

Monaca

UIフレームワーク for Cordova

<https://ja.monaca.io/>

ハイブリッドアプリの背景

問題

Android は Java、iOSは Objective-C/Swiftで開発する必要がある。



Go で両方かけるらしいが。。。

PhoneGap社

一つの言語、Javascript/HTML5で書けないか？
Webブラウザでも動くし。



解決!

Javascript/HTML5をパッケージして、ネイティブアプリを作れば良い！

極端に言えば、Javascript/HTML5 が、もれなく付いてくるWebブラウザアプリが作れる。
(インターネットからHTMLをダウンロードする必要なし。かつ、カメラとかにアクセスもできる!)

Monacaとは？

PhoneGap社 -> オープンソース化



Apache が受け入れ、Cordovaに名称変更



Android/iOSとのプラグイン(Javascript API)、カメラ、GPS、プッシュ通知機能など提供



けど、UIは、自分たちで頑張ってね



Monaca社、Ionic社

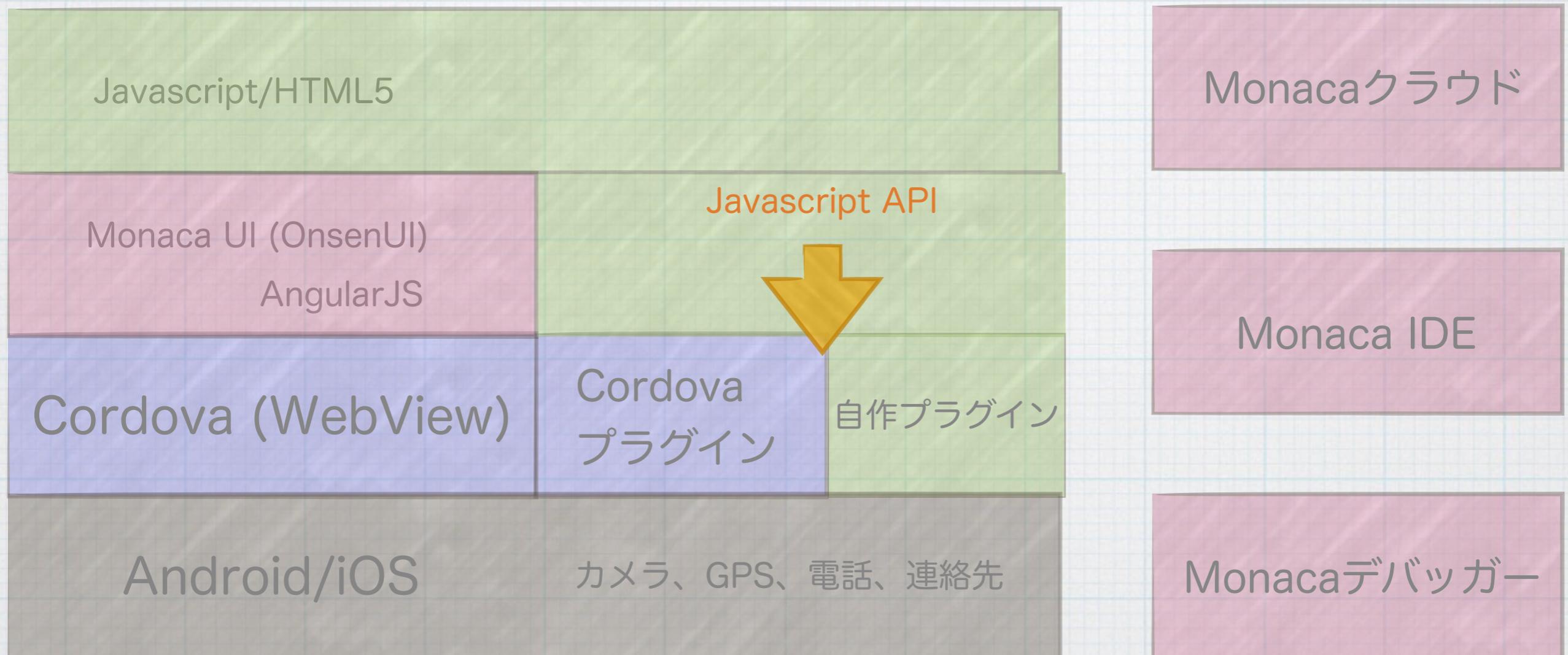
では、うちから、UIコンポーネント、開発ツール、クラウドサーバ(MBaaS)を提供しましょう。

Monacaの位置付け

基本的には、HTML、JSの処理は、Cordova(WebView)で処理される。

カメラ、Push通知等のOSネイティブの機能は、Cordovaプラグイン(Javascript API)が提供される。

自作プラグインを、Objective-C/Swiftで書くことも可能。



Monacaクラウドを使って見る

アカウントを登録する

<https://monaca.mobi/ja/register/start>

The screenshot shows the registration start page for Monaca Cloud. The main heading is "さっそくMonacaを使ってみよう!" (Let's use Monaca!). Below it is the subtitle "クロスプラットフォーム時代のアプリ開発" (Cross-platform application development). On the left, there is a diagram illustrating the platform's features: a central cloud icon contains a laptop, an HTML5 logo, and a smartphone icon, with "Complete development platform that runs on your browser" written above it. Four arrows point from this central icon to four mobile devices below: "Cross platform support (iOS, Android, Windows 8)" points to an iPhone; "PhoneGap support" points to an Android phone; "Integrated Backend API" points to a Windows Phone; and "..." points to a tablet. To the right is a large input form for account creation, containing fields for email, password, and confirmation. A note at the bottom states "Monacaの利用規約に同意の上、アカウントの作成をお願いします。" (Please agree to the Monaca Terms of Service to create an account.) An orange button at the bottom right says "いますぐ登録 (無料)" (Register now (free)).

さっそくMonacaを使ってみよう!

クロスプラットフォーム時代のアプリ開発

Complete development platform
that runs on your browser

PhoneGap support

Cross platform support
(iOS, Android, Windows 8)

Integrated Backend API

アカウント作成

Monacaの利用規約に同意の上、アカウントの作成をお願いします。

いますぐ登録 (無料)

ダッシュボード

ホームページ。左側の「Hello Worldアプリ」の「開く」をクリックしてみる。

The screenshot shows the Monaca User Dashboard interface. On the left, there's a sidebar titled 'すべてのプロジェクト' (All Projects) with tabs for 'オンライン' (Online) and 'アーカイブ' (Archive). A large yellow arrow points to the '開く' (Open) button next to the 'Hello World' project card. The card itself has a small icon, the title 'Hello Worldアプリ', a brief description, and a 'NEW' badge. To the right of the sidebar is a main content area featuring a banner for 'Monaca × mobile backend' and a section for 'Monaca ブログ'. The blog section lists three articles: 'Amazon Mobile Analyticsを使ってアプリ利用状況を解析しよう' (Published 2016-01-13), 'jQueryから離れるためのJavaScript代替ライブラリまとめ' (Published 2015-12-28), and 'MonacaアプリでReactを使ってみよう' (Published 2015-12-24). At the bottom right, there's a 'お問い合わせ・ヘルプ' (Contact Support) button.

Hello World

開発ツールがWebブラウザで動く。「Start Demo」をクリックしてみる。

The screenshot shows the Monaca IDE interface. On the left, there's a sidebar with developer tools and a message about the local environment. The main workspace shows the 'Hello World' project structure and the content of 'index.html'. The code includes standard HTML tags like DOCTYPE, head, and body, along with a script section containing PhoneGap event handling logic. In the preview pane on the right, the text 'Welcome to Monaca!' is visible, followed by a blue button labeled 'Start Demo'. A yellow arrow points to this button, indicating where to click.

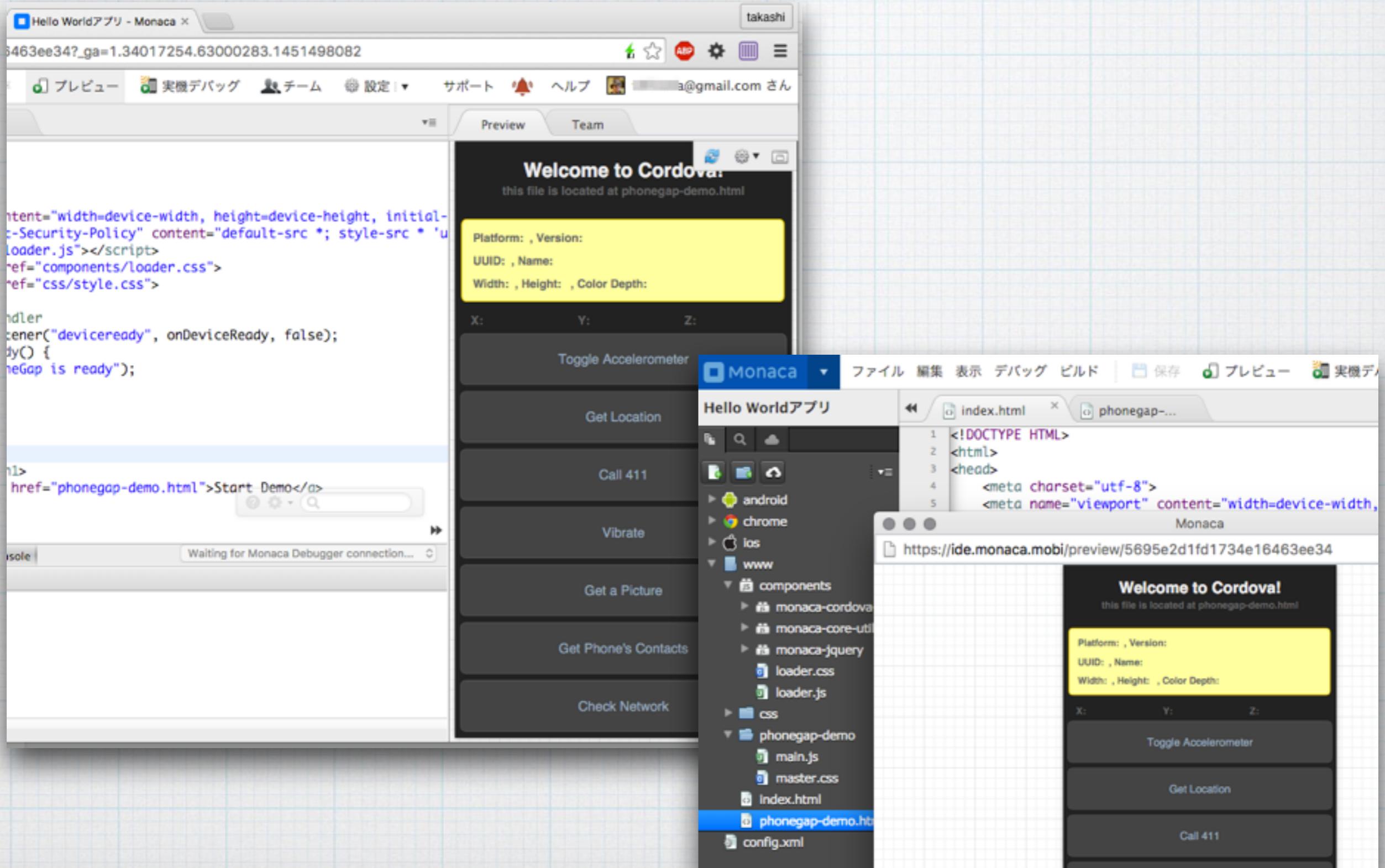
```
<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, height=device-height, initial-scale=1.0, maximum-scale=1.0">
    <meta http-equiv="Content-Security-Policy" content="default-src *; style-src * 'unsafe-style-src'">
    <script src="components/loader.js"></script>
    <link rel="stylesheet" href="components/loader.css">
    <link rel="stylesheet" href="css/style.css">
    <script>
        // PhoneGap event handler
        document.addEventListener("deviceready", onDeviceReady, false);
        function onDeviceReady() {
            console.log("PhoneGap is ready");
        }
    </script>
</head>
<body>

    <h1>Welcome to Monaca!</h1>
    <a class="button--large" href="phonegap-demo.html">Start Demo</a>

</body>
</html>
```

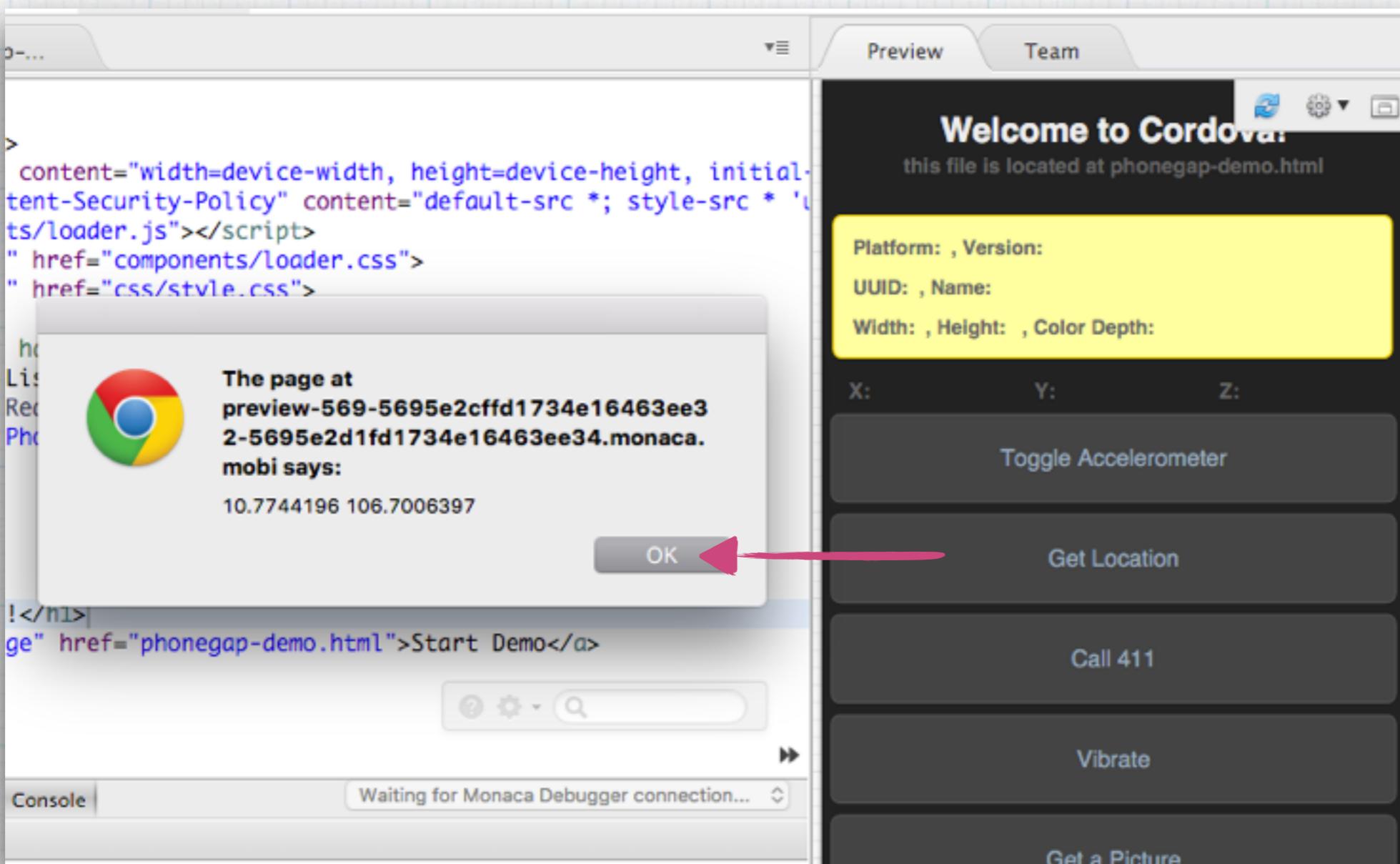
Hello World

エミュレータがWebブラウザで動く。ディタッチも出来る。



Get Location

「Get Location」をクリックします。アラートで、緯度、経度が表示される。



ブラウザ上なので、カメラ、バイブ、アクセロメータは無理か。。。 Facetimeは起動する。

ソースを書き換えてみる

The screenshot shows the Monaca IDE interface. On the left, a file tree displays the project structure for a "Hello World アプリ". The "phonegap-demo.html" file is selected. The main workspace shows the HTML code for this file. A pink arrow points from the file tree to the selected file. Another pink arrow points from the code editor to the bottom of the page, where two numbered steps are described.

① 左のファイル一覧から、phonegap-demo.htmlを選択。

② ソースに1行追加する。

③ 保存を押す。

④ リフレッシュする。

The code editor shows the following HTML:

```
<html>
  <head>
    </script>
    <link rel="stylesheet" href="phonegap-demo/master.css" type="text/css" media="scr
    <script type="text/javascript" charset="utf-8" src="phonegap-demo/main.js"></scri
  </head>
  <body onload="init();" id="stage" class="theme">

    <h1>Welcome to Cordova!</h1>
    <h2>this file is located at phonegap-demo.html</h2>
    <div id="info">
      <h4>Platform: <span id="platform"> &ampnbsp</span>, Version: <span id="versio
      <h4>UUID: <span id="uuid"> &ampnbsp</span>, Name: <span id="name">&ampnbsp</sp
      <h4>Width: <span id="width"> &ampnbsp</span>, Height: <span id="height">&nbs
      <span>, Color Depth: <span id="colorDepth"></span></h4>
    </div>
    <dl id="accel-data">
      <dt>X:</dt><dd id="x">&ampnbsp</dd>
      <dt>Y:</dt><dd id="y">&ampnbsp</dd>
      <dt>Z:</dt><dd id="z">&ampnbsp</dd>
    </dl>
    <a href="#" class="btn large" onclick="toggleAccel();">1行追加してみる</a>
    <a href="#" class="btn large" onclick="toggleAccel();">Toggle Accelerometer</a>
    <a href="#" class="btn large" onclick="getLocation();">Get Location</a>
    <a href="tel:411" class="btn large">Call 411</a>
    <a href="#" class="btn large" onclick="vibrate();">Vibrate</a>
  </body>
</html>
```

The preview panel on the right shows the updated content: "Welcome to Cordova!" and "1行追加してみる".

① 左のファイル一覧から、phonegap-demo.htmlを選択。 ② ソースに1行追加する。

③ 保存を押す。

④ リフレッシュする。

実機デバックしてみる。

iOSで。

デバッガーの説明とインストール

デバッグ用アプリ「Monacaデバッガー」



Monacaでは、「Monacaデバッガー」というスマートフォン端末またはパソコン上でアプリのデバッグを簡単に行うことができるアプリケーションを用意しています。Monacaデバッガーの詳細についてはドキュメントを参照してください。

簡単な動作確認は「プレビュー」で

ネイティブ環境ではなく、Monaca上で簡単な動作確認を行いたい場合は「プレビュー」機能をお使いください。プレビューの詳細についてはドキュメントを参照してください。

Monacaデバッガのインストール

デバッグする環境に合わせてMonacaデバッガーをインストールできます。



Android向け
デバッガー
インストール



iOS向け
デバッガー
インストール



Chrome向け
デバッガー
インストール

ビルドしてインストールで。

iOS向けデバッガーのインストール

iOS向けMonacaデバッガー

デバッガーを利用するには、App Storeで提供しているデバッガーをインストールする方法、もしくはMonaca上でビルドしたものをインストールする方法の2通りがあります。

App Storeからインストール

バージョン 5.0.0

ストアからインストールしたデバッガー上で簡単にアプリの動作確認ができます。デバッガーにはMonacaが標準で提供している基本Cordovaプラグインをすべて搭載しています。詳細は[こちら](#)。

ビルドしてインストール

バージョン 5.0.0

プラグイン管理画面で有効にしたCordovaプラグインを組み込んだデバッガーをビルドできます。またUSBデバッグにも対応し、Monaca CLIやLocalkitを用いたJavaScriptの高度なデバッグが可能になります。詳細は[こちら](#)。

お約束の証明書の設定が必要。。。。



実機にインストールするため、証明書の面倒な設定が必要。

http://docs.monaca.mobi/cur/ja/quick_start/localkit/building_app/

証明書の作成中。。。.

iOSアプリ設定

Project

- Cordovaプラグイン
- JS/CSSコンポーネント
- 外部サービス連携

Build

- Androidアプリ
- Androidビルド
- iOSアプリ**
- iOSビルド
- Windows
- Chrome Apps

iOSアプリケーションをビルトするために必要な項目を設定します。このページの設

アプリケーション設定

アプリケーション名:

App ID: Androidのパッケージ名も変更されます。

バージョン:

バンドルバージョン:

iOSビルド設定

iOSアプリケーションをビルトするために必要な項目を設定します。このページの設定はユ

Monaca IDE

iOS Dev Centerに登録したユーザー名とメールアドレスを入力してください。

ユーザー名
(例: Monaca Tarou):

メールアドレス
(例: example@monaca.mobi):

Apple Developer Center にて作業

Apple Developer Center のサイトへ行き、

証明書(cer)の作成、App IDの登録、プロビジョニングファイルの作成、をする。

詳細はこちら。

http://docs.monaca.mobi/cur/ja/manual/build/ios/build_ios/

プロビジョニングファイルをアップロード

作成したプロビジョニングファイルをアップロードする。

iOS向けデバッガーのインストール

ビルド インストール デバッガー接続 実機デバッグ

 プロビジョニングファイルを登録してください

iOS Provisioning Portalを使用して、インストール先の端末が含まれたプロビジョニングファイルを作成しアップロードしてください。プロビジョニングでは以下の設定が必要です。

iOS Provisioning Portalでの設定内容

| | |
|------------------------|-------------------------------------|
| プロビジョニングの種類 | Development Distribution |
| プロファイル名 (Profile Name) | 任意のアルファベット (例: Monaca Profile) |
| 証明書 (Certificates) | Monacaに登録した証明書 |
| App ID | Explicit App ID または Wildcard App ID |
| 端末 (Devices) | インストールする端末が含まれていること |

プロビジョニングファイル：
C:\fakepath\Provision_for_APNS.mobileprovision 参照...

戻る 次へ

ビルドが開始する。

iOS向けデバッガーのインストール

ビルド インストール デバッガー接続 実機デバッグ

ビルド中です...

現在アプリをビルドしています。しばらくお待ちください。
この作業には数分かかることがあります。

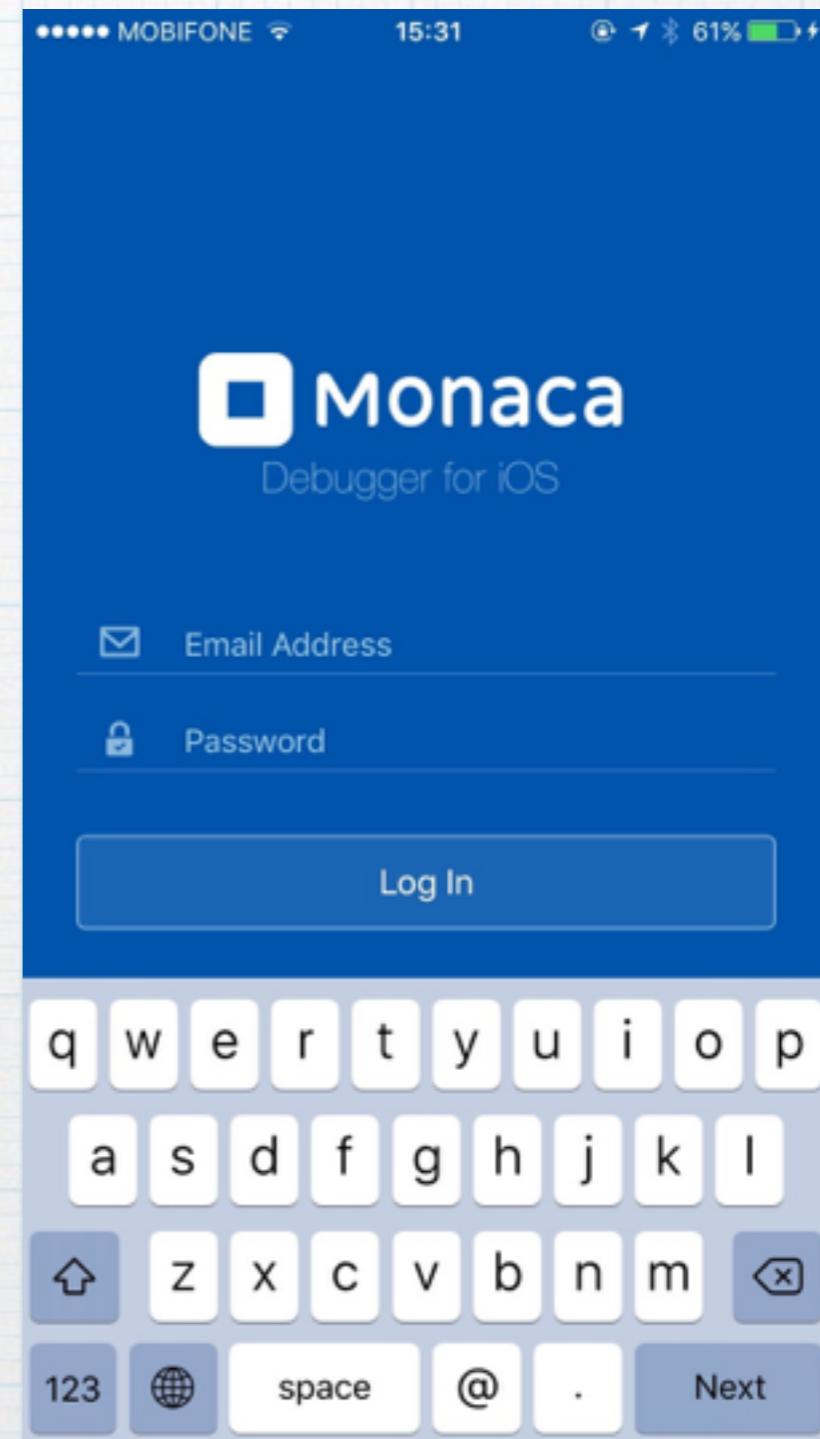
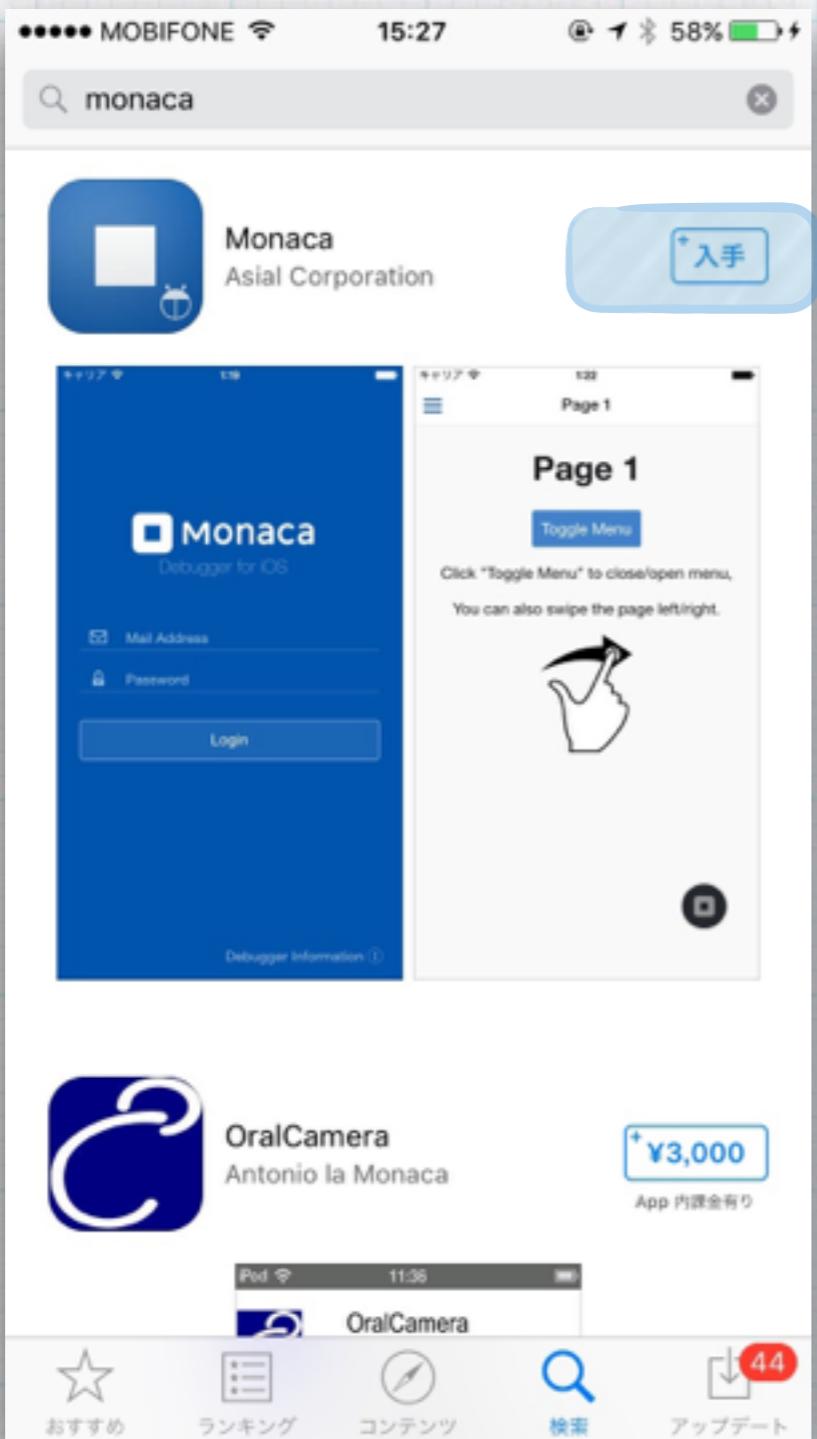
ビルドが完了したら、自動的に次の画面に進みます。

キャンセル

デバッガーアプリをインストール

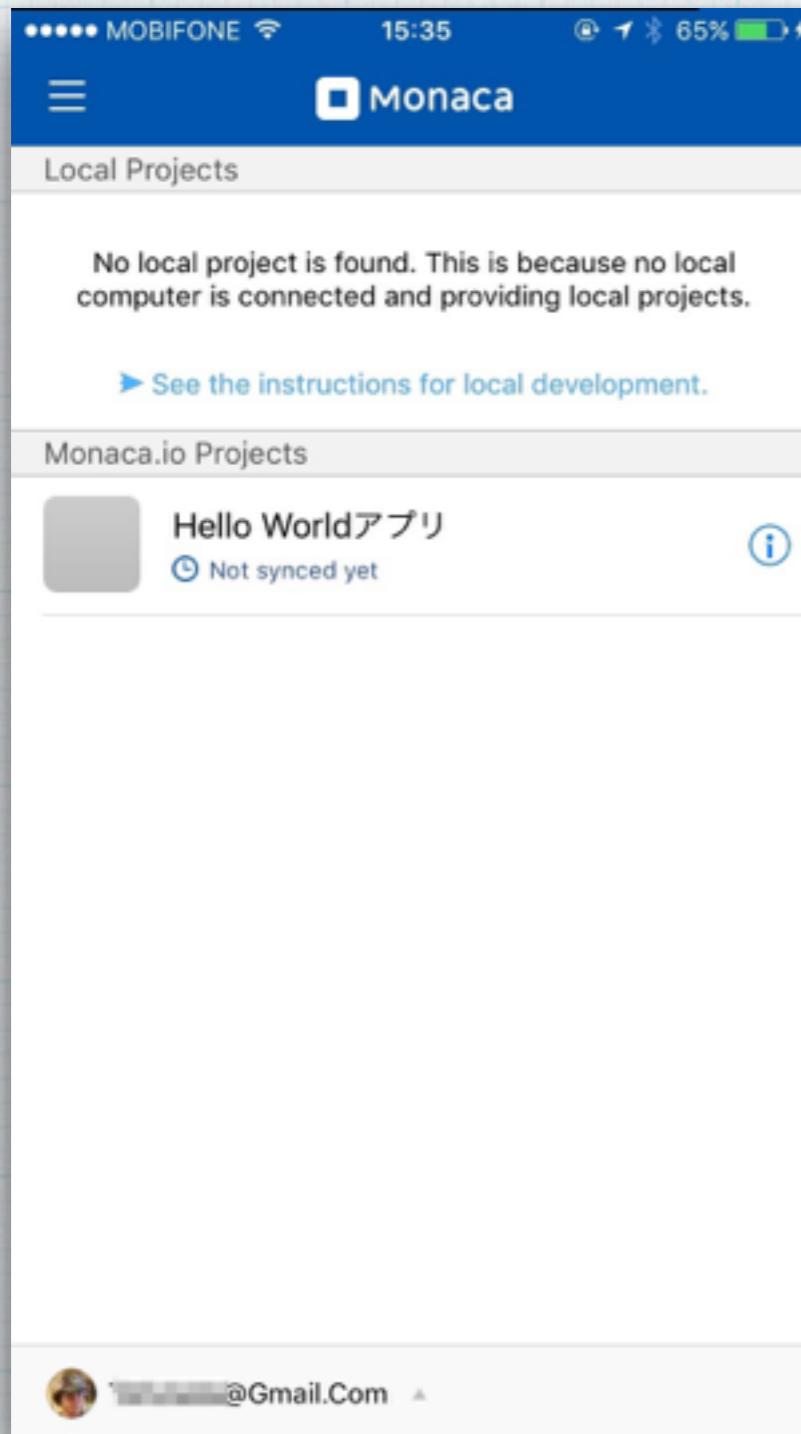
App StoreからiPhoneにインストール。

ログインしましょう。



デバッガーアプリにて。

ログインすると、先ほどビルドしたアプリが表示されるはず。



再度、ブラウザから、「実機デバッグ」をクリック。



アプリ起動！

カメラ、連絡先、GPS、電話にアクセスできた！

