

國元 祐二

株式会社ミガロ.

RAD事業部 技術支援課 顧客サポート

# Business4Mobile:jQuery連携で機能拡張

スマートフォンから社内 DB にアクセスできる「Business4Mobile」。スマートデバイスを、さらにビジネス活用するためのテクニックを紹介する。

- はじめに
- Business4Mobile とは
- Business4Mobile の活用
- JavaScript (jQuery) 連携による機能拡張
- スマートフォン連携の機能拡張
- 画像表示の機能拡張
- おわりに



略歴

1979年03月27日生  
2002年追手門学院大学文学部  
アジア文化学科卒  
2010年10月株式会社ミガロ、入社  
2010年10月RAD事業部配属

現在の仕事内容

Delphi/400 や JC/400、  
Business4Mobile の製品試験やサ  
ポート業務などを専門に担当している。

## 1. はじめに

ここ数年で、iPhone や iPad などのスマートフォンやタブレットが広く普及してきた。もちろん、スマートデバイスの活用は個人だけでなく、ビジネスにおいても急速に注目されるようになってきている。

Web プラットフォームの業務システムも現在では一般に使用されているが、これも Web 環境の普及の結果である。同様に今、注目を浴びているスマートデバイスも、これからのビジネス進展の中で必要な環境の1つとなっていくのだろう。

ミガロ. では今年4月、スマートデバイスを業務で有効に活用するためのソリューションとして「Business4Mobile (ビジネスフォーモバイル)」という新製品を発売した。【図1】

本稿では、この Business4Mobile という製品の魅力を伝えるとともに、すでに製品を使用している方々に、さらにビジネス活用するための具体的なテクニックをいくつか紹介したい。

## 2. Business4Mobile とは

Business4Mobile はスマートフォンやタブレットなどのスマートデバイスを利用して、外出先から社内のデータベースを参照することが可能なソフトである。この Business4Mobile には、大きく分けると次の3つの製品特徴がある。

### (1) マルチスマートデバイス対応

1つ目の特徴は、すべてのスマートデバイスから利用できるという点である。

#### 【図2】

Business4Mobile のアプリケーションは、特定のスマートフォンへの対応ではなく、Web ブラウザを使って利用する。そのため、iPhone や iPad、Android などの機種を問わず、あらゆるスマートデバイスから活用できる。もちろん、PC の Web ブラウザでも利用可能である。

### (2) マルチデータベース対応

2つ目の特徴は、さまざまなデータバ

スで利用できるという点である。【図3】

Business4Mobile で使用できるデータベースは、IBM の DB2、DB2/400、Informixをはじめ、Oracle、SQLServer、PostgreSQL など多種多様なデータベースに対応している。

最近では、企業内で複数のデータベースを運用している場合も多い。つまり、Business4Mobile は、上記のデータベースであれば複数のデータベースに接続することができ、スマートデバイス上から利用することが可能になる。

### (3) プログラミング不要

3つ目の特徴は、プログラミングなしで、簡単に利用できるという点である。

Business4Mobile では「クエリー」と呼ばれる条件設定を作成しておくだけで、スマートデバイスからデータベースを参照することができる。

なお、Business4Mobile のクエリーは、IBM i の Query (クエリー) や Microsoft Access のクエリをご存知の方であれば、同じようなものだと考えて

かまわない。【図4】

具体的には、クエリーの設定は Business4Mobile に付属するツール上で、参照したいファイルや項目を選択するだけで設定することができる。プログラミングが必要ないため、プログラム開発者でなくても簡単に使えることが魅力である。

また本稿後半で、プログラム開発者向けに「JavaScript (jQuery)」を使った機能拡張のテクニックを紹介するが、Business4Mobile の標準機能を使う場合には、プログラミングは一切必要としないので安心してほしい。

### 3. Business4Mobile の活用

Business4Mobile を使えば、スマートフォンやタブレットから、社内データベースを利用できるという製品概要を説明してきた。

ここからは、Business4Mobile を使えば、どんな画面や用途に活用できるのかを例を挙げて紹介していきたい。

#### (1) 客先での出荷状況確認

例えば、客先で商品出荷状況を確認したい、といった場合に活用することができる。

図5のように、スマートフォンから条件を指定して製品を検索し、発注情報(出荷番号)から詳しい出荷状況などを入手することで、リアルタイムの情報をお客様に案内することが可能になる。【図5】

#### (2) 急なアポイント情報の取得

例えば、営業が社外に出る場合、今までは担当者自身が顧客の連絡先情報などを詳細に、しかも事前に把握しておく必要があった。

しかし Business4Mobile を使えば、外出先から社内のデータベースを参照することが可能になるので、得意先担当者のアポイント情報などが急に必要になっても社内の人間に依頼したり、いったん帰社するといったことをせずとも情報を確認できるようになる。【図6】

スマートフォンでこうした活用ができれば、外出先でのビジネスをさらに効率よく行えるようになるだろう。

#### (3) メール送信やグラフ活用

Business4Mobile では、前述したように、必要なデータを簡単に抽出することができる。さらに、クエリー結果画面の右下には、メールボタンも用意されている。つまり、このボタンを使用することで、結果を PDF の添付ファイルとしてメール送信することができる。【図7】

またデータを抽出して、表形式で参照するだけでなく、グラフで表示できる機能もついている。そのため、ビジュアル形式のわかりやすい分析に展開して、さらなる業務活用に利用することができる。【図8】

本稿で、Business4Mobile の詳細をすべて説明することはできないが、製品のイメージを多少はつかんでいただけたらどうか。Business4Mobile では、上記のような画面や充実した機能を、だれもが(プログラム開発者でなくとも)簡単に扱うことができる。

そして、プログラム開発者であれば、Business4Mobile を便利なソフトとして使うだけでなく、開発ツールのように扱えないのだろうか、自分のほしい機能を追加していきたいと考える方がおられるかもしれない。

以降からは、そのようなさらなる Business4Mobile 活用を希望されるプログラム開発者に向けて、「jQuery」という JavaScript 連携による機能拡張の可能性について紹介していきたい。

### 4. JavaScript (jQuery) 連携による機能拡張

今回の機能拡張では「jQuery (jQuery (ジェイクエリー))」という JavaScript ライブラリを使用する。jQuery は、HTML を拡張するライブラリである。これを利用することで、さまざまな機能が少しのコードで手軽に実装可能になる。

JavaScript はブラウザで利用できる簡易な言語だが、ブラウザの種類ごとにプログラムの実装方法が異なるといった場合がある。しかし、jQuery ではブラウザの種類による違いを吸収する仕組みを持つため、プログラマーはロジックの実装に専念できるという利点がある。

加えて、JavaScript のライブラリは各種存在するが、jQuery はほかのライ

ブラリと比べると利用者が多く、Web 上での情報取得も容易であり、比較的簡単に利用できる。

また、jQuery のライセンスはフリーのオープンソースのため、ライブラリ内の著作権表示さえきちんと残せば、商用・非商用を問わず自由に利用することが可能である。

さらに、世界中のさまざまな有志が jQuery を利用したサンプルを公開しており、難しい機能でもそれらを参考にすることで、短期間に実装可能になる。

#### jQuery の利用方法

jQuery の利用方法は、最初にライブラリファイルを jQuery 公式サイトにアクセスして、画面右側中央付近にある Download (jQuery) ボタンからダウンロードしておこう。【図9】

ダウンロードのボタンを選択すると、現時点の最新版ファイルが取得できる。ダウンロードしたファイルを HTML で読み込むと jQuery の機能が利用可能となる。

#### 【jQuery 公式サイト】

<http://jquery.com>

#### jQuery の記述方法

jQuery を利用した JavaScript の記述方法は、セレクター API により、操作する要素 (HTML 中のタグ) を選択して、その要素に対して「コマンド」を実行する。また、コマンドは互いを「連鎖」させることができるため、コマンドの結果を、別のコマンドに送ることが可能になる。これらの組み合わせにより、少ないコード記述でさまざまな効果を生み出すことができる点が特徴である。

少し詳しく述べると、通常の JavaScript で要素を取得する場合は、document.getElementById メソッドで要素を取得するが、これはブラウザにより挙動が異なる場合がある。しかし、jQuery では、`$()` 関数を利用して `$("#要素のID名")` で要素を取得可能である。

また、それだけでなく `$()` 関数では、クラス名やタグ名での要素取得やさまざまな「フィルター」を利用することで、簡単に HTML の要素を取得できる仕組みになっている。

図1

**Business4Mobile**

スマートフォン活用 データ検索ツール

# Business4Mobile

(ビジネスフォーモバイル) BUSINESS4mobile

簡単な設定で  
スマートフォンから社内データベースにアクセス

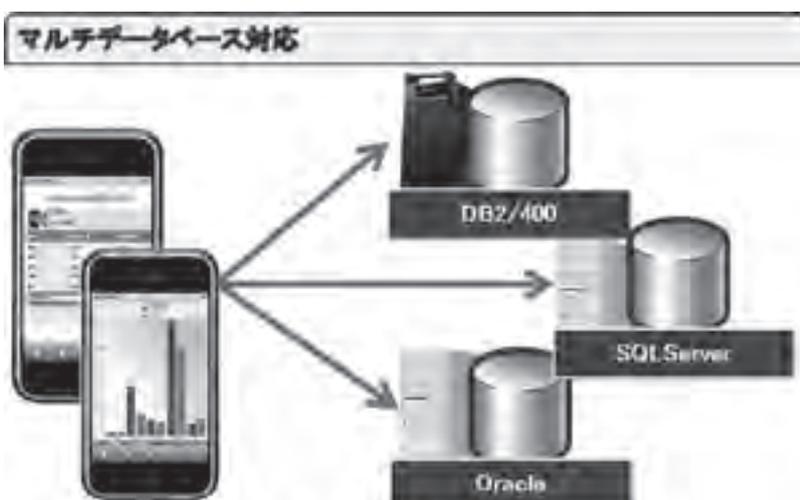


図2

**マルチスマートデバイス対応**



図3



さらに前述したように、jQuery のコマンドの「連鎖 (メソッドチェーン)」もある。メソッドチェーンは名前のおり、メソッドを実行して、その結果に対してさらにメソッドを実行する機能である。

今回の Business4Mobile の拡張でも、上記のさまざまな機能を利用しながら実装を行っていく。

## 5. スマートフォン連携の機能拡張

ここから、Business4Mobile を用いて、スマートフォンのブラウザ機能をさらに連携活用するための機能拡張について説明していきたい。

具体的には、スマートフォン上に表示される実行結果の画面に、リンクを追加し、図 10 に示すように、以下の 3 つの機能呼び出すこととする。【図 10】

### (1) GoogleMap 表示の機能

表示された住所を選択すると、その位置を示す GoogleMap ページを開く。

### (2) 電話発信の機能

表示された電話番号を選択することで直接、電話発信する。

### (3) メール送信の機能

メールアドレスを選択すると、メールアドレスを送信先にした状態で、メールソフトを起動する。

処理のフローは図 11 のようになる。【図 11】

- ①クエリーの実行結果画面の項目タイトル名を確認
- ②キーワードが存在する場合は機能追加処理
- ③キーワードの項目タイトル名を消去
- ④画面表示

そして、実行結果画面は図 12 のようになる。列項目のタイトルに含まれるキーワード (@GMAP) などが削除されており、表示データには、キーワードに対応したリンク機能が追加されている。【図 12】

### 事前設定

JavaScript による実装の事前設定として、以下を行う。

### ●外部 JavaScript ファイルの読み込

Business4Mobile の実行結果画面は、約 10 種類の HTML テンプレートファイルから作成されている。

最初に、Business4Mobile の実行結果画面の HTML テンプレートファイルを変更する。実行結果画面の HTML テンプレートファイルは、resultlist\_template.html である。このファイルに、jQuery への参照と今回追加する外部 JavaScript ファイルの参照を記述する。

まず、jQuery ファイルをリネームする。

jQuery-1.6.2.min.js → jQuery.js

その後、Business4Mobile がインストールされた Apache サーバーに、jQuery のディレクトリを作成して配置する。

インストール先ディレクトリ：  
/jQuery/jQuery.js

続けて、今回追加する JavaScript は、ext ディレクトリを作成して配置する。

インストール先ディレクトリ：  
/ext/MigaroExtension.js

最後に、追加した 2 つのファイルを、外部 JavaScript として HTML から読み込むように設定を行う。【ソース 1】

### ●div タグの追加

Business4Mobile で処理された結果は \$result に設定されるため、id 属性に Extension と設定した div タグで囲む。JavaScript 内で、この id の要素を取得して処理を実行する。【ソース 2】

### ●キーワード定義

実行結果画面に表示される列項目タイトルの見出しを判別して、さまざまな機能を付加する。そのためには、Business4Mobile のツールからクエリーを設定する時点で、キーワードとなる項目名の設定が行われている必要がある。

それらのキーワードは JavaScript で定義している。【ソース 3】

### ●キーワード設定

Business4Mobile のツール上でクエ

リーを設定する際には、図 13 のように、列の項目タイトル名に @GMAIL や @TEL、@EMAIL などのキーワードをつけておく。そして、キーワードの有無で表示される項目に対して、機能の追加処理を行う。【図 13】

### JavaScript の実装

いよいよ JavaScript の実装に入る。

### ●「一覧形式」or「詳細形式」の判別

Business4Mobile の実行結果画面は、図 14 のように「一覧形式」と「詳細形式」の 2 種類が用意されている。

一覧形式は、表形式のデータ表示で上部に項目のタイトル表示があり、その下に複数のレコードを表示することができる。一方、詳細形式は、1 レコードのデータを縦に並べ、たくさんの項目を表示することができる。【図 14】

一覧形式と詳細形式では、出力される Table タグの構造が異なるため、実行結果画面をプログラムで判別する必要がある。識別には、出力される HTML 内の特定のタグの存在有無で判断する。実行結果は一覧形式、詳細形式ともに、HTML の Table 要素として出力されている。

一覧形式の実行結果内の <td> 要素には、Business4Mobile が出力する html に titleWhite クラスが存在する。そのため、<td> 要素に titleWhite のクラスが存在する場合は一覧形式であり、存在しない場合には詳細形式と判断する。【ソース 4】

### ●jQuery を利用した要素取得

詳細形式を表示する場合には、左側に列の項目、右側にデータが表示されるため、左側の列の項目の <td> 要素を jQuery を利用して取得する。

このような場合、jQuery の \$() のフィルター機能を利用することで、簡単に要素を取得することができる。

今回、判別のために取得したい内容は、列の項目タイトル名のため、左側の <td> 要素のみとなる。今回は、even フィルター (偶数の要素取得) を利用して取得する。

\$("#Extension td:even")

図4

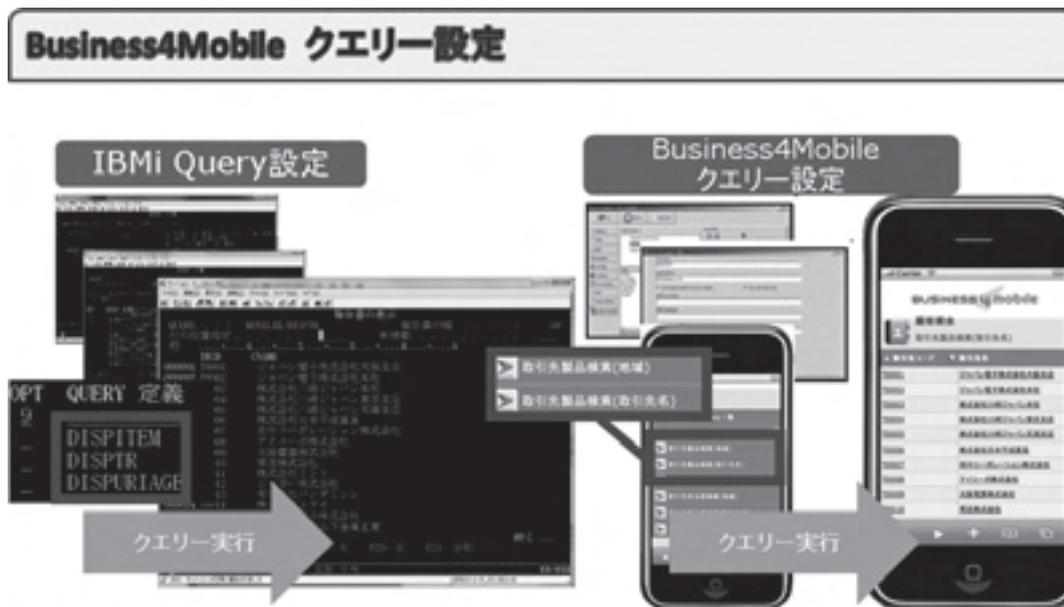


図5



図6



このセレクターの内容は“id 属性が Extension の要素内に含まれる <td> 要素で、偶数の <td> タグのみを取得する”という内容となっている。【ソース 5】

なお詳細を後述するが、取得した <td> 要素内の列項目を取得後に、項目のタイトルの見出しの文字を走査して特定のキーワードが設定されている場合には、タグを書き換える操作を行う。

また前述したように、@GMAP などのキーワード文字に、タイトル項目名での設定内容が一致すると、そのキーワードに対応する機能を追加する処理が行われる仕組みとなっている。

それらタグ要素への操作も、jQuery を利用することで容易に行える。

以降から、データを表示している <td> 要素に対して、アンカー機能をそれぞれ追加して、GoogleMap へのリンクや電話番号のリンク、メーカー起動のリンクなどを実現していこう。各リンク機能についておのおの説明する。

## GoogleMap へのリンク

GoogleMap へのリンク機能は、データタグ要素を、GoogleMap へのリンクが設定された jQuery の wrapInner メソッド（各要素の子要素を、HTML で作成した要素で囲む）を利用して <a> タグではさみこみ、要素が選択された際に GoogleMap のページを表示する仕組みになっている。

注意点としては、GoogleMap への URL の引数に住所を渡す際に、URL エンコーディングが必要となる点が挙げられる。URL エンコーディングには、JavaScript の encodeURIComponent 関数を利用する。【ソース 6】

処理前

```
<td> 大阪府浪速区湊町 2-1-57</td>
```

処理後

```
<td><a href='http://maps.google.co.jp/maps?q='  
( エンコードされた住所)'' target  
='_blank'> 大阪府浪速区湊町 2-1-57</a></td>
```

jQuery を用いることで、少しの実装で GoogleMap へのリンク機能を追加す

ることが可能となっている。また、ほかのリンク機能もこの応用で作成できる。

## 電話発信へのリンク

スマートフォンからの電話発信のリンク機能は、電話番号のデータ要素に <a> タグを追加後、href 属性に「tel: 電話番号」と設定することで実現できる。【ソース 7】

処理前

```
<td>06-6631-8601</td>
```

処理後

```
<td><a href='tel:06-6631-8601'>06-6  
631-8601</a></td>
```

このように記述しなければ、電話発信はスマートフォンのブラウザ機能に依存してしまう。端末の種類によっては利用できない可能性が生じるが、機能を実装することにより、スマートフォンの機種を問わず電話発信が可能となる。

## メール送信のリンク

スマートフォンからのメール送信のリンク機能も、電話発信と同様に、メールアドレスのデータ要素に <a> タグを追加する。メーカーの起動は、href 属性に「mailto: メールアドレス」と設定することで利用可能となる。【ソース 8】

処理前

```
<td>info@test.co.jp</td>
```

処理後

```
<td><a href='mailto: info@test.co.jp'  
> info@test.co.jp </a></td>
```

以上で、3つのリンク機能が追加された。

## 列タイトル項目名置換

最後に、キーワードが項目名に表示されてしまうため、キーワード項目の文字の置換を行い、実行結果画面へ表示しないように処理を行っておこう。【ソース 9】

処理前

```
<td>@GMAP 地図 </td>
```

処理後

```
<td> 地図 </td>
```

以上の手順により、Business4Mobile に GoogleMap 連携、電話発信、メール送信とリンク機能を簡単に追加することができた。

そのほかにもこのリンク機能を応用することで、Web サーバーに配置された Word や Excel、PDF ファイルの参照なども設定次第で容易に実現可能である。

## 6. 画像表示の機能拡張

Business4Mobile のデータベースのデータの出力結果を加工して、図 15 のように画像を表示する機能を追加する。【図 15】

### img タグの追加

Business4Mobile の実行結果画面に画像を表示するには、まず、クエリー作成時に列項目見出しにキーワード「@IMG」を設定しておき、また、ファイルの列のデータにあらかじめ、Web サーバー上に配置してある画像のパス (URL) を入力しておく。

設定した列項目タイトルの見出しをプログラムが判別した場合には、img タグを追加して、img タグの src 属性にファイルの URL を指定することで画像を表示する。

これら要素の追加も、jQuery を利用すれば簡単に行える。

```
$( '<img/>' ).attr( 'src', 画像パス ).  
appendTo( $( argDataCol ) );
```

このたった1行のコードで実装ができる。これは、jQuery のメソッドチェーンの機能を利用しているため、img タグを追加後、src 属性に画像パスの URL を設定して、img タグを現在の列要素 (td 要素) へ追加している。

このように複雑な内容の処理も、少ない行数で記述できるのである。【ソース 10】

処理前

```
<td>http://exampleServer.jp/example.  
jpg</td>
```

処理後

```
<td><img src='http://exampleServer.jp/  
example.jpg'/></td>
```

図7



図8



図9



今回の例のように、動的に HTML の要素が追加できるため、実装によっては、Business4Mobile で Flash や動画などを埋め込むという拡張も考えられる。

## 7. おわりに

Business4Mobile は、もともとパッケージソフトなのでプログラミングを必要とせず、製品の機能だけで、スマートフォンやタブレットから簡単に活用することができる。本稿では最初に、この Business4Mobile の特徴や機能を述べた。

続いて、Business4Mobile に対して、jQuery を利用した JavaScript のプログラミングについていくつか紹介したが、プログラミングによって、こうした標準以上の機能を独自に追加できる拡張性を知っていただければ幸いである。

JavaScript を使われたことがある方であれば、そのような Business4Mobile の機能拡張も簡単に実現できるであろう。また、JavaScript は習得するのも比較的容易なプログラム言語なので、初めての方もぜひ、本稿を参考にして機能拡張にチャレンジしていただきたい。

**M**

図10



図11

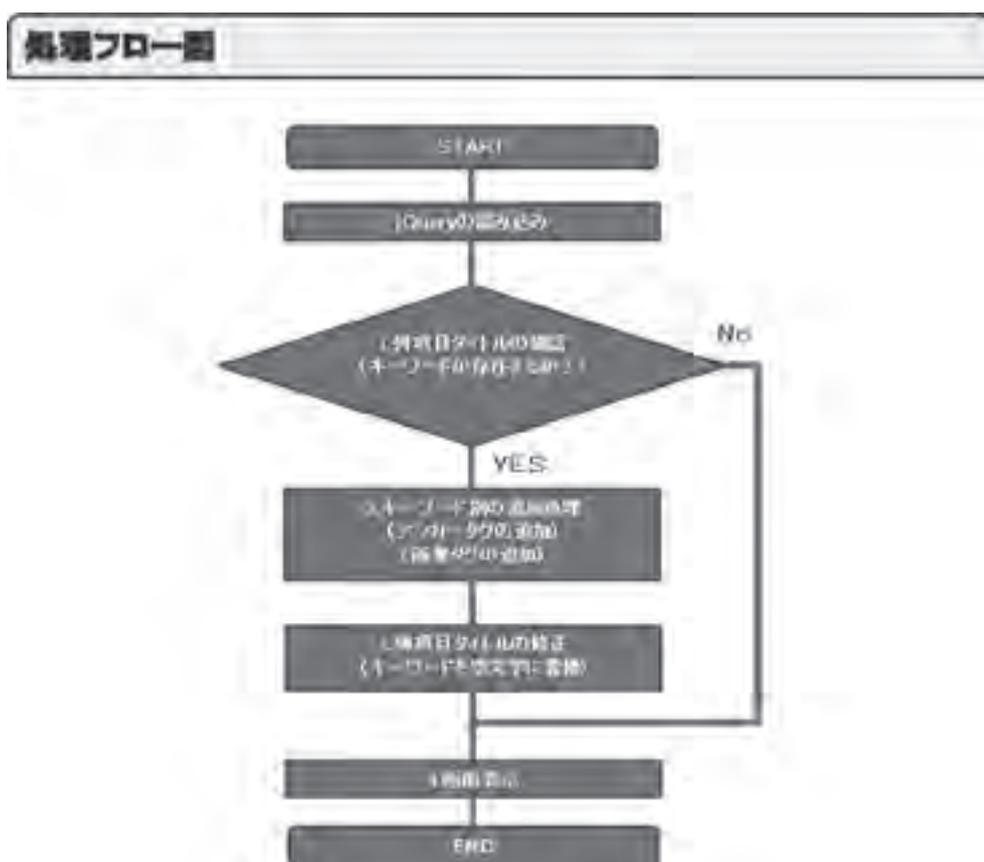


図12



ソース1

### HTMLから外部JavaScriptファイルの読み込み設定

◆resultlist\_template.htmlの編集

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/
html>"+
<head>+
<title>Business4mobile - Results</title>+
<meta http-equiv="Content-Type" content="text/html; charset=x-sjis">+
<meta name="GENERATOR">+
  content="WebSphere Development Studio Client for iSeries">+
<meta name="viewport" content="width=device-width" />+
<meta name="viewport" content="height=device-height" />+
<meta name="apple-mobile-web-app-capable" content="yes" />+
<meta name="apple-mobile-web-app-status-bar-style" content="black" />+
<script type="text/javascript" src="./query/jquery.js"></script> +
<script type="text/javascript" src="./ext/NigaroExtension.js"></script> +
<meta name="format-detection" content="telephone=yes,address=yes,email=yes" />+
+
<link rel="stylesheet" href="JP/theme/waster.css" type="text/css">+
</head>+
<body >+
```

ソース2

### divタグの追加

◆resultlist\_template.htmlの編集

```
+
<tr>+
+
  <td+
    <table width="100%" border="0" cellpadding="0" cellspacing="0">+
      <tr class="header">+
        <td align="left" height="60" colspan="2">${foldername}</td>+
      </tr>+
    </table>+
  </td>+
</tr>+
<tr>+
  <td align="left"><div id="Extension">${result}</div></td>+
</tr>+
<tr>+
  <td align="center" class="backline" height="35" valign="middle">${query_list}</td>+
</tr>+
<tr class="footer">+
  <td height="50" align="right" valign="middle">${buttons_bar}</td>+
</tr>+
</table>+
</left>+
</form>+
</body>+
</html>+
```

### キーワード定義

◆ MigaroExtension.js

```
//キーワード配列
var _KEYWORD_ARR;
_KEYWORD_ARR = new Array();
//クエリの列項目タイトル「キーワード」
_KEYWORD_ARR[COL_NORMAL_LINK] = "LINK" //通常リンクキーワード
_KEYWORD_ARR[COL_POPUP_LINK] = "ePOP" //ポップアップリンクキーワード
_KEYWORD_ARR[COL_TEL_LINK] = "eTEL" //電話番号リンクキーワード
_KEYWORD_ARR[COL_MAIL_LINK] = "eEMAIL" //メールリンクキーワード
_KEYWORD_ARR[COL_GMAP_LINK] = "eGMAP" //マップ表示キーワード
_KEYWORD_ARR[COL_IMG_DISP] = "eIMG" //画像表示キーワード
```

図13



図14



#### ソース4

### 「一覧形式」 or 「詳細形式」の判別

◆MigaroExtension.js

```
/**
 * 一覧形式 or 詳細形式の判別
 * @since 2011/08/09
 * @update
 * @author Migaro.support
 * @param
 * @return DISPLAY_LIST List形式 or DISPLAY_RECORD Record形式
 */
function checkListorRecord(){
  var selector=$("#titleWhite");
  //一覧形式の場合は、titleWhiteクラスが存在する。
  if(selector.length == 0){
    return DISPLAY_RECORD; //詳細形式
  }else{
    return DISPLAY_LIST; //一覧形式
  }
}
```

#### ソース5

### jQueryを利用した要素取得

◆MigaroExtension.js

```
}else if(DisplayType == DISPLAY_RECORD){
  //列項目タイトルのid要素を取得するセレクタ
  var selector=$("#ExtensionId:even");
  selector.each(function(){
    //列項目タイトルの見出しを取得
    var coltext=$(this).text();
    //拡張タイプの取得
    var resultCustomType = checkTitleFunction(coltext);
    //通常表示以外の場合
    if(resultCustomType != OOL_NORMAL){
      //拡張表示
      customtag(resultCustomType, $(this).next());
      //見出しの修正
      replaceTagLeaderTitle($(this));
    }
  });
}
```

#### ソース6

### GoogleMapのリンク追加

◆MigaroExtension.js

```
/**
 * @log追加 googleMap
 * @since 2011/08/13
 * @update de
 * @author Migaro.support
 * @param argDataCol - idタグ要素
 * @param argAddress - 住所
 * @return
 */
function createMapLink(argDataCol, argAddress){
  var temp = "<a href='http://maps.google.co.jp/maps?qs="
  + encodeURIComponent(argAddress) + "' target='_blank'></a>";
  $(argDataCol).wrapInner(temp);
}
```

## ソース7

## 電話発信のリンク追加

◆MigaroExtension.js

```

/**
 * tag追加 電話番号Link
 * @since 2011/06/13
 * @update
 * @author Migaro-support
 * @param argDataCol - tdタグ要素
 * @param argLink - linkURL
 * @return
 */
function createTelLink(argDataCol, argLink){
    var temp = "<a href='tel:' + argLink + '></a>";
    $(argDataCol).wrapInner(temp);
}

```

## ソース8

## メールのリンク追加

◆MigaroExtension.js

```

/**
 * tag追加 EmailAddress
 * @since 2011/06/13
 * @update
 * @author Migaro-support
 * @param argDataCol - tdタグ要素
 * @param argEmailAddress - EmailAddress
 * @return
 */
function createMailLink(argDataCol, argEmailAddress){
    var temp = "<a href='mailto:' + argEmailAddress + '></a>";
    $(argDataCol).wrapInner(temp);
}

```

## ソース9

## 列タイトル項目名置換

◆MigaroExtension.js

```

/**
 * Header項目変更
 * @since 2011/06/13
 * @update
 * @author Migaro-support
 * @param argTitleCol - <td>タグ要素
 * @return
 */
function replaceTagHeaderTitle(argTitleCol){
    if( $(argTitleCol).context.firstChild.id.toLowerCase() != "table"){
        for( key in _KEYWORD_ARR ){
            $(argTitleCol).text( $(argTitleCol).text().replace(_KEYWORD_ARR[key], ""));
        }
    }else{
        var tag = $(argTitleCol).context.firstChild.tagName;
    }
}

```

図15



ソース10

```

imgタグの追加
◆MigaroExtension.js
/****
 * 画像表示
 * @since 2011/06/13
 * @update
 * @author Migaro-support
 * @param argDataCol - tdタグ要素
 * @param argImgSrc - 画像ソース
 * @return
 */
function createImg(argDataCol, argImgSrc) {
    $(argDataCol).text("");
    $('<img />').attr('src', argImgSrc).appendTo($(argDataCol));
}

```