

Valinta Cocos2D

Integration Instructions

V0.5

Android

Android integration needs modifications to Android.mk file in JNI folder.
Add following lines (modify paths accordingly)

Android.mk

```
LOCAL_LDLIBS := -libzemeho_cocos_valinta_android.a
```

#These curl prebuilt files are found from your Cocos2D installation (external) path:

```
LOCAL_LDLIBS += -l<<<CURL PATH TO YOUR COCOS2D EXTERNAL>>/curl/prebuilt/android/armeabi-v7a/libcurl.a  
LOCAL_LDLIBS += -l<<<CURL PATH TO YOUR COCOS2D EXTERNAL>>/curl/prebuilt/android/armeabi-v7a/libssl.a  
LOCAL_LDLIBS += -l<<<CURL PATH TO YOUR COCOS2D EXTERNAL>>/curl/prebuilt/android/armeabi-v7a/libcrypto.a
```

Google Advertisement support

Audio ads networks needs GAID (in Android) in order to give more rated ads. There fore Google ad services are needed and couple of java files to fetch the GAID:

build.gradle:

Add following bolded lines to dependencies:

```
dependencies {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    compile 'com.google.android.gms:play-services-base:9.6.1'  
    compile 'com.google.android.gms:play-services-identity:9.6.1'  
    compile 'com.google.android.gms:play-services-ads:9.6.1'  
    compile project(':libcocos2dx')  
    compile 'com.google.android.gms:play-services-appindexing:9.6.1'  
}
```

Java Files:

Copy (from the sdk package) following files to your app folder: **src/com/zemeho/valinta**

NativeInterface.java

AdvertisingIdClient.java

iOS

For iOS integration add libzemeho_cocos_valinta_ios.a and libcurl.a as statically linked to your project and you are good to go like in example.

iOS plugin needs Ad.network framework into the xcode project.

macOS

For macOS integration add `libzemeho_cocos_valinta_mac.a` as statically linked to your project and you are good to go like in example.

Cocos2D C++ API

Function API:

```
void create(cocos2d::Layer *m_scene, string app_id, string client_id, string ad_id );  
void select_genre(int genre_id);  
void play_next_song();  
void stop_playback();  
bool play_next_ad();  
bool is_ad_available();  
void banner_ad_clicked( int ad_id );  
Void clear_cache();
```

Callbacks which developer needs to declare in own app

```
virtual void genres_received(const vector<unordered_map<string, string>> genrelist);  
virtual void now_playing(const unordered_map<string, string> song_info)  
virtual void now_playing_ad(const vector<unordered_map<string, string>> banner_info) ;  
virtual void audio_ad_available(bool available)  
virtual void ready_to_play()  
virtual void playback_stopped()  
virtual void playback_status(long position, long duration)
```

Cocos2D C++ API Usage

```
#include "zemeho_cocos_valinta_if.h"
```

//1. Inherit ValintaPlugin from the zemeho_cocos_valinta_if.h:

```
class MyValinta : public ValintaPlugin {
public:
    void genres_received(vector<unordered_map<string, string>> genrelist);
    void now_playing(unordered_map<string, string> song_info);
    void playing_song(vector<string> song_info);
    void now_playing_ad(const vector<unordered_map<string, string>> banner_info);
    void ready_to_play();
    void playback_status(long position, long duration);
    void playback_stopped();
    void audio_ad_available(bool available);
    void valinta_started(string m_advertisement_id);

};
static MyValinta *m_valinta;

// 2. Implement virtual functions (see full example from sdk):
void MyValinta::valinta_started(string m_advertisement_id){
    std::cout << "VALINTA STARTED!" ;
}
}
```

// 3. Start plugin with create command in your cocos2d scene class:

// NOTICE: You will need application ID from zemeho.valinta.com

```
GameScene::GameScene()
{
    m_valinta = new MyValinta();
    m_valinta->create((cocos2d::Layer*)this,"<<appname from
zemeho.valinta.com>>","<leave empty>","<leave empty>");
}

// 4. select genre and when ready to play, play next song
void MyValinta::genres_received(vector<unordered_map<string, string>>
genrelist)
{
    std::cout << "GENRES RECEIVED" << endl;
    if (genrelist.size() > 1) select_genre(atoi(genrelist[0]["id"].c_str()));
}
void MyValinta::ready_to_play()
{
    std::cout << "READY TO PLAY" << endl;
    play_next_song();
}
}
```