Device Connect 1.0

HTML5 開発チュートリアル

Ver 1.00

変更履歴

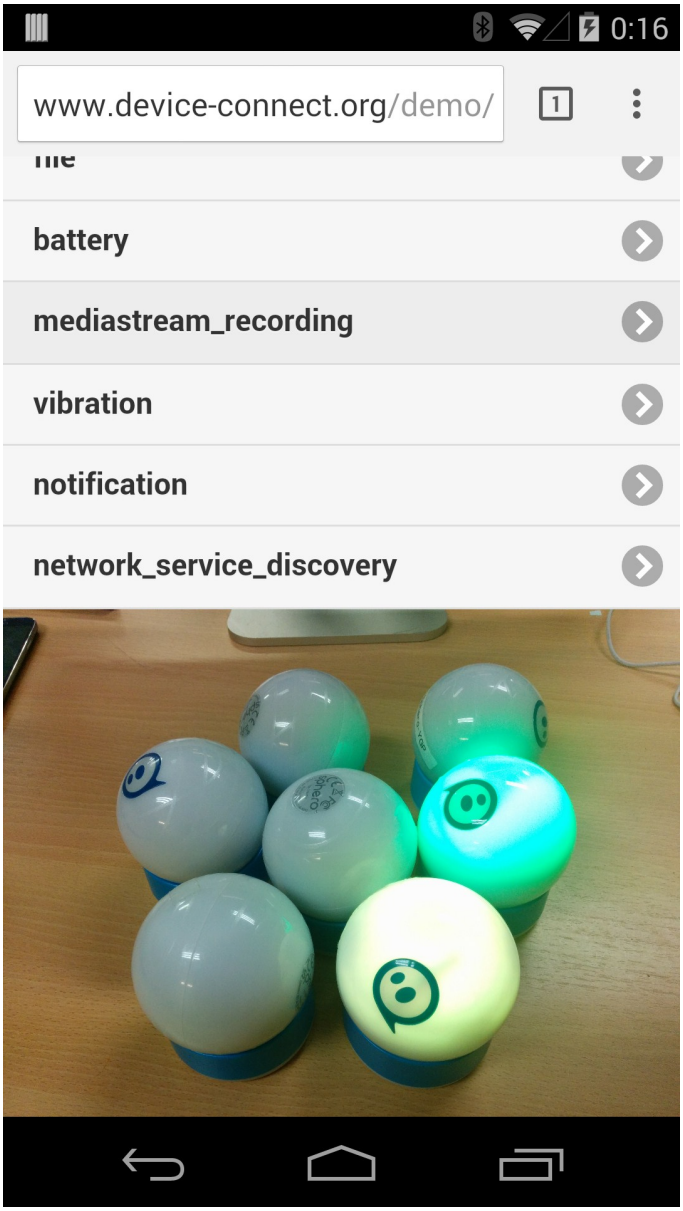
変更日付	変更内容	変更担当者
2014/09/24	初版作成。	佐々木

1. はじめに

本ドキュメントは、Device Connect API 1.0 を用いて HTML5 アプリを開発する手法について解説する。HTML5 に関する一般的な知識と、JavaScript に関する一般的な知識を保持しているのを前提に、解説していく。

本ドキュメントでは、解説を通し、media_recording プロファイルを用いて写真を撮影する HTML5 アプリの作成をおこなう。

図 0. 完成イメージ



2. Device Connectについて

Device Connectでは、RESTfullのやり取りで、命令やデータの送受信を通してデバイスの制御や値の取得が可能である。各処理は、Profileという形でURIが割り振れ、そのURIをJavaScriptで呼び出す事で、処理をおこなう。

2.1 Device Connect対応のProfile

Device Connectで、サポートしている標準Profileは、以下のProfileである。

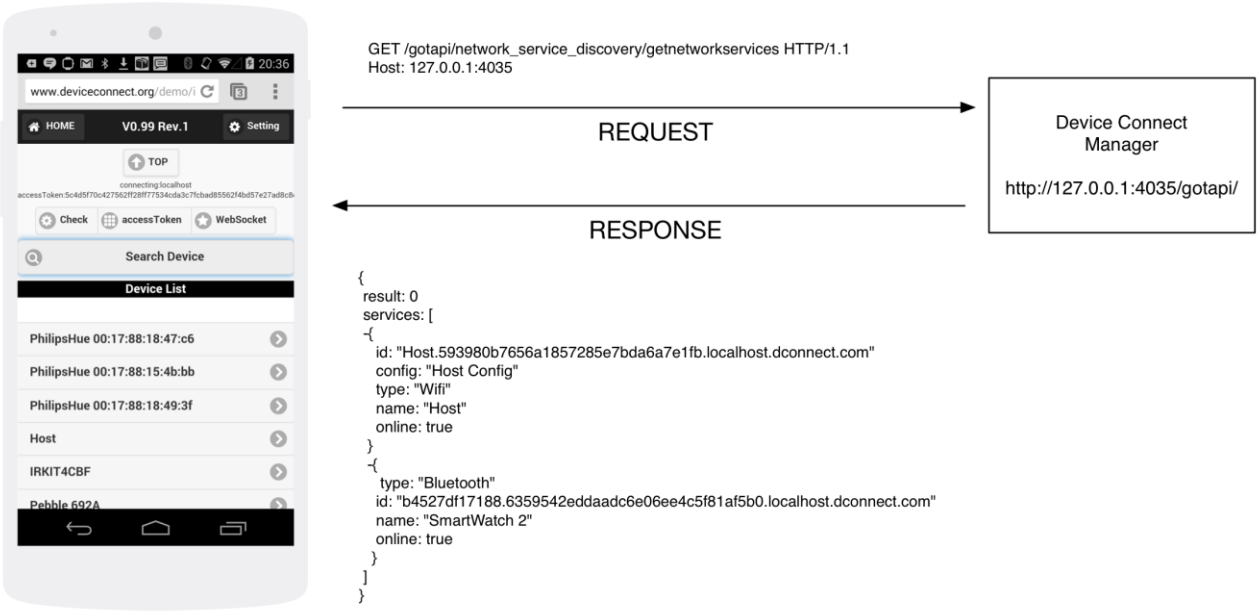
System Profile	Device Connect本体の情報や、プラグインの情報、プラグインの設定画面の起動などができる。
Battery Status Profile	バッテリーの状態を保持できる。
Proximity Profile	近接の状態を取得する。
Media Player Profile	メディアの再生(音声, 音楽, 動画)を行う。
Network Service Discovery Profile	デバイスの対応Profile一覧を取得する。
Vibration Profile	デバイスをバイブレートする命令を送る。
Notification Profile	デバイスに通知(Notification)をおこなう。
Setting Profile I	デバイスの設定(音量, 画面輝度, 画面スリープ時間等)を行う。
DeviceOrientation Profile	デバイスの加速度等を取得できる。
File Profile	デバイスとFileの送信、受信等がおこなえる。
File Descriptor Profile	デバイスにFileの生成、書き込み/読み込み等がおこなえる。
MediaStream Recording Profile	メディアの録音(音声,音楽,動画)をおこなう。
Phone Profile	デバイスに電話発信の命令を送る。

各Profile詳細に関しては、「Device Connect 1.0 RESTful API仕様書」を参照ください。

2.2 Device Connect への接続

それでは、早速、Device Connect に接続し値を取得するサンプルを作りながら解説していく。
Device Connect では、HTML5 で開発されるアプリケーションは、スマートデバイス内のローカル IP で起動している httpd サーバとの HTTP プロトコルで通信をおこなう事で、処理をおこなう。

図 1. Device Connect Manager への問い合わせ



2.3 サンプルの作成

それでは、簡単なサンプルを作りながら解説していく。JavaScript で、AJAX を用いて、Device Connect Manager に問い合わせし、結果の JSON を取得するサンプルを作成する。下記の index.html を入力し、任意のサーバにアップロードする。

index.html

```
<html>
<head>
    <title>Sample</title>
    <script>
        function getVersion() {

            var uri = “http://127.0.0.1:4035/gotapi/system”;

            var xhr = new XMLHttpRequest();
            xhr.open("GET", uri, false);
            xhr.onreadystatechange = function() {
                if (xhr.readyState === 4) {
                    if (xhr.status === 200 || xhr.status == 0) {
                        var json = JSON.parse(xhr.responseText);
                        alert(json.version);
                    } else {

                    }
                }
            };
            xhr.send(null);
        }
    </script>
</head>
<body>
    <input type="button" value="button" onclick="getVersion();">
</body>
</hrml>
```

任意のサーバにアップロードした index.html に、Android の Chrome ブラウザを起動し、接続します。ボタンを押して、Device Connect の Version が取得できたら成功である。


20:57
www.device-connect.org/demo/
button


20:56
www.device-connect.org/demo/
button
ページ www.device-connect.org の記述:
1.0
OK

このように Request では、Host 名と URI を指定し、HTTP 1.1 の GET プロトコルで Device Connect Manager に問い合わせをおこなう。Response では、問い合わせに対する返答が、JSON フォーマットで返ってくる。

リクエスト 例)

```
GET /gotapi/system HTTP/1.1
Host: 127.0.0.1:4035
```

レスポンス 例)

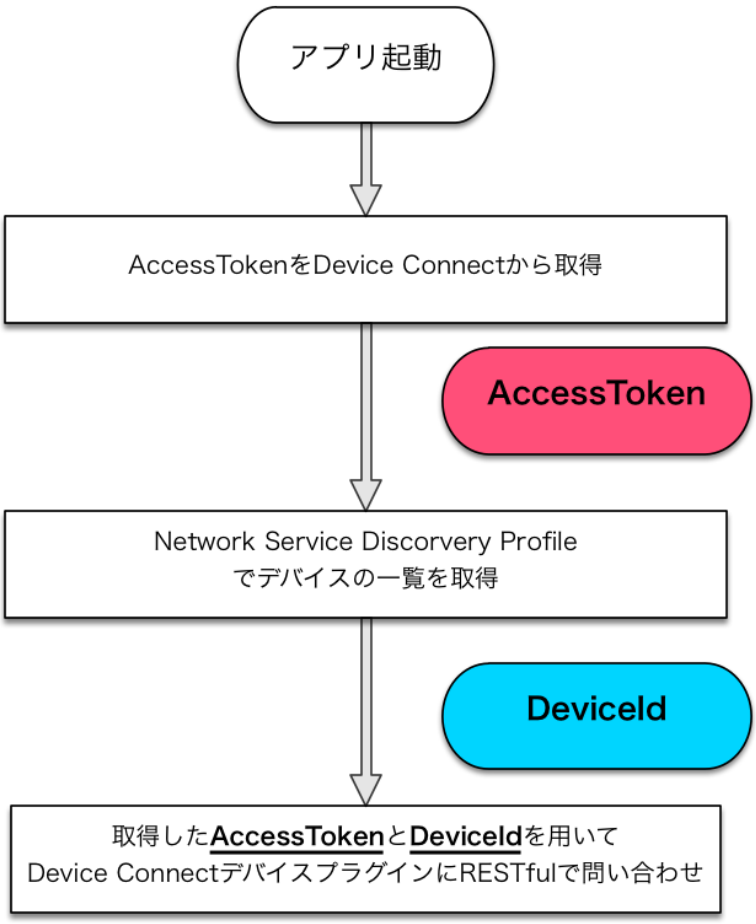
```
{
plugins: [
    {
       id: “604ffa70c537ae49471c6babb16bb4.localhost.dconnect.com",
       name: "Sony Camera"
     },
    {
      id: “593980b7656a1857285e7bda6a7e1fb.localhost.dconnect.com",
      name: "Host"
    },
],
result: 0,
supports: [
     “files",
     “network_service_discovery",
     “system",
     “authorization"
],
version: "1.00"
}
```

返り値の JSON は、正常処理の時は、result: 0 が、異常処理の場合は、result:1 が返ってくる。System Profile に関しては、Device Connect 1.0 RESTful API 仕様書(Device Connect API System Profile)にフォーマット詳細が記載されている。

3 SDK を用いた HTML5 アプリの開発

Device Connect との通信には、いくつかの手順を行う必要がある。HTML5 でアプリを開発する際に、最初に AccessToken を取得する。また、接続したハードウェアを特例するために、各デバイスをユニークに示す DeviceId を Network Service Discovery Profile に問い合わせし取得する。AccessToken と DeviceId の 2 つの値を用いて、操作したいハードウェアに対応した Device Connect デバイスプラグインに対して、処理の実施をおこなう。

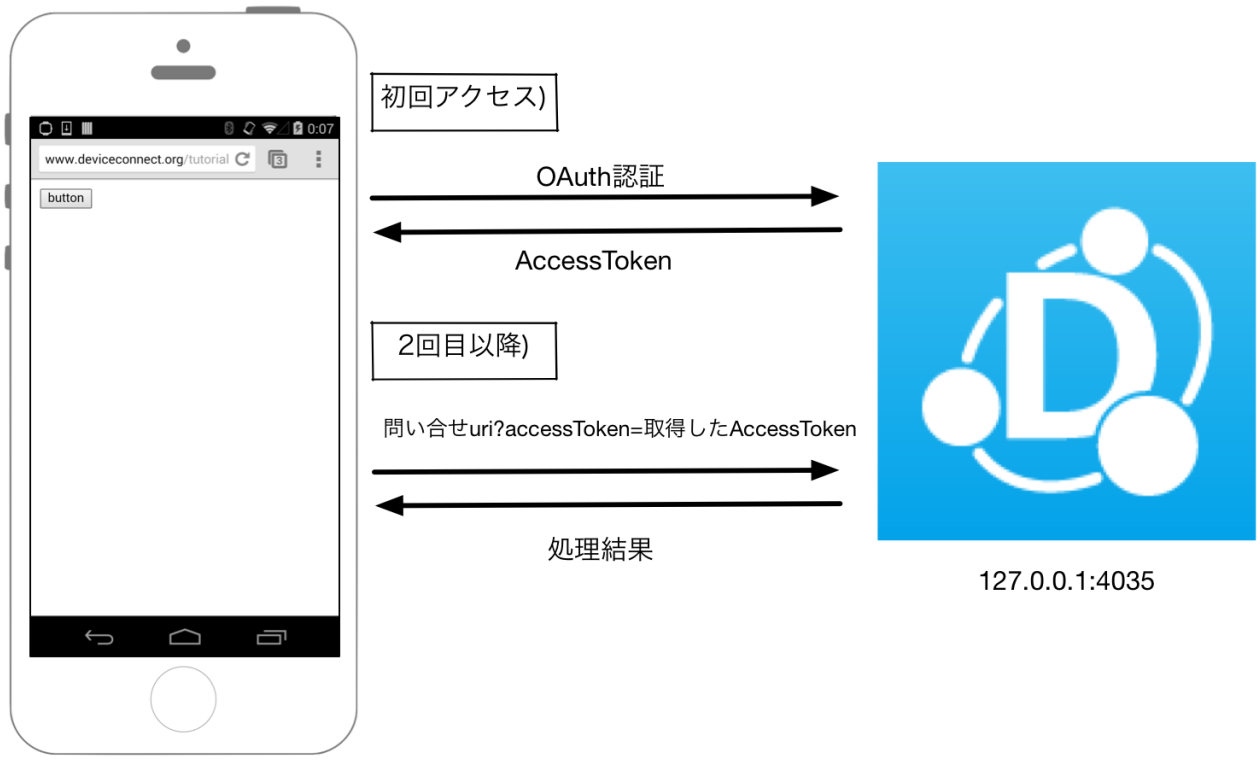
図 2. Device Connect デバイスプラグインにアクセスするまでの処理



3.1 AccessToken を Device Connect から取得

Device Connect にアクセスするアプリは、Device Connect から発行される OAuth Token を用いて Device Connect にアクセスする。そのため、HTML5 で開発するアプリケーションでは、最初に OAuth プロトコルを用いて API 利用の認証を受ける必要がある。

図 3. 初回の OAuth 認証の取得と 2 回目以降の流れ



それでは、早速、OAuth 認証をおこなうサンプルを作成する。

今回編集するファイル群

index.html	HTML 関連処理
js/main.js	JavaScript の処理
js/dconnectsdk-1.0.0.js	Device Connect JavaScript SDK
js/jquery-1.11.0.min.js	JQuery 1.11

index.html

```
<html>
<head>
    <title>Sample</title>
    <meta name="viewport" content="width=device-width, user-scalable=no" />
    <script src="js/dconnectsdk-0.1.0.js" type="text/javascript"></script>
    <script src="js/jquery-1.11.0.min.js" type="text/javascript"></script>
    <script src="js/main.js" type="text/javascript"></script>
</head>
<body>
    <input type="button" value="Get accessToken" onclick="authorization();">
</body>
</hrml>
```

js/main.js

```
// accessToken を保存
var accessToken = "";

// アプリ名
var myAppName = "com.test.html5.app1";

function authorization(){

       var scopes = Array("deviceorientation", "mediastream_recording");

       dConnect.authorization('http://www.deviceconnect.org/demo/', scopes, myAppName,
              function(clientId, clientSecret, newAccessToken) {
                     // accessToken の取得
                     accessToken = newAccessToken;
              },
              function(errorCode, errorMessage) {
                     alert("Failed to get accessToken.");
       }
    );
}
```

OAuth の認証では、HTML5 アプリが利用したいプロファイルの認証をおこなう。

```
var scopes = Array("deviceorientation", "mediastream_recording");
```

本例では、deviceorientation と mediastream_recording のプロファイルへのアクセス要求をおこなっている。ユーザが「同意する」ボタンを押す事で、指定したプロファイルへのアクセスが可能な AccessToken が発行される。以後、ここで取得した AccessToken を用いる事で、HTML5 アプリは、プロファイルを利用する事が可能になる。

3.2 DeviceIDの取得

前項までで、AccessTokenを取得した事で、同意を得たプロファイルにアクセスできるようになった。本校では、特定デバイスへのアクセスに必要なDeviceIDの取得方法に関して解説する。

DeviceIDは、Network Service Discovery Profileを用いて取得可能である。Network Service Discovery Profileでは、Device Connectが接続可能なデバイスの一覧を取得してくる。

本項目から、取得した値は、JQuery Mobileで、画面に表示しながら、取得方法の解説をおこなう。

今回編集するファイル群

index.html	HTML関連処理
js/main.js	JavaScriptの処理
js/dconnectsdk-1.0.0.js	Device Connect JavaScript SDK
js/jquery-1.11.0.min.js	JQuery 1.11
js/jquery.mobile-1.4.2.min.js	JQuery Mobile 1.4.2 JavaScript
css/jquery.mobile-1.4.2.min.css	JQuery Mobile 1.4.2 CSS

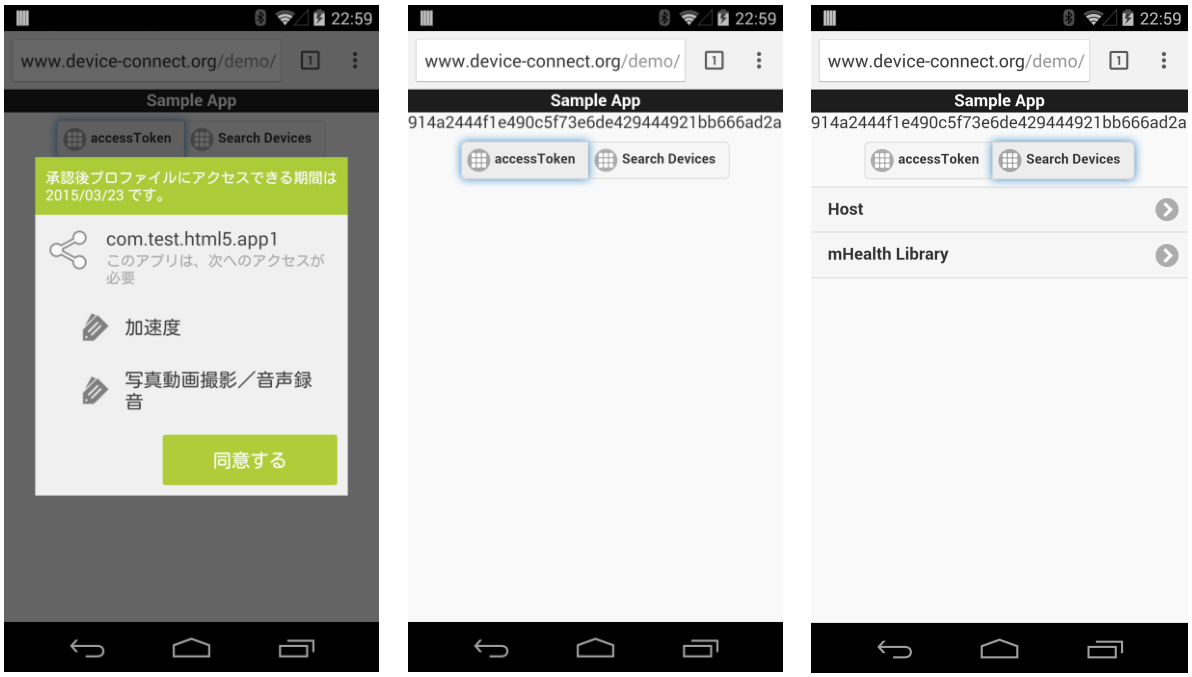
index.html

```
<html>
<head>
    <title>Sample</title>
    <meta name="viewport" content="width=device-width, user-scalable=no" />
    <script src="js/jquery-1.11.0.min.js" type="text/javascript"></script>
    <script src="js/jquery.mobile-1.4.2.min.js" type="text/javascript"></script>
    <script src="js/dconnectsdk-1.0.0.js" type="text/javascript"></script>
    <script src="js/main.js" type="text/javascript"></script>
    <link href="css/jquery.mobile-1.4.2.min.css" rel="stylesheet" />
</head>
<body>
     <div id="page1" data-role="page">
        <div data-role="header" data-theme="b" style="text-align:center">
                  Sample App
        </div>

        <div id="accessToken">

        </div>

        <div data-role="controlgroup" data-type="horizontal" style="text-align:center" >
            <input data-icon="grid" data-inline="true" data-mini="true"
onclick="javascript:authorization();" type="button" value="accessToken" />
            <input data-icon="grid" data-inline="true" data-mini="true"
onclick="javascript:searchDevice();" type="button" value="Search Devices" />
        </div>

        <ul class="list" data-role="listview" id="list"></ul>
    </div>
</body>
</hrml>
```

js/main.js

```
// accessTokenを保存
var accessToken = "";

// アプリ名
var myAppName = "com.test.html5.app1";

function authorization(){

        var scopes = Array("deviceorientation", "mediastream_recording");

        dConnect.authorization('http://www.deviceconnect.org/demo/', scopes, myAppName,
                function(clientId, clientSecret, newAccessToken) {
                        // accessTokenの取得
                        accessToken = newAccessToken;
                },
                function(errorCode, errorMessage) {
                        alert("Failed to get accessToken.");
        }
    );
}

// デバイスの検索
function searchDevice() {
    dConnect.discoverDevices(function(status, headerMap, responseText) {
        var str = "";
        var obj = JSON.parse(responseText);
        if (obj.result == 0) {
            for (var i = 0; i < obj.services.length; i++) {
                str += '<li><a href="javascript:searchSystem(¥'' + obj.services[i].id +
'¥');"';
                str += 'value= “' + obj.services[i].name + '">' + obj.services[i].name +
'</a></li>';
            }
        }

        var listHtml = document.getElementById('list');
        listHtml.innerHTML = str;
        $("ul.list").listview("refresh");

    }, function(readyState, status) {

    });
}
```

図 5. OAuth 認証ダイアログボックスと DeviceID の取得



accessToken ボタンを押す事で、accessToken を取得し、SearchDevice ボタンを押す事で、DeviceID を取得してきている。

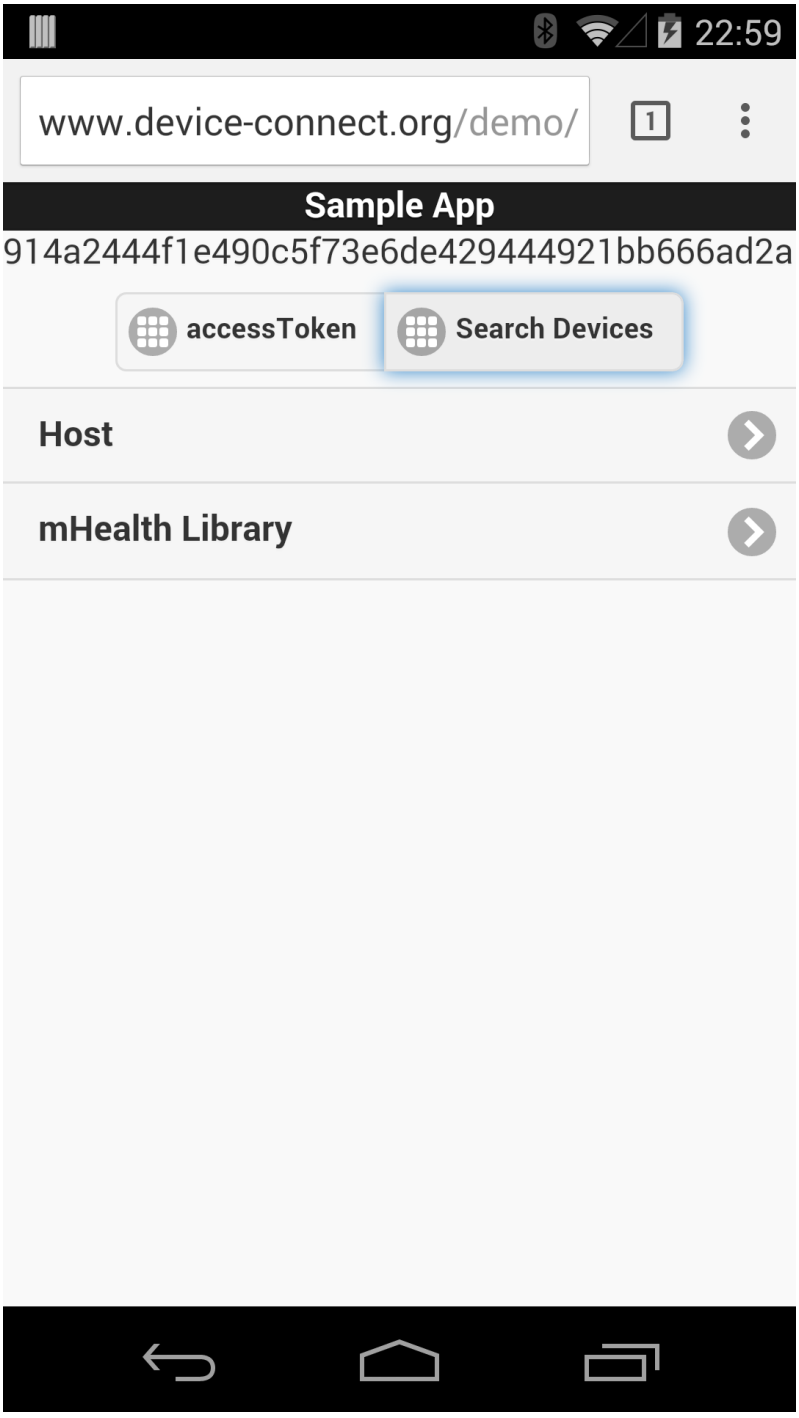
デバイスの検索は、下記に抜粋した js/main.js の searchDevice()でおこなっている。

```
// デバイスの検索
function searchDevice() {
    dConnect.discoverDevices(function(status, headerMap, responseText) {
        var str = "";
        var obj = JSON.parse(responseText);
        if (obj.result == 0) {
            for (var i = 0; i < obj.services.length; i++) {
                str += '<li><a href="javascript:searchSystem(¥'' + obj.services[i].id +
'¥');"';
                str += 'value= "' + obj.services[i].name + '">' + obj.services[i].name +
'</a></li>';
            }
        }

        var listHtml = document.getElementById('list');
        listHtml.innerHTML = str;
        $("ul.list").listview("refresh");

    }, function(readyState, status) {

    });
}
```

seacchDevice()では、dConnect.discoverDevice()を呼び出す事で、接続可能な端末の一覧を取得してくる。本例では、取得したデバイスListは、JQuery MobileのListViewに格納し、一覧を表示している。

```
<li><a href=“javascript:searchSystem(deviceId)” value=“プラグイン名”>プラグイン名
</a>
```

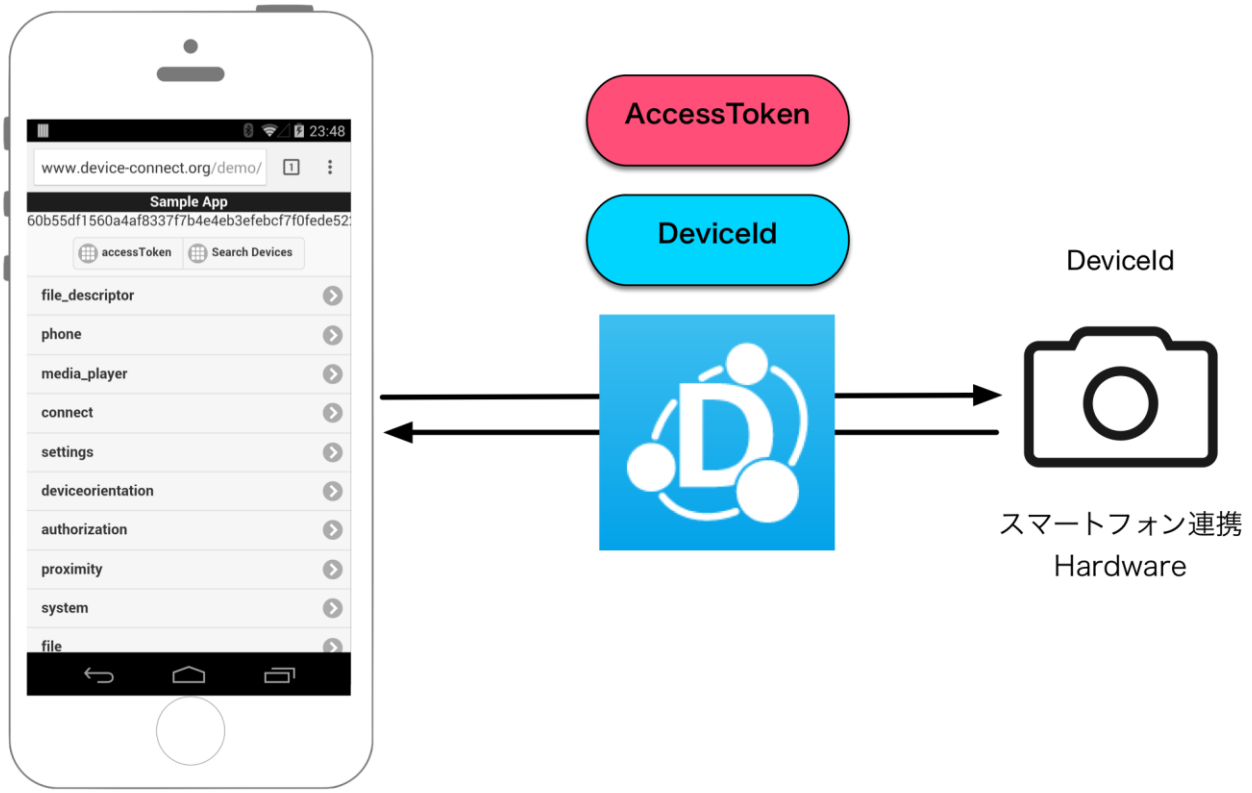
図 5. DeviceIDとデバイス名の一覧



3.3 AccessToken と DeviceId を用いてプラグインに問い合わせ

前項までに、AccessToken と DeviceID を取得した。この 2 つを用いる事で、Device Connect デバイスプラグインへのアクセスが可能になる。Device Connect デバイスプラグインでは、対応するプロファイルを通して、ハードウェアへのアクセスが可能になる。

図 6. AccessToken と DeviceId とスマートフォン連携 Hardware



今回編集するファイル群

index.html	HTML 関連処理
js/main.js	JavaScript の処理
js/dconnectsdk-1.0.0.js	Device Connect JavaScript SDK
js/jquery-1.11.0.min.js	JQuery 1.11
js/jquery.mobile-1.4.2.min.js	JQuery Mobile 1.4.2 JavaScript
css/jquery.mobile-1.4.2.min.css	JQuery Mobile 1.4.2 CSS

js/main.js

```
// accessTokenを保存
var accessToken = "";

// アプリ名
var myAppName = "com.test.html5.app1";

function authorization(){

        var scopes = Array("deviceorientation", "mediastream_recording");

        dConnect.authorization('http://www.deviceconnect.org/demo/', scopes, myAppName,
                function(clientId, clientSecret, newAccessToken) {
                        // accessTokenの取得
                        accessToken = newAccessToken;

                        // accessTokenをWebに表記
                        $('#accessToken').html(accessToken).trigger('accessToken');
                },
                function(errorCode, errorMessage) {
                        alert("Failed to get accessToken.");
        }
    );
}

// デバイスの検索
function searchDevice() {
    dConnect.discoverDevices(function(status, headerMap, responseText) {
        var str = "";
        var obj = JSON.parse(responseText);
        if (obj.result == 0) {
            for (var i = 0; i < obj.services.length; i++) {
                str += '<li><a href="javascript:searchSystem(¥'' + obj.services[i].id +
'¥');"';
                str += 'value= “' + obj.services[i].name + '">' + obj.services[i].name +
'</a></li>';
            }
        }

        var listHtml = document.getElementById('list');
        listHtml.innerHTML = str;
        $("ul.list").listview("refresh");

    }, function(readyState, status) {

    });
}

// デバイスのプロファイルの一覧
function searchSystem(deviceId) {

    var builder = new dConnect.URIBuilder();
    builder.setProfile("system");
    builder.setAttribute("device");
```

```
    builder.setDeviceId(deviceId);
    builder.setAccessToken(accessToken);
    var uri = builder.build();

    dConnect.execute('GET', uri, null, null, function(status, headerMap, responseText) {
        var json = JSON.parse(responseText);

        if (json.result == 0) {
            var str = "";
            for (var i = 0; i < json.supports.length; i++) {
                str += '<li><a href="javascript:searchProfile(¥'' + deviceId + '¥', ';
                str += '¥'' + json.supports[i] + '¥');" ';
                str += 'value="' + json.supports[i] + '">';
                str += json.supports[i] + '</a></li>';
            }

            var listHtml = document.getElementById('list');
            listHtml.innerHTML = str;
            $("ul.list").listview("refresh");

        } else {

        }
    }, function(xhr, textStatus, errorThrown) {

    });
}
```

本例では、デバイスプラグインのプロファイル一覧を取得するために、searchSystem メソッドを js/main.js に追加している。本例では、プロファイルへの接続は、URIBuilder を用いて接続している。

URI が

```
system/device
```

のプロファイル、つまり System プロファイルへアクセスしたい場合は、dconnectsdk を用いる事で、下記のように記述する事が可能になる。

```
var builder = new dConnect.URIBuilder();
    builder.setProfile("system");
    builder.setAttribute("device");
    builder.setDeviceId(deviceId);
    builder.setAccessToken(accessToken);
    var uri = builder.build();
```

上記の builder で生成される生成される URL は、下記の通りである。

```
http://127.0.0.1/4035/gotapi/system/device?deviceId=取得した DeviceID&accrssToken=取得したアクセストークン
```

この URI を用いて、dConnect.execute()で、その URI への接続をおこなう。レスポンスは、JSON 形式で送られてくる。

3.4 プロファイルへのアクセス

ここから先は、Device Connect デバイスプラグイン「Host」プラグインを用いて解説していく。「Host」プラグインでは、mediastream_recording プロファイルへのアクセスが可能である。本項目では、写真を撮影するサンプルを解説する。

今回編集するファイル群

index.html	HTML 関連処理
js/main.js	JavaScript の処理
js/jquery-1.11.0.min.js	JQuery 1.11
js/dconnectsdk-1.0.0.js	Device Connect JavaScript SDK
js/jquery.mobile-1.4.2.min.js	JQuery Mobile 1.4.2 JavaScript
css/jquery.mobile-1.4.2.min.css	JQuery Mobile 1.4.2 CSS

今回必要な DeviceConnect デバイスプラグイン

「Host」プラグイン	Host プラグインを事前インストールしておく

index.html

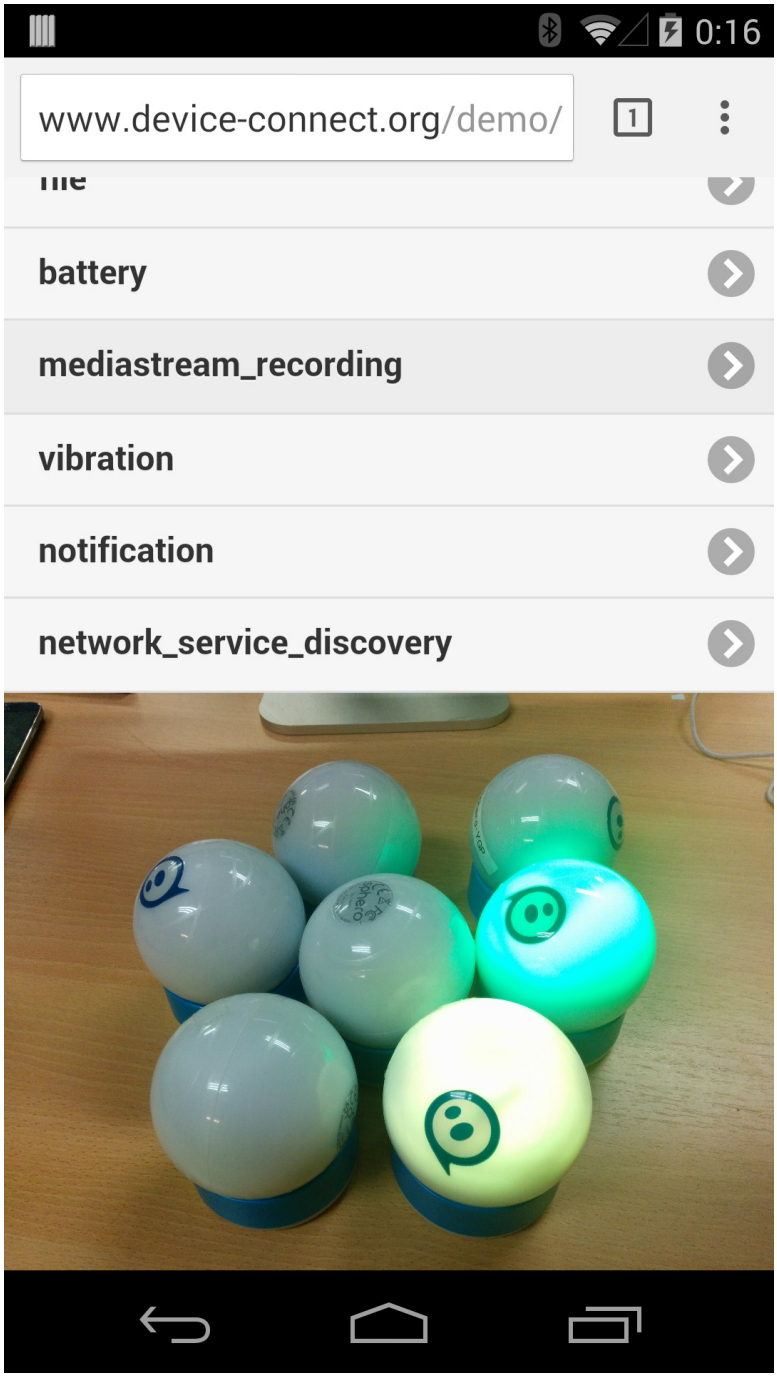
```
<html>
<head>
    <title>Sample</title>
    <meta name="viewport" content="width=device-width, user-scalable=no" />
    <script src="js/jquery-1.11.0.min.js" type="text/javascript"></script>
    <script src="js/jquery.mobile-1.4.2.min.js" type="text/javascript"></script>
    <script src="js/dconnectsdk-1.0.0.js" type="text/javascript"></script>
    <script src="js/main.js" type="text/javascript"></script>
    <link href="css/jquery.mobile-1.4.2.min.css" rel="stylesheet" />
</head>
<body>
     <div id="page1" data-role="page">
        <div data-role="header" data-theme="b" style="text-align:center">
                  Sample App
        </div>

        <div id="accessToken">

        </div>

        <div data-role="controlgroup" data-type="horizontal" style="text-align:center" >
            <input data-icon="grid" data-inline="true" data-mini="true"
onclick="javascript:authorization();" type="button" value="accessToken" />
            <input data-icon="grid" data-inline="true" data-mini="true"
onclick="javascript:searchDevice();" type="button" value="Search Devices" />
        </div>

        <ul class="list" data-role="listview" id="list"></ul>

          <div id="result" data-theme="a"></div>
    </div>
</body>

</hrml>
```

js/main.js

```
// accessTokenを保存
var accessToken = "";

// アプリ名
var myAppName = "com.test.html5.app1";

function authorization(){

        var scopes = Array("deviceorientation", "mediastream_recording");

        dConnect.authorization('http://www.deviceconnect.org/demo/', scopes, myAppName,
                function(clientId, clientSecret, newAccessToken) {
                        // accessTokenの取得
                        accessToken = newAccessToken;

                        // accessTokenをWebに表記
                        $('#accessToken').html(accessToken).trigger('accessToken');
                },
                function(errorCode, errorMessage) {
                        alert("Failed to get accessToken.");
        }
    );
}

// デバイスの検索
function searchDevice() {
    dConnect.discoverDevices(function(status, headerMap, responseText) {
        var str = "";
        var obj = JSON.parse(responseText);
        if (obj.result == 0) {
            for (var i = 0; i < obj.services.length; i++) {
                str += '<li><a href="javascript:searchSystem(¥'' + obj.services[i].id + '¥');"';
                str += 'value= "' + obj.services[i].name + '">' + obj.services[i].name + '</a></li>';
            }
        }

        var listHtml = document.getElementById('list');
        listHtml.innerHTML = str;
        $("ul.list").listview("refresh");

    }, function(readyState, status) {

    });
}

// デバイスのプロファイルの一覧
function searchSystem(deviceId) {

    var builder = new dConnect.URIBuilder();
    builder.setProfile("system");
    builder.setAttribute("device");
    builder.setDeviceId(deviceId);
    builder.setAccessToken(accessToken);
    var uri = builder.build();

    dConnect.execute('GET', uri, null, null, function(status, headerMap, responseText) {
        var json = JSON.parse(responseText);
```

```
        if (json.result == 0) {
            var str = "";
            for (var i = 0; i < json.supports.length; i++) {
                str += '<li><a href="javascript:searchProfile(¥'' + deviceId + '¥', ';
                str += '¥'' + json.supports[i] + '¥');" ';
                str += 'value="' + json.supports[i] + '">';
                str += json.supports[i] + '</a></li>';
            }

            var listHtml = document.getElementById('list');
            listHtml.innerHTML = str;
            $("ul.list").listview("refresh");

        } else {

        }
    }, function(xhr, textStatus, errorThrown) {

    });
}
// Profile別処理の分岐
function searchProfile(deviceId, profile) {
        if (profile == "mediastream_recording") {
        doTakePhoto(deviceId);
    }
}

// 写真を撮影しWebに表示
function doTakePhoto(deviceId) {

    var builder = new dConnect.URIBuilder();
    builder.setProfile("mediastream_recording");
    builder.setAttribute("takephoto");
    builder.setDeviceId(deviceId);
    builder.setAccessToken(accessToken);

    var uri = builder.build();

    dConnect.execute('POST', uri, null, null, function(status, headerMap, responseText) {
        var json = JSON.parse(responseText);

        if (json.result == 0) {
            var str = "";
            str += '<img src="' + json.uri + '" width="100%">';
            str += '</center><br>';
            $('#result').html(str).trigger('create');

        } else {

        }

    }, function(xhr, textStatus, errorThrown) {
    });

}
```

図 7. デモ


0:16
www.device-connect.org/demo/
battery
mediastream_recording
vibration
notification
network_service_discovery