

PhoneGap でつくる WebView アプリの作り方

PhoneGap は WebView を簡単に扱えるフレームワークと呼ばれるもの。これを導入すると iPhone や Android にサイトページを内蔵して閲覧できるアプリをつくることができます。アプリ本体に組み込めば、電車の中などのオフラインでアプリを起動して見る事が可能。逆に外部サイトにリンクを貼り閲覧するようにすることも可能（この場合オンライン必須）。また、JavaScript を組み込むことも可能なので jQueryMobile を内蔵させることも可能。

PhoneGap のバージョンが違くと動かない場合があるので注意。2.0.0 を対象に進める。

<http://phonegap.com/download>



- ・他に必要なもの (Android)

- JDK1.6 か 1.7

- Eclipse

- Android SDK r18

- (最新版は r21 だが、プロジェクト構成などが違うので少し古い r18 をつかうとよし)

- ・他に必要なもの (iOS)

- Xcode

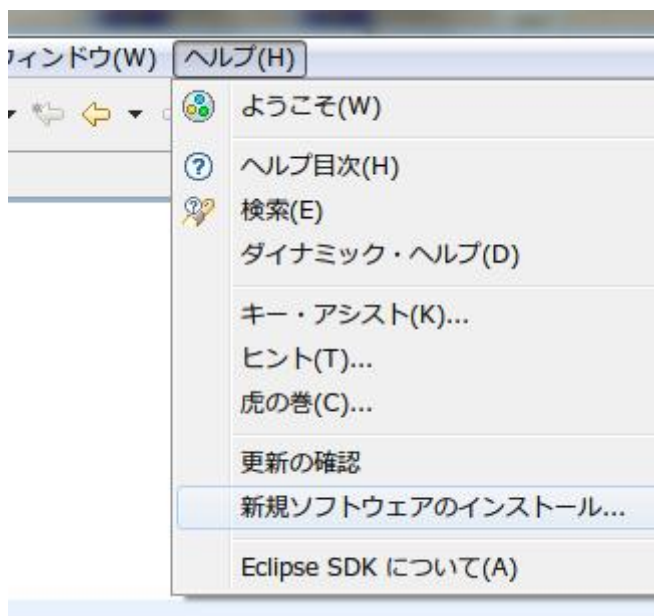
- ・ PhoneGap ヘルプ

- http://docs.phonegap.com/en/2.0.0/guide_getting-started_index.md.html

1、インストール

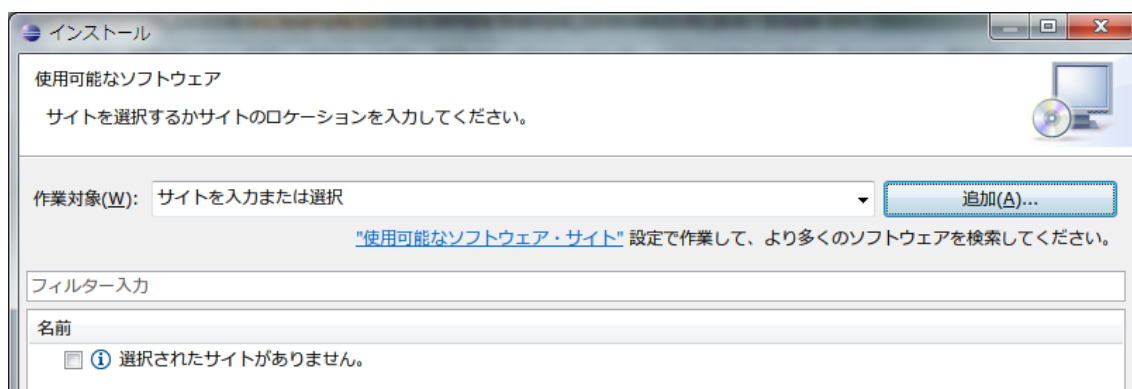
JDK を最初にダウンロードしてインストールしておく。

<http://java.sun.com/javase/ja/6/download.html>



eclipse はホームフォルダに解凍しておく (Vista 以降の OS では書き込みの権限などがうるさいのでホームフォルダにアプリを展開すると問題がすくないため)。また、解凍しただけで eclipse のインストールは完了する。

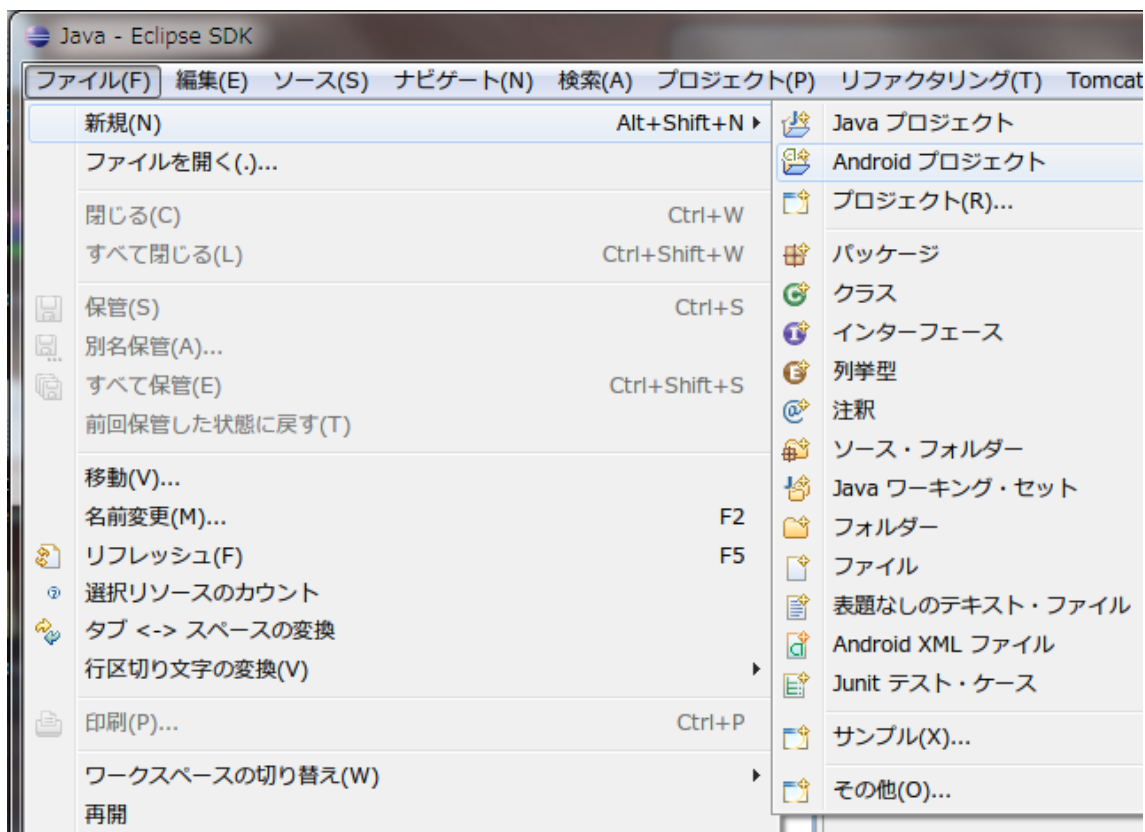
次に Android SDK をインストールするには eclipse を実行。eclipse のヘルプから新規ソフトウェアのインストールを選択。

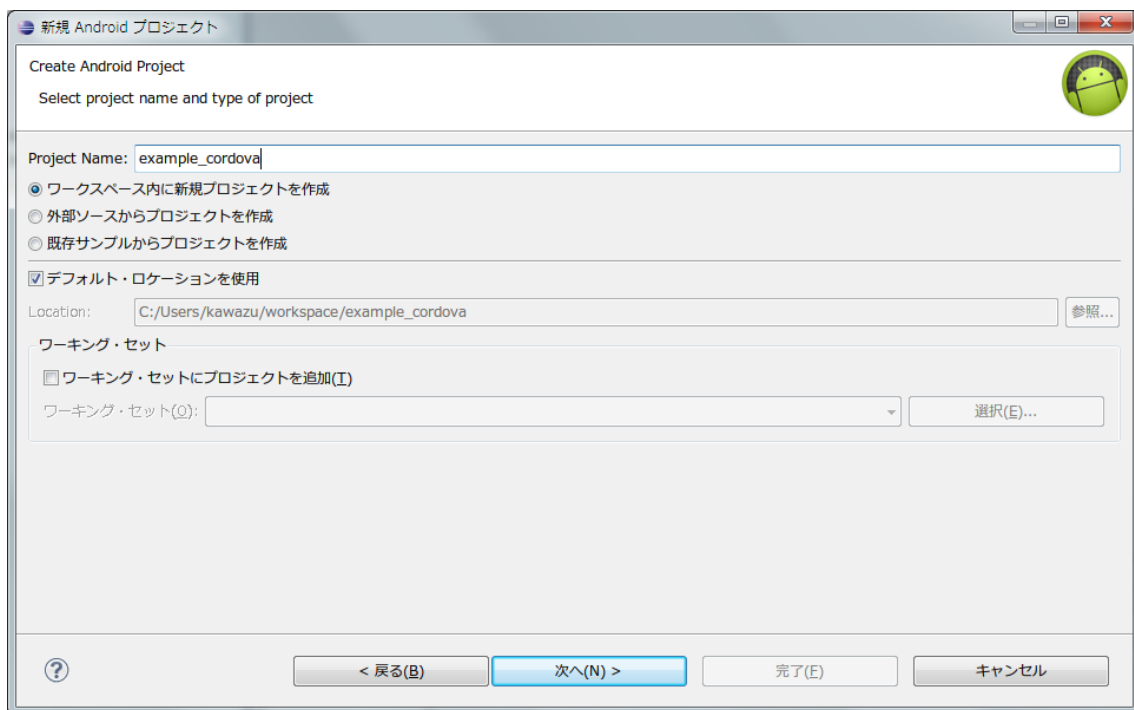


作業対象に <https://dl-ssl.google.com/android/eclipse/> を入力すると最新版がインストールされます。Android SDK Manager を起動し 2.3.3 用の SDK とエミュレータをダウンロードしてください。

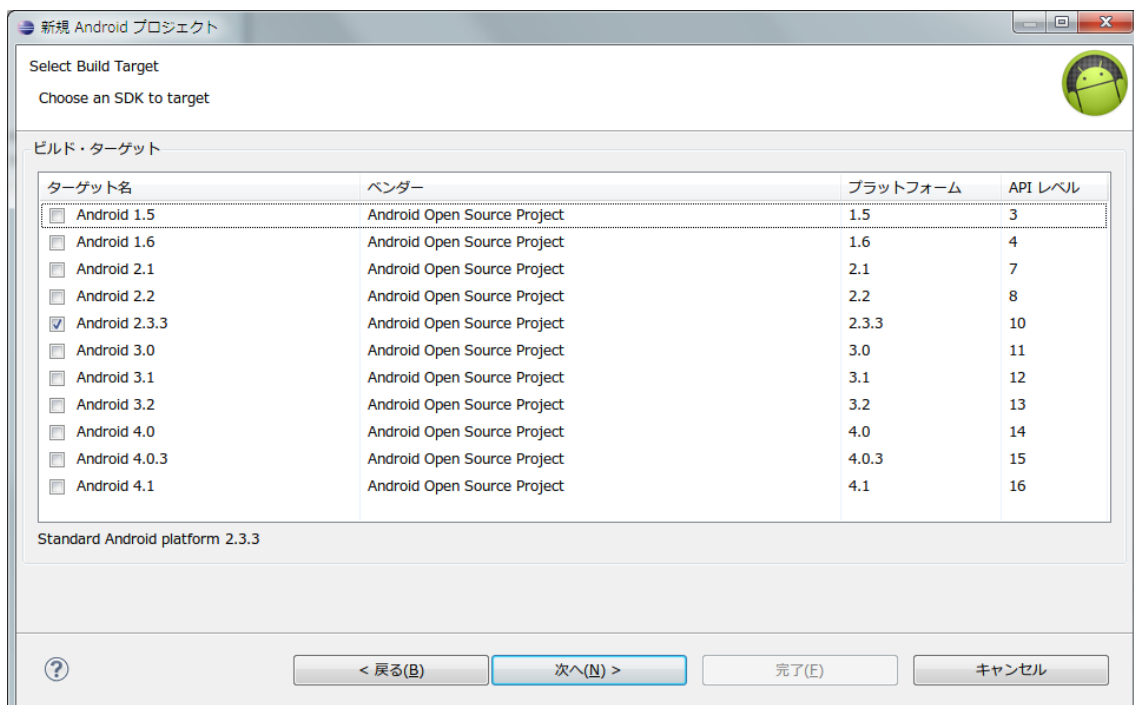
2、 Android のプロジェクトをつくる。

eclipse を起動して新規作成。アプリのアップデートする場合はしなくてよし。当然、新規にアプリをつくる時だけ。

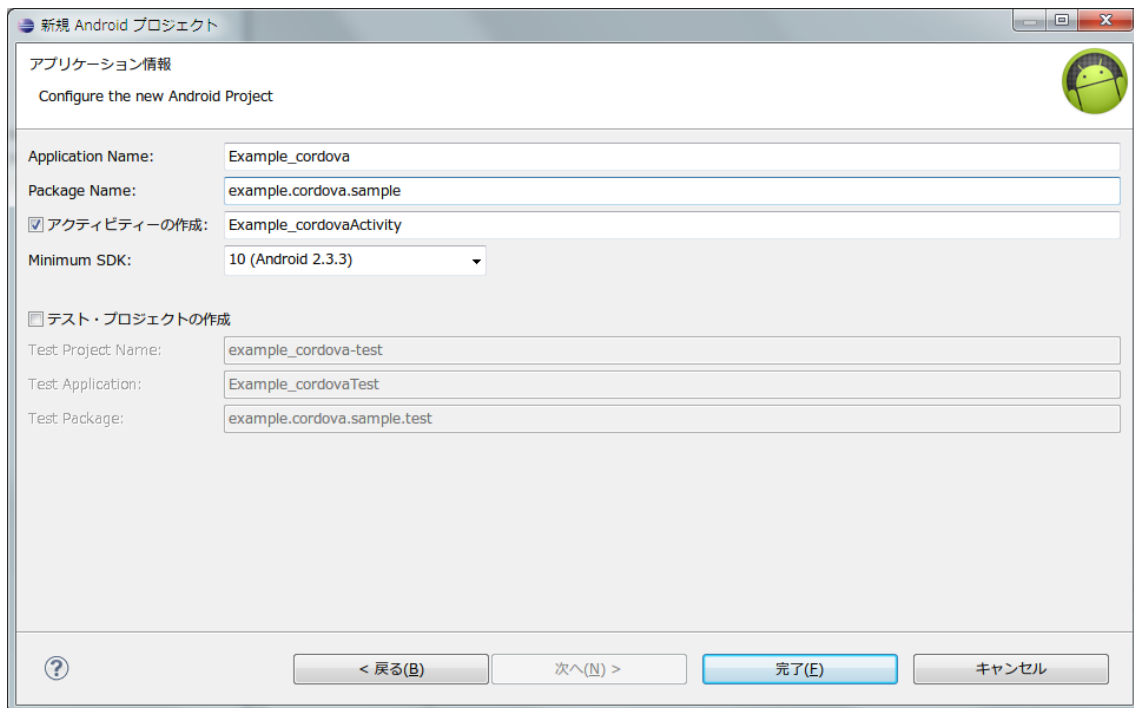




ワークスペースに新規プロジェクトを作成。プロジェクト名がアプリを入れるフォルダ名になる。



ビルド・ターゲットは Android 2.3.3 にする。それ以下だと動かないので注意。



パッケージ名は他のアプリとかぶらないユニークな名前にすること。

端末にインストールされているアプリはパッケージネームをユニークにして管理しているのでパッケージ名がかぶるアプリにするとすでにある同名のアプリは上書きインストールされて消えてしまいます。

以上で空プロジェクトの作成が終了。

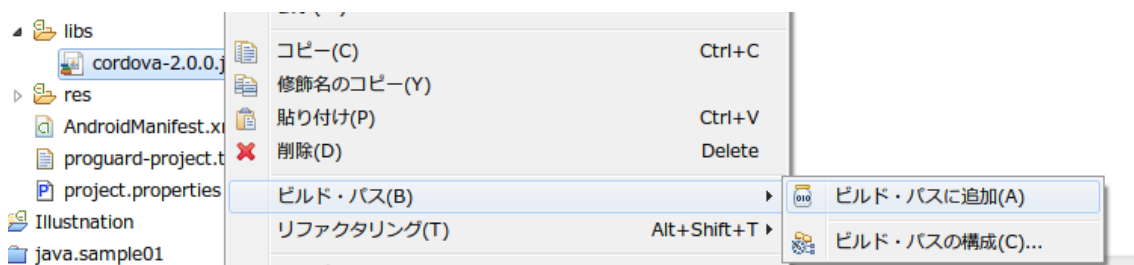
PhoneGap はこれにいろいろなファイルを追加したり既存のファイルを修正します。

3、 PhoneGap を設置。

ダウンロードした PhoneGap を解凍。

¥phonegap-phonegap-2dbbdab¥lib¥android¥example を見る。assets, res, libs の3フォルダをコピーする。さっき作った空のプロジェクトがホームフォルダの workspace のプロジェクト名フォルダにあるので、エクスプローラで開いて、上書き貼り付けする。Android SDK r18 だと上書き貼り付けだけで設置は完了ですが、r20 からだとファイル構成が違うようなので差分をとってマージしてください。

次に eclipse を立ち上げるか、すでに起動していたら、パッケージエクスプローラからプロジェクトをクリックして F5 して更新する。ファイルが更新されるので、libs にある cordova-2.0.0.jar を右クリック、ビルド・パス、ビルド・パスに追加を選択する。



パッケージエクスプローラの参照パスに cordova-2.0.0.jar が追加されているのを確認してください。

次にソースフォルダの java を編集します。

```
src
├── example.cordova.sample
│   └── Example_cordovaActivity.java
```

以下に書き換えます。

```
package example.cordova.sample;

import android.app.Activity;
import android.os.Bundle;
import org.apache.cordova.*;

public class Example_cordovaActivity extends DroidGap
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        super.loadUrl("file:///android_asset/www/index.html");
    }
}
```

`superloadUrl` を外部に飛ばすように書くと外のサイトを読む設定になります。
今回は内蔵なので内側をしめています。
パッケージ名とパブリッククラス名はオリジナルを設定してください。

次にルートにある、`AndroidManifest.xml` を編集します。

以下のコードを参考に書き換えます。

```
<manifest
xmlns:android=http://schemas.android.com/apk/res/android
  android:windowSoftInputMode="adjustPan"
  package=" example.cordova.sample "
  android:versionName="1.1" android:versionCode="5">
  <supports-screens
    android:largeScreens="true"
    android:normalScreens="true"
    android:smallScreens="true"
    android:xlargeScreens="true"
    android:resizeable="true"
    android:anyDensity="true"
  />

  <uses-permission android:name="android.permission.CAMERA" />
  <uses-permission android:name="android.permission.VIBRATE" />
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission android:name="android.permission.RECEIVE_SMS" />
  <uses-permission android:name="android.permission.RECORD_AUDIO" />
  <uses-permission android:name="android.permission.RECORD_VIDEO"/>
  <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
  <uses-permission android:name="android.permission.READ_CONTACTS" />
  <uses-permission android:name="android.permission.WRITE_CONTACTS" />
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
  <uses-permission android:name="android.permission.GET_ACCOUNTS" />
  <uses-permission android:name="android.permission.BROADCAST_STICKY" />
```



```
<application android:icon="@drawable/icon" android:label="@string/app_name"
    android:debuggable="true">
    <activity android:name="cordovaExample" android:label="@string/app_name"
        android:configChanges="orientation | keyboardHidden">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

<uses-sdk android:minSdkVersion="5" />
</manifest>
```

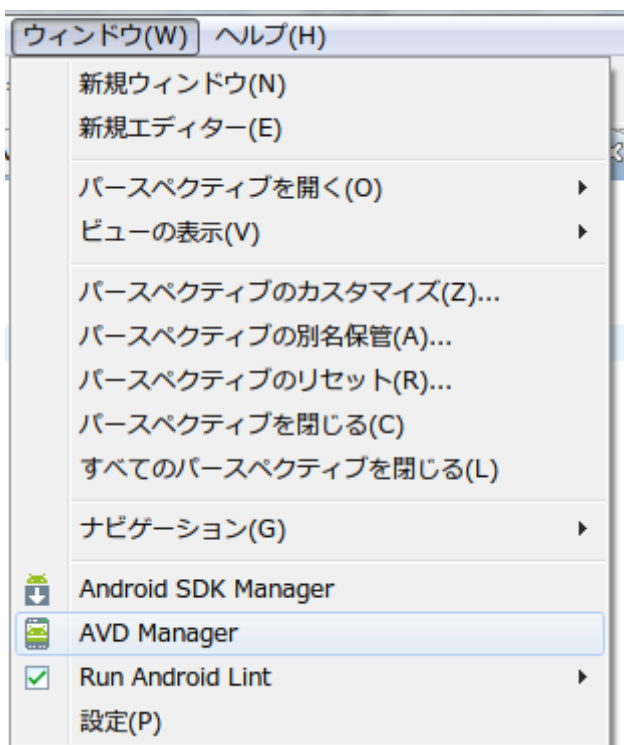
パッケージ名とアプリ名はオリジナルを使用。あとは部分的に書き換えや追記します。次に `AndroidManifest.xml` に `android:versionName="1.1" android:versionCode="5"` と書かれています。これがアプリのバージョンを管理する番号です。アプリをアップグレードするときはこの値をインクリメントします。 `versionName` がユーザーから見えるもので、 `versionCode` が内部で識別するナンバーです。Android システム自体はこの値を使用しませんが、ここをいじらないとアプリのバージョンがわからなくなってしまいます。

次に `android:icon="@drawable/icon" android:label="@string/app_name"` と書かれています。これがアイコンと表示されるアプリ名を参照しているパスです。アイコンは PNG 形式で、一つあれば大丈夫です。Android が端末の画面サイズに合わせて引き伸ばしたり縮小したり実行した時に自動でやってくれます。この場合は、 `/res/drawable/icon/icon.png` を置き換えます。デフォルトだと 96 ピクセルの正方形サイズになってます。アイコンを変えるときはこれを置き換えます。次にアプリ名ですが、 `/layout/values/strings.xml` の値を書き換えます。

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">cordovaExample</string>
</resources>
```

string タグに囲まれたところを書き換えます。

4、 エミュレータでテスト



eclipse のウインドウ、AVDManager を開きます。

新規で以下のように入力して ROM 設定をつくります。

最初に ROM 設定をつくって実行するときは起動に 10 分ぐらいかかります。

ROM データは Android SDK Manager でダウンロードできます。

Create new Android Virtual Device (AVD) ✕

名前:

ターゲット:

CPU/ABI:

SD Card:

サイズ:

ファイル:

Snapshot:

使用可能

Skin:

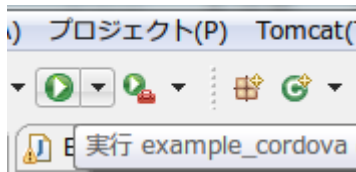
ビルトイン:

解決: x

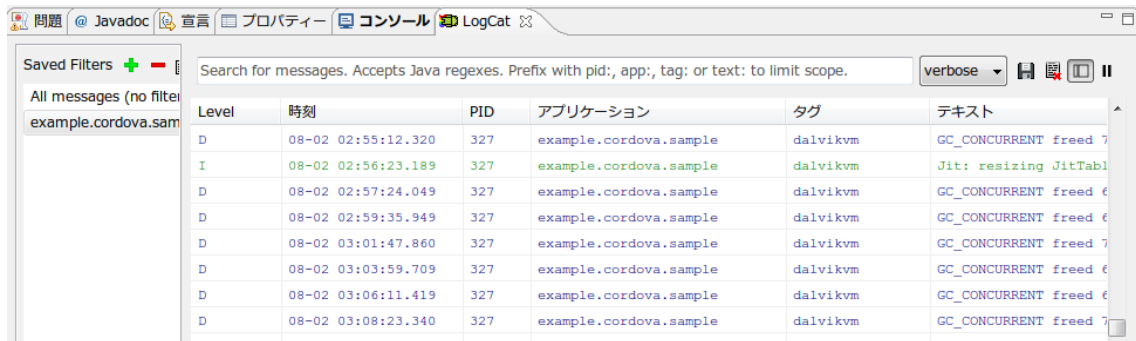
ハードウェア:

プロパティ	値	
Abstracted LCD density	240	
Max VM application h...	24	
Device ram size	256	

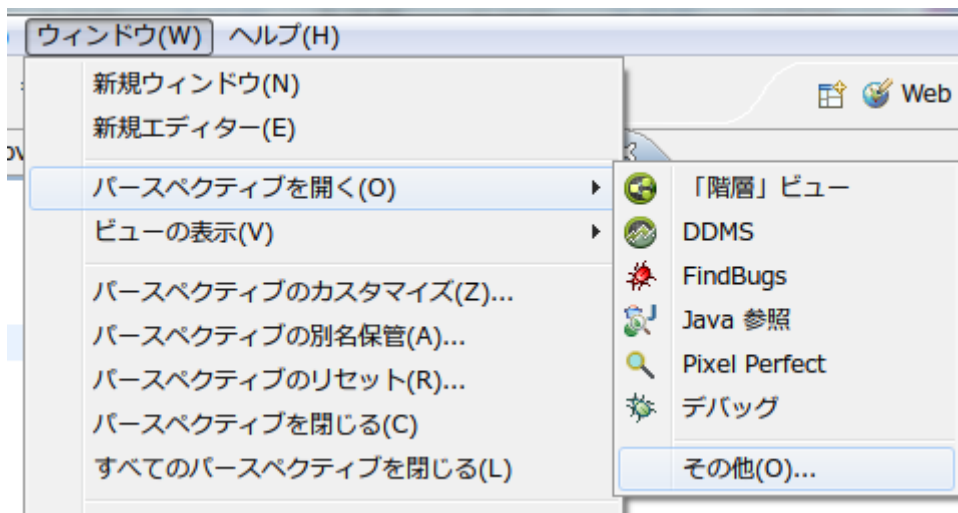
Override the existing AVD with the same name



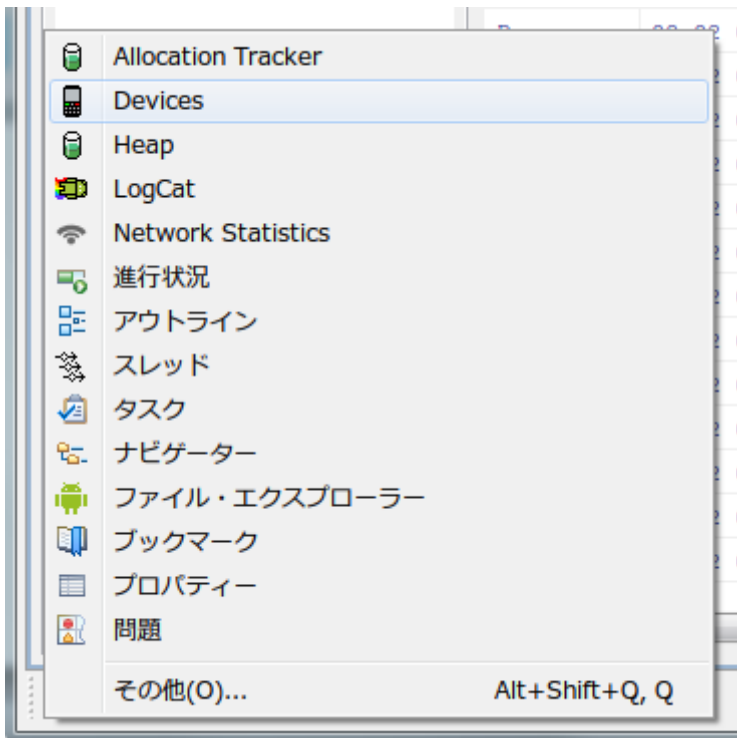
エミュレータが起動したら実行をクリックします。



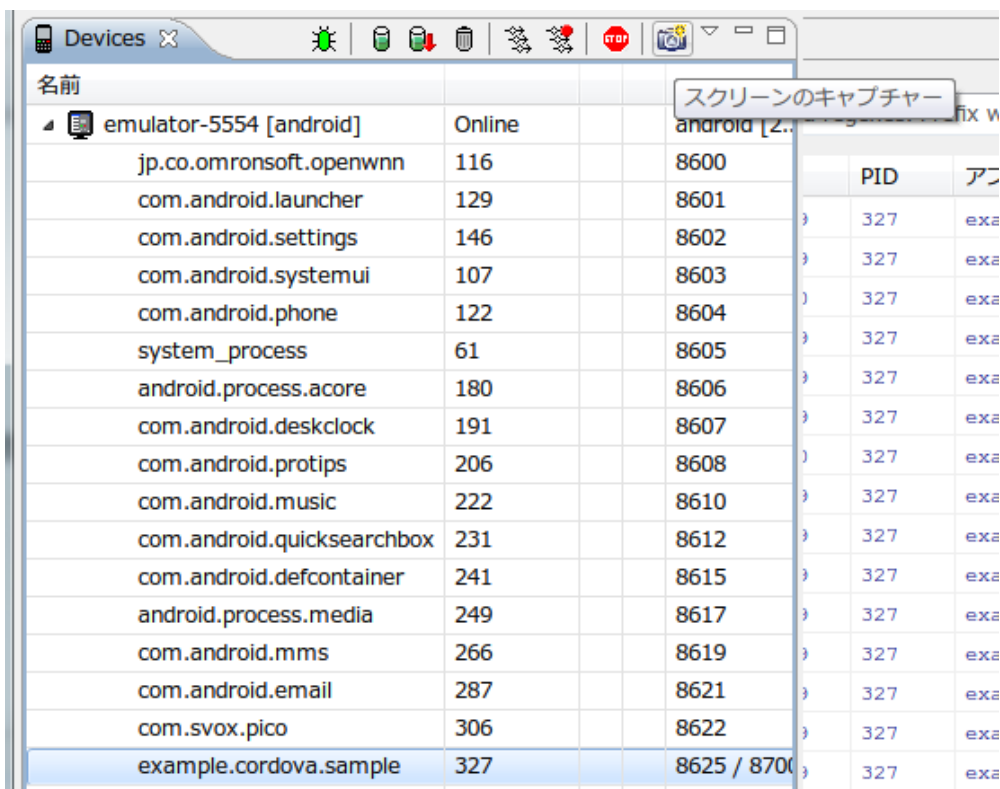
LogCat というタブに実行しているアプリのプロセスの過程が表示されます。エラーが出た場合はこれを元にデバッグします。また無限ループなどになっているとログが永遠と出続けます。これもデバッグの手がかりにしてください。

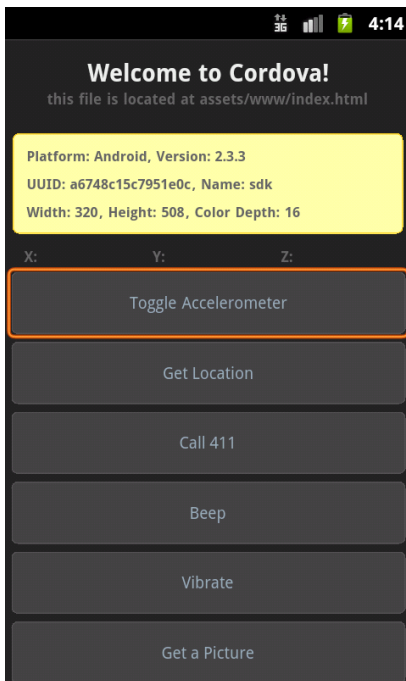


LogCat の詳細をみる時はウインドウ、パースペクティブを開く、DDMS を開きます。



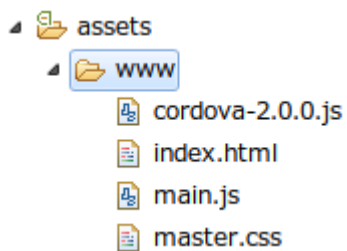
左下のアイコンをクリックすると実行しているアプリの詳細が見れます。
Device を選択、プロセスを選んで画面のキャプチャを取ることも可能。





エミュレータで起動しているアプリ画面のキャプチャを取ることができます。

Ctrl キーと **F11** を押すと画面が横向きになりテストすることが可能です。

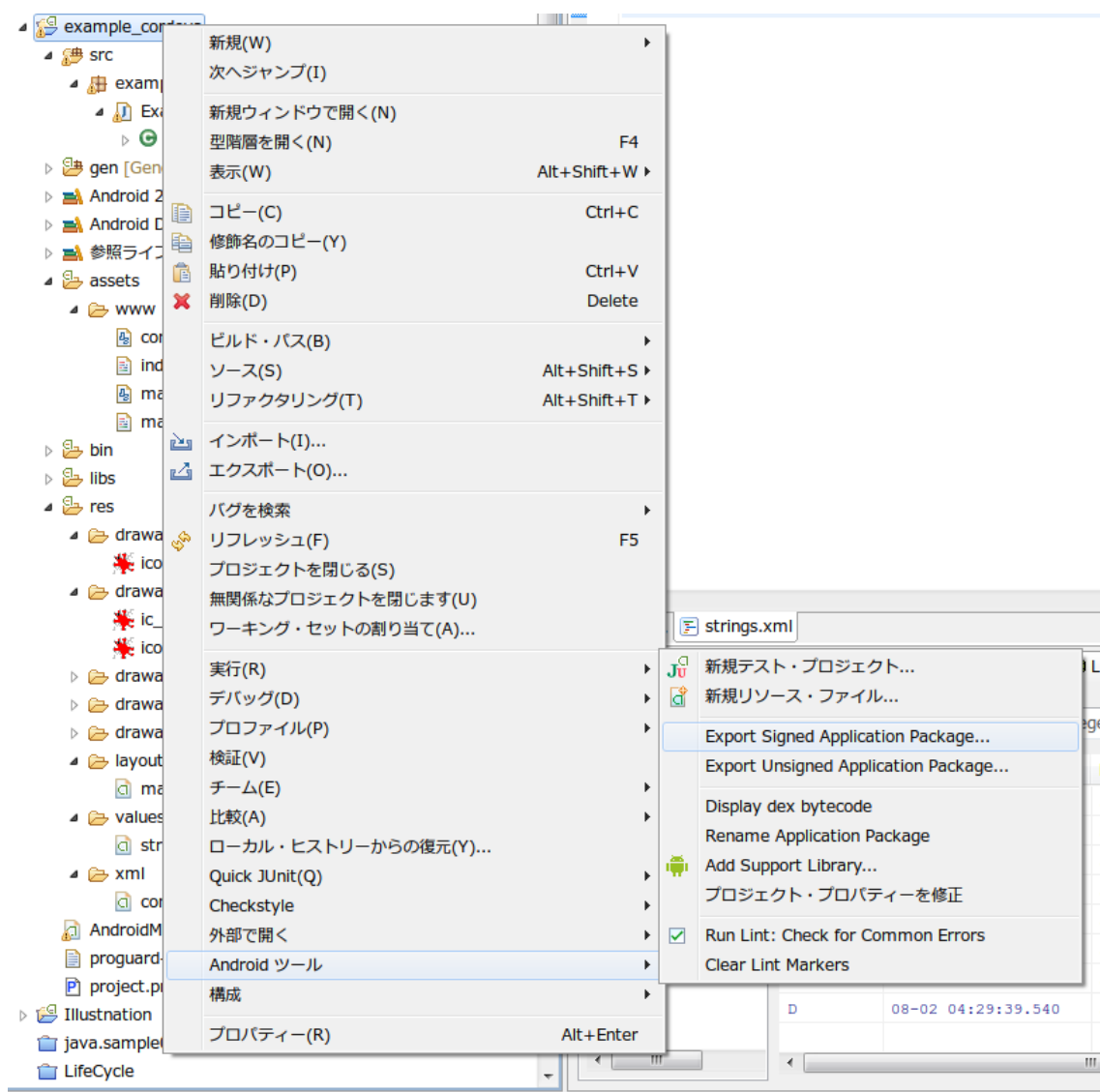


www フォルダの中の **html** ファイルが参照されているのでファイルをエクスプローラーで差し替えれば違う **html** ファイルを閲覧できます。指定は **index.html** になっているので起動時に最初に見れるのは **index.html** です。

ファイルを置換したらパッケージエクスプローラを選択して **F5** で **eclipse** に反映できます。あとはエミュレータで確認してください。ちなみにプロジェクトをコピーして量産しようとしてもパッケージ名が同じなので端末には別個にインストールできません。必ず新規からプロジェクトをつくってください。

5、マーケットに公開できる形式にする。

これでエミュレータで動く形になりましたが、実機でテストする場合は専用のケーブルで直結するかアプリを適当なサーバーにアップして、実機のブラウザから落としてインストールするか、面倒くさい場合は、実機に **DropBox** アプリを入れてそれを経由してインストールすることが可能です。マーケットに公開する場合はこれとは別にデジタル署名というのが必要です。



デジタル署名をするときはプロジェクトを右クリック、**Android ツール**、**Export Signed Application Package** を選択してデジタル署名をしていきます。

6、 iOS に PhoneGap を組み込む

iOS での設置は Android ほど難しくありません。圧縮ファイルにある `Cordova-2.0.0.pkg` をインストールすると Xcode に PhoneGap のプロジェクトが出来るようになります。後は手順通りすすめていきます。内蔵する `html` を `asset` フォルダなどにまとめ、つくっているプロジェクトをファインダーで開き、そこに設置します。ローカルファイルを読むようにプログラムを宣言したら、Xcode に反映させます。プロジェクトを更新するか `asset` フォルダを Xcode のプロジェクトにドラッグして認識させます。あとは起動時のスプラッシュ画面とアイコンを差し替えれば終わりです。