

SmartBrowser Update

for iOS / Version 1.5.9 build121

2020年6月



株式会社 ブルーテック
BLUETECH CORPORATION

更新内容-概要



iOS Version 1.5.9 build121 の更新内容について

【追加変更した機能一覧】

1. 指定枠内をスキャンするバーコード読取API【新機能】
2. フラッシュライトを制御するAPI【新機能】
3. スキャン時カメラズーム機能【新機能】
4. ナビゲーションバーの表示切り替え(meta指定)【新機能】
5. 手書きパッドを制御するAPI【新機能】



指定枠内をスキャンするバーコード読取API

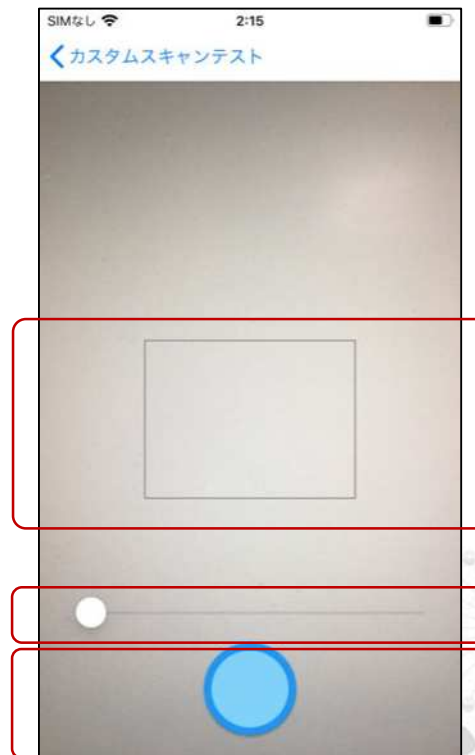
1. 指定枠内をスキャンするバーコード読取API（カスタムスキャン） **デモサイト**

デモサイト：

<https://www.bluetech.co.jp/iweb/smartbrowser/sample/customscan.html>



1.1 画面説明



①バーコード読み取りエリア

スキャンボタンをクリックすると赤枠になり、赤枠内のバーコードをスキャンボタンでスキャンします。
読み取りエリアはAPIで枠サイズを指定可能です。

②拡大ズームバー：スケール値1～10で拡大ズームします

③スキャンボタン：赤枠内のバーコードを読み取ります

指定枠内をスキャンするバーコード読取API

1. 指定枠内をスキャンするバーコード読取API

1.2 APIコール

smartbrowser().callAPIで「path:'scan/continurous」を呼び出しカスタムバーコード読み取りAPIを制御します

path	Params	Value	notes
scan/continurous	width_ratio	読み取り枠サイズの幅を指定 (画面幅に対して) 'width_ratio':0.5	未指定の場合は画面幅
	aspect_ratio	読み取り枠サイズの高さを指定 (width_ratioに対して) 'aspect_ratio':0.75	未指定の場合は画面高さ
	scanformat	スキャンするバーコード種別を指定 ※指定値は次ページ参照	初期値はauto
	camerafacing	auto : 端末設定 front : フロントカメラ back : バックカメラ	初期値はauto
	zoom	float値 1(等倍)~	初期値は1
	flashlight	true(点灯) / false(消灯)	初期値はfalse
	auto_scan	true(スキャンボタン非表示) / false(スキャンボタンを表示)	初期値はfalse

※ scanformat でスキャンするバーコード種別を指定すると読み取りすると、誤読取率が減少します
また、スキャン速度が向上します

指定枠内をスキャンするバーコード読取API

1. 指定枠内をスキャンするバーコード読取API

1.3 読み取りバーコード種別

属性値	モード	内容	記述方法	SmartBrowser iOS版 標準スキャン	SmartBrowser iOS版 指定枠内スキャン
scanformat	auto	読み取り可能な すべての種類	'scanformat':'auto'	✓	✓
	UPC_A	UPC-A	'scanformat':'UPC_A'	-	✓
	UPC_E	UPC-E	'scanformat':'UPC_E'	✓	✓
	EAN_8	EAN-8	'scanformat':'EAN_8'	✓	✓
	EAN_13	EAN-13	'scanformat':'EAN_13'	✓	✓
	CODE_39	Code 39	'scanformat':'CODE_39'	✓	✓
	CODE_93	Code 93	'scanformat':'CODE_93'	✓	✓
	CODE_128	Code 128	'scanformat':'CODE_128'	✓	✓
	ITF	ITF	'scanformat':'ITF'	✓	✓
	RSS_14	GS1 DataBar	'scanformat':'RSS_14'	-	✓
	RSS_EXPANDED	GS1 Databar Expanded	'scanformat':'RSS_EXPANDED'	-	✓
	CODABAR	Codabar	'scanformat':'CODABAR'	-	✓
	AZTEC	Aztec Code	'scanformat':'AZTEC'	✓	✓
	MAXCODE	MaxiCode	'scanformat':'MAXCODE'	-	✓
	PDF_417	PDF417	'scanformat':'PDF_417'	✓	✓
QR_CODE	QR code	'scanformat':'QR_CODE'	✓	✓	
DATA_MATRIX	Data Matrix	'scanformat':'DATA_MATRIX'	✓	✓	

指定枠内をスキャンするバーコード読取API

1. 指定枠内をスキャンするバーコード読取API

1.4 APIコールバック

Name	内容	Value(sample)
data	バーコードの読み取り値	4201341043541
orientation	端末の向き	180
system	device_id	インストール時に設定される端末識別番号
	device_name	設定画面で入力した端末名
	os_name	OS名
	application_version_name	SmartBrowserのバージョン
	os_version	OSバージョン
datetime	バーコードの読み取り日時	2019/11/12 17:37:00
timestamp	バーコードの読み取り日時(Unixtime)	1573547820136
location	longitude	経度 西経がマイナス、東経がプラス 単位：度(degree)
	latitude	緯度 南緯がマイナス、北緯がプラス 単位：度(degree)
	accuracy	精度 単位：メートル
	timestamp	GPSデータ取得日時(Unixtime)
format_name	バーコードの種類	EAN_13

1.4 APIコールバック(jsonデータサンプル)

```
{
  "data": "4201341043541",
  "system": {
    "application_version_name": "1.5.3",
    "os_name": "iOS",
    "device_id": "15fe294cfb",
    "device_name": "テスト端末 1",
    "os_version": "13.2"
  },
  "request": {
    "width_ratio": 0.5,
    "aspect_ratio": 0.75,
    "auto_scan": false
  },
  "format_name": "EAN_13",
  "datetime": "2019/11/12 17:37:00",
  "timestamp": 1573547820136,
  "location": {
    "timestamp": 1573545807624,
    "latitude": 35.165565515561134,
    "longitude": 136.89802241157463,
    "accuracy": 65
  }
}
```



指定枠内をスキャンするバーコード読取API

1. 指定枠内をスキャンするバーコード読取API

デモサイト

1.5 サンプルソースコード

```
<html lang="jp">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>カスタムスキャンテスト</title>
  <script>

    function customScan() {
      smartbrowser().callAPI({
        'path': 'scan/continurous',
        'params': {
          'width_ratio': 0.5,
          'aspect_ratio': 0.75,
          //'scanformat':'QR_CODE',
        }
      });
    }

    function onSmartBrowserBarcodeScanned(data) {
      document.getElementById('data').innerText = JSON.stringify(data);
    }
  </script>
</head>
```

```
<body>
  <hr>
  <h2>カスタムスキャンテスト</h2>
  <hr>
  読み取りエリア : width_ratio:0.5 / aspect_ratio:0.75<br />
  <button onclick="customScan()">バーコード読み取り</button>
  <hr>
  <div id="data"></div>
</body>
</html>
```





フラッシュライトを制御するAPI

2. フラッシュライトを制御する

デモサイト：

<https://www.bluetech.co.jp/iweb/smartbrowser/sample/flashlight.html>

デモサイト



2.1 HTML属性 指定

スキャン時にフラッシュライトが点灯します

```
<input type="text" flashlight="true" scanmode="direct" />
```

属性名	パラメタ内容	動作	記述方法	備考
flashlight	true	点灯	flashlight="true"	
	false	消灯	flashlight="false"	

2.2 APIコール

smartbrowser().callAPIで「path:'flashlight'」を呼び出しフラッシュライトを制御します

path	Params	Value	notes
flashlight	Enable	true or false (true:点灯、false : 消灯)	



フラッシュライトを制御するAPI

2. フラッシュライトを制御する

2.3 サンプルソースコード

```
<html lang="jp">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <title>フラッシュライトテスト</title>
  <script>
    function flashlight(flg){
      smartbrowser().callAPI({
        path: 'flashlight',
        params: {enable:flg }
      });
    }
  </script>
</head>
<body>
  <hr>
  <h2>フラッシュライト(TextBox連動)</h2>
  <input type="text" flashlight="true" scanmode="direct" />
  <hr>
  <h2>フラッシュライト(API起動)</h2>
  <hr>
  <input type="checkbox" onclick="flashlight(this.checked);">チェックONで点灯
  <hr>
</body>
</html>
```

デモサイト





スキャン時カメラ画像の拡大

3. スキャン時カメラ画像の拡大

デモサイト：

<https://www.bluetech.co.jp/iweb/smartbrowser/sample/zoom.html>

デモサイト



3.1 HTML属性 指定

スキャン時にカメラ画像をスケール値で拡大します

属性名	パラメタ内容	動作	記述方法	備考
zoom	スケール値を指定 1-10 の数値 (小数点以下記述可能)	スキャン時カメラ画像を ズームスケール値で拡大	zoom="1" zoom="2.5"	10以上を指定できませんが、 読み取り精度はカメラの性能に よります





スキャン時カメラ画像の拡大

3. スキャン時カメラ画像の拡大

3.3 サンプルソースコード

```
<html lang="jp">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>スキャン時カメラ画像の拡大縮小</title>
</head>
<body>
  <hr>
  <h2>カメラ画像の拡大縮小(TextBox連動)</h2>
  <hr>
  スケール1:<input type="text" scanmode="direct" zoom="1" /><br />
  スケール2:<input type="text" scanmode="direct" zoom="2" /><br />
  スケール3:<input type="text" scanmode="direct" zoom="3" /><br />
  スケール4:<input type="text" scanmode="direct" zoom="4" /><br />
  スケール5:<input type="text" scanmode="direct" zoom="5" /><br />
  スケール6:<input type="text" scanmode="direct" zoom="6" /><br />
  スケール7:<input type="text" scanmode="direct" zoom="7" /><br />
  スケール8:<input type="text" scanmode="direct" zoom="8" /><br />
  スケール9:<input type="text" scanmode="direct" zoom="9" /><br />
  スケール10:<input type="text" scanmode="direct" zoom="10" />
  <hr>
</body>
</html>
```

デモサイト





ナビゲーションバーの表示切り替え

4. ナビゲーションバーの表示切り替え

デモサイト：

<https://www.bluetech.co.jp/iweb/smartbrowser/sample/navbar.html>

デモサイト



4.1 HTML属性 指定 (meta)

SmartBrowserのナビゲーションバーの表示を切り替えます

meta name	要素	値	説明	備考
smartbrowser-settings	navigationbar	true	ナビゲーションバーを表示	
	navigationbar	false	ナビゲーションバーを非表示	

4.2 記述例

```
<meta name='smartbrowser-settings' navigationbar="false">
```



手書きパッドを制御するAPI

5. 手書きパッドを制御するAPI

デモサイト:

<https://www.bluetech.co.jp/iweb/smartbrowser/sample/freedraw.html>

デモサイト



5.1 API起動

smartbrowser().callAPIで「path:'image/free_draw'」を呼び出し手書きパッドを起動します

path	Params	Value	notes
image/free_draw	アプリ側の指定パラメータはなし 個別に指定したパラメータはコールバックに引き継がれます	なし	フリードローの手書きパッドを表示します。手書きした内容は画像として保存します。

5.2 APIコールバック

手書き画像データはPNG形式でURLスキームとなります。データはサーバへ送信して処理を行ってください。

Return value	Json label	Json value
data	result	読み取り結果
	image	読み取り画像のURLスキーム (sample "image": "data:image/png;base64;iVBORw0KGoAAAANSUUhEUgAA.....")
	request	読み取り起動時のParamsの内容

5.3 APIコールバック(jsonデータサンプル)

```
{"request": {}, "result": {"image": "data:image/png;base64,iVBORw0KGgoAAAANSUUhEUgAA....."}}
```

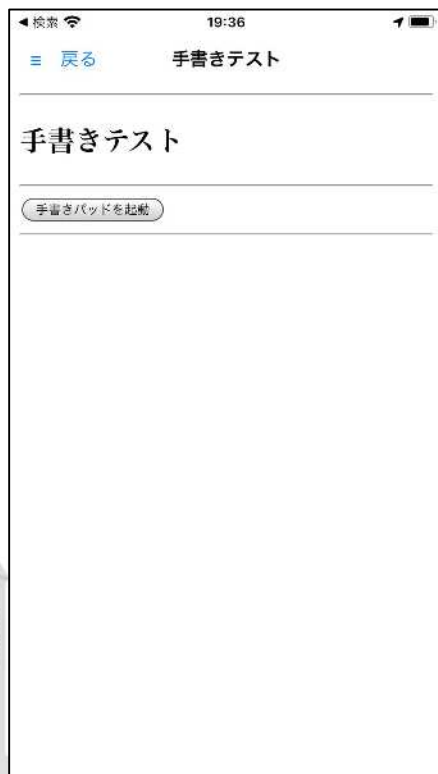


手書きパッドを制御するAPI

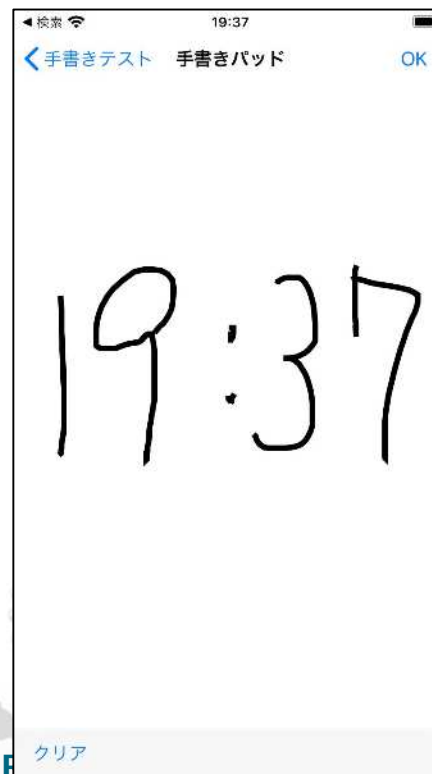
5. 手書きパッドを制御するAPI

5.4 画面イメージ

手書きパッド起動



手書きパッド表示



手書き確定 (画像表示)





手書きパッドを制御するAPI

5. 手書きパッドを制御するAPI

5.5 サンプルソースコード

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>手書きテスト</title>
  <script>
    function freeDraw() {
      smartbrowser().callAPI({
        path: 'image/free_draw',
        params: {
        }
      });
    }

    function onSmartBrowserImageFreeDrawn(data) {
      document.getElementById('freeDrawnImage').src = data.result.image;
      document.getElementById('response').innerText = JSON.stringify(data);
    }

  </script>
</head>
<body>
  <hr>
  <h2>手書きテスト</h2>
  <hr>
  <button onclick="freeDraw();">手書きパッドを起動</button>
  <hr>
  <img id="freeDrawnImage" /><br />
  <div id="response"></div>
</body>
</html>
```

デモサイト

