
MORE Volume Mute Controller SDK

Version 0.16.08.15

2016.08.15

Table of Contents

MORE Volume Mute Controller SDK	1
<i>Version 0.16.03.06</i>	1
1. Introduction.....	3
1.1. Overview	3
2. Service.....	4
2.1. Volume Mute Controller SDK	4
2.2. Use SDK Library	4
2.3. Volume Mute Function	7
3. Parameter Definition	7
3.1. OnCallbackResult Function Parameter.....	7
3.2. Handler 與 Callback listener 參數對應.....	8

1. Introduction

1.1. Overview

MORE Volume Mute Controller SDK 主要提供 Smart Mobile Device 更快速與簡潔的開發模組，APP 開發人員只要將 MORE Volume Mute Controller SDK 加入到自己的 APP 專案裡，透過 API 的呼叫即可讓智慧型裝置達到靜音的效果，目前提供 Android 開發版本。

2. Service

2.1. Volume Mute Controller SDK

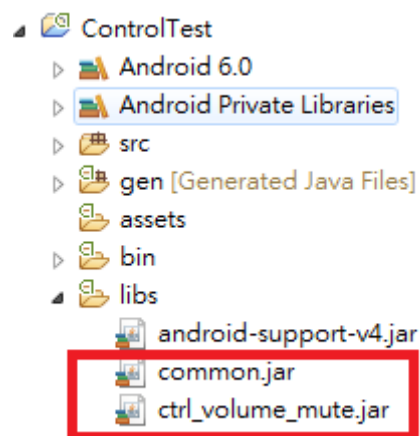
Volume Mute Controller SDK 主要提供給開發者一個能只要程式不被 Activity onDestroy()都能使智慧型裝置音效靜音。

2.2. Use SDK Library

使用 Volume Mute Controller SDK 不須任何權限，此模組會透過 Handler 或 Callback Listener 是否有成功且正確的執行靜音模式。

使用步驟：

1. 先將以下 jar 檔加入到 Android 開發專案的 libs 資料夾



2. 匯入以下 library

```
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import sdk.ideas.common.CtrlType;
import sdk.ideas.common.OnCallbackResult;
import sdk.ideas.ctrl.mute.MuteVolumeHandler;
```

3. 設定 Volume Mute Handler，建議設定為static，這樣不會重複new讓多個物件去搶一個音效裝置。

```
static MuteVolumeHandler mMuteVolumeHandler = new MuteVolumeHandler(this);
```

4. 本SDK 提供Handler 和Callback Listener方法來去知道程式是否有成功執行，開發者可以以自己喜好去做設定，以下是Handler 方法範例

1) 先宣告Handler

```
private Handler theHandler = new Handler()
{
    @Override
    public void handleMessage(Message msg)
    {
        switch (msg.what)
        {
            case CtrlType.MSG_RESPONSE_VOLUME_HANDLER:
                Log.d("handler response: ", "Result: " + String.valueOf(msg.arg1) + " From: "
                    + String.valueOf(msg.arg2) + " Message: " +
                    ((HashMap<String,String>)msg.obj) );
                break;
            default:
                break;
        }
    }
};
```

2) 設定此 handler

```
mMuteVolumeHandler.setHandler(theHandler);
```

5. Callback Listener 方式

```
mMuteVolumeHandler.setOnCallbackResultListener(new OnCallbackResult()
{
    @Override
    public void onCallbackResult(int result, int what, int from, HashMap<String,
String> message)
    {
        Log.d("Listener response: ", "Result: " + String.valueOf(result) + " What:
" + String.valueOf(what)
            + " From: " + String.valueOf(from) + " " + message);
    }

});
```

6. 使裝置靜音

```
mMuteVolumeHandler.startMute();
```

7. 使裝置開啟聲音

```
mMuteVolumeHandler.stopMute();
```

2.3. Volume Mute Function

Location Controller API	
Function	Description
MuteVolumeHandler (Context mContext)	建構元
void startMute()	使裝置靜音
void stopMute()	使裝置開啟聲音
void setHandler(Handler handler)	設定 Handler，參數對應可參考 3.2
void setOnCallbackResultListener(OnCallbackResult listener)	設定 callback listener，其參數參考 3.1

3. Parameter Definition

3.1. OnCallbackResult Function Parameter

result :

Value	Description
1	SUCCESS
0	UNKNOWN ERROR

what :

Value	Description
1032	VOLUME MUTE CONTROLLER HANDLER

from :

Value	Description
0	VOLUME MUTE ENABLE

	FUNCTION
1	VOLUME MUTE DISABLE FUNCTION

message :

key	Description
“message”	敘述執行成功或錯誤訊息

3.2. Handler 與 Callback listener 參數對應

Handler	OnCallbackResult
msg.what	int what
msg.arg1	int result
msg.arg2	int from
msg.obj	HashMap<String, String> message