの長さによってはさらに行数を要する可能性もあります。

2.2.13 スマートフォン用コンテンツでは相応しくないと思われる表現

スマートフォンの画面サイズは PC 向けサイトよりも限られています。フィーチャーフォン向けサイトよりも表現力豊かに見せることができます。このため、スマートフォンに最適化されたコンテンツを制作するには、すでに一般的に用いられている表現方法が相応しくない場合が考えられます。例えば、フットパス(パンくず)はスマートフォン向けサイトでは効果的な表現方法とされないことが考えられます。また、ヘルプコンテンツについても表示領域の関係から導入すべきではないかもしれません。ぱっと見で使い方がすぐわかるように配慮されたサイトではこれらの表現は不要である場合が多いからです。

2.2.14 表示する情報の優先度を考える

ページに表示する情報には優先度によって開閉式コンテンツが適用できないか検討を行ってください。ユーザーが見るべき情報は表示領域を割いて表示すべきですが、補足情報や重要性の低いものについて、開閉式コンテンツによって初期状態は折りたたんだ状態にしておくこと、ユーザーは利便性が高いと感じます。開閉式コンテンツを効果的に活用することによって、ページが縦長になってしまうのを防ぐ、また、開閉式コンテンツの見出しをわかりやすく工夫することによってユーザーはそれが何なのかを見出しを見ただけで理解できるようになります。



開閉式メニューの例。『サブメニュー』の左側の『-』をタップすることで『並べ替え』と『最新情報』のボタンを折りたたんで非表示にすることができます。

2.2.15 『戻るキー』について

本フレームワークが提供する画面において、ブラウザの[戻る]及び[進む]ボタンは使用しないでください。また、スマートフォン本体の[戻る]ボタンも使用しないでください。

2.2.16 応答時間と画面の設計

一般的な PC 向け Web サイトと同様、スマートフォン向け Web サイトでもできる限り迅速にユーザーへ応答を返すような設計が必要です。

ただし、サイズの大きいデータを取り扱う場合や、設計段階で電波状況の良くない場所で使用することがわかっている場合等、やむを得ないケースも考えられます。このような場合は、Ajax を効果的に活用し、データ部分の HTML は Ajax による非同期通信で取得するように設計し、それ以外の部分の HTML は即座にユーザーに応答できるようにします。Ajax による通信中はローディングメッセージを出力し、ユーザーに処理中であることをアピールすることで『固まっているように見える』ということがなくなります。

2.2.17 サーチフィルターの適用方針

サーチフィルターを使用する場合は、表示対象となるデータの件数と、データベース上に存在するデータの件数に注意が必要です。ページ上にロードされているデータから絞り込むための部品ですが、ユーザーによってはデータベース上に存在するすべてのデータから絞り込んでしまうと勘違いしてしまうかもしれません。このため、サーチフィルターは検索結果一覧のデータが決まり切っている場合や、すべてのデータを読み込んでいる場合に使用するようにします。

2.2.18 スライド遷移とボタン操作イメージ

通常のスライド遷移では、次ページに遷移する際に右から次のページが表示されるようにアニメーション処理されます。このとき、ユーザは『右へ進む』ように感じます。また、前のページに戻る場合は『左へ戻る』ように感じます。このため、『戻る』ボタンは画面の左上に配置し、ボタンの配置場所と画面が遷移する方向を一致させるようにしてください。

2.2.19 トグルスイッチの適用方針

トグルスイッチは、2つの相反する選択肢をユーザに求める場合に使用し、ユーザに迷わせないようにします。『はい/いいえ』『ON/OFF』等の簡潔な選択肢にします(長い文言はトグルスイッチが見つづらくなります)。例えば、『○×業務連携機能』という項目に『ON/OFF』の二者択一の選択肢であれば、それぞれの選択肢が何を意味するのか分かりやすいでしょう。一方、『○×業務』という項目に『ON/OFF』の選択肢とした場合、どちらを選ぶべきか明確にユーザに伝えることはできません。

3 IM-Mobile Framework プログラミングガイド

3.1 基本的な記法

3.1.1 基本的な構成

スマートフォン向け Web サイトを制作するために、どのようなタグを使用したらよいのかについては、本フレームワークのタグリファレンスや jQuery Mobile のリファレンスを参照してください。本書では、スマートフォン向け Web サイトの HTML をどのように記述したらよいのか、その概念について説明します。

まず、おおまかに HEAD タグと BODY タグから構成される点はこれまでの Web サイトと変わりありません。HEAD タグおよび BODY タグの中にスマートフォン向けのタグを正しく記述していくことによってスマートフォン向けコンテンツを制作することができます。下記に本フレームワークをベースとした、最もシンプルなスマートフォン向け HTML の例を示します。

```
1 <!doctype html>
2 <html>
3 <head>
4     <meta http-equiv="content-type" content="text/html; charset=UTF-8">
5     <imart type=" spIncludeJQM" ></imart>
6 </head>
7 <body>
8     <div data-role="page">
9         <imart type=" spHeaderWithBack" page=" path/to/page" ></imart>
10        <div data-role="content">
11            <p>ここにコンテンツを記述します</p>
12        </div>
13        <imart type=" spFooterWithLogout" ></imart>
14    </div>
15 </body>
16 </html>
```

5 行目では HEAD タグの中にスマートフォン用のライブラリを使用するための宣言を記述します。

8～14 行目までがコンテンツです。まず、data-role 属性が"page"の div タグを記述します。その div タグの内部にヘッダ部(9 行目)、コンテンツ部(10～12 行目)、フッタ部(13 行目)を記述してください。例では、ヘッダ部とフッタ部に本フレームワークから提供しているタグを記述していますが、<div data-role="header"> や <div data-role="footer">を記述することにより、個別にヘッダとフッタを定義することもできます。

ここでは、HEAD タグの中にライブラリを使用するための宣言が必要であることと、コンテンツはヘッダとコンテンツとフッタの 3 つから構成されることを理解してください。

3.1.2 HEADタグ内部の記述

HEAD タグでは、スマートフォン用ライブラリを使用するための宣言を記述します。この宣言を簡易に行うために、本フレームワークでは `spIncludeJQM` というタグを提供しています。多くの場合はこのタグを使用することで十分な機能を実現することができるようになっていますが、jQuery Mobile の記法に従って個別にこの宣言を記述することも可能です。ただしこの場合には、ソースコードの共通化という観点で、複数の記述法でサイトが構成されている点に注意してください。

3.1.3 BODYタグ内部の記述

BODY タグ内部には、ヘッダ、コンテンツ、フッタを記述します。さらにコンテンツ部にはテキストボックス等の入力部品やデータベースからの検索結果を表示するためのリスト等、スマートフォン用のタグを使用してコンテンツをマークアップしていきます。コンテンツを記述するためのタグは多種に渡りますが、それらの説明については、本フレームワークのリファレンスや jQuery Mobile のリファレンスを参照してください。

3.2 業務スケルトンの使い方

intra-mart e Builder の業務スケルトンウィザードを使用することで、対話形式でスマートフォン用の画面を作成することができます。作成した後はボタン処理等を適切に追加してください。本項では、業務スケルトンウィザードから作成することができる画面とその概要を示します。

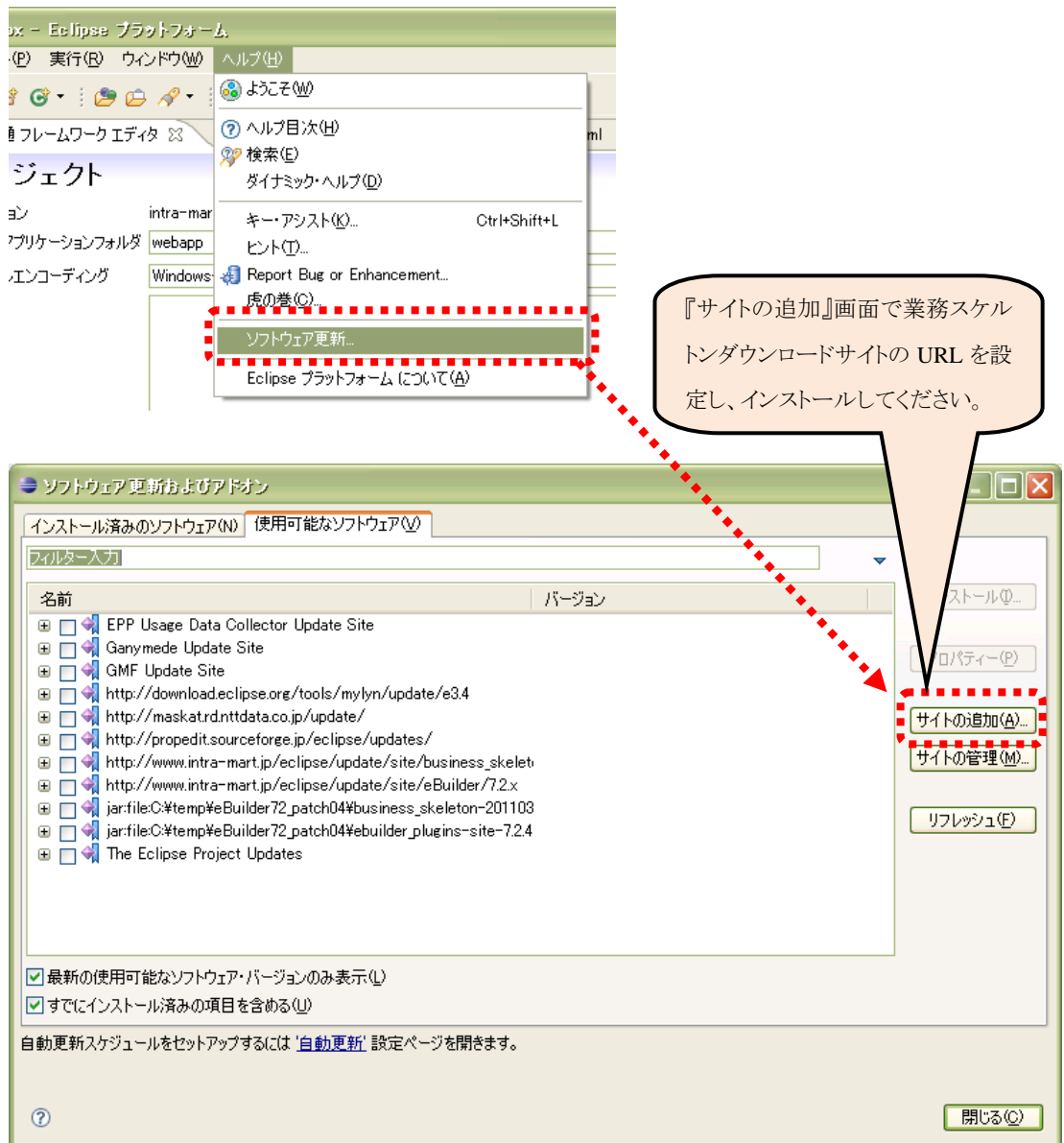


3.2.1 業務スケルトンのインストール

業務スケルトンは eclipse のソフトウェア更新マネージャを使用して、intra-mart の更新サイトから intra-mart e Builder へインストールします。業務スケルトンのインストールは下記に示す URL からダウンロードしてください。

■IM-Mobile Framework 業務スケルトンダウンロードサイト

http://www.intra-mart.jp/eclipse/update/site/business_skeleton/mfw



3.2.2 一覧

データベースから検索したレコードセットを一覧形式のリストで表示するための画面を作成することができます。オプションを指定することで、一覧をソートするためのキーを定義することもできます。

下記は業務スケルトンウィザードで作成した画面のイメージです。実際に作成する画面は、業務要件やウィザードの入力内容によって異なります。



3.2.3 参照

データベースから検索したデータを読み取り専用の単票形式で表示するための画面を作成することができます。オプションを指定することで、当該レコードの削除機能を付加することもできます。

下記は業務スケルトンウィザードで作成した画面のイメージです。実際に作成する画面は、業務要件やウィザードの入力内容によって異なります。



3.2.4 登録

新規にレコードを登録するための単票形式の入力画面を作成することができます。

下記は業務スケルトンウィザードで作成した画面のイメージです。実際に作成する画面は、業務要件やウィザードの入力内容によって異なります。

The image shows a mobile application registration screen. At the top, there is a dark header bar with a back arrow and the text '戻る' (Back) on the left, and 'ヘッダータイトル' (Header Title) in the center. Below the header is a light gray content area. At the top of this area is a blue bar with the text 'コンテンツタイトル' (Content Title). The main content area contains two sections: '項目1' (Item 1) with a single text input field, and '項目2' (Item 2) with two radio button options labeled 'radioltem1' and 'radioltem2'. At the bottom of the content area is a large blue button labeled '登録' (Register). The bottom of the screen features a dark navigation bar with two buttons: 'MENU' with a home icon and 'ログアウト' (Logout) with a circular arrow icon.

3.2.5 編集

データベースに登録済みの詳細情報を編集するための単票形式の入力画面を作成することができます。オプションを指定することで、当該レコードの削除機能を付加することもできます。

下記は業務スケルトンウィザードで作成した画面のイメージです。実際に作成する画面は、業務要件やウィザードの入力内容によって異なります。



3.3 IM-Mobile Framework共通画面の使い方

本フレームワークをインストールすると、作成済みの共通画面を使用することができるようになります。本項では、共通画面とその概要を示します。

3.3.1 IM-Mobile Framework共通画面一覧

下記に本フレームワークにて提供するスマートフォン向け共通画面の一覧を示します。

項番	画面名	概要
1	ログイン	スマートフォン向けログイン画面です。
2	メニュー	スマートフォン向けメニュー画面です。 グループ管理者画面で設定したメニュー項目を表示します。
3	ポップアップメニュー	ポップアップ形式で表示するメニューです。 グループ管理者画面で設定したメニュー項目のうち、pop 属性が設定されている項目を表示します。
4	パスワード変更	スマートフォン向けパスワード変更画面です。 当画面は初回ログイン時やパスワード期限切れの時に表示します。
5	メッセージ画面(アクセス拒否)	スマートフォン向けメッセージ表示画面です。
6	メッセージ画面(セッション無効)	スマートフォン向けメッセージ表示画面です。
7	メッセージ画面(認証失敗)	スマートフォン向けメッセージ表示画面です。
8	メッセージ画面(ログイングループなし)	スマートフォン向けメッセージ表示画面です。
9	メッセージ画面(ログイン許可なし)	スマートフォン向けメッセージ表示画面です。
10	メッセージ画面(アカウントロック)	スマートフォン向けメッセージ表示画面です。
11	メッセージ画面(ログイン禁止)	スマートフォン向けメッセージ表示画面です。
12	メッセージ画面(メンテナンス中)	スマートフォン向けメッセージ表示画面です。
13	メッセージ画面(ページアクセス権限なし)	スマートフォン向けメッセージ表示画面です。
14	メッセージ画面(ログインセッション無効)	スマートフォン向けメッセージ表示画面です。
15	メッセージ画面(セッション不整合)	スマートフォン向けメッセージ表示画面です。
16	メッセージ画面(セッションタイムアウト)	スマートフォン向けメッセージ表示画面です。
17	メッセージ画面(システムエラー)	スマートフォン向けメッセージ表示画面です。
18	メッセージ画面(二重ログイン)	スマートフォン向けメッセージ表示画面です。
19	メッセージ画面(汎用)	スマートフォン向けメッセージ表示画面です。汎用的に開発者が指定するメッセージを表示することができます。
20	ユーザ検索画面	スマートフォン向けユーザ共通検索画面です。IM-共通マスタ上のユーザを検索することができます。

3.3.2 ポップアップメニューについて

ポップアップメニューは、スマートフォン版メニューを任意の画面からポップアップ表示するための画面です。ポップアップメニューが表示する項目は、グループ管理者画面で設定したメニュー項目のうち、pop 属性が設定されているものを対象とします。

ポップアップメニューの表示例:



3.3.2.1 使用方法

<%Server_Path%/doc/imart/WEB-INF/classes フォルダに XML ファイル「service-config-mobile_fw.xml」を作成し、内容を下記のようにします。

1	<?xml version="1.0" encoding="UTF-8" standalone="no"?>
2	<service-config>
3	<service>
4	<service-id>toPopUpMenu</service-id>
5	<next-page>
6	<page-path>/system/security/user/mobile_fw/popup_menu.jssp</page-path>
7	</next-page>
8	</service>
9	</service-config>

ポップアップメニューの表示対象とする項目には pop 属性の設定を行います。

ログイングループ管理者メニューのメニュー設定より、任意のメニューに対し引数を与えます。

引数は、キーが「pop」、値を「true」とします。

メモ帳

ページ内部URL

メニューID(必須)

表示名(必須)(国際化)

クライアントタイプ

URL(必須)

引数

<input type="checkbox"/>	キー	値	編集
<input type="checkbox"/>	pop	true	編集

アイコン画像ファイルパス

備考

権限リスト

ロール	組織	役職	パブリックグループ
<input type="checkbox"/>	ロール名	表示名	
<input type="checkbox"/>	level1	レベル1ユーザ	
<input type="checkbox"/>	level2	レベル2ユーザ	
<input type="checkbox"/>	level3	レベル3ユーザ	

ポップアップメニューは、呼び出し元となる画面の HTML/JSP にポップアップメニューを起動するためのリンクを記述する必要があります。下記にポップアップメニューの記述例を示します。

<記述例>

```

...
<a href="/imart/mobile_fw-toPopUpMenu.service" data-rel="dialog" data-role="button">ポ
ップアップメニュー</a>
...

```

※スクリプト開発の場合、jQuery Mobileの問題によりポップアップメニューの戻るボタンが動作しないことがあります。3.6.3 data-rel="back"時の画面遷移エラーについて を参考に別途スクリプトを配置してください。

3.3.3 ユーザ共通検索画面

IM-Mobile Framework Version7.2.1 より、共通画面にユーザ共通検索画面が加わりました。

下記に使用方法について示します。

※本機能を利用するには IM-共通マスタのインストールが必要です。

アプリケーション共通マスタには対応しておりませんのでご注意ください。

ユーザ共通検索画面の表示例:



3.3.3.1 使用方法

<%Server_Path%>/doc/imart/WEB-INF/classes フォルダに XML ファイル「service-config-mobile_fw.xml」を作成し配置します。（既にファイルがある場合は<service>要素内を追記します）

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <service-config>
3   <service>
4     <service-id>toImACMSearchSp</service-id>
5     <next-page>
6       <page-path>/smartphone_search/search/search.jssp</page-path>
7     </next-page>
8   </service>
9 </service-config>

```

スクリプト開発の場合:使用する HTML ファイル内の<HEAD>タグ内で

<IMART type=" imACMSearchSp">タグを定義します。

```

1 <head>
2     <imart type="spIncludeJQM" />
3     <imart type="imACMSearchSp" />
4 </head>
5 ...

```

JavaEE 開発の場合:使用する JSP ファイル内の<HEAD>タグ内で

<imsp:imACMSearchSp>タグを定義します。

```

1 <head>
2     <imsp:includeJQM />
3     <imsp:imACMSearchSp />
4 </head>
5 ...

```

画面を表示する際は、任意のイベント処理内で以下のようなクライアント JavaScript を実装して呼び出します。

パラメータの target 属性は固定で"jp.co.intra_mart.im_master.search.user.smartphone"を指定し、

callback 属性にはコールバック関数名を指定します。

その他指定可能なパラメータについては 3.6.2 初期状態・検索の絞り込み条件の設定 を参照してください。

```

1 //検索画面に渡すパラメータ定義
2 var param = {
3     target : "jp.co.intra_mart.im_master.search.user.smartphone",
4     callback_function: "resultSearch"
5 };
6 //パラメータをエンコード
7 var parameter_json_string = escape(ImJson.toJSONString(param));
8
9 //検索画面へ遷移
10 $.mobile.changePage("/imart/mobile_fw-toImACMSearchSp.service", {
11     data      : { parameter : parameter_json_string },
12     role      : 'dialog',
13     method:"POST"
14 });
15

```

※スクリプト開発の場合、jQuery Mobile の問題によりポップアップ画面の戻るボタンが動作しないことがあります。 3.6.3 data-rel="back"時の画面遷移エラーについて を参考に別途スクリプトを配置してください。

3.3.3.2 初期状態・検索の絞り込み条件の設定

ユーザ選択画面の初期状態及び検索条件を設定する場合、引数 parameter に対し Object を JSON 文字列に変換した値を渡して設定します。Object に設定可能なプロパティを以下一覧に示します。

プロパティ名	型	必須	初期値	備考
target	String	○	-	使用するプラグイン名を指定します。 値は固定で “jp.co.intra_mart.im_master.search.user.smartphone” を指定してください。
basic_area	String	×	none	基本情報描画領域の表示／非表示を切り替えます。 “true”：表示する “false”：表示しない
tabs	Array	×	-	target に検索対象を設定せず、実行時に使用する検索画面タブを設定したい場合にこの項目を使用します。詳細については 3.3.3.2.1 検索画面タブ Plugin 指定を参照してください。
wnd_title	String	×	ユーザ検索	ページタイトルを設定します。
message	String	×	-	ダイアログのヘッダタイトルを設定します。
callback_function	String	○	-	コールバック関数名を設定します。
default_tab_id	String	×	-	画面を初期表示したときに表示するタブを選択します。設定可能なタブについては 3.3.3.2.1 検索画面タブ Plugin 指定を参照してください。
type	String	×	multiple	検索結果を単一選択とするか、複数選択とするかを設定します。 “single”：単一選択 “multiple”：複数選択
default_selected	Array	×	-	選択一覧で選択済のオブジェクトを設定します。 戻り値の形式については 3.6.3 コールバック・返却値を参考にしてください。
department_main_only	Boolean	×	-	検索対象を、主所属組織が設定されたユーザのみとするかを設定します。
target_date	Date	×	システム日付	期間化された情報を取得する際基準にする日付を指定します。
deleted_data	Boolean	×	false	論理削除状態のデータを検索対象にするかどうかを設定します。 true :論理削除データを取得 false:論理削除データを取得しない
target_locale	String	×	現在のログインユーザのロケール	表示ロケールを設定します。検索結果もこのロケール表示で返します。
criteria	Object	×	-	検索時の暗黙的な絞り込み条件を設定します。 詳細については 3.3.3.2.2 暗黙条件を参照してください。
prop	Array	×	-	検索画面タブ毎に、取得するテーブルのカラム名を設

				定めます。詳しくは 3.3.3.2.1 検索画面タブ Plugin 指定を参照してください。
keyword_search_target	Array	×	-	キーワードの検索対象を指定します。複数指定することができます。 “code”:コード “name”:名称 “search_name”:検索名 ※キーワード検索のみ有効な属性です。
keyword_search_type	String	×	-	キーワード検索の検索方法を指定します。 “prefix”:前方一致 “full” :完全一致 “part” :部分一致 ※キーワード検索のみ有効な属性です。

3.3.3.2.1 検索画面タブPlugin指定

指定可能な検索画面タブ Plugin は以下のとおりです。

検索画面タブ	プラグイン ID
ユーザ検索(キーワード)	jp.co.intra_mart.im_master.app.search.tabs.user.user.keyword.smartphone
ユーザ検索(組織)	jp.co.intra_mart.im_master.app.search.tabs.user.department.tree_with_list.smartphone
ユーザ検索(パブリックグループ)	jp.co.intra_mart.im_master.app.search.tabs.user.public_group.tree_with_list.smartphone
ユーザ検索(プライベートグループ)	jp.co.intra_mart.im_master.app.search.tabs.user.private_group.tree.smartphone
ユーザ検索(ロール)	jp.co.intra_mart.im_master.app.search.tabs.user.role.tree_with_list.smartphone

■ タブセットの指定方法

実行時に使用する検索画面タブを設定したい場合、検索画面に渡すパラメータの tabs 属性を指定します。

tabs 属性の指定方法は以下の通りです。

プロパティ名	型	必須	初期値	備考
tabs	Array	×	-	
配列インデックス	Object	×	-	
id	String	○	-	検索画面タブ Plugin ID
title	String	○	-	タブの表示
<pre> “tabs” : [{ id: “jp.co.intra_mart.im_master.app.search.tabs.user.user.keyword.smartphone”, name:”キーワード” }] </pre>				

■ 各タブのユーザ情報返却値の指定方法

コールバック時に取得するユーザ情報に IM 共通マスタのユーザテーブルの任意のカラムを追加したい場合、検索画面に渡すパラメータの prop 属性にタブのプラグイン ID とカラム名を指定します。

prop 属性の指定方法は以下の通りです。

ユーザテーブルの詳細については IM-共通マスタ テーブル定義書を参照してください。

プロパティ名	型	必須	初期値	備考
prop	Object	×	-	
検索タブ Plugin ID	Array	×	-	
配列インデックス	String	○	-	取得したいカラム名
<pre>prop: { "jp.co.intra_mart.im_master.app.search.tabs.user.user.keyword.smartphone": ["email_address1"] }</pre>				

3.3.3.2.2 暗黙条件

ユーザ検索(キーワード)タブで検索を行う際に、ユーザ操作とは別に指定する条件です。検索結果はこの暗黙条件の範囲で自動的に絞り込まれます。

■ 会社

ユーザ検索時に指定の会社で絞り込みます。ただし、組織リストまたは組織を設定している場合、会社は利用されません。

プロパティ名	型	必須	初期値	備考
company	Array	×	-	
配列インデックス	Object	×	-	
company_cd	String	○	-	
<pre>"criteria": { company: [{ "company_cd": 会社コード }, { "company_cd": 会社コード }, ...] }</pre>				

■ 組織セット表示設定

会社毎に表示する組織セットを設定します。表示する組織セットは会社毎にデフォルト組織セットリストで指定することができ、組織セット名の表示/非表示を選択できます。

プロパティ名	型	必須	初期値	備考
department_set_disp	Object	×	-	
type	String	○	hide	hide、display、allのいずれか
list	Array	○	-	
配列インデックス	Object	×	-	
company_cd	String	○	-	
department_cd	String	○	-	
<pre>"criteria": { "department_set_disp": { "type": "hide", "list": [{ "company_cd": 会社コード, "depatrment_set_cd": 組織セットコード }, ...] } }</pre>				

■ 組織

ユーザ検索時に指定の組織で絞り込みます。ただし、組織リストが指定されている場合は、組織リスト指定が優先されます。

プロパティ名	型	必須	初期値	備考
department_set	Object	×	-	
company_d	String	○	-	
department_set_cd	String	○	-	
department	Object	×	-	
department_cd	String	○	-	
compare	String	×	le	gt、ge、lt、le のいずれか
<pre>"criteria": { "department_set": { "company_cd": 会社コード, "department_set_cd": 組織セットコード, "department": { "department_cd": 組織コード, "compare": "le" } } }</pre>				

■ 組織役職

ユーザ検索時に指定の役職で絞り込みます。

プロパティ名	型	必須	初期値	備考
department_set	Object	×	-	
company_cd	String	○	-	
department_set_cd	String	○	-	
post	Object	×	-	
post_cd	String	○	-	
compare	String	×	le	gt、ge、lt、le のいずれか
<pre>"criteria": { "department_set": { "company_cd": 会社コード, "department_set_cd": 組織セットコード, "post": { "post_cd": 役職コード, "compare": "le" } } }</pre>				

■ 組織分類項目

ユーザ検索時に指定の組織分類項目を持つ組織で絞り込みます。

プロパティ名	型	必須	初期値	備考
dept_ctg_item	Array	×	-	
配列インデックス	Object	×	-	
company_cd	String	○	-	
category_cd	String	○	-	
category_item_cd	Array	○	-	
配列インデックス	String	○	le	gt、ge、lt、le のいずれか
<pre>"criteria": { "dept_ctg_item": [{"company_cd": 会社コード, "category_cd": 組織分類コード, "category_item_cd": [組織分類項目コード, ...]}, ...] }</pre>				

■ パブリックグループ

ユーザ検索時に指定のパブリックグループで絞り込みます。

プロパティ名	型	必須	初期値	備考
public_group_set	Object	×	-	
public_group_set_cd	String	○	-	
public_group	Object	×	-	
public_group_cd	String	○	-	
compare	String	×	le	gt、ge、lt、le のいずれか
<pre>"criteria": { "public_group_set": { "public_group_set_cd": パブリックグループセットコード, "public_group": { "public_group_cd": パブリックグループコード, "compare": "le" } } }</pre>				

■ パブリックグループ役割

ユーザ検索時に指定の役割で絞り込みます

プロパティ名	型	必須	初期値	備考
public_group_set	Object	×	-	
public_group_set_cd	String	○	-	
public_group_role	Object	×	-	
role_cd	String	○	-	
compare	String	×	le	gt、ge、lt、le のいずれか


```
"criteria": {
  "public_group_set": {
    "public_group_set_cd": パブリックグループセットコード,
    "public_group_role": {
      "role_cd": パブリックグループコード,
      "compare": "le"
    }
  }
}
```

■ パブリックグループ分類項目

ユーザ検索時に指定のパブリック分類項目を持つパブリックグループで絞り込みます。

プロパティ名	型	必須	初期値	備考
public_group_ctg_item	Array	×	-	
配列インデックス	Object	×	-	
category_cd	String	○	-	
category_item_cd	Array	○	-	
配列インデックス	String	○	-	

```
"criteria": {
  "public_group_ctg_item":
    [{"category_cd": パブリックグループ分類コード,
      "category_item_cd": [パブリックグループ分類項目コード, ...], ...}
  ]
}
```

■ ユーザ分類項目

ユーザ検索時に指定のユーザ分類項目で絞り込みます。

プロパティ名	型	必須	初期値	備考
user_ctg_item	Array	×	-	
配列インデックス	Object	×	-	
category_cd	String	○	-	
category_item_cd	Array	○	-	
配列インデックス	String	○		

```
"criteria": {
  "user_ctg_item":
    [{"category_cd": ユーザ分類コード,
      "category_item_cd": [ユーザ分類項目コード, ...], ...}
  ]
}
```

■ ロール

ユーザ検索時に指定のロールで絞り込みます。

プロパティ名	型	必須	初期値	備考
role	String	×	-	

```
"criteria": {
```

```
"role" : ロール ID
```

■ プライベートグループ

プライベートグループ検索時に指定のプライベートグループオーナーで絞り込みます。

プロパティ名	型	必須	初期値	備考
private_group_owner	String	×	ログイン ユーザID	

```
criteria" : {
  "private_group_owner" : ユーザ ID
```

3.3.3.3 コールバック・返却値

ユーザ選択画面で選択したユーザ情報は、コールバック関数名で指定した関数の引数として返却されます。

戻り値は以下形式オブジェクトの配列になります。

プロパティ名	型	説明
type	String	このオブジェクトの型を表す。主にアプリケーション側で型の判別ができるように提示するもの。アプリケーション共通マスタが標準で提供する機能の範囲では、取得した情報元のテーブル名を設定する。
keyFields	Array	文字列の配列。 data 内で一意性を表すキーとなるプロパティのプロパティ名を配列として保持する。検索画面の選択内容一覧において重複選択を避けるためのガイドとして使用する。 具体的には data から keyFields に設定されたの名前のプロパティを取得し、同一 type を含めて比較して重複をチェックする。
displayName	String	オブジェクトを画面に表示する際に使用する表示文字列
data	Objcet	実際にデータベースから取得したレコードの内容がオブジェクトとして設定される。引数の「取得する情報(prop)」に指定された項目をデータベースから取得し、その名称のプロパティを設定する。

3.3.3.4 大量データモード

大量データモードとは、全件検索やあいまい検索などレスポンス悪化を招く検索を制限するためのものです。具体的には、大量データモードを使用すると以下の制約が発生します。

■ ユーザ検索(キーワード)

- キーワードの入力が必須になります。つまり、全件検索はできなくなります。
- ユーザ検索時に表示数制限(ユーザ)を超えたデータの取得ができなくなります。

■ ユーザ検索(組織)

- 組織階層の移動ごとの検索になります。
- ユーザ検索時に表示数制限(ユーザ)を超えたデータの取得ができなくなります。

- ユーザ検索(パブリックグループ)
 - パブリックグループ階層ごとの検索になります。
 - ユーザ検索時に表示数制限(ユーザ)を超えたデータの取得ができなくなります。
- ユーザ検索(ロール)
 - ユーザ検索時に表示数制限(ユーザ)を超えたデータの取得ができなくなります。

3.3.3.4.1 設定方法

<%Server_Path%>/conf/parameter.xml の以下の箇所を設定します。

大量データモードを使用する場合、<param-value>を true に設定します。

設定は PC 版ユーザ共通検索と共用のパラメータとなりますので十分注意して設定してください。

```
<param>
  <param-name>appcom.search.large_data_mode.user.enabled</param-name>
  <param-value>true</param-value>
</param>
```

3.4 IM-Mobile Framework画面共通IMARTタグの使い方

本フレームワークをインストールすると、3.4.1に示すスマートフォン用タグライブラリを使用できるようになります。APIドキュメント示す書式に従ってHTMLファイルにタグを埋め込むことで、スマートフォン用の入力部品等を使用することができます。

3.4.1 IM-Mobile Framework 画面共通IMARTタグ一覧

下記に本フレームワークにて提供するスマートフォン向けタグの一覧を示します。各タグの詳細な書式や属性の使い方については、APIドキュメントを参照してください。

項番	タグ名	機能概要
1	spIncludeJQM	HTML ファイル内でスマートフォン向け入力部品を使用するためのライブラリの取り込みを行います。
2	spHeaderWithBack	『戻る』ボタンを持つ共通ヘッダを表示します。
3	spHeaderWithMenu	『MENU』ボタンを持つ共通ヘッダを表示します。
4	spCommonFooter	『MENU』ボタン、『ログアウト』ボタンを持つ共通フッタを表示します。
5	spFooterWithLogout	『ログアウト』ボタンを持つ共通フッタを表示します。
6	spFieldContain	入力部品にラベルを生成し、入力部品とラベルの関連付けを行います。
7	spControlGroup	複数の入力部品をグループ化します。
8	spRadioButton	ラジオボタンを表示します。
9	spCheckbox	チェックボックスを表示します。
10	spToggleSwitch	トグルスイッチを表示します。
11	spSlider	つまみを左右に動かすことで数値を入力するための入力部品を表示します。
12	spSearchbox	検索用のテキストボックスを表示します。
13	spPagingButton	複数のページを持つリスト形式の一覧のページ移動を行うためのツールバーを表示します。
14	spCollapsibleList	見出しと詳細情報を持つ開閉式のリストを表示します。
15	spDownload	ストレージサービスからファイルダウンロードの行うための部品です。
16	spDatePicker	日付文字列を参照入力するためのインタフェースを提供します。
17	spCalendar	ユーザーが拡張可能なカレンダーレイアウトを表示します。
18	spCollapsible	開閉型のブロック形式で見出しと詳細情報を取り扱うインタフェースを提供します。

3.4.2 チェックボックスとラジオボタンの実装について

チェックボックス、ラジオボタンは、従来の IMART タグで提供されている `<imart type="input" style="checkbox|radio">` で記述することができます。この方法で記述する場合は、さらに `<label>` タグの記述が必要になり、マークアップ作業が煩雑になりがちになります。このため、本フレームワークから提供している `spCheckbox` および `spRadioButton` の使用をおすすめします。これらのタグはスマートフォン向けに最適化されており、また、HTML の行数を減らすことにも貢献できます。`spCheckbox` および `spRadioButton` の使い方については、API ドキュメントを参照してください。

3.4.3 ファイルのダウンロード用部品の実装方針

スマートフォン端末では、ファイルダウンロードの仕組みが OS ごと(機種ごと)に仕様が異なります。このため、ファイルをダウンロードするための実装方式を考える場合、なるべく端末依存を吸収するような実装方法がないか検討を行う必要があります。

ここでは実装例として、ファイルの存在チェックのコードを示します。下記の例のように実装することで、ファイル存在チェック処理に関しての端末依存を吸収することができますようになります。

< sample_file_exists.html >

```

1 <!doctype html>
2 <html>
3 <head>
4   <imart type=" spIncludeJQM" ></imart>
5   <script src="csjs/im_ajax_request.js"></script>
6   <script src="csjs/im_json.js"></script>
7 </head>
8 <body>
9   <div data-role="page">
10    <imart type="spHeaderWithBack" page="" headerText=$messages.headerTitle/>
11    <div data-role="content">
12      <a id="downloadButton" data-role="button" data-icon="refresh" data-inline=
"true">ファイルダウンロード</a>
13      <script type="text/javascript">
14        jQuery(document).ready(function($) {
15          $('#downloadButton').tap(function() {
16            $.mobile.showPageLoadingMsg();
17            var ajaxUrl = "<imart type="string" value=$url.checkFile/>";
18            ImAjax.requestAsyncSend(
19              ajaxUrl,
20              {
21                "path":$('input[name=path]:hidden').val(),
22                "fileName":$('input[name=fileName]:hidden').val(),
23                action);

```

```

24         });
25         function action(res) {
26             if(res.data.isExist) {
27                 $('form[name=downloadForm]').submit();
28             }else{
29                 alert("ファイルが存在しません");
30             }
31             $.mobile.hidePageLoadingMsg();
32         }
33     });
34     </script>
35 </div>
36 <imart type="spDownload" style="form" name="downloadForm" path="sample/download/"
fileName="sample_file.txt" downloadName="サンプルファイル"/>
37 <imart type="spCommonFooter" />
38 </div>
39 </body>
40 </html>

```

<sample_file_exists.js>

```

1 var $url = new Object();
2 function init() {
3     $url.checkFile = new URL("sample/check_file_exists_ajax").location();
4 }

```

<check_file_exists_ajax.js>

```

1 function init(request) {
2     // JSON形式の文字列からJavaScriptオブジェクトに変換
3     var messageBody = request.getMessageBody("UTF-8");
4     var receiveData = ImJson.parseJSON(messageBody);
5
6     // ファイルパス名
7     var path = receiveData.path;
8     // パスの最後が区切り文字 ("/") ではない場合、 "/" を追加。
9     // パスが空文字列の場合は何もしない。
10    path = path.lastIndexOf("/") == path.length - 1 ? path : path + "/";
11
12    // ファイル名
13    var fileName = receiveData.fileName;
14
15    // ファイルの取得処理
16    var oFile = new VirtualFile(path + fileName);
17
18    // ファイルの存在結果 (true : false)
19    var isExist = oFile.exist();
20

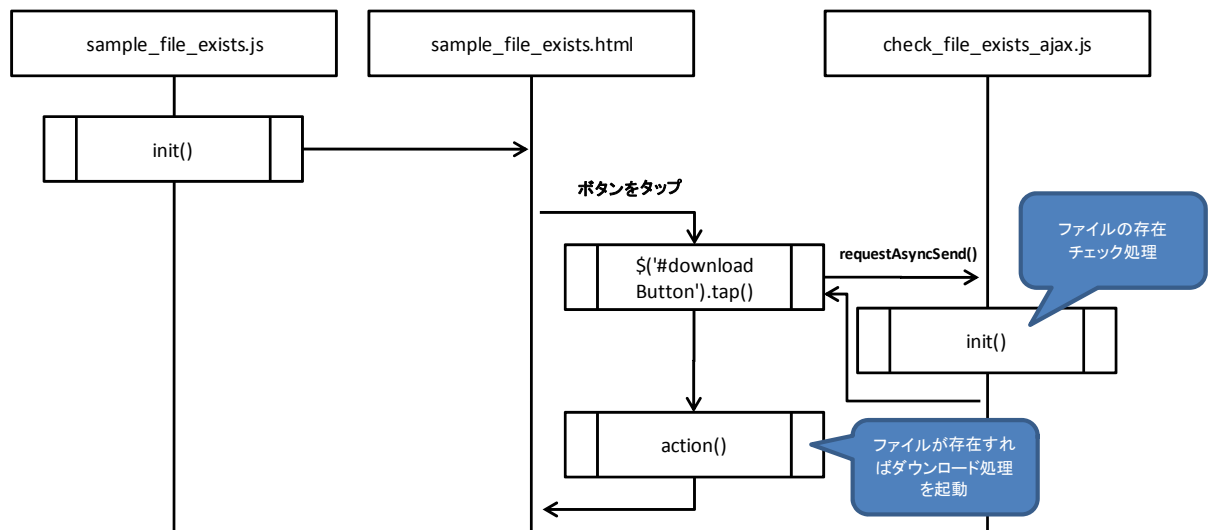
```

```

21 // レスポンスデータを作成
22 var responseData = new Object();
23 responseData.isExist = isExist;
24
25 // クライアントにファイルの存在結果を返却
26 var jsonString = ImJson.toJSONString(responseData);
27 var response = Web.getHTTPResponse();
28 response.setContentType("application/json; charset=UTF-8");
29 response.sendMessageBodyString(jsonString);
30 }

```

<sample_file_exists.html> 12 行目にマークアップされている『ファイルダウンロード』ボタンをタップすることによって、Ajax によるファイル存在チェックを起動します(15 行目～)。チェックの結果は 25 行目の action 関数によって判定を行います。ファイルが存在していた場合は form として定義されている spDownload タグ(36 行目)を submit し(27 行目)、ファイルが存在していない場合は alert を表示します(29 行目)。なお、例ではファイルが存在していない場合のハンドリングとして alert の表示を行っていますが、必要に応じて環境別のハンドリングを行ってください。



3.4.4 日付と時刻の入力について

時刻の入力を行うには、DateBox タグを使用してください。下記にリファレンスサイトを示します。日付の入力に関しては spDatePicker タグを使用することができます(DateBox タグでも日付入力を行うことができます)。spDatePickerを使用する場合は業務カレンダーとの連携も可能です。spDatePickerタグについてはAPIドキュメントを参照してください。

■jQuery Mobile datebox plugin

<http://dev.jtsage.com/jQM-DateBox/>

なお、intra-mart の業務カレンダーとの連携はしていません。

ここでは DateBox タグの概要について説明します。

(1) HEAD タグ内の記述

DateBox タグの実行ファイルは本フレームワークをインストールすることで同時にインストールされます。

DateBox タグを利用するには、各 HTML ファイルの HEAD タグ内で DateBox タグ実行ファイルおよび jQuery Mobile へのリンクを記述してください。下記に記述例を示します。

<記述例1(sample.html)>

1	...
2	<head>
3	<link rel="stylesheet" type="text/css" href="<IMART type="string" value=contextPath> </IMART>/css/mobile_fw/jquery.mobile-1.0b1.min.css" />
4	<link rel="stylesheet" type="text/css" href="<IMART type="string" value=contextPath> </IMART>/css/mobile_fw/jquery.mobile.datebox.css" />
5	<script type="text/javascript" src="<IMART type="string" value=contextPath> </IMART>/csjs/mobile_fw/jquery-1.6.1.min.js"></script>
6	<script type="text/javascript" src="<IMART type="string" value=contextPath> </IMART>/csjs/mobile_fw/jquery.mobile-1.0b1.min.js"></script>
7	<script type="text/javascript" src="<IMART type="string" value=contextPath> </IMART>/csjs/mobile_fw/jquery.mobile.datebox.js"></script>
8	...
9	</head>

HTML 中で使用している contextPath を取得するために JavaScript を記述してください。

<記述例(sample.js)>

1	var contextPath = "";
2	function init(request) {
3	contextPath = Web.getContextPath();
4	}

sample.html の 4 行目と 7 行目が DateBox タグを使用するための宣言です。3 行目、5 行目、6 行目は jQuery Mobile を使用するための宣言です。カスタムスタイルシートやカスタムスクリプトを使用する場合は適宜、記述の追加を行ってください。あとは DateBox タグを BODY タグ内に記述することで DateBox タグを使用することができます。

下記リンクもご参照ください。

■jQuery Mobile datebox plugin – Download and use

<http://dev.jtsage.com/jQM-DateBox/demos/install.html>

sample.html は、下記のように記述することもできます。

<記述例2(sample.html)>

```
1  ...
2  <head>
3    <imart type="spIncludeJQM" />
4    <link rel="stylesheet" type="text/css" href="<IMART type="string" value=contextPath>
      </IMART>/css/mobile_fw/jquery.mobile.datebox.css" />
5    <script type="text/javascript" src="<IMART type="string" value=contextPath>
      </IMART>/csjs/mobile_fw/jquery.mobile.datebox.js"></script>
6  ...
7  </head>
```

(2) DateBox タグのイメージ

ここでは DateBox タグでできることについて簡単に説明します。属性値を指定することにより、様々な見た目や振る舞いを持ちます。属性値の詳細についてはリファレンスサイトをご参照ください。



日付入力の例。DateBox タグを『Calendar Mode』で動作させているところです。いずれかの日付をタップすると、呼び出し元のテキストボックスにその日付が入力されます。『2011 年 8 月』と表記されている部分や曜日ヘッダの部分、日付ボタンの色等はカスタマイズすることができます。



日付入力の例。DateBox タグを『Android Mode』で動作させているところです。『+』や『-』で日付を調整し『設定』ボタンをタップすると、呼び出し元のテキストボックスにその日付が入力されます。他のモードと同様、属性値の設定によってさらにカスタマイズすることができます。



日付入力の場合。DateBox タグを『Slide Mode』で動作させているところです。年月日別に3段にスライド可能なポップアップになっていますので、各項目をスライドして日付を調整し『設定』ボタンをタップすると、呼び出し元のテキストボックスにその日付が入力されます。他のモードと同様、属性値の設定によってさらにカスタマイズすることができます。



日付入力の場合。DateBox タグを『Flip Mode』で動作させているところです。年月日別に3列にフリップ可能なポップアップになっていますので、各項目をフリップして日付を調整し『設定』ボタンをタップすると、呼び出し元のテキストボックスにその日付が入力されます。他のモードと同様、属性値の設定によってさらにカスタマイズすることができます。



時刻入力の例。DateBox タグを『Flip Time Mode』で動作させているところです。時分別に2列にフリップ可能なポップアップになっていますので、各項目をフリップして時刻を調整し『設定』ボタンをタップすると、呼び出し元のテキストボックスにその時刻が入力されます。他のモードと同様、属性値の設定によってさらにカスタマイズすることができます。



時刻入力の例。DateBox タグを『Time Mode』で動作させているところです。『+』や『-』で時刻を調整し『設定』ボタンをタップすると、呼び出し元のテキストボックスにその日付が入力されます。他のモードと同様、属性値の設定によってさらにカスタマイズすることができます。

3.5 IM-Mobile Framework画面共通JavaEEタグの使い方

IM-Mobile Framework Ver7.2.1 から、3.5.2 に示すスマートフォン用タグライブラリを使用できるようになりました。APIドキュメント示す書式に従って JSP ファイルにタグを埋め込むことで、スマートフォン用の入力部品等を使用することができます。

3.5.1 JSPタグディレクティブ

本フレームワーク JavaEE タグを使用する場合、タグライブラリを使用する JSP のページで以下のような taglib ディレクティブを指定する必要があります。

```
<%@ taglib prefix="imsp" uri="http://www.intra-mart.co.jp/taglib/imsp" %>
```

詳しくは API リスト「画面共通 JavaEE タグライブラリ」を参照してください。

3.5.2 IM-Mobile Framework画面共通JavaEEタグ一覧

下記に本フレームワークにて提供するスマートフォン向けタグの一覧を示します。各タグの詳細な書式や属性の使い方については、APIドキュメントを参照してください。

項番	タグ名	機能概要
1	includeJQM	HTML ファイル内でスマートフォン向け入力部品を使用するためのライブラリの取り込みを行います。
2	headerWithLink	ヘッダ部左端に、任意のページに遷移のボタンを備えたヘッダを表示します。
3	commonFooter	『MENU』ボタン、『ログアウト』ボタンを持つ共通フッタを表示します。
4	footerWithLogout	『ログアウト』ボタンを持つ共通フッタを表示します。
5	fieldContain	入力部品にラベルを生成し、入力部品とラベルの関連付けを行います。
6	controlGroup	複数の入力部品をグループ化します。
7	radioButton	ラジオボタンを表示します。
8	checkbox	チェックボックスを表示します。
9	toggleSwitch	トグルスイッチを表示します。
10	slider	つまみを左右に動かすことで数値を入力するための入力部品を表示します。
11	searchbox	検索用のテキストボックスを表示します。
12	pagingButton	複数のページを持つリスト形式の一覧のページ移動を行うためのツールバーを表示します。
13	collapsibleList	見出しと詳細情報を持つ開閉式のリストを表示します。
14	download	ストレージサービスからファイルダウンロードの行うための部品です。
15	datePicker	日付文字列を参照入力するためのインターフェースを提供します。
16	calendar	ユーザが拡張可能なカレンダーレイアウトを表示します。
17	collapsible	開閉型のブロック形式で見出しと詳細情報を取り扱うインターフェースを提供します。

3.6 TIPS

3.6.1 ページ遷移について

通常のページ遷移を行う場合は<IMART type="link">を使用しますが、ファンクションコンテナ内のページへ遷移する場合(PC 用サイトやフィーチャーフォン用サイトのページ)は注意が必要です。この場合は Ajax を OFF にする必要があります。下記にサンプルコードを示します。

なお、スマートフォン用サイト内でのページ遷移における<IMART type="link">を使用する場合は、Ajax を OFF にする必要はありません。

<サンプルコード>

1	<!-- page_pc は PC 用のページです -->
2	
3	<!-- この例はページ遷移が動作しません -->
4	<imart type="link" page="path/to/page_pc">PC サイトへのリンク</imart>
5	
6	<!-- このように記述してください -->
7	<imart type="link" page="path/to/page_pc" data-ajax="false">PC サイトへのリンク</imart>
8	
9	<!-- このように記述することもできます -->
10	<imart type="link" page="path/to/page_pc" rel="external">PC サイトへのリンク</imart>
11	

jQuery Mobile の mobileinit()内で Ajax 処理を OFF にしている場合は、さらに注意が必要です。この場合のページ遷移の挙動について下表に示します。項番3の場合は Ajax によるページ遷移は行われない点に注意してください。

項番	mobileinit()の設定	各 HTML での記述	Ajax の状態	ページ遷移の挙動
1	ajaxEnabled="true" (既定値)	data-ajax="true" (既定値)	有効	Ajax によるページ遷移が行われる
2	ajaxEnabled="true" (既定値)	data-ajax="false"	無効	通常のページ遷移が行われる
3	ajaxEnabled="false"	data-ajax="true" (既定値)	無効 (ajaxEnabled の値が優先される)	通常のページ遷移が行われる
4	ajaxEnabled="false"	data-ajax="false"	無効	通常のページ遷移が行われる

3.6.2 Debug.Browseについて

3.6.1と関連しますが、Debug.browseで表示される画面はAjaxがONになっている場合は表示されないため、Debug.browseを使用したい場合は一時的にAjaxをOFFにしてください。

3.6.3 data-rel="back"時の画面遷移エラーについて

スクリプト開発モデルの場合、異なるプレゼンテーションページ間で Ajax 画面遷移を行い
遷移先の画面から初期表示時の画面へ戻る際、jQuery Mobile 内部の URL 自動変換エラーによる
スクリプトエラーが発生し処理が中止される事象が確認されています。

上記が原因の問題に遭遇した場合、以下のスクリプトを配置することで事象を回避できます。

```
1 <script>
2 (function($) {
3   var firstPage = null;
4   $(document).unbind("pagebeforechange").bind("pagebeforechange", function(e, data) {
5     if (typeof data.toPage === "string" &&
6         data.toPage.indexOf("#") === -1 &&
7         window.location.href.indexOf(data.toPage) !== -1) {
8       data.toPage = $(document.body).children("[data-role='page']:first");
9     }
10  });
11 })(jQuery);
12 </script>
13
14
15
16
```

4 各工程におけるポイント

4.1 開発着手前(要件定義まで)

4.1.1 端末の選定

システムに接続するスマートフォン端末の選定作業は非常に重要です。搭載されている OS が同じ Android だからといっても、機種が異なれば機種依存の問題が発生する可能性があるからです。端末選定の方針としては、できるだけ絞り込む方向性とするをおすすめします。『Android 2.x 系であればどんな端末を使用してもよい』という場合に許容範囲を広げてしまうと、設計側が想定していない端末で利用者側がシステムに接続しようとすることも考えられるからです。このときにシステムが正常に動作するかは設計者側では保証できません。

- 機種を限定できることがベスト
- OS の種類(iOS/Android/BlackBerry/Windows Phone…)はできるだけ絞り込むようにする
- OS のバージョン(Android 1.x 系/2.x 系/3.x 系…)はできるだけ絞り込むようにする
- さらに OS のバージョンだけでなく、リビジョン(Android 2.2/2.3…)でも絞り込むようにする

端末の絞り込みができない場合は、以下の方針で検討を進めるようにします。

- そのシステムが準拠する規格を明確にする(HTML5 でコンテンツ制作する)
- 試験工程で動作確認した機種を利用者に開示し、それらの機種でのシステム利用を推奨する
- クライアントリソースへのアクセスを伴う業務処理等、機種によって実現が難しい業務がないかあらかじめ把握しておく
- システム側で制御可能な機種依存の現象であれば、回避コードを実装できないか検討しておく

4.1.2 画面イメージ

顧客から業務要件をヒアリングし、システム化範囲を検討する際、画面イメージを使用したすり合わせが行われることもあります。このとき、できるだけ画面モックアップ等を作成してすり合わせを行うことをおすすめします。システム動作イメージに近いすり合わせを行うことで、顧客との意識のギャップを埋めることができるだけでなく、設計時のギャップを埋めることもできるからです。また、この段階で実機を利用すると、さらに有効性を高めることができます。なお、画面モックアップは業務スケルトンを活用することで簡単に作成することができます。

4.2 設計工程

4.2.1 画面の設計

2章 スマートフォン向けサイト制作方針をご参照ください。

4.2.2 ビジネスロジックの設計

ビジネスロジックの設計手法としては、PC 向けサイトを構築する場合と大きく変わりはありません。PC 版サイトと同じビジネスロジックを再利用し、生産性と品質を確保する、といった事項を検討し、効率的に設計してください。

4.3 試験工程

4.3.1 試験端末の選定

試験の対象とするスマートフォン端末は、そのシステムの調達範囲に含まれるすべての端末を対象とすべきです。このため、どの機種を調達範囲とするかは非常に重要な決定となります。

実機の代わりにエミュレータを使用することはなるべく避けてください。単体試験のような比較的初期の試験工程であっても特定の機種による不具合が抽出される頻度が最も高い試験工程となりますので、試験端末は原則として実機を使用するようにしてください。このため、試験環境を構成する機器としては、サーバ、スマートフォン端末が必要になります。また、試験環境にインターネット経由でアクセスできない場合は、無線LAN環境も必要になります。

4.3.2 試験観点について

試験観点として抽出すべき観点について、ここでは一例を示します。

試験工程の序盤、単体試験工程では、PC向けサイトの場合と同様、以下の観点で試験を実施してください。

- 分岐網羅
- 繰り返し網羅
- 正常系／異常系の入力値のチェック、境界値チェック等
- ソースコードの全ルート通過

加えて、スマートフォン独自の操作や見栄えに対する試験を実施するようにしてください。

- 画面に表示されている入力部品はスマートフォン用に最適化されたものになっているか(入力部品の見栄え、表示位置、部品の振る舞い)
- 画面遷移のエフェクトが想定した動作になっているか
- フリップ・スライド・端末回転等の操作に問題がないか

それ以降の試験工程でもPC向けサイトの場合と基本的な考え方は同様ですが、総合試験工程等で電波状況に応じた試験を実施してください。

スマートフォン開発ガイドライン Ver. 7.2

2013/4/5 第4版

Copyright© 2013 NTT DATA INTRAMART CORPORATION

TEL: 03-5549-2821

FAX: 03-5549-2816

E-MAIL: info@intra-mart.jp

URL: <http://www.intra-mart.jp/>