

【セッションNo. 2】

SP4i モバイルアプリ開発テクニック

株式会社ミガロ。
RAD事業部 技術支援課
尾崎 浩司



【アジェンダ】

- SP4i におけるモバイルアプリ開発
- HTMLの概要とHTML5の機能
- モバイルに適した画面レイアウト作成テクニック
- HTML5を活用したオフラインアプリ開発手法
- まとめ



SP4iにおけるモバイルアプリ開発



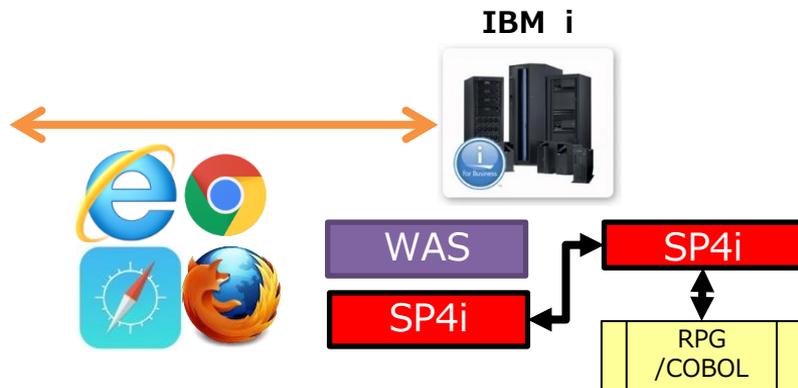
■ SP4iにおけるモバイルアプリ開発

- RPG/COBOLスキルでWeb化を実現するSmartPad4i (SP4i)
 - **マルチブラウザ対応**により、PCブラウザだけでなく、モバイルブラウザからも使用可能
 - モバイルアプリの画面作成に役立つ**HTML5**が使用可能
 - iOS/Androidに対応した**ハイブリッドアプリ**の作成

PC (Windows等) での使用



モバイル (iOS/Android等) での使用



■ SP4iにおけるモバイルアプリ開発

- 4つの簡単ステップ → モバイルアプリでも同様に開発可能



1. 画面の作成



2. 画面連携とプログラムの自動生成



3. プログラムロジックの追加とコンパイル



4. メニュー登録と実行

HTMLで画面作成
(市販のHTMLエディタで簡単に作成!)

```
桁 . . . . . 1 / 1 走査検索
SEU=>
FMT * ..... * 1 ...+... 2 ...+... 3 ...+... 4 ...+...
085.00 C010 * </YOURCODE>
087.00 C * INIT TAG
088.00 C * INITIALIZATION (RESERVED JAC1400)
089.00 C * YOUR INITIALIZATION EXSR JCINIT
090.00 C * EXSR YRINIT
091.00 C *
092.00 C * T100 TAG
093.00 C * UPDATE YOUR DATA BEFORE SEND EXSR YRDATA
094.00 C
095.00 C
```

骨組みプログラムに
RPG/COBOLでロジック追加

プログラムをメニューに
登録して完成

DSPF定義と同様に
属性・桁数等を定義して
RPG/COBOL配布



■ SP4iにおけるモバイルアプリ開発

• ハイブリッドアプリ

- iOS/AndroidのアプリストアよりSP4iアプリをインストールして使用
- アプリ上からSP4iプログラムを実行する為、通常のブラウザアプリでは困難なデバイス機能の活用が容易に可能



カメラと連動した写真アップロード

バーコード/QRコード読み取り

GPSを活用した現在地の取得

アドレス帳を利用したメール送信



■ SP4iにおけるモバイルアプリ開発

- 今回のポイント
 - SP4iにおけるモバイルアプリ開発に役立つテクニックを紹介
 - SP4iにおける画面レイアウト作成で使用するHTMLの基本
 - モバイルアプリ開発に役立つHTML作成テクニック
 - HTML5の活用（オフラインアプリの実装）

目的

従来のPC向けHTML画面作成との違いを理解し、
モバイルに最適化した画面や機能の実装を習得する



HTMLの概要とHTML5の機能



■ HTMLの概要とHTML5の機能



• HTML

- Webページを作成する為のマークアップ言語
- ホームページビルダーやDreamweaver等 HTMLエディタで容易に編集可能
- HTMLタグ(<〇〇>や 開始<〇〇>～ 終了</〇〇>)の形式で記述

input タグ
type = "text"
id = "INPTA"

input タグ
type = "button"
id = "BTN01"

HTML記述例

```
<div id="content">←  
<h2>JavaScriptのサンプル 必須入力チェック例</h2>←  
<span class="LBL">項目A</span>←  
<input type="text" id="INPTA" class="INP">←  
<span class="MSG">※必須入力</span><br>←  
<span class="LBL">項目B</span>←  
<input type="text" id="INPTB" class="INP">←  
<span class="MSG">※必須入力</span><br>←  
<span class="LBL">項目C</span>←  
<input type="text" id="INPTC" class="INP">←  
<span class="MSG"></span><br>←  
<span class="LBL">項目D</span>←  
<input type="text" id="INPTD" class="INP">←  
<span class="MSG"></span><br>←  
<input type="button" id="BTN01" value="実行">←  
</div>←
```

画面の基本構造（レイアウト）を作成するのがHTML



■ HTMLの概要とHTML5の機能



• JavaScript

- Web上でインタラクティブな表現をする為に開発されたスクリプト言語
- HTML内にプログラムを埋め込むことで、Webページに様々な機能を付加できる
(Webブラウザの動作を定義できる)
- クライアント (ブラウザ上) で実行される為、Webサーバーに負担をかけずに実行できる
- スクリプト言語の為、コンパイルは不要



画面上の動作が定義できるのがJavaScript



■ HTMLの概要とHTML5の機能

- HTML5とは
 - **動画や音声、図形描画などがプラグイン無しで実現可能**になった
 - videoタグ, audioタグ, canvasタグ 等
 - HTML4に比べてより明確に文書構造が記述できるようにタグが精査された
 - headerタグ, footerタグ, sectionタグ 等
 - **フォーム入力機能に新しい属性が追加**された
 - type="date" (カレンダー入力) , type="number" (数値入力) 等
 - 各種**APIの追加**
 - Geolocation API(位置情報取得) ,ファイルAPI(ローカルファイル利用) 等



HTML5では、HTML4では実現することの難しかった機能が簡単に作成できる



■ HTMLの概要とHTML5の機能

• HTML5の機能

- audio タグ (プラグインなしで音声ファイルを再生)

audioタグで音声ファイルをソースに指定

```
<audio id="beep1" src="音声ファイルパス" ></audio>
```

```
document.getElementById('beep1').play();
```

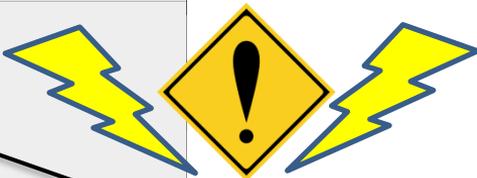
JavaScriptを使用して、audioタグに指定した要素のplayメソッドを実行して、音声再生

会員管理システム 終了

Migaro Technical Seminar

会員情報登録

会員名	<input type="text" value="山田 太郎"/> (漢字)
	<input type="text" value="やまだ たろう"/> (カナ) <small>全角カタカナで入力してください。</small>
性別	<input checked="" type="radio"/> 男性 <input type="radio"/> 女性
生年月日	<input type="text" value="1990/01/01"/> <input type="text"/>
入会日	<input type="text" value="2020/12/01"/> <input type="text"/>
住所	<input type="text" value="〒5560017"/> ...
	<input type="text" value="大阪府大阪市浪速区湊町"/> <input type="text" value="2-1-57 難波サンケイビル13F"/>
メールアドレス	<input type="text" value="YAMADA@MIGARO.CO.JP"/>



【例】
入力誤りがある場合に、エラー音を再生する



HTMLの概要とHTML5の機能

HTML5の機能

- フォーム機能の拡張 input type属性

```
<input type="date" id="INENT" >
```

従来のようにJavaScriptで独自のカレンダーを実装することなく、ブラウザの機能としてカレンダーが使用可能

Safari (iOS)



Chrome



Firefox



実行するブラウザによって
カレンダーの表示は異なる
(IEは非対応)



HTMLの概要とHTML5の機能

- HTML5の機能

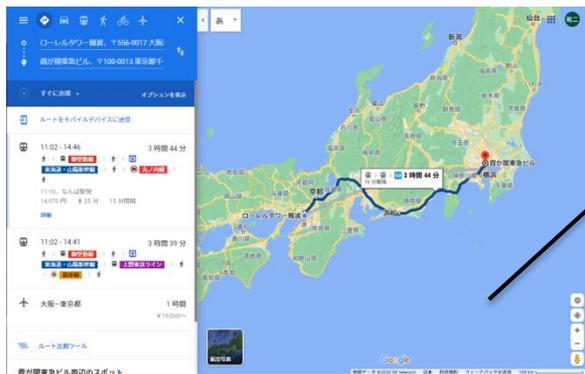
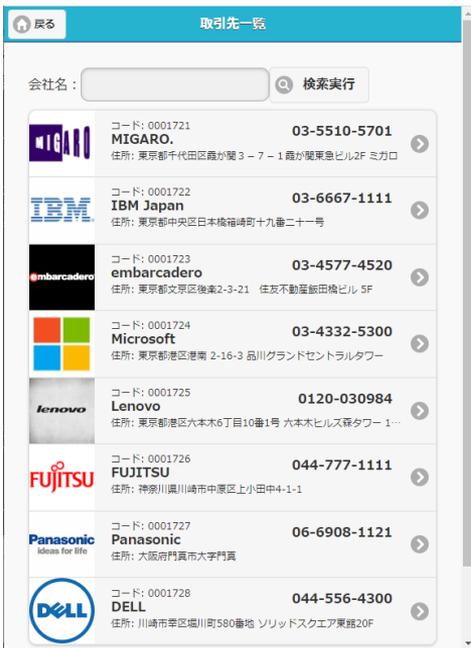
- 位置情報の取得 Geolocation API

現在位置の緯度と経度を取得

latitude : 緯度

longitude : 経度

```
navigator.geolocation.getCurrentPosition(function(position) {  
    saddr = position.coords.latitude + ',' +  
            position.coords.longitude;  
});
```



応用すると、GoogleMapの経路検索を使用して、現在位置からマスターに登録された住所地点までのルート表示が可能



モバイルに適した画面レイアウト 作成テクニック



■ モバイルに適した画面レイアウト作成テクニック

• PCとモバイルとの違い

• 画面サイズ

- PCのモニターと異なり、**表示域が小さい場合**が多い
使用デバイスを、意識したインターフェース作成が必要

• 操作方法

- PCでは**キーボードとマウス**入力
スマートデバイスでは、**ソフトウェアキーボードとタッチ**操作

• 表示の違い（ウィンドウ表示、全画面表示）

- PCではモニター上に**複数のウィンドウを並べて表示**が可能
スマートデバイスでは、**常に単一画面**のみ表示



■ モバイルに適した画面レイアウト作成テクニック

• レイアウトの工夫

- スマートフォンのアプリの場合は縦に要素を並べる



スマートフォンの場合、要素を横に並べると目線が左右に動くことになる

- ボタンは、アイコン等を使用してシンプルなものとし、直感的でわかりやすくする

- Font Awesome (<https://fontawesome.com/>)
(広く使用されている Webアイコンフォント)



色、大きさをcssで調整可能、
画像ではないため、拡大時でも綺麗

14:49 192.168.0.21

検品情報入力 Exit

Migaro.Technical Seminar

品番: A0001

キズ: 有 無

適正寸法:

備考を入力する

寸法が正しくない

備考:

登録件数: 0



■ モバイルに適した画面レイアウト作成テクニック

• 入力要素の工夫

- 入力項目は少なくする – (簡単に入力できるインターフェースをなるべく使用)
 - ラジオボタンやチェックボックス、コンボボックスを使用してキーボード入力を減らす
 - 表示する項目を少なくすることも効果的

検品情報入力
Migaro Technical Seminar

品番: A0001

キズ: 有 無

適正寸法:

備考を入力する

登録件数:

検品情報入力
Migaro Technical Seminar

品番: A0001

キズ: 有 無

適正寸法:

備考を入力する

備考:

登録件数: 5

入力しない場合がある項目は、
デフォルトで非表示にすることで、
画面上のエリアを確保

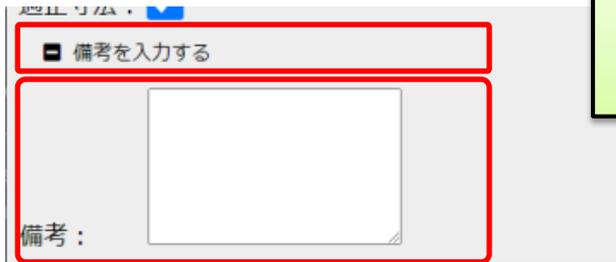


キーボード入力を少なくして、ユーザーの負担を減らす



■ モバイルに適した画面レイアウト作成テクニック

● 実装例



remark 要素クリックにより
アイコン (田,日) の切替
次の要素の表示/非表示切替

HTML記述例

```
<div class="checklist" id="remark" style="margin-left:20px">  
  <i id="closelcon" class="fas fa-plus-square active" ></i>  
  <i id="openlcon" class="fas fa-minus-square" style="display:none"></i>  
  <span style="margin-left:10px">備考を入力する</span>  
</div>  
<div class="checklist" style="display:none;margin-left:20px">  
  <label class="label">備考:</label>  
  <textarea id="TXT01" rows="5" style="width:200px"></textarea>  
</div>
```

Java Script部

```
<script>  
  ~省略~  
  //備考表示  
  $('#remark').on('click',function(e){  
    $(this).next().slideToggle();  
    $('#closelcon,#openlcon').toggle();  
  });  
  ~省略~  
</script>
```

```
<script src="jquery.min.js"></script>
```

jQueryライブラリの読み込み



■ モバイルに適した画面レイアウト作成テクニック

- 入力要素の工夫
 - 入力フォームはtype属性を設定

```
<input type="date" xxxxxxxxxxxx>  
<input type="number" xxxxxxxxxxxx>  
<input type="email" xxxxxxxxxxxx>
```

Type="number"の場合
数値キーボードが表示される

type="email"の場合
メールアドレス用のキーボード

Type="date"の場合
カレンダーが表示される

type属性に合わせた適切なインターフェースが表示される



■ モバイルに適した画面レイアウト作成テクニック

• Viewport

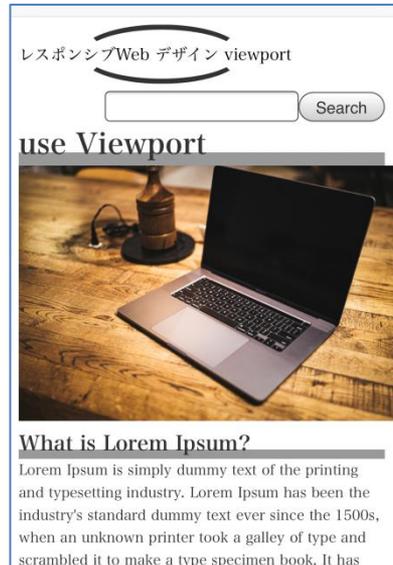
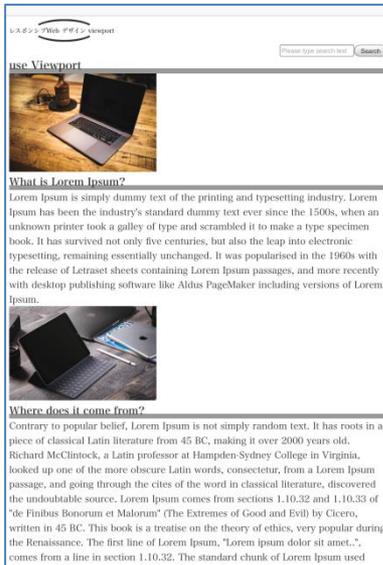
- スマートデバイスで適切な表示領域を表示するための設定

Headタグ内に記述する
使用しているデバイスの幅に
合わせてビューポートが指定される

```
<meta name="viewport" content="width=device-width" />
```

スマートフォンで実行

左：ビューポート無
右：ビューポート有



viewportを指定することで、
スマートデバイスでも適切な
サイズで画面を表示できる



■ モバイルに適した画面レイアウト作成テクニック

• レスポンシブWebデザイン

- 画面サイズの異なるPCとスマートデバイスの両方で同じ画面(HTML)を使用
 - 昔はPC用、スマートフォン用のようにHTMLを分けて作成されていたが、近年では、HTMLを分けず、メディアクエリを使用して画面サイズに合わせてレイアウトを変更する作成方法が一般的になった。

スマートフォンでは自動的にレイアウトが変わる

画面例 PC

No.	会員名(漢字)	会員名(カナ)	性別	生年月日	入会日
00000009	寺脇 育二	テラワキ イクジ	男性	1947/01/09	2012/05/22
00000010	高見 浩正	タカミ ヒロマサ	男性	1955/02/15	2010/05/28
00000011	村井 莉央	ムライ リオ	女性	1992/06/09	2010/06/12
00000012	高井 雄大	タカイ ユウタ	男性	1932/04/18	2010/06/12
00000013	福沢 愛梨	フクザワ アイリ	女性	1984/12/06	2010/06/14
00000014	藤井 奈々	フジイ ナナ	女性	1946/03/03	2010/07/19
00000015	楠 ひろ子	クスノキ ヒロコ	女性	1947/05/18	2010/07/25
00000016	おかやま 芳正	オカヤマ ヨシマサ	男性	1948/10/04	2010/07/26
00000017	吉田 徹	ヨシダ トオル	男性	1949/05/14	2012/05/11
00000018	宮坂 十樹	ミヤサカ レンキ	男性	1978/05/15	2012/05/20

画面例 スマートフォン

選択	No.	会員名(漢字)
+ (選択)	00000001	細川 エリカ
+ (選択)	00000002	大原 結衣
+ (選択)	00000003	藤澤 南朋
+ (選択)	00000004	松田 恵麻
+ (選択)	00000005	有村 理紗

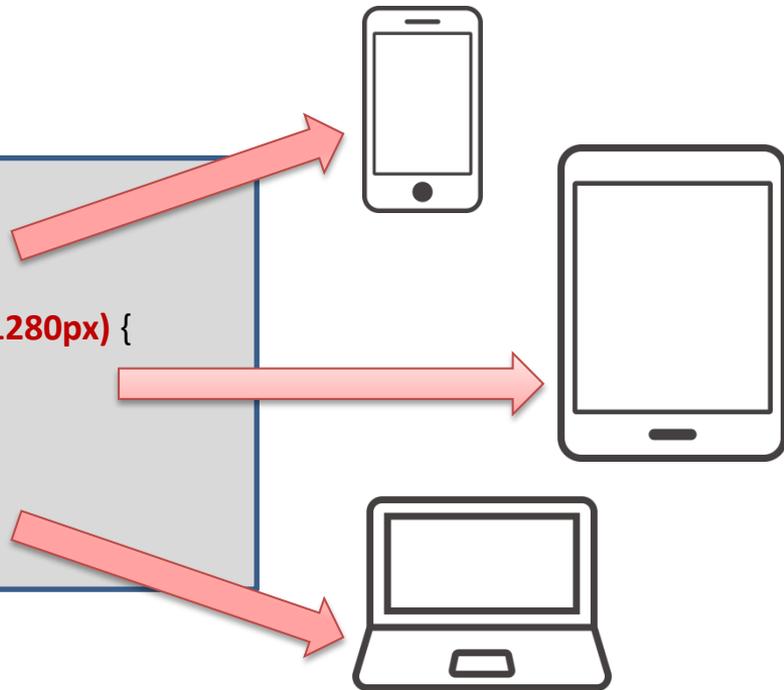
会員名(カナ): オオハラ ユイ
性別: 女性
生年月日: 1994/03/20
入会日: 2012/02/12

■ モバイルに適した画面レイアウト作成テクニック

• レスポンシブWebデザイン

• メディアクエリ

```
@media screen and (max-width: 767px) {  
  /*画面サイズ 767以下をスマートフォンのスタイル*/  
}  
@media only screen and (min-width: 767px) and (max-width: 1280px) {  
  /*画面サイズ 768～1280pxをタブレットのスタイル*/  
}  
@media screen and (min-width:1280px) {  
  /* PCのサイズ */  
}
```



メディアクエリを利用することで、画面サイズごとのスタイルを設定可能



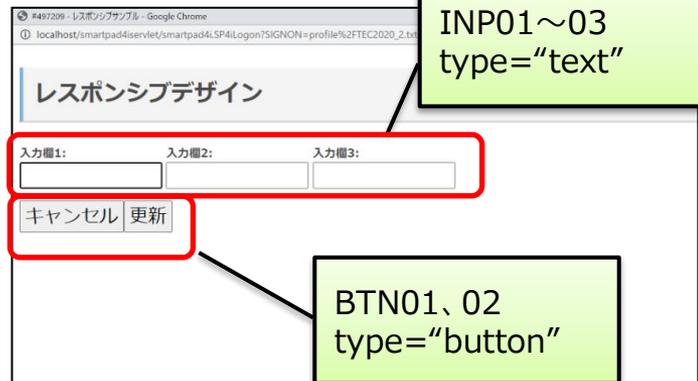
■ モバイルに適した画面レイアウト作成テクニック

- レスポンシブWebデザイン
 - メディアクエリの例

HTML例

```
<div class="Content">
<!-- Content start --->
  <ul class="col3" >
    <li><label for="INP01">入力欄1:</label><input type="text" id="INP01"></li>
    <li><label for="INP02">入力欄2:</label><input type="text" id="INP02"></li>
    <li><label for="INP03">入力欄3:</label><input type="text" id="INP03"></li>
  </ul>
</div>
<input type="button" id="BTN01" value="キャンセル">
<input type="button" id="BTN02" value="更新">
</div>
<!-- Content end --->
</div>
```

HTMLレイアウト



■ モバイルに適した画面レイアウト作成テクニック

- レスポンシブWebデザイン
 - メディアクエリ スマートフォンで適用されるcss

css例 /*スマートフォン */

```
@media screen and (max-width: 767px) {
```

```
li{  
  width:100%;  
}
```

入力欄を横幅に合わせる

```
input[type="text"]{  
  width:100%;  
}
```

```
input[type="button"]{  
  padding:20px;  
}
```

ボタン内の余白を大きくする

画面例 PC



スマートフォンで実行した場合、入力欄が縦に並び

画面例 スマートフォン



HTML5を活用した オフラインアプリ開発手法



■ HTML5を活用したオフラインアプリ開発手法

- 一般的なWebアプリ
 - ネットワークに接続されていることが前提
 - 一般的にはオフライン時にサーバーへアクセスするとエラーになる



【社内利用時】

(倉庫、工場 etc) 場所によっては
事務所と違い、Wifiが弱かったり入らない可能性

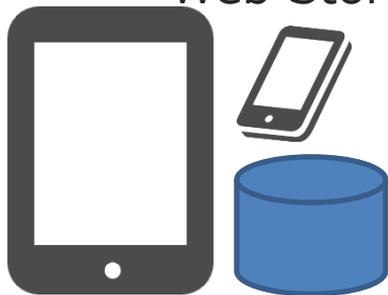
→ オフラインで使用できることにより
活用方法が増える。



■ HTML5を活用したオフラインアプリ開発手法

• HTML5の機能

- Web Storage ストレージ (sessionStorage , localStorage)



localStorageは
デバイスに保存される



◆ sessionStorage

```
sessionStorage.setItem('key', 'value'); // データを保存する  
var data = sessionStorage.getItem('key'); // データを取得する  
sessionStorage.removeItem('key'); // データを削除する
```

◆ localStorage

```
localStorage.setItem('key', 'value'); // データを保存する  
var data = localStorage.getItem('key'); // データを取得する  
localStorage.removeItem('key'); // データを削除する
```

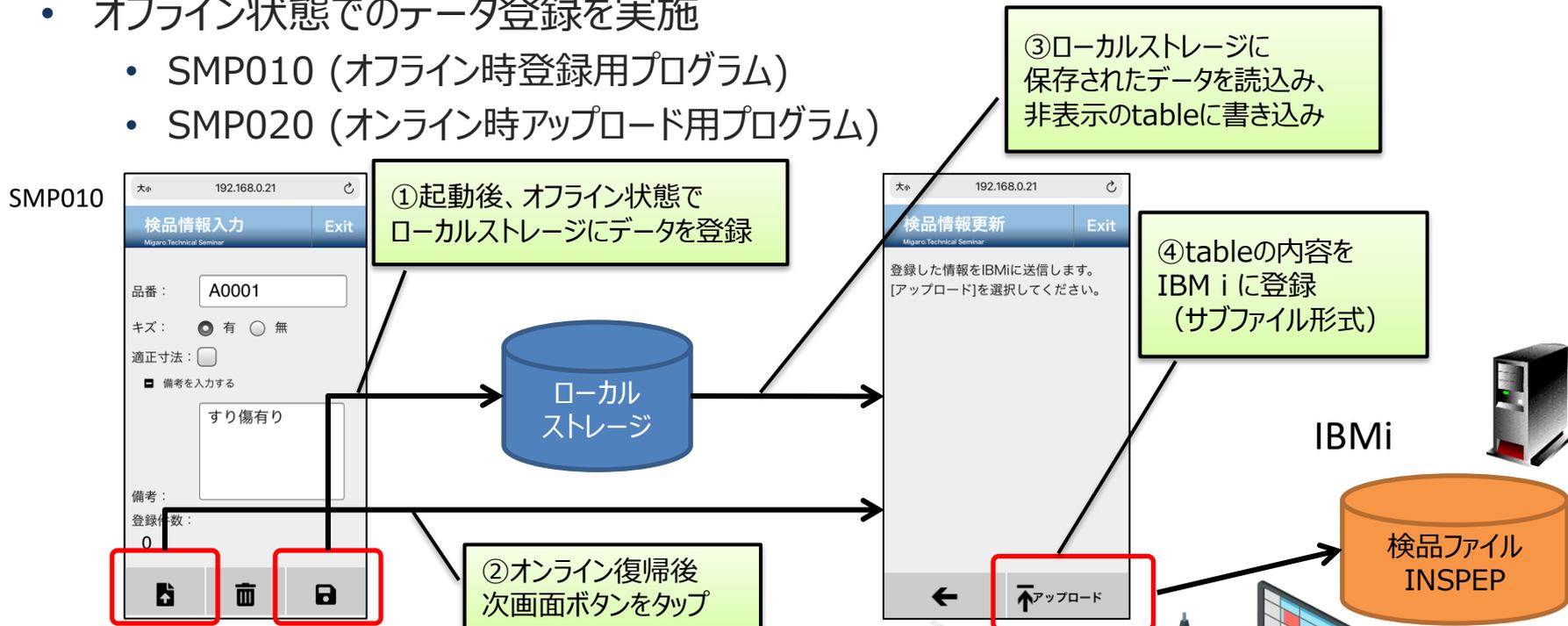
sessionStorageはブラウザ終了後クリア、localStorageは永続的に保存される



■ HTML5を活用したオフラインアプリ開発手法

• オフラインアプリデモ

- オフライン状態でのデータ登録を実施
 - SMP010 (オフライン時登録用プログラム)
 - SMP020 (オンライン時アップロード用プログラム)



■ HTML5を活用したオフラインアプリ開発手法

- ローカルストレージへ保存【SMP010】
 - オフライン時には、端末のローカルストレージにデータを保存



```
//登録ボタン時処理
```

```
$('.save').on('click',function(e){
```

```
var dataArray = [];
```

```
var getArray = localStorage.getItem('SP4TEC20_SMP010');
```

```
if(getArray){
```

```
    dataArray = JSON.parse(getArray);
```

```
}
```

```
var obj = {};
```

```
obj.cd = $('#INP01').val();
```

```
obj.scratch = ($('#[data-scratch]:eq(0)').prop('checked'))? 1:0;
```

```
obj.size = ($('#CHK01').prop('checked'))? 'Y':'N';
```

```
obj.remarks = $('#TXT01').val();
```

ローカルストレージのデータを取得し、配列(dataArray)として保持する。

画面上の各入力要素の値を取得して、変数obj にセット

次頁へ続く



■ HTML5を活用したオフラインアプリ開発手法

- ローカルストレージへ保存 【SMP010】
 - オフライン時には、端末のローカルストレージにデータを保存

```
//追加  
dataArray.push(obj);  
var setjson = JSON.stringify(dataArray);  
localStorage.setItem('SP4TEC20 SMP010', setjson);  
//入力欄のクリア  
$('#INP01').val("");  
$('#[data-scratch]').prop('checked',false);  
$('#CHK01').prop('checked',false);  
$('#TXT01').val("");  
//登録件数表示  
countShow();  
});
```

設定したobj変数の値を配列(dataArray)に追加

配列をJSONオブジェクトに変換し、
ローカルストレージに保存

次レコードの情報を入力する為に
画面入力項目の要素をクリア



■ HTML5を活用したオフラインアプリ開発手法

- オフライン時の対応 【SMP010】
 - オフライン時にはサーバーへ接続できない旨を表示する方法



```
function checkServer(){
  //オンライン・オフラインをチェック
  if (navigator.onLine) {
    //オンラインです
    return true;
  } else {
    //オフラインです
    window.alert('オフライン状態です。接続後に処理してください。');
    return false;
  }
  return false;
}
```

NavigatorのonLineプロパティで接続状況をチェック

オフライン時には、警告メッセージを表示

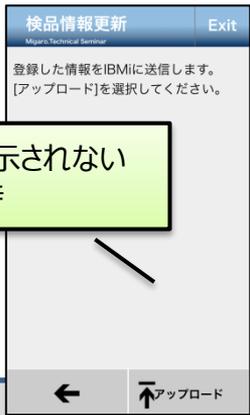
SP4iのサブミット時関数 (cansubmit) から、checkServer関数を呼び出すことで、送信/中断の制御が可能



■ HTML5を活用したオフラインアプリ開発手法

- ローカルストレージのデータを復元 【SMP020】
 - SMP010でローカルストレージに保存したデータをSMP020へ復元
 - SMP020では送信用のサブファイル入力欄を作成(非表示に設定)

画面には表示されない
tableを保持



HTML

```
<table id="SFL01" width="100%" border="1px" cellspacing="0" cellpadding="5px" data-table="SFL01" style="display:none;margin-bottom:40px;table-layout:fixed" class="PAGE">
<thead>
~省略~
</thead>
<tbody>
<tr>
<td><input type="text" id="INP01" data-cd = "true" ></td>
<td><input type="text" id="INP02" data-scratch = "true" ></td>
<td><input type="text" id="INP03" data-size = "true" ></td>
<td><textarea id="TXT01" data-remarks = "true" ></textarea></td>
</tr>
</tbody>
</table>
```

サブファイルのテーブル要素を非表示とする

`<input type="text" id="INP01" data-cd = "true" ></td>`
`<input type="text" id="INP02" data-scratch = "true" ></td>`
`<input type="text" id="INP03" data-size = "true" ></td>`
`<textarea id="TXT01" data-remarks = "true" ></textarea></td>`

ローカルストレージに保存された
複数件の入力情報を保持する為に
各項目のフィールドを定義



■ HTML5を活用したオフラインアプリ開発手法

• ローカルストレージのデータを復元 【SMP020】

- SMP010で登録したデータをSMP020へ復元
 - 送信用プログラムへ登録したデータ件数を送信
登録件数分のサブファイルレコードを作成

RPG (SMP020)

```
0078.00 * <YOURCODE>
0079.00 * YOUR CODE
0080.00 C      *ENTRY          PLIST
0081.00 C          PARM          PCOUNT      3 0
0082.00 C          PARM          PBACK       1
```

SMP010で登録を行った件数を受け取り
SMP020を実行する

```
0522.00 * <YOURCODE>
0523.00 * PLEASE REMOVE THIS BLOC C510B IF YOU ARE BUILDING A TEMPLATE
0524.00 C          Z-ADD      1          SPL102      4 0
0525.00 C          Z-ADD      PCOUNT      SPL902      4 0
0526.00 * </YOURCODE>
```

非表示のtableに対し、実行時に
レコード件数分の入力欄の枠をセット



■ HTML5を活用したオフラインアプリ開発手法

• ローカルストレージのデータを復元 【SMP020】

- SMP010で登録したデータをSMP020へ復元
 - 送信用のサブファイル入力欄へローカルストレージデータを復元

```
var dataArray = [];  
var getArray = localStorage.getItem('SP4TEC20_SMP010');  
if(getArray){  
    dataArray = JSON.parse(getArray);  
}
```

```
var data_cd = $('[data-cd]');  
var data_scratch = $('[data-scratch]');  
var data_size = $('[data-size]');  
var data_remarks = $('[data-remarks]');
```

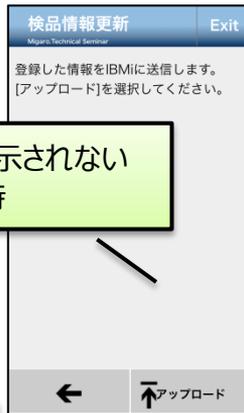
```
$.each(dataArray, function(index, val) {  
    $(data_cd[index]).val(val.cd);  
    $(data_scratch[index]).val(val.scratch);  
    $(data_size[index]).val(val.size);  
    $(data_remarks[index]).val(val.remarks);  
});
```

ローカルストレージのデータを取得し、配列(dataArray)として保持する。

table上の入力欄の要素をセット

ローカルストレージのデータをtable上の入力欄に復元

画面には表示されないtableを保持



■ HTML5を活用したオフラインアプリ開発手法

• ローカルストレージのデータを復元 【SMP020】

- SMP010で登録したデータをSMP020へ復元
 - IBMiへ送信

IBM i側では、通常のサブファイルレコードの入力として処理を行う

非表示で保持しているtable (サブファイル)
入力欄の情報をIBMiへ送信

品番	傷	寸法	備考
A0001	有	True	
A0002	無	True	
A0003	無	True	-----
A0004	無	false	
A0005	無	true	
A0006	無	true	-----

検品情報更新
Migaro Technical Seminar

Exit

登録した情報をIBMiに送信します。
[アップロード]を選択してください。

←
↑
アップロード

RPG (SMP020)

```

0550.00 *-----
0551.00 * SB0010  登録処理
0552.00 *-----
0553.00 C      SB0010      BEGSR
0554.00 C              OPEN      INSPEP
0555.00 C              DO        PCOUNT      I      4 0
0556.00 C      I              OCCUR      SOIF02
0557.00 C              MOVEL(P)  I INP01      SPPCD
0558.00 C              MOVEL(P)  I INP02      SPSCR
0559.00 C              MOVEL(P)  I INP03      SPSIZ
0560.00 C              MOVEL(P)  ITXT01      SPREM
0561.00 C              MOVEL      *DATE      SPDAY
0562.00 C      SPSCR      IFEQ      '無'
0563.00 C      SPSIZ      ANDEQ      'Y'
0564.00 C              MOVEL(P)  'Y'      SPRSL
0565.00 C              ENDIF
0566.00 C              WRITE      INSPER
    
```



まとめ



■ まとめ

- SP4iにおけるモバイルアプリ開発テクニックを紹介
 - HTMLの概要とHTML5の機能
 - モバイルに適した画面レイアウト作成テクニック
 - レイアウトの工夫
 - 入力要素の工夫
 - Viewport
 - レスポンシブWebデザイン
 - HTML5を活用したオフラインアプリ開発手法
 - ローカルストレージの活用



ご清聴ありがとうございました。

