

GotAPIによる Webからのデバイス活用

2015/9/17

1. GotAPIとは？
2. あえて未インストールでGotAPI体験(Gclue社デモサイト)
3. GotAPIの色々な機能を試す
(GotAPIのOSS実装: デバイスコネクトWebAPI)
4. curlによるREST API動作確認

1. GotAPIとは？

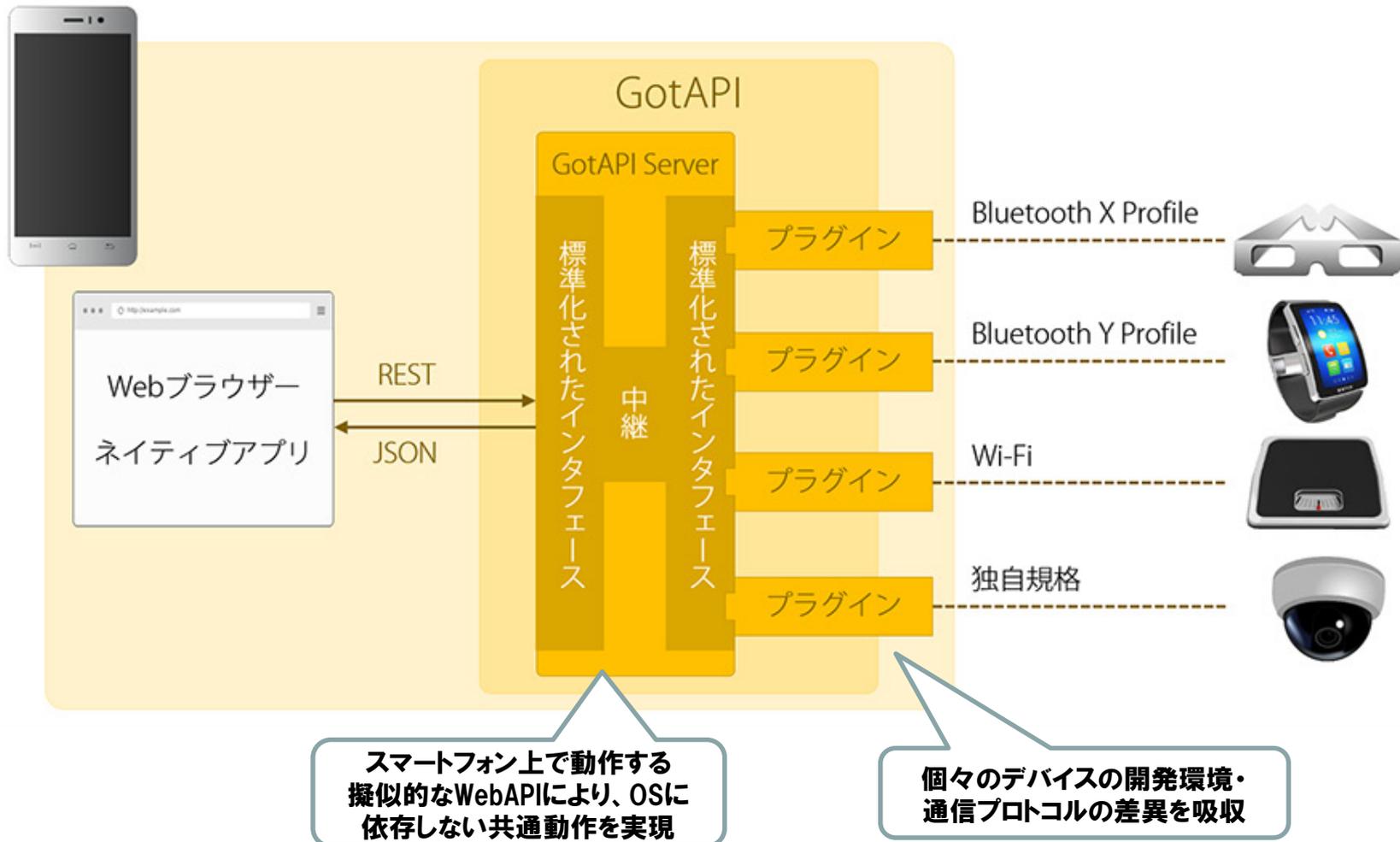
2. あえて未インストールでGotAPI体験(Gclue社デモサイト)

3. GotAPIの色々な機能を試す
(GotAPIのOSS実装: デバイスコネクトWebAPI)

4. curlによるREST API動作確認

1.1 GotAPI (Generic Open Terminal API)とは

- 多種多様なデバイスを統一したWebインタフェースに統合
- プラグイン方式により柔軟なデバイス追加を可能に
- 詳しくはコンソーシアムWebサイトで <http://device-webapi.org/gotapi.html>



スマートフォン上で動作する
擬似的なWebAPIにより、OSに
依存しない共通動作を実現

個々のデバイスの開発環境・
通信プロトコルの差異を吸収

1.2 APIアクセスのデザインパターン

- **One shot data** : HTTP GET/POSTでの単純アクセス
例: アクセスした瞬間の加速度センサーの値を取得(繰り返し値が欲しい場合はポーリング)
- **Event driven data** : PUT/DELETE、WebSocketでのイベント処理
例: 加速度センサーの値に変化があった瞬間の値を連続的に自動取得
- **Streaming data** : URIの直接参照
例: カメラ映像を要求した場合に、OSやGotAPIの内部構造を経由せずに映像リソースとしてアプリから直接利用。

シンプルで手軽なHTTPアクセスと、効率的なWebSocketによるイベント処理を両立

1.3 GotAPI仕様の入手先

以下のURL(OMA※のWebサイト)から無料で入手可能

Full Specification

<http://technical.openmobilealliance.org/Technical/technical-information/release-program/current-releases/generic-open-terminal-api-framework-1-0>

White Paper

<http://openmobilealliance.hs-sites.com/gotapi-making-the-internet-of-things-interoperable>

※OMA(Open Mobile Alliance): 携帯電話関連の標準化団体

1.4 デバイスアクセスのためのAPI

・GotAPIの規定はWebインターフェースの実現についてのみ

⇒ 基本的な機能セットはOSSのGotAPI実装「デバイスコネクトWebAPI」で規定し、プラグイン開発のためのライブラリも整備



ServiceDiscovery
接続デバイスの発見



Proximity
近接検知



Battery
バッテリー情報取得



Media Player
音楽・動画の再生制御

Authorization
アプリ認証

DeviceOrientation
デバイスの傾き

Vibration
バイブレーション

MediaStream Recording
録画・録音・写真撮影

ServiceInformation
デバイス情報

Temperature
温度検知

Light
ライト点灯と消灯

※ 上記はAPIの一例です

1.5 プラグインのラインナップ

・オープンな開発環境を提供するデバイスをプラグイン化



※ 上記はプラグインの一例です

1.6 スマートフォンのWebブラウザからの利用

Gclue社Webサイトから試せます

<http://www.gclue.io/dwa/demo/#/>

Webの標準的な仕組みのみで、Webサイト上にGotAPI利用のための導線
(アプリストアからのインストール・個々の機器設定)を提供可能

⇒ プリインストールでなくても利用でき、個々の機器設定もサポート

Device Web API Manager

Home News API Demo SDK ▾ Link

Device Web API Manager

Device connection with RESTful

ANDROID APP ON Google play

Android OS 4.2.x以降

1. GotAPIとは？

2. あえて未インストールでGotAPI体験(GClue社デモサイト)

3. GotAPIの色々な機能を試す
(GotAPIのOSS実装: デバイスコネクトWebAPI)

4. curlによるREST API動作確認

2.1 デモ:あえて未インストールでデモサイトにアクセス

- 1.あえてアプリをアンインストールしてから、ブラウザでGClue社デモWebサイトを開く
<http://www.gclue.io/dwa/demo/#/>
- 2.[確認する]を選ぶとGooglePlayのリンクが開くので、アプリ（デバイスWebAPIマネージャ）をインストール
- 3.Webサイトに戻って再度[確認する]を選ぶとマネージャの起動確認画面になり、[launch]を選ぶとデモWebサイトの画面に

※マネージャが有効になっている場合は最初からデモWebサイトが表示される

- 4.[心拍計]を選ぶと、プラグインのインストール画面になるのでインストール
- 5.画面の指示に従ってデバイスをセットアップ（Bluetoothの有効化、ペアリング等）
- 6.再度デモWebサイトで[心拍計]を選ぶと心拍計デモ画面に

※プラグインがインストールされている場合は最初からデモ画面が表示される



Webサイトからの導線に完結する形で、必要アプリのインストールからデバイスのセットアップまで実現

各デバイスのアプリストアのアカウントを登録し、Device WebAPIのマネージャーアプリとホストプラグインをインストール

Vuzix M100の場合 (<http://www.vuzix.jp/appstore-howtouse.html>)

0.[M100/PC]最新バージョンへのアップデート(Ver2.3)

- 1.[スマホ or PC]Vuzixアプリストアのアカウントを登録し、“DeviceWebAPI Manager”、“host Device Web API Plug in”をライブラリに追加
- 2.[M100]設定のApp Updaterからアカウント情報を入力
- 3.[M100]”Check for New Apps...”からアプリを選んでしばらく待つ(一つだけ選ぶ)
- 4.[M100]確認画面が出たら”Install”を選んでインストール(上手く行ったらもう片方も)

Epson BT-200の場合



- 1.MOVERIO Apps Marketのアカウントを登録
- 2.”DeviceWebAPI Manager”、“host (Device Web API Plug in)”をインストール

ポイント:開発用ファームウェアへの書換え不要でWeb開発が可能に

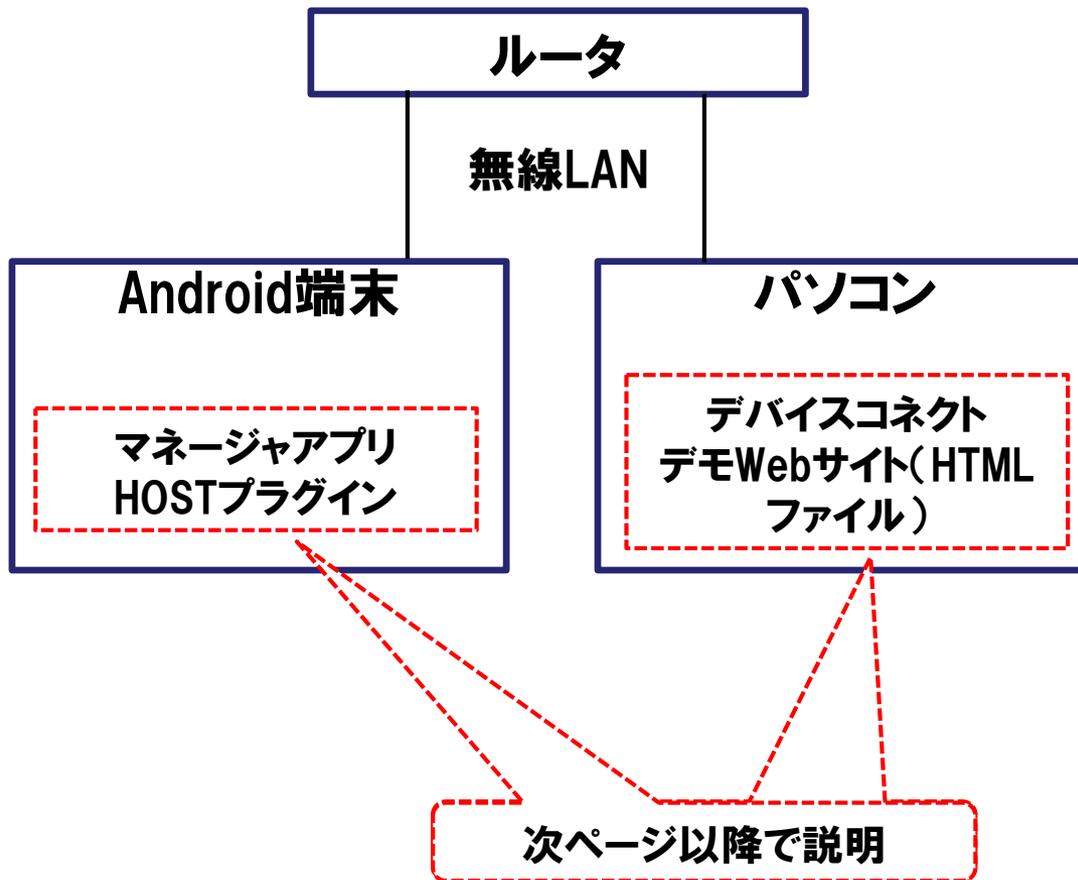
1. GotAPIとは？

2. あえて未インストールでGotAPI体験(Gclue社デモサイト)

3. GotAPIの色々な機能を試す
(GotAPIのOSS実装: デバイスコネクトWebAPI)

4. curlによるREST API動作確認

1. 同一ネットワークのパソコンとAndroid端末を用意



3.2 STEP:1 ADBを利用したアプリインストール

0.Android端末、PC開発環境の準備

1.GitHubからPCにデモ用ファイルのダウンロード

<https://github.com/DeviceConnect/DeviceConnect-Docs/blob/master/Bin/demoWebSite.zip>

2.「ADB Install」コマンドでマネージャアプリ、Hostプラグイン)をAndroid端末にインストール(開発環境がなければGClue社アプリでもOK)

> adb install .%demoWebSite%apk%dConnectManager.apk [改行]

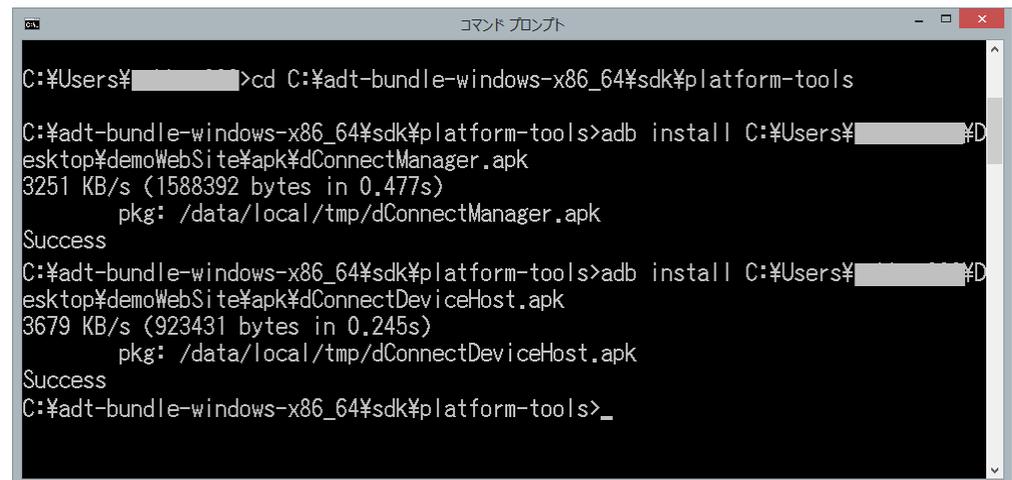
Successと出たら成功

> adb install .%demoWebSite%apk%dConnectDeviceHost.apk [改行]

Successと出たら成功

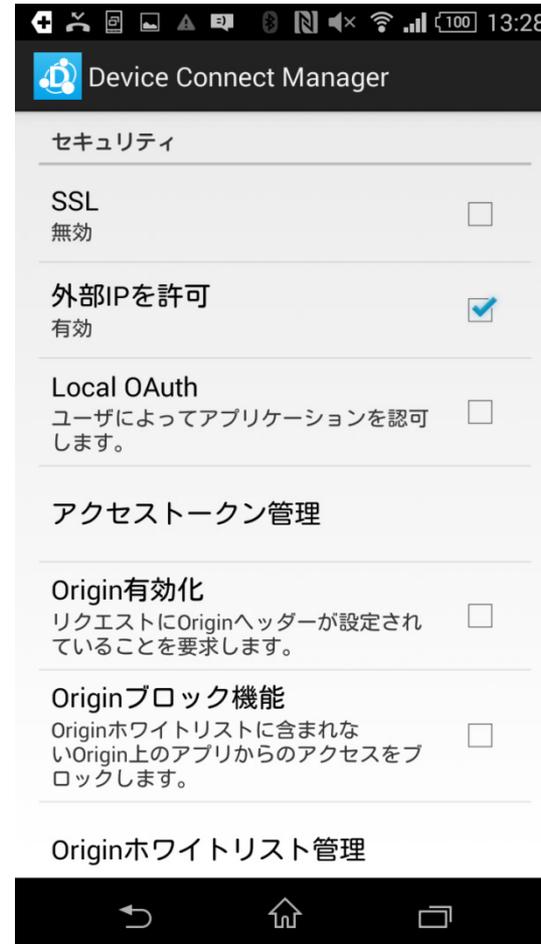
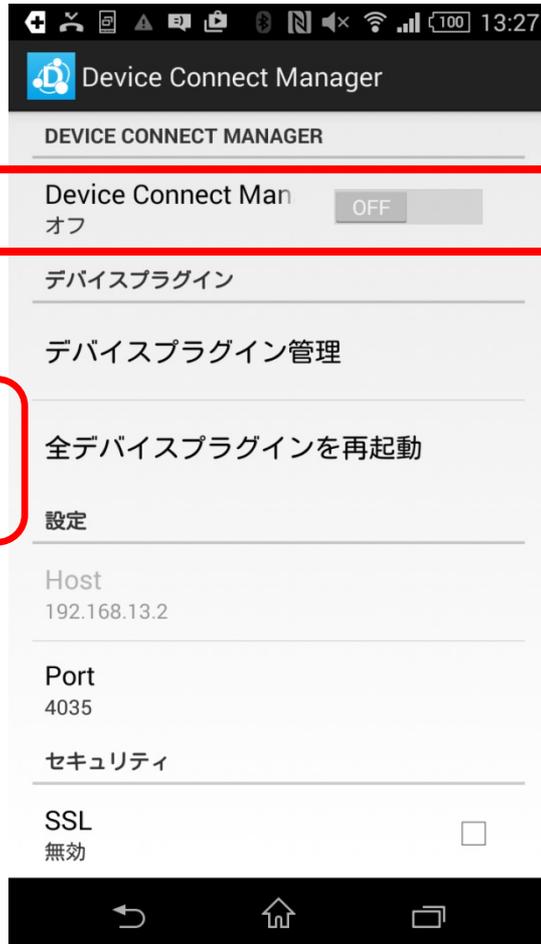
ダウンロードファイル

```
demoWebSite
├── [apk]
│   ├── dConnectManager.apk
│   ├── dConnectDeviceHost.apk
│   └── その他のAPKファイル
├── [css]
├── [js]
└── index.html
```



```
コマンド プロンプト
C:\Users\%>cd C:\adt-bundle-windows-x86_64\sdk\platform-tools
C:\adt-bundle-windows-x86_64\sdk\platform-tools>adb install C:\Users\%demoWebSite%apk%dConnectManager.apk
3251 KB/s (1588392 bytes in 0.477s)
  pkg: /data/local/tmp/dConnectManager.apk
Success
C:\adt-bundle-windows-x86_64\sdk\platform-tools>adb install C:\Users\%demoWebSite%apk%dConnectDeviceHost.apk
3679 KB/s (923431 bytes in 0.245s)
  pkg: /data/local/tmp/dConnectDeviceHost.apk
Success
C:\adt-bundle-windows-x86_64\sdk\platform-tools>_
```

とりあえず試すために、マネージャアプリのセキュリティを無効にする
※セキュリティ設定の説明はまた別の機会に…



マネージャアプリを開いてサーバ機能をオフにし、セキュリティ設定の外部IPのみを有効にしてからサーバ機能をオンにする

3.4 STEP:3 PCでデモ用Webサイトを開く

補足

ChromeブラウザVer.45以降、PC・AndroidともにFileスキーム(file:///)でのデモの動作確認が出来なくなったため、**暫定対処**としてLocalhost上にWebコンテンツを置けるようにしています

1. ダウンロードファイルの中身をadb pushでAndroid端末のストレージにコピー



ダウンロードしたZIPファイルをフォルダに展開

> adb push **demoWebSite** ← PCにダウンロードしたファイルのフォルダ (ドラック&ドロップでOK)
/sdcard/org.deviceconnect.android.manager/demoWebSite [改行]

2. Webブラウザで以下のURLを入力して開く

“http://[Android端末のIPアドレス]:8080/demoWebSite/index.html?ip=[Android端末のIPアドレス]#demo”

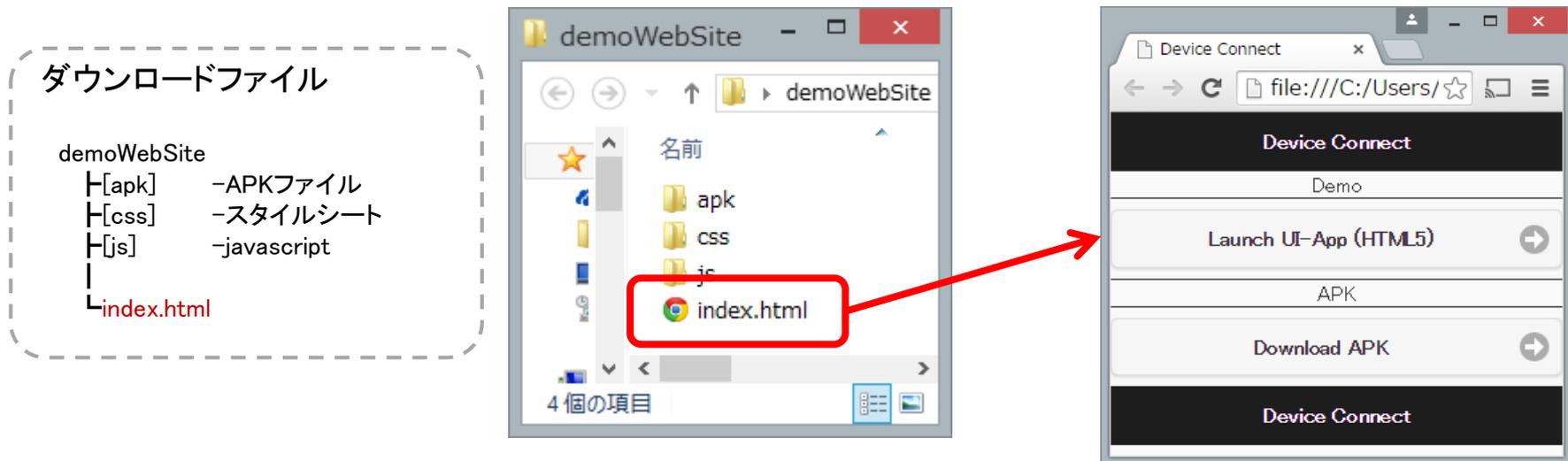
インストールしたAndroid端末上で開く場合は

“http://localhost:8080/demoWebSite/index.html#demo”

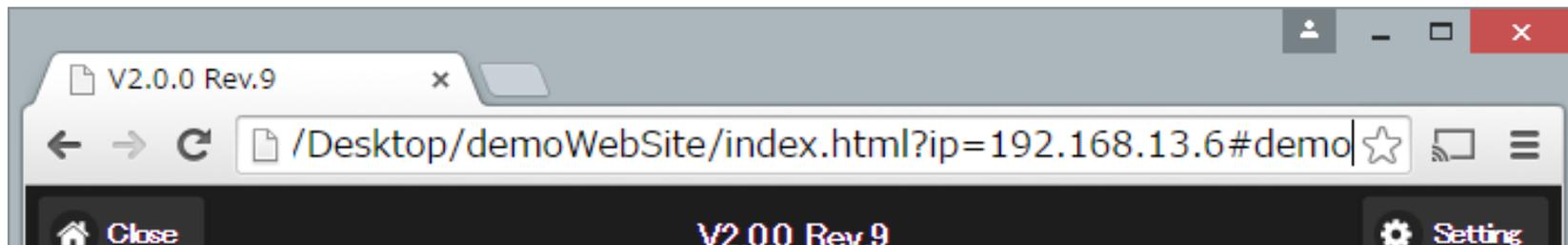
操作対象のIPアドレスとして
パラメータを付加

1. Firefoxブラウザでは、ダウンロードファイルのindex.htmlを開くだけで確認できる

※ChromeブラウザVer.45以降、PC・AndroidともにFileスキーム(file:///)でのデモの動作確認が出来なくなっております。

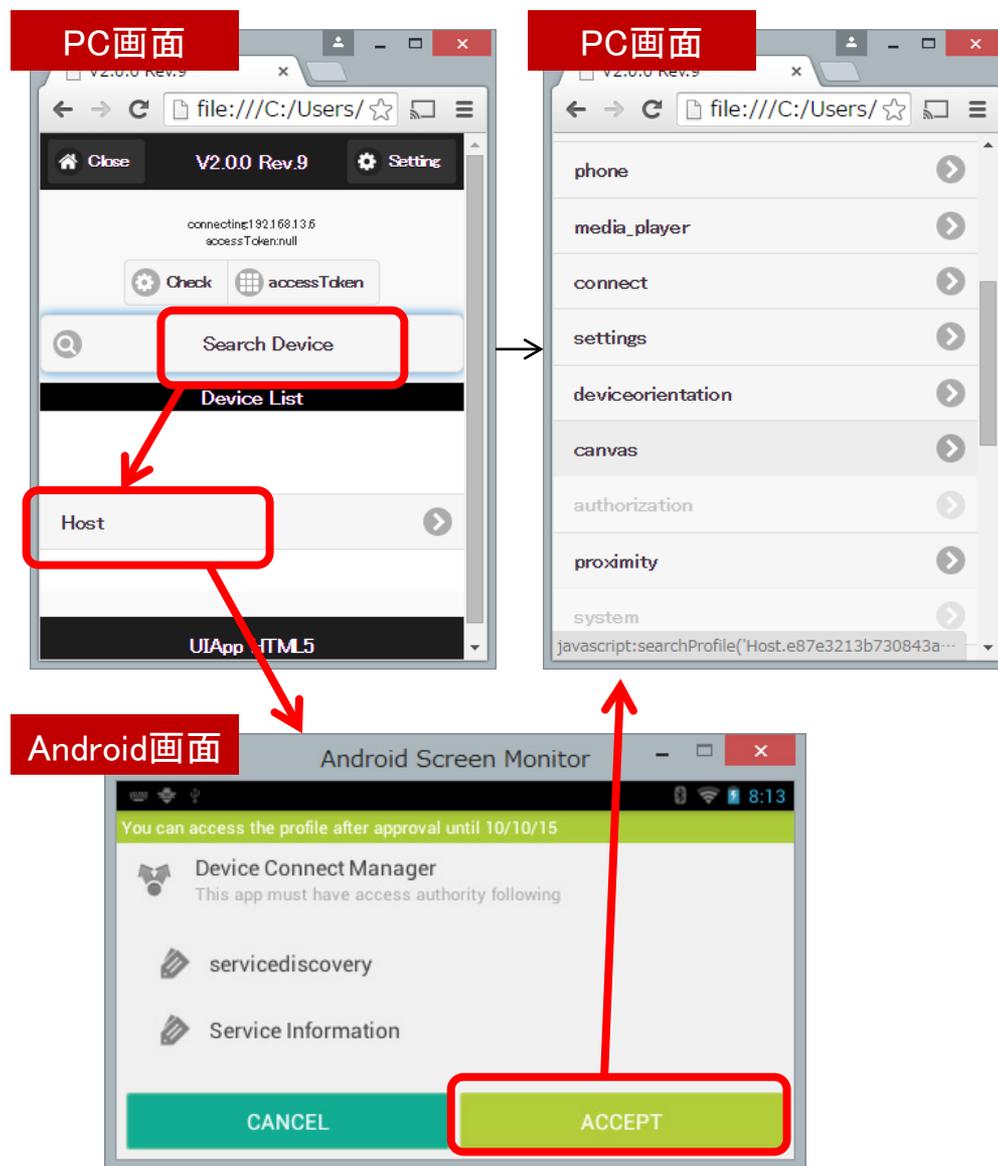


2. URLの最後に"?ip=[Android端末のIPアドレス]#demo"と追加して移動

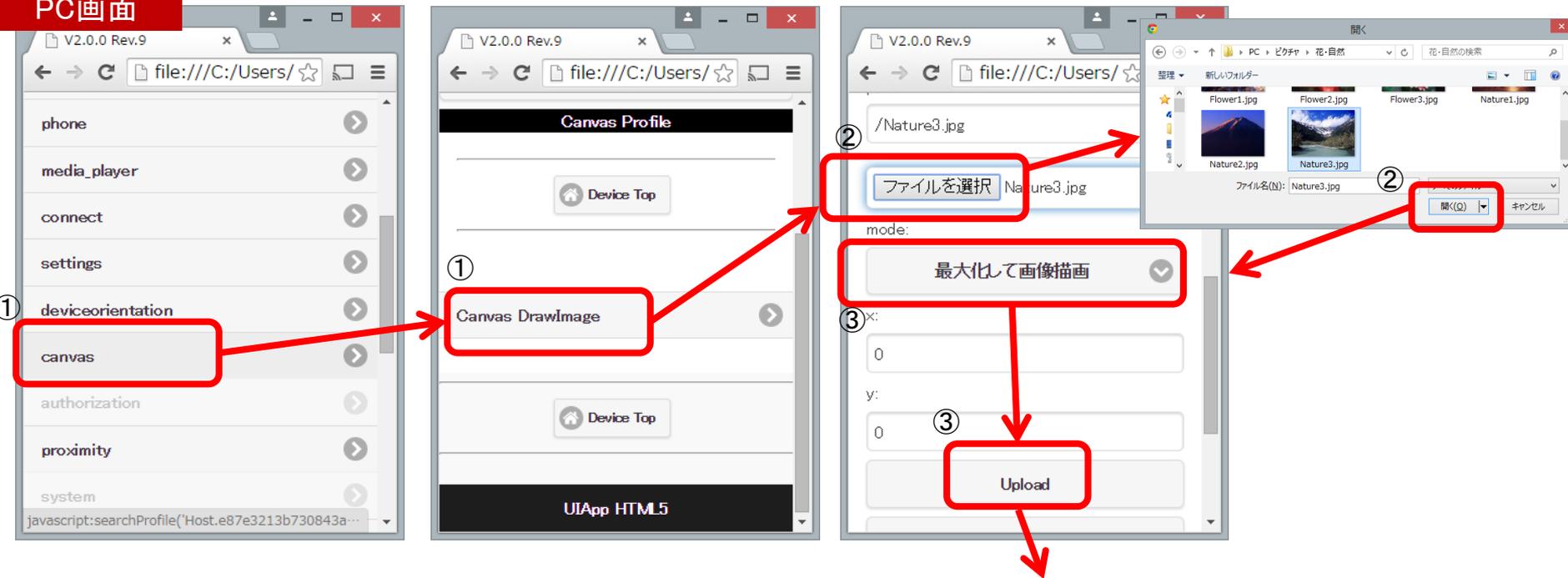


1.デモ画面でSearch Deviceを
選ぶとDevice ListにHostが
表示される。

2.Android端末の画面で
[同意する] を選ぶと
デモ画面に機能のリストが
表示される



PC画面



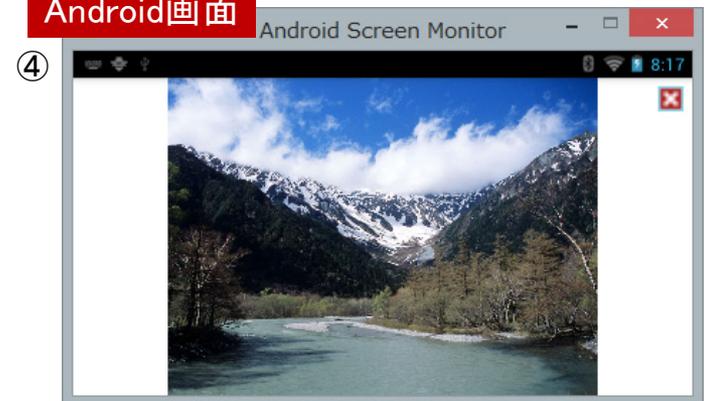
1.機能リストからcanvas、Canvas DrawImageを選択

2.ファイルを選択で画像を選択

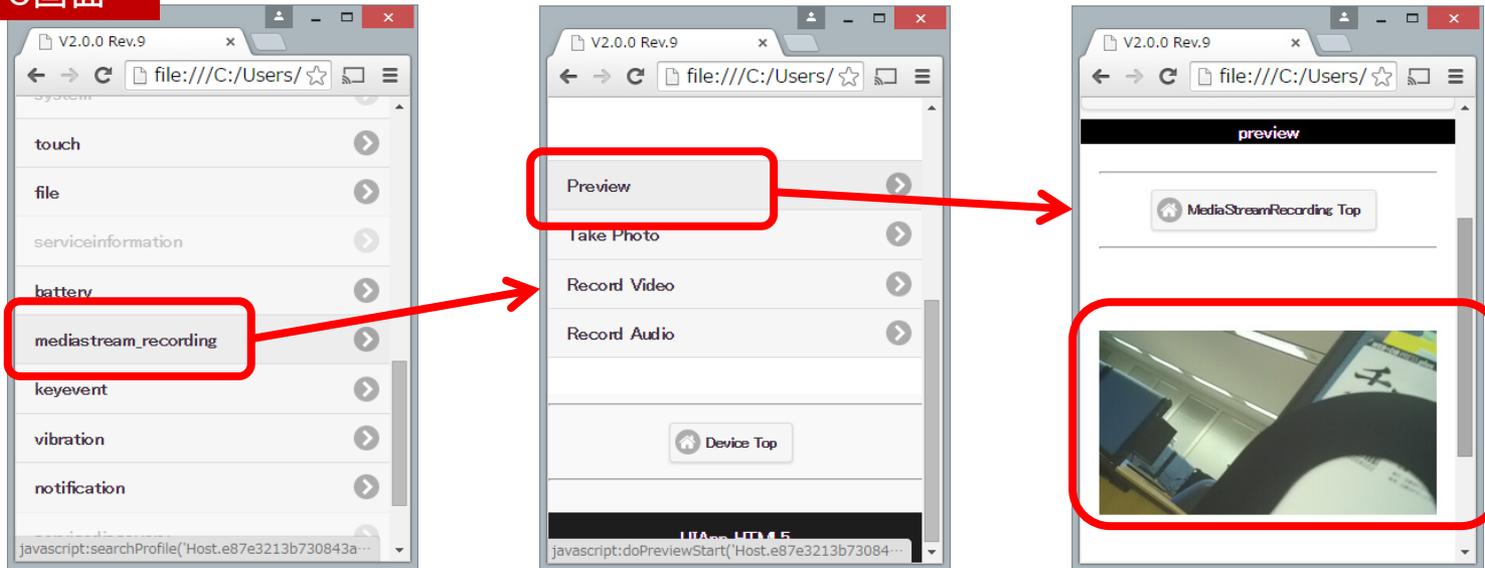
3.mode:「最大化して画像描画」に変更してUploadを選択

4. 画像がAndroidの画面に表示される

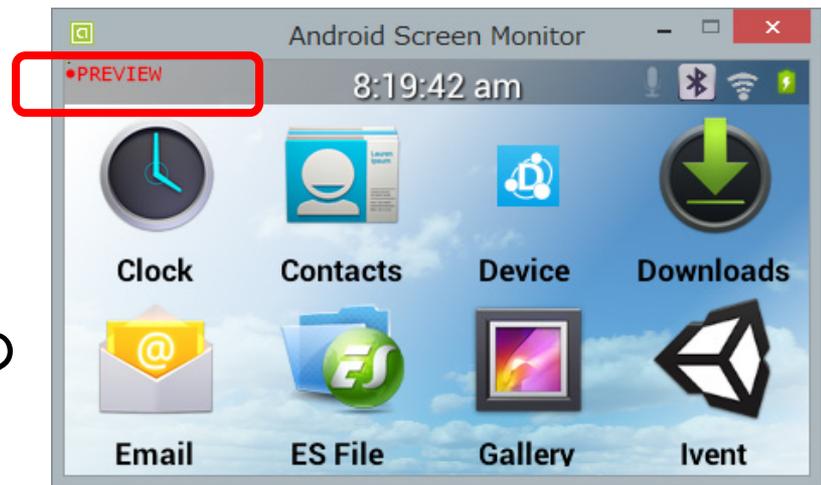
Android画面



PC画面



Android画面



- 1.機能リストからmediastream_recording、Previewを選択
- 2.Webブラウザの画面にAndrpod端末のカメラ映像が表示される(撮影中はAndroidの画面の左上にPREVIEWと表示される)

1. GotAPIとは？
2. あえて未インストールでGotAPI体験(Gclue社デモサイト)
3. GotAPIの色々な機能を試す
(GotAPIのOSS実装: デバイスコネクトWebAPI)
4. curlによるREST API動作確認

4.1 REST APIを試す下準備

WebAPI (REST API) として利用できる = 開発環境にも依存しない

⇒ cURL (コンソールからHTTPリクエストが行えるツール <http://curl.haxx.se/>) により動作を確かめます

※簡略化のために、セキュリティ無効設定を前提として進めます。

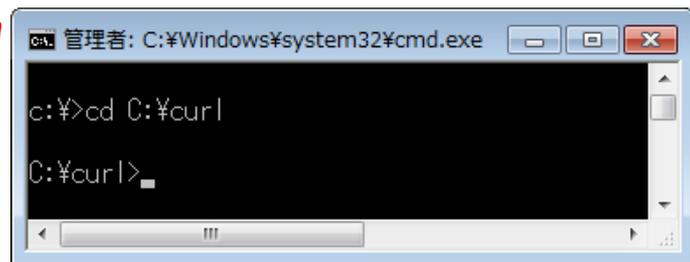
※現状のバージョンでは、セキュリティ動作の都合上、スライドP.18(3.5 STEP:4)の手順で、プラグインの機能確認のダイアログで[同意する]を選んでいる必要があります

準備 (Windowsの場合)

1. プログラムとファイルの検索で
”cmd”と入力してコンソールの起動



2. cURLをダウンロードしてフォルダを作成
“cd [フォルダの場所]”でcurl.exeのパスに移動



4.2 REST APIで利用できる機能の確認

デバイスコネクトWebAPIが使えるかどうかの確認(availability)

リクエスト

```
>curl -X GET http://192.168.13.6:4035/gotapi/availability
```

レスポンス

```
{"product":"Device Connect Manager","result":0,"version":"2.0.0"}
```

利用できるデバイスリストの確認(servicediscovery)

リクエスト

```
>curl -X GET http://192.168.13.6:4035/gotapi/servicediscovery
```

レスポンス

```
{"product":"Device Connect Manager","result":0,
"services":[{"id":"F3:3B:15:7E:29:BB.e9484eb5107adfe1af6a0dc65c03232.localhost.deviceconnect.org",
"config":"","type":"BLE","name":"MIO GLOBAL","online":true},
{"id":"Host.e87e3213b730843a437ff6c676899df0.localhost.deviceconnect.org",
"scopes":["file_descriptor","phone","media_player","connect","settings","deviceorientation","canvas","authorization","proximity","system",
"touch","file","serviceinformation","battery","mediastream_recording","keyevent","vibration","notification","servicediscovery"],
"config":"HostConfig","type":"Wifi","name":"Host","online":true}],
"version":"2.0.0"}
```

} サービスID


 デバイスの名前

```
>curl -X GET -H "Origin: http://localhost" http://192.168.13.6:4035/gotapi/servicediscovery
```


 curlコマンド


 HTTP GET


 Originの指定
(セキュリティ設定の
Originを有効化した場合)


 AndroidのIPアドレス
4035ポートの利用はOMAで規定

※セキュリティ設定の Originを無効化している場合、curlを使わずにブラウザのアドレス欄からでも確認可能

4.3 REST APIで利用できる機能の確認

プラグインリストの確認(system)

リクエスト

```
>curl -X GET http://192.168.13.6:4035/gotapi/system
```

レスポンス

```
{ "product": "Device Connect Manager", "result": 0,
  "supports": ["files", "system", "authorization", "servicediscovery", "availability"], } 本体のAPI※1
  "plugins": [ { "packageName": "org.deviceconnect.android.deviceplugin.heartrate",
                 "id": "e9484eb5107adfef1af6a0dc65c03232.localhost.deviceconnect.org", } プラグインID※2
                 "supports": ["servicediscovery", "serviceinformation", "system", "health"],
                 "name": "HeartRate(BLE) (Device Web API device plug-in)" },
                { "packageName": "org.deviceconnect.android.deviceplugin.host",
                 "id": "e87e3213b730843a437ff6c676899df0.localhost.deviceconnect.org", } プラグインID※2
                 "supports": ["servicediscovery", "serviceinformation", "system", "battery",
                              "notification", "vibration", "settings", "deviceorientation", "file",
                              "file_descriptor", "proximity", "connect", "mediastream_recording",
                              "phone", "media_player", "proximity", "canvas", "keyevent", "touch"
                              ],
                 "name": "Host 2.0.0 "
                },
  ],
  "version": "2.0.0 "
}
```

※1 本体APIの“files”はシステム内部のプラグイン処理で、“authorization”はセキュリティ処理で利用のため省略

※2 ひとつのプラグインに複数のデバイスがあるため、プラグインID≠サービスID、プラグインの名前≠デバイスの名前となっている

4.4 REST APIで利用できる機能の確認

指定したデバイスで使える機能の確認(serviceinformation)

リクエスト

```
>curl -X GET http://192.168.13.6:4035/gotapi/serviceinformation
?serviceId=Host.e87e3213b730843a437ff6c676899df0.localhost.deviceconnect.org
```

レスポンス

```
{"product":"Device Connect Manager","result":0,
"supports":["file_descriptor","phone","media_player","connect","settings","deviceorientation",
"canvas","authorization","proximity","system","touch","file","serviceinformation","battery",
"mediastream_recording","keyevent","vibration","notification","servicediscovery"],
"connect":{"},"version":"2.0.0"}
```

} 利用できる
機能のリスト

プラグインの設定をする(system/device/wakeup)

リクエスト

```
>curl -X PUT http://192.168.13.6:4035/gotapi/system/device/wakeup
?pluginId=e9484eb5107adef1af6a0dc65c03232.localhost.deviceconnect.org
```

レスポンス

```
{"product":"Device Connect Manager","result":0,"version":"2.0.0"}
```



←のようなプラグインの設定画面が開きます

4.5 REST APIとしての動作確認:機能の利用例

バッテリーの残量を知る

リクエスト

```
>curl -X GET http://192.168.13.6:4035/gotapi/battery  
?serviceId=Host.e87e3213b730843a437ff6c676899df0.localhost.deviceconnect.org }サービスID
```

レスポンス

```
{“charging”:true,“product”:“Device Connect Manager”,“result”:0,  
“level”:0.8700000047683716,“version”:“2.0.0”}
```

残り87%

Androidに通知を送る

https://github.com/DeviceConnect/DeviceConnect-JS/wiki/2.13.Notification#notification_post

リクエスト

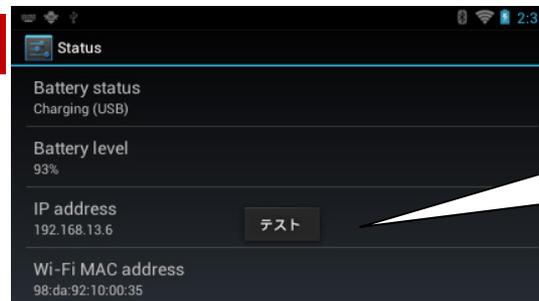
```
>curl -X POST http://192.168.13.6:4035/gotapi/notification/notify  
-F "serviceId=Host.e87e3213b730843a437ff6c676899df0.localhost.deviceconnect.org" }サービスID  
-F "type=3" -F "body=%E3%83%86%E3%82%B9%E3%83%88"
```

URLエンコードされた文字列（ここでは「テスト」を表している）

レスポンス

```
{"product":“Device Connect Manager”,“notificationId”:69699,“result”:0,“version”:“2.0.0”}
```

Android画面

Android端末上に
通知が表示される
(画面はM100の場合)

- Webの標準的な技術のみで、デバイスアクセスのためのWebインターフェースを実現、Webサイトから利用導線も用意できるのでプリインストールも不要
- 仕様をOMAで標準化し、自由なライセンス(MIT)でオープンソース提供、コンテンツ開発やデバイス対応のためのSDKも整備
- シンプルで手軽なHTTPでのアクセスから、セキュリティ対策を施したWebSocketによる効率的なアクセスまで、目的に応じた機能アクセスが可能

<http://device-webapi.org/>

まずはお気軽にお試しください