Valence

Valence モバイルアプリケーション開発テクニック

株式会社ミガロ.
プロダクト事業部技術支援課
尾崎　浩司

略　歴
生年月日:1973年8月16日
最終学歴:1996年　三重大学 工学部卒業
入社年月:1999年10月　株式会社ミガロ. 入社
社内経歴:
1999年10月　システム事業部配属
2013年04月　RAD事業部(現プロダクト事業部)配属

現在の仕事内容:
ミガロ. が取り扱う3つの開発ツールのセミナー講師や技術支援を主に担当している。

1.はじめに
2.Valence Mobileアプリの使用方法
3.Valenceモバイルアプリケーション作成の基本
4.デスクトップとモバイルの共用テクニック
5.デバイス機能活用テクニック
6.さいごに

1.はじめに

Valenceは、IBM i(AS/400)を使用したモダナイゼーションツールである。Valenceは登場当時、Senchaと呼ばれるJavaScriptをベースとしたUIフレームワークを使用して画面を設計し、ビジネスロジックをRPGでコーディングする事でアプリケーションを開発するものであった。当時よりモダンなWebアプリケーションが作成できる環境として好評であったが、開発にはSenchaのスキルが必須で、若干敷居が高かったのも事実である。そこで、より簡単にアプリケーション開発が行えるようにと、2018年に登場したValence 5.2からは、「Valence App Builder」が追加で搭載され、新たにローコード開発機能がサポートされるようになった。Valence App Builderの登場により、Valenceを使用するアプリケーション開発生産性は大幅に向上し、現在ではこれが、Valenceの主力機能になっている。
これまでにも、本テクニカルレポートにて、Valence App Builderの開発手法を紹介してきたが、それらは、主にPCブラウザ(デスクトップ)向けアプリケーションのトピックが中心であった。Valenceは、PCブラウザ向けアプリケーションだけでなく、もちろんモバイルアプリケーションの開発も可能である。そこで、今回は、Valence App Builderを使用したモバイルアプリケーション開発に焦点を絞って、その開発テクニックを紹介する。

2.Valence Mobileアプリの使用方法

Valenceにおいて、スマートフォンやタブレット等のモバイルデバイスを使用する場合、ブラウザではなく、専用のアプリから実行する。Valence5.2以前では、「Valence Portal」という専用アプリを使用していたが、現在はValence 6.0以降に対応した新しい「Valence Mobile」アプリに変更されている。iOSやAndroidの公式アプリストアの中にValence Mobileアプリが登録されているので、始めにアプリをデバイスにダウンロードし、インストールすればよい。【図1】

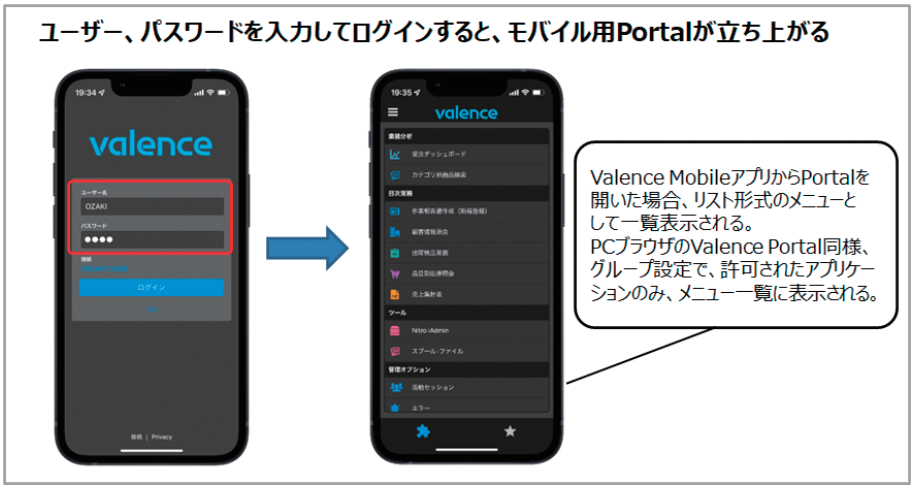
図 1　Valence Mobileアプリ

アプリをインストールしたら、ホーム画面から「Valence Mobile」アイコンをタップしてアプリを起動する。初回起動時に初期設定として、IBM iへの接続先登録画面が表示されるので、「接続先名」と「URL」を登録する。「URL」は、PCブラウザで、Valence Portalにアクセスする場合と同様に、IBM iのIPアドレスおよびValenceを実行するインスタンスに相当するポート番号をURL形式で指定すればよい。【図2】

図 2　Valence Mobileアプリ 初期設定

初期設定が完了すると、Valence Portalへのログイン画面が表示されるので、PCブラウザの場合と同様に、有効なユーザーとパスワードを入力してValenceにログインすると、モバイル用のValence Portalが立ち上がる。PCブラウザの場合は、タイル形式のアイコンメニューとして開くが、Valence Mobileアプリの場合は、リスト形式のメニューとして表示される。**【図3】**

図 3　Valence Mobileアプリ Portalへのログイン



もちろん、[ポータル管理]→[グループ]で設定した権限設定が有効な為、許可されたアプリケーションのみがメニューにリスト表示される。さらに[ポータル管理]→[アプリ]の中で、個別のアプリケーション設定を開き、設定画面の中で、[デスクトップ]と[モバイル]を個別にValence Portalに含めるかどうか指定できるようになっている。つまりValence Mobileアプリから利用した時のみ、実行を許可するという設定も可能である。**【図4】**

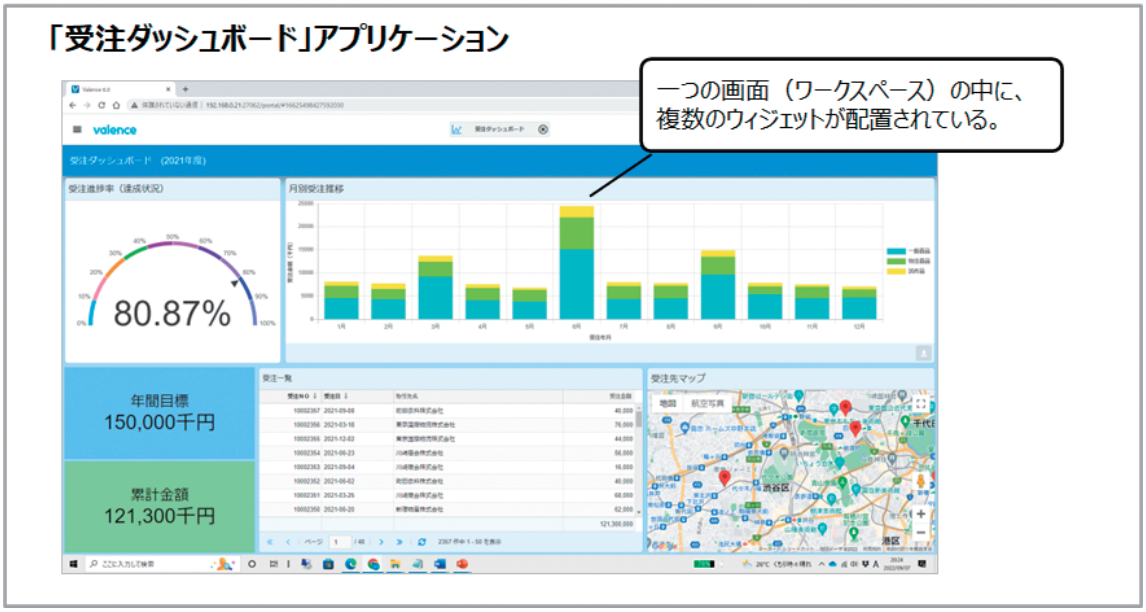
図 4　[ポータル管理]→[アプリ]設定画面

Valence MobileアプリからValence Portalを開けば、後はPCブラウザ同様に実行したいアプリケーションをメニュー一覧からタップすればよい。
なお、Valence MobileアプリからValenceを利用する場合、Wi-Fiや4G/5G通信回線からのアクセスになる点には注意してほしい。社内ネットワーク内のWi-Fiを使用するのであれば、そのままIBM iに接続できるが、外部から通信回線を使用したアクセスを想定する場合は、予めVPNを設定して、社内ネットワークに事前にログインしておく必要がある。
ちなみにVPNを使用せずに直接インターネット経由でValence Mobileアプリを使用したい場合は、別途SSL環境を構築して運用する方法もあるが、本稿では取り上げない。
興味のある方は、Valence開発元CNX社のblog記事(https://www.cnxcorp.com/blog/setting-valence-ibm-i-external-access-ssl)等を参考にしてほしい。

3.Valence モバイルアプリケーション作成の基本

本節よりValence App Builderを使用した具体的なモバイルアプリケーション作成方法を紹介する。
まず、PCブラウザ(デスクトップ)向けアプリケーションをそのままモバイルで実行するとどうなるかを確認する。ここでは例として、「受注ダッシュボード」アプリケーションを使用する。**【図5】**

図 5　PCブラウザ(デスクトップ)アプリケーション例



このアプリケーションは、一つの画面（ワークスペース）にChartやGrid、Map、Single KPIといった複数のウィジェットを合計6個配置して、複数の要素を一括表示するものである。

同じアプリケーションをValence Mobileアプリから、スマートフォンで実行する。その実行結果が、**【図6】**である。

図 6　PCブラウザ向けアプリケーションをそのまま実行

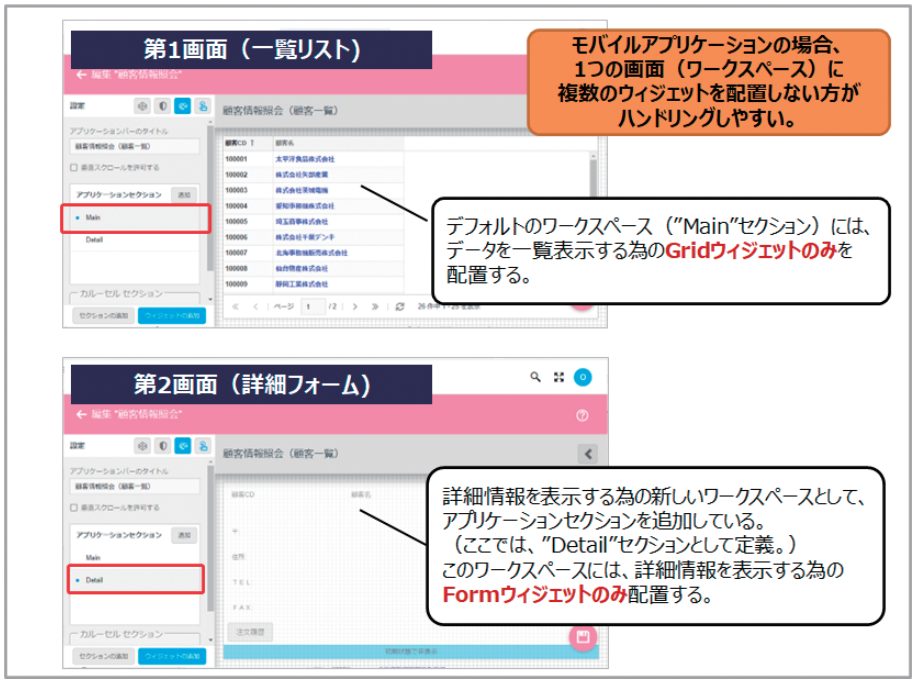
Valence Mobileアプリから実行すると、ワークスペース内の各ウィジェットが自動的に分割され、ウィジェット単位で画面表示される。そして、各ウィジェットはスワイプ操作や画面下部の点（インジケーターアイコン）のタップ操作で画面切り替えを行う形になる。このようにValenceでは、特に意識しなくても、自動的にモバイル用に画面が最適化されるのである。

Valenceでは、実行環境に合わせて表示方法を自動的に変更するので便利である。しかし、画面は自動分割となる為、PCでの画面のレイアウトだけを意識して作成したものを、そのままモバイルデバイスから実行すると意図しないレイアウトや挙動になる可能性もある。

つまり、モバイル向けアプリケーションとして作成する場合、意図的にモバイルでの利用を前提として開発を行った方が、想定通りの動作にできる。

具体的には、モバイルの場合、一つの画面には、同時に一つのウィジェットだけが表示される仕様となる点を意識すればよい。例として、Gridウィジェットに一覧出力されたデータの中から、行を選択し、詳細情報をFormウィジェットで確認するといった動きのアプリケーションを考えてみる。この場合、App Builderのアプリケーション作成ステップにおいて、2つのアプリケーションセクション（ワークスペース）を設定し、1つ目のワークスペースには一覧リストを表示するGridウィジェットを、2つ目のワークスペースには詳細情報を表示するFormウィジェットを配置する構成にすればよい。**【図7】**

図 7　モバイルに適したアプリケーション設定



そして、アプリケーションの[動作内容]で、次のような設定を行う。

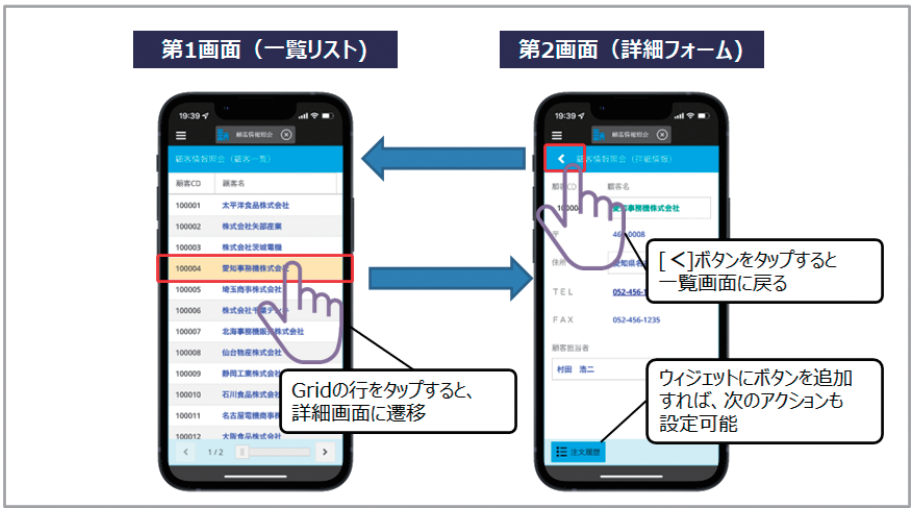
1つ目のワークスペースにあるGridウィジェットの行クリック時に、Formウィジェットに対する[フィルタアクション]と[ウィジェットの表示/非表示]のアクションを設定する。

2つ目のワークスペース上には、前画面に[戻る]ボタン(ここでは、スマートフォンで分かり易いよう[<]マークのアイコンを設定)を追加する。追加したボタンのクリック時に、[ウィジェットの表示/非表示]アクションを設定し、1つ目のワークスペース上のGridウィジェットを再表示させる。**【図8】**

図 8　アプリケーション設定[動作内容]画面

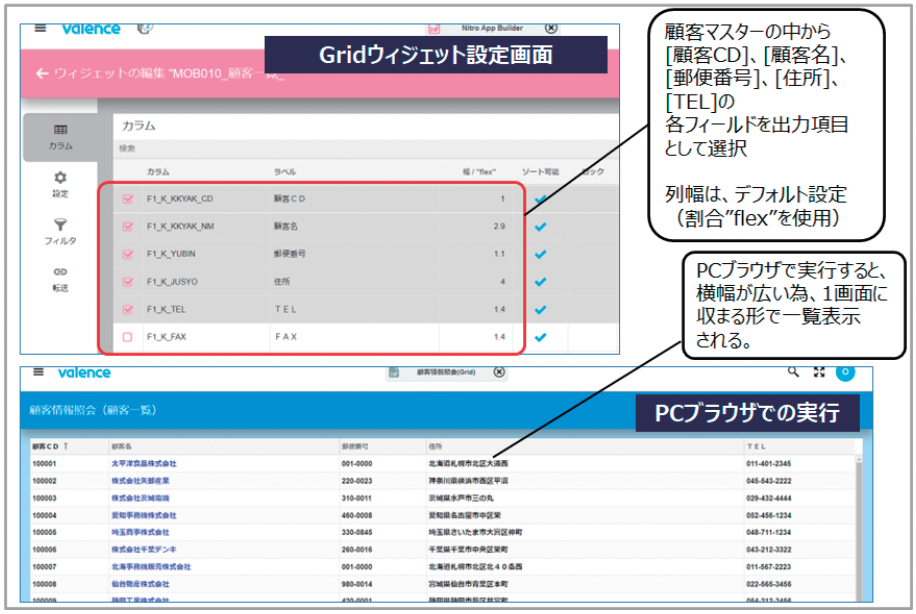
このように、明示的に必要な画面の遷移を[動作内容]に定義することで、意図しない画面遷移を防ぐことができ、モバイルに最適化した画面遷移とする事が可能である。**【図9】**

図 9　モバイルアプリケーション実行例



次にウィジェット作成におけるモバイルでの考慮点を説明する。Valenceのウィジェットの中で一番多用するのが、データを一覧形式で出力するGridウィジェットだろう。例えば、**【図10】**は、顧客情報を一覧表示するGridウィジェットの設定例と、それをPCブラウザで実行した場合の実行画面である。

図 10　Gridウィジェット設定とPCブラウザでの実行例



同じアプリケーションをValence Mobileアプリから実行した場合の画面が、**【図11】**である。同じ内容をもちろん表示できるのだが、特に横方向の画面幅の狭いスマートフォンでは、Grid内の表部分をスワイプ操作で横スクロールしながら閲覧する必要がある。

図11 スマートフォンでの実行例

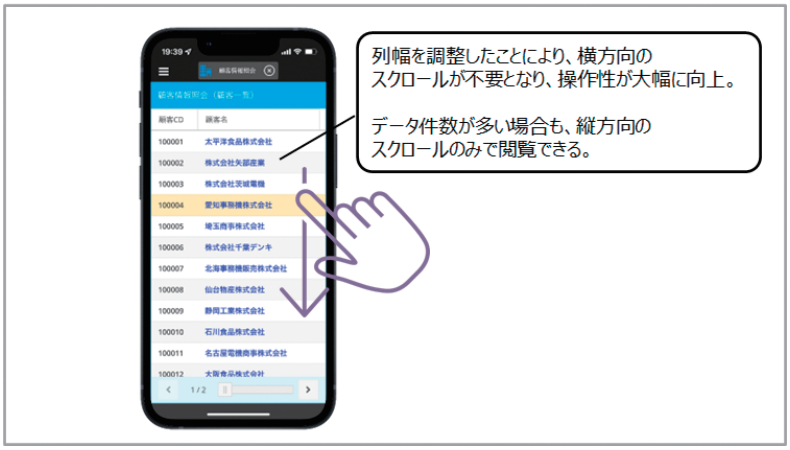
片手操作が基本となるスマートフォンにおいて、横方向のスワイプによるGridの横スクロールはお世辞にも使い勝手が良いとは言えない。では、Gridの画面設計をどうすればよいかであるが、これは各カラムの列幅をスマートフォンの画面サイズにあわせて、明示的にピクセル値を設定して調整すればよい。

筆者が使用しているiPhone13miniをターゲットデバイスとした場合では、各カラムの列幅の合計値を360ピクセル程度に収めると、横スクロール無しに、データを表示させる事が可能である。ここでは、出力する項目を2項目に絞り込み、実際に出力されるデータ文字長から勘案して、[顧客CD]を100ピクセルに、[顧客名]を260ピクセルに設定した。**【図12】**

図12 スマートフォンに最適化したGridウィジェット設定

スマートフォン用に調整した顧客一覧画面では、横方向のスクロールは無くなり、縦方向のみのスクロールでデータが参照できる。**【図13】**

図13　列幅調整後のスマートフォンでの実行例



モバイルの場合、デバイスにより画面のサイズや解像度が異なる為、開発の際には、ターゲットデバイスによる実機テストを行う事を推奨するが、一般的なiOSやAndroidのスマートフォンであれば、横幅は、合計で300～400ピクセル程度になるように設定すればよいだろう。

Gridウィジェットの次に使用頻度が高いのが、詳細情報を表示するのに最適なFormウィジェットだろう。Formウィジェットも、基本的な考え方は、同様だが、原則としては、一つの項目を1行に配置し、横方向に複数項目を配置しない事である。次の図は、顧客に関する詳細情報を表示するFormウィジェットの設定例である。**【図14】**

図14　スマートフォンに最適化したFromウィジェット設定

Formウィジェットの場合、横幅をピクセル設定しなくても、1行に1項目であれば、Flex(割合)=1としておけば、デバイスにあわせて横幅が調整される。1行に2項目の場合でも、例えば、1:2のような割合で指定すればよい。

【図14】のFormウィジェットの設定をValence Mobileアプリから実行した画面が【図15】である。スマートフォン上で違和感のない配置になっている事がわかる。

図15 Fromウィジェットのスマートフォンでの実行例

本節では、主にスマートフォンからValence Mobileアプリを実行する事を前提に作成ポイントを紹介した。スマートフォンの場合、デバイスを縦向きで使用する事が前提となるが、タブレットの場合、横向きでの使用が可能な場合もある。例えば、本節冒頭で例にあげた「受注ダッシュボード」アプリケーションは、iPadであれば横向きで実行する事により、PCブラウザの場合と同様に複数ウィジェットが同時表示される事がわかる。【図16】

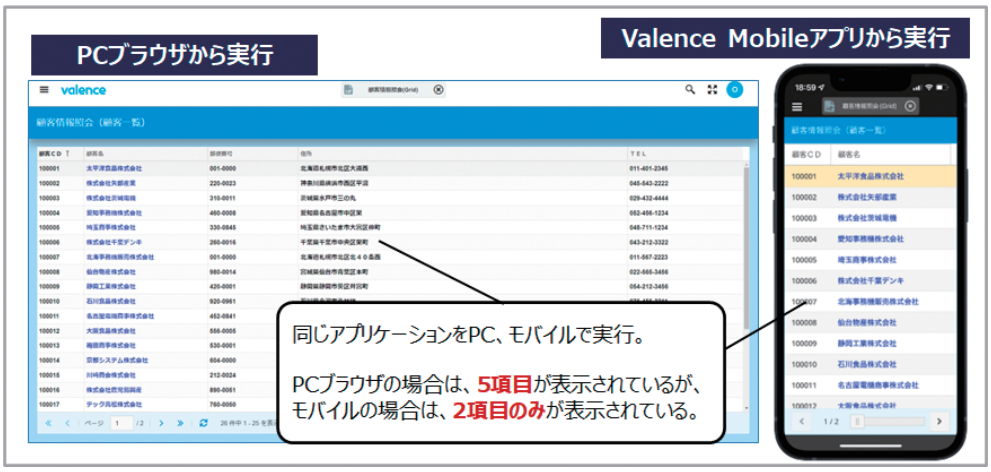
図16 タブレット(iPad)横向きでの利用

4.デスクトップとモバイルの共用テクニック

前節では、モバイルに特化したアプリケーションを作成する場合の手法を紹介した。モバイル専用のアプリケーションを作成する場合は問題ないが、一つの同じアプリケーションをPCブラウザとモバイルとで共用しながらも、実行環境にあわせて画面を自動的に最適化したい場合もあるだろう。本節では、アプリケーションをPCブラウザ(デスクトップ)とモバイルで共用するテクニックを紹介する。

ここでは、前節での例でも使用した顧客情報を一覧リスト出力するGridウィジェットを用いたアプリケーションを使用する。**【図17】**が完成したアプリケーションの実行例である。同じアプリケーションであるにも関わらず、PCブラウザで実行した時と、Valence Mobileアプリから実行した時とで、Gridウィジェットの出力カラムが異なる事がわかる。

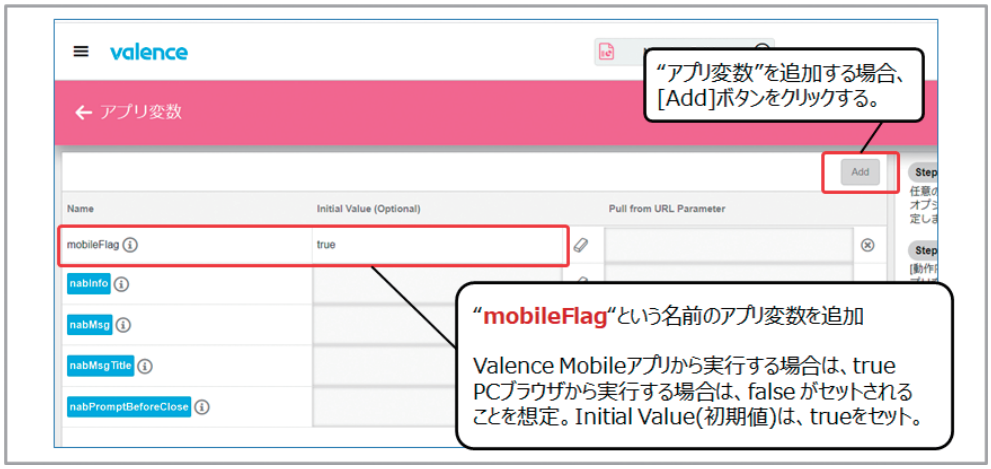
図 17　アプリケーションの共用(デバイスにあわせた最適化)



これは、実行する環境(デスクトップかモバイルか)にあわせて項目単位で、表示/非表示を動的に切り替える事で実現をしている。その実装方法を紹介する。

Valence App Builderではアプリケーション毎に[アプリ変数]と呼ばれる独自の変数を定義し利用する事が可能である。ここでは、実行環境を判別する為のフラグとなる[mobileFlag]変数を一つ追加する。**【図18】**

図 18　アプリ変数を追加



次に、アプリケーションの[動作内容]画面を開く。そして画面上部にある[Startup]イベントに、[スクリプトの実行]アクションを追加する。[Startup]イベントは、アプリケーション起動時に初期実行されるものである。ここに処理を追加することで、実行環境の判別を行う。**【図19】**

図 19　[Starup]イベントに[スクリプトの実行]を追加

Valence App Builderでは、[スクリプトの実行]アクションを使用すれば、ブラウザあるいはValence Mobile アプリ上で実行可能なJavaScript処理を追加する事ができる。単にJavaScriptが記述できるだけでなく、Valenceの為の専用APIも用意されており、これらを活用する事がポイントである。ここでは、実行環境の判別が可能なAPIである”Valence.mobile.Access.isNativePortal”メソッドを使用する。このメソッドは、Valence Mobileアプリから実行した場合に、trueが返却されるものである。(PCブラウザから実行した場合は、falseが返却される。)
このAPIを使用したJavaScriptの記述例が**【ソース1】**である。

ソース1　Valence Mobileアプリからの実行かどうかを判別

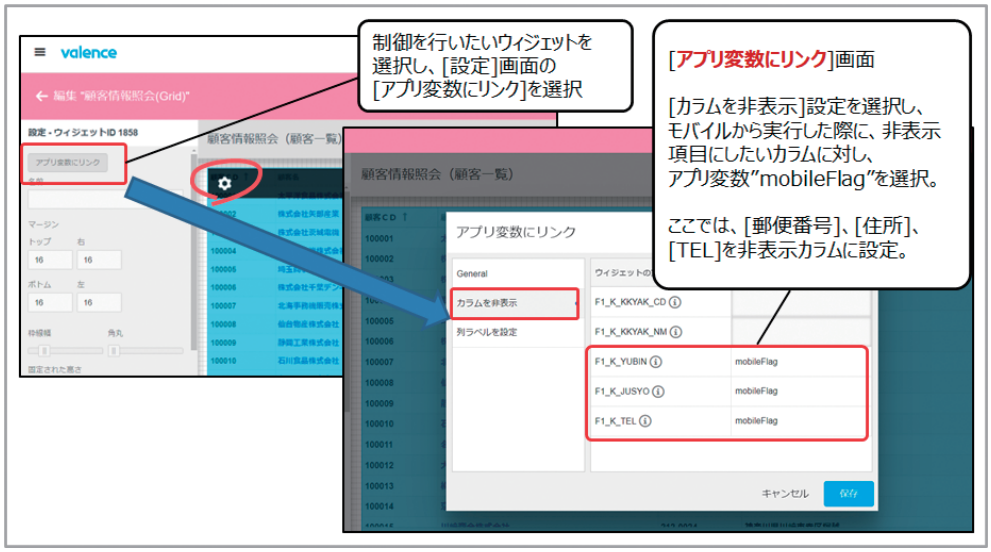
```
//----- Valence Mobileアプリからの実行かどうかを判定
//   ※PCブラウザで実行した場合は、アプリ変数mobileFlagをFalseに変更
if (!Valence.mobile.Access.isNativePortal()) {
    setAppVar('mobileFlag', false);
}
```

Valence

このプログラムにより、アプリケーションをPCブラウザから実行した場合、setAppVarメソッドを使用して、アプリ変数”mobileFlag”（初期値は、true）の値をfalseに変更する事ができる。

あとは、アプリケーションの設計画面で表示に関する制御を行いたいウィジェットを選択し、[アプリ変数にリンク]機能を設定すれば良い。今回の場合、顧客一覧リストを表示するGridウィジェットに対し、[アプリ変数にリンク]画面の”カラムを非表示”設定画面を使用し、Valence Mobileアプリから実行時に非表示とする列（カラム）に対し、アプリ変数”mobileFlag”を割り当てればよい。**【図20】**

図 20　アプリ変数を使用した非表示カラムの設定

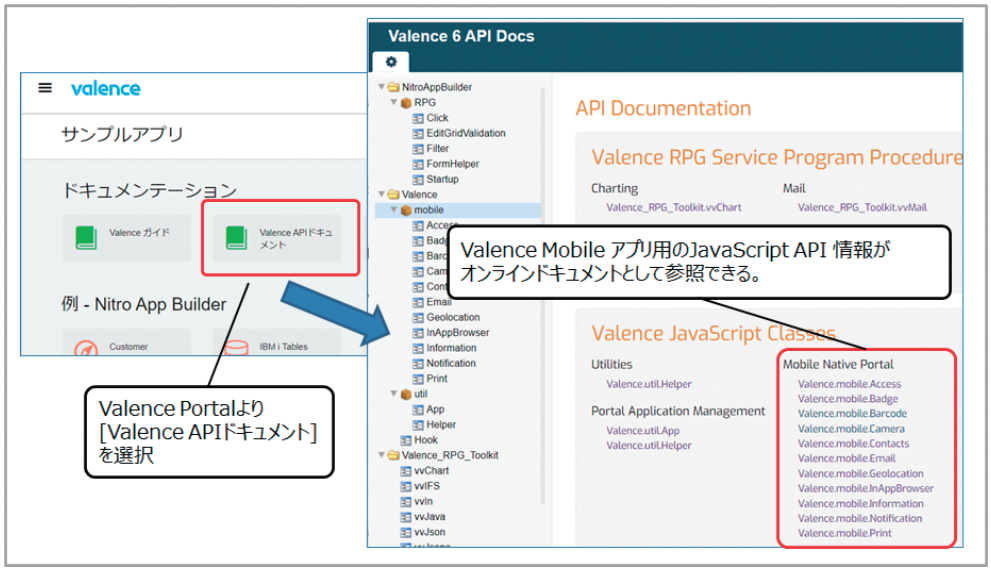


“カラムを非表示”設定画面では、アプリ変数の値が”true”となるカラムを非表示にする事ができる。この設定により、Valence Mobileアプリから実行した場合に、一部の列（カラム）のみを表示するようにカスタマイズできるのである。このように、アプリ変数を使用することで、ウィジェットの表示方法を制御できる。工夫をすれば、PCブラウザ（デスクトップ）とモバイルとで、一つの同じアプリケーションを共用しながらも、それぞれの実行環境にあわせた画面のカスタマイズが可能となる。今回は、Gridウィジェットのカラムについて制御を行ったが、他のウィジェットもアプリ変数による制御が可能なので、試してみてほしい。

5.デバイス機能活用テクニック

モバイルデバイスを業務で活用するメリットの一つが、PCでは難しかったカメラ等のデバイス機能が活用できることである。Valenceには、Valence Mobileアプリから使用できる専用のJavaScript APIが用意されている。PCブラウザのValence Portalから、[Valence APIドキュメント]を開くと、オンラインドキュメントが呼び出せる。ドキュメントトップページの[Valence JavaScript Classes]が、ブラウザあるいはValence Mobileアプリ上で使用できるクライアント側のJavaScript API一覧となる。この中にある[Mobile Native Portal]の部分が、Valence Mobileアプリで使用可能なモバイルデバイス用API一覧である。**【図21】**

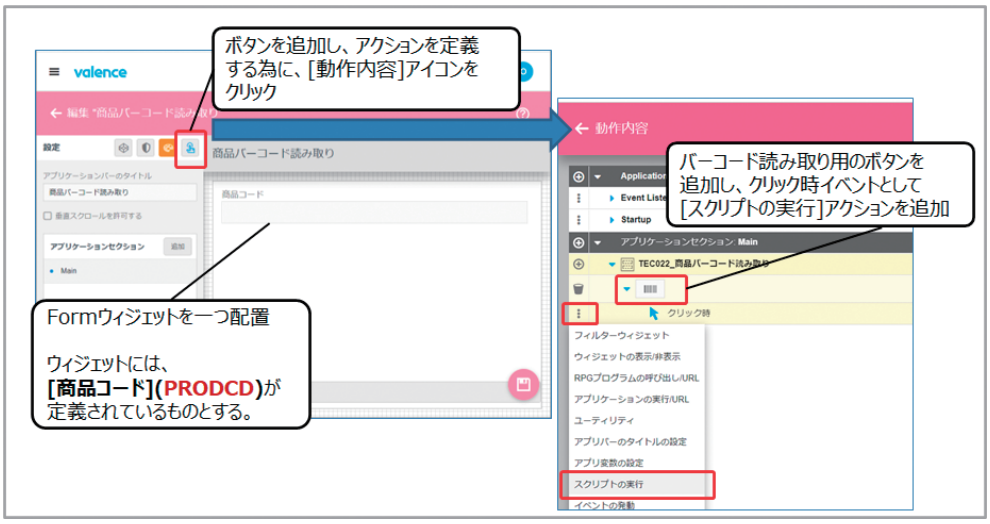
図 21　Valence APIドキュメント(英語)



なお、前節の”Valence.mobile.Access.isNativePortal”メソッドも、このモバイルデバイス用APIの一つである。本節では、デバイス機能の中でも使用頻度が高いと思われるカメラを活用した”バーコードの読み取り機能”の実装および”GPSを使用した現在位置(ロケーション)取得”の実装方法を紹介する。具体的には、”Valence.mobile.Barcode”と”Valence.mobile.Geolocation”の2つのAPIが該当する。

1つ目として、”バーコード読み取り機能”を紹介する。ここでは、[動作内容]画面を使用して、Formウィジェットにある商品コード欄[PRODCD]の箇所にボタンを追加し、このボタンのクリックイベントに[スクリプトの実行]アクションを追加する。**【図22】**

図 22　バーコード読み取り機能の実装(アクション設定)



エディタの中に、”Valence.mobile.Barcode” API を使用したコードを追加する。コードの記述例が、**【ソース 2】**である。

ソース2　バーコード読み取り機能の実装例

```
//===バーコード読み取り処理 (Valence Barcode APIを使用)===
var me = this;
Ext.Viewport.mask();
//----- スキャン開始
Valence.mobile.Barcode.scan({
    scope    : me,
    callback : function (response) {
        Ext.Viewport.unmask();
        //----- レスポンスが空の場合
        if (Ext.isEmpty(response)) {
            Ext.Msg.alert('バーコードスキャン', 'レスポンスがありません');
            return;
        }
        if (response.success) {
            //----- レスポンスがある場合
            if (!response.data.cancelled) {
                //----- 読み取り完了の場合
                cmp.setValues({
                    PRODCD : response.data.text
                });
            } else {
                //----- 読み取りキャンセルの場合
                Ext.Msg.alert('バーコードスキャン', 'キャンセル');
            }
        }
    }
});
```

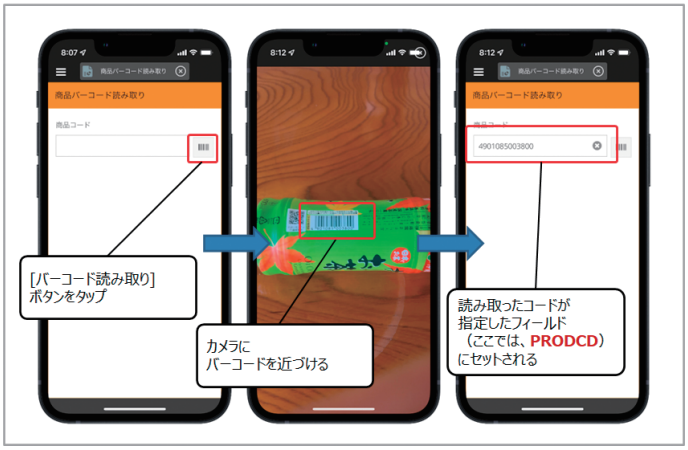
2-①　2-②　2-③

2-①の部分が、APIを使用してカメラを起動しバーコードスキャンを開始する部分である。このメソッドは、バーコードスキャン結果を2-②のようにcallback関数として処理を行う。処理結果をresponse変数として受け取る為、スキャンを正常終了したかどうかをresponse.successで判断する。その上で、読み取りキャンセルでなければ、バーコードの読み取り結果がresuponse.data.textから取得出来る。

2-③のようなsetValuesメソッドを使用すれば、取得した読み取り結果をウィジェット上の特定のフィールド(ここでは"PRODCD"フィールド)の画面値を更新することができる。

このように作成したアプリケーションをValence Mobileアプリから実行すると、デバイスのカメラを活用したバーコードの読み取りが可能になる。**【図23】**

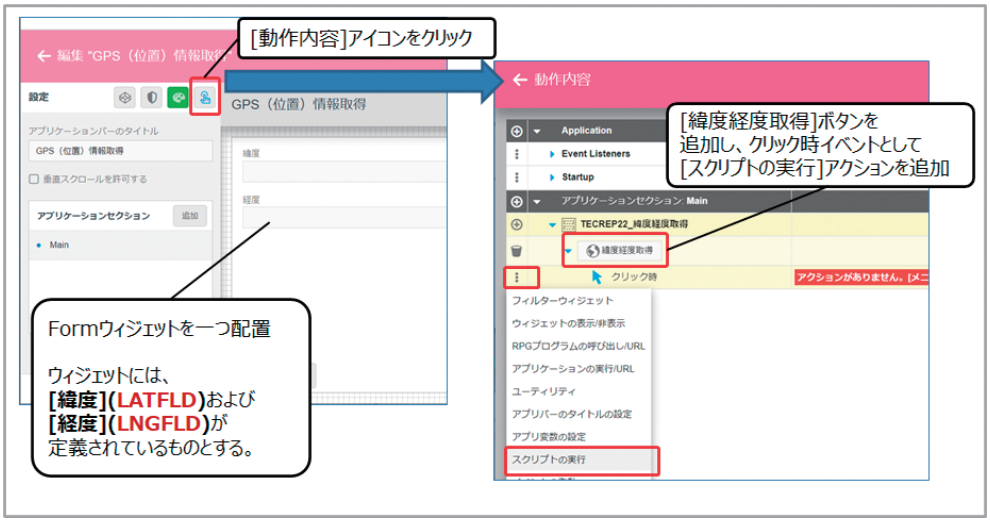
図 23 バーコード読み取りアプリケーション実行例



この実行例では、商品に貼付されたJANコード(1次元バーコード)を読み取っているが、他にもQRコード(2次元バーコード)を読み取ることも可能である。

2つ目に"GPSを使用した現在位置(ロケーション)取得"を紹介する。まず、Formウィジェット上には、GPSから取得したロケーション情報である緯度と経度を保持する為に、2つのフィールド(LATFLD, LNGFLD)を定義する。次に[動作内容]画面で、Formウィジェット上に[緯度経度取得]ボタンを追加し、このボタンのクリックイベントに[スクリプトの実行]アクションを追加する。**【図24】**

図 24 現在位置取得機能の実装(アクション設定)



エディタの中に、”Valence.mobile.Geolocation”APIを使用したコードを追加する。コード記述例が、**【ソース3】**である。

ソース3　現在位置取得（GPS）機能の実装例

```
//----- GPS(現在値)を取得 (Geolocation APIを使用)
var me = this;
Ext.Viewport.mask();
//----- ロケーション取得を開始
Valence.mobile.Geolocation.getCurrentPosition({
    scope    : me,
    callback : function (response) {
        Ext.Viewport.unmask();
        //----- レスポンス無しの場合、処理終了
        if (Ext.isEmpty(response)) {
            return;
        }
        //----- レスポンス結果の確認
        if (response.success) {
            //----- 正しく情報が取得出来た場合
            if (response.coords) {
                //----- 緯度をセット
                if (response.coords.latitude) {
                    cmp.setValues({
                        LATFLD : response.coords.latitude
                    });
                }
                //----- 経度をセット
                if (response.coords.longitude) {
                    cmp.setValues({
                        LNGFLD : response.coords.longitude
                    });
                }
            }
        } else {
            //----- 取得出来なかった場合、エラーメッセージを返す
            Ext.Msg.alert('Error', response.message);
        }
    }
});
```

3-①　3-②　3-③

ソースの考え方は、先程紹介したバーコード読み取りとほぼ同じである。3-①の部分が、APIを使用してGPSを起動し、現在位置の取得を開始する部分である。このメソッドも、取得結果を3-②のようにcallback関数として処理を行う。処理結果をresponse変数として受け取る為、情報の取得が正常終了したかどうかをresponse.successで判断する。その上で、情報が正しく取得できた場合、結果がresponse.coords.latitude（緯度）およびresponse.coords.longitude（経度）から取得出来る。3-③のようにsetValuesメソッドを使用し、取得した結果をウィジェット上のフィールド画面値に反映することができる。

最後に、Formウィジェット上に［Google　Map表示］ボタンを追加し、このボタンのクリックイベントに［アプリケーションの実行/URL］アクションを追加する。このアクションは、任意のURLより外部ブラウザを開く事ができる機能となる。ここで、Formウィジェットにセットされた緯度経度の値をパラメータにGoogle Mapを呼び出せば完成である。**【図25】**

図 25　外部ブラウザ起動（アクション設定）

完成したアプリケーションをValence　Mobileアプリから実行する。まず、[緯度経度取得]ボタンをタップすると、GPSを使用して位置情報を取得し、その結果となる緯度経度情報がFormウィジェットに表示される。次に、[Google Map表示]ボタンをタップすると、外部ブラウザが立ち上がり、現在位置にピンが打たれた地図が表示されることがわかる。**【図26】**

図 26　現在位置取得（GPS）機能アプリケーション実行例

このように、”Valence.mobile.Geolocation”APIを使用すれば、位置情報を取得出来る為、外部ブラウザへ連動すれば、簡単にGoogle Mapを表示させられることがわかった。

このアプリケーションでは、外部ブラウザを使用してGoogle Mapを表示させているが、なぜValence App Builder標準のMapウィジェットを使用せずに、わざわざ外部ブラウザを呼び出しているのだろうか？
理由は、ValenceのMapウィジェットは、現在位置を表示させる為には、緯度・経度ではなく、その地点の住所情報をセットする必要があるからだ。
つまり、ValenceのAPIで取得した位置情報を元に、住所情報に変換する事ができれば、直接Mapウィジェットを使用する事が可能となる。ただ残念ながら、Valenceの専用APIには、住所変換機能が無い為、住所情報を取得するには、独自に処理を実装する必要がでてくる。本稿の最後のトピックとして、この位置情報から住所情報に変換するテクニックを紹介する。
この変換処理は、Google Map APIを使用すれば実現可能である。Valenceが使用するMapウィジェット自体もGoogle Map APIを使用している為、ValenceでMapウィジェットを使用しているのであれば、Valenceの[ポータル管理]→[設定]の中にある[Google Map用APIキー]を設定しているはずである。この設定があれば、Valence App Builderの[スクリプトの実行]で記述するJavaScriptからでもGoogle Map APIがそのまま使用できる。
今回実装したい地点から住所への変換処理を行う為には、“Geocoding API”を使用すればよい。

具体的な実装手法を紹介する。先程のアプリケーションに[ADDRESS]という名前のアプリ変数を一つ追加し、さらにFormウィジェット上にも、住所を表示する為のフィールド(ADDRESS)を追加する。そして、[現在位置住所取得]ボタンを新たに追加し、そのボタンのクリック時イベントに[スクリプトの実行]を追加する。その中に住所変換ロジックを追加する。ソースの記述例が、**【ソース4】**である。

ソース4　位置情報から住所情報に変換処理を行う実装例

```
//----- フォーム上の緯度と経度を取得
var lat = rec.get('LATFLD');
var lng = rec.get('LNGFLD');

//----- 緯度経度情報を変数として保持
var curlatLng = new google.maps.LatLng(lat, lng);

//----- Google map geocode api を使用して、緯度経度情報から住所を取得
var geocoder = new google.maps.Geocoder();
geocoder.geocode({
    latLng: curlatLng
}, function(results, status) {
    //----- 正常に取得できた場合
    if (status == google.maps.GeocoderStatus.OK) {
        if (results[0].geometry) {
        // 住所を取得
        var adr = results[0].formatted_address;
        //----- フォーム上の[住所]欄に取得した住所をセット
        cmp.setValues({
            ADDRESS : adr
        });
        //----- アプリ変数ADDRESSにも取得した住所をセット
        setAppVar('ADDRESS', adr);
        }
    }
    //----- 正常に取得できなかった場合、ステータスに応じたエラーを返す
    else if (status == google.maps.GeocoderStatus.ZERO_RESULTS) {
        Ext.Msg.alert("住所が見つかりませんでした。");
    } else if (status == google.maps.GeocoderStatus.ERROR) {
        Ext.Msg.alert("サーバ接続に失敗しました。");
    } else if (status == google.maps.GeocoderStatus.INVALID_REQUEST) {
        Ext.Msg.alert("リクエストが無効でした。");
    } else if (status == google.maps.GeocoderStatus.OVER_QUERY_LIMIT) {
        Ext.Msg.alert("リクエストの制限回数を超えました。");
    } else if (status == google.maps.GeocoderStatus.REQUEST_DENIED) {
        Ext.Msg.alert("サービスが使えない状態でした。");
    } else if (status == google.maps.GeocoderStatus.UNKNOWN_ERROR) {
        Ext.Msg.alert("原因不明のエラーが発生しました。");
    }
});
```

4-①　4-②　4-③　4-④　4-⑤

4-①の部分が、Formウィジェットに表示されている緯度と経度を取得する処理である。4-②では、取得した緯度・経度をGoogle Map APIが定めた形式の緯度経度情報の変数に変換している。そして、4-③でGeocoding APIを呼び出して、4-④で変換処理を実行している。geocodeメソッドの実行パラメータとして、4-②で取得した変数値をセットし、結果を関数として受け取っている。受け取った関数内で住所情報を取得するのが、4-⑤の部分である。Results[0].formatted.addressに住所情報が格納される為、この結果をFormウィジェット上の住所欄(ADDRESSフィールド)ならびにアプリ変数"ADDRESS"にセットするという内容である。

これで住所情報を取得できるので、後はアプリ変数"ADDRESS"に格納された住所情報を使って、Mapウィジェットを表示させれば、アプリケーションは完成である。完成したアプリケーションを実行すると、先程同様の手順で緯度経度を取得後に、[現在位置住所取得]ボタンをタップすると、住所欄に変換された住所がセットされることがわかる。最後に、[Map表示]ボタンで、Mapウィジェットを表示している。ここでは、Valenceのウィジェットを使用して、直接アプリケーション上に地図を表示している為、先程の外部ブラウザのパターンのような外付け感のない自然な地図表示になっている事がわかる。**【図27】**

図 27　住所情報を取得するアプリケーション実行例

本節では、デバイス機能の活用として、バーコード読み取りと現在位置取得(GPS)の使用方法を紹介した。ここでは、単体の機能に関する実装方法を紹介したが、最後に実際のアプリケーションでこれらの機能を活用したデモを紹介する。とは言っても実行している実際の様子をこの紙面でお伝えするのは難しい為、今回弊社ホームページ上にある[Migaro. 技術Tips]サイトに、モバイルアプリケーションのデモ動画をアップした。次のURLから動画が確認できるので、Valence Mobileアプリを使用したApp Builderアプリケーションの実装例として是非確認してほしい。

URL:https://www.migaro.co.jp/tips/1640/

6.さいごに

本稿では、Valenceにおけるモバイルアプリケーション開発テクニックとして、アプリケーション作成の基本からモバイルデバイス機能の活用方法までを具体例をあげながら紹介してきた。これまで、ValenceについてはPCブラウザの機能が中心で、モバイル開発に関する情報が少なかったので、本稿を足掛かりとして、皆様もValenceを使用したモバイルアプリ開発にチャレンジ頂ければ幸いである。