



首頁 / Android 文章 / 正文

Android 3.0 r1中文API文檔(103) —— InputMethodManager [輸入法]

原創

鎖晴

2018-08-23 14:08

前言

本章內容是android.view.inputmethod.InputMethodManager，為輸入法相關章節，版本為Android 3.0 r1，翻譯來自"六必治"，歡迎大家訪問他的博客：<http://www.cnblogs.com/zcmky/>，再次感謝"六必治"！期待你加入Android API 中文的翻譯，聯繫我over140@gmail.com。

聲明

歡迎轉載，但請保留文章原始出處)

博客園：<http://www.cnblogs.com/>

Android中文翻譯組：<http://goo.gl/6vJQI>

正文

一、結構

*public final class **InputMethodManager** extends **Object***

java.lang.Object

android.view.inputmethod.InputMethodManager

二、類概述



整個輸入法框架（IMF）結構的核心API，應用程序之間進行調度和當前輸入法交互。你可以用[Context.getSystemService\(\)](#)取得這一接口的實例。

架構總述(Architecture Overview)

輸入法框架（IMF）共有三個主要部分：

- * **輸入法管理器**，管理各部分的交互。它是一個客戶端API，存在於各個應用程序的context中，用來溝通管理所有進程間交互的全局系統服務。
- * **輸入法(IME)**，實現一個允許用戶生成文本的獨立交互模塊。系統綁定一個當前的輸入法。使其創建和生成，決定輸入法何時隱藏或者顯示它的UI。同一時間只能有一個IME運行。
- * **客戶應用程序**，通過輸入法管理器控制輸入焦點和IME的狀態。一次只能有一個客戶端使用IME。

應用程序(Applications)

大多數情況下，使用標準TextView或其子類的應用程序只要做少量工作就可以讓軟鍵盤(soft input methods)正常工作。你要注意的是：

- * 正確設置你的可編輯文本視圖，這樣輸入法有足夠的上下文幫助用戶向其輸入文本。
- * 當輸入法顯示時妥善處理隱藏屏幕空間。理想的情況下應用程序應處理窗口變小，如必要你可以利用系統執行平移窗口。你應在活動（activity）中設置windowSoftInputMode屬性或調整創建窗口的相應值，以便系統決定平移或縮放（它會嘗試自動調整但可能出錯）。
- * 你還可以通過相同的windowSoftInputMode控制窗口的首選軟鍵盤狀態（打開、關閉等）。

通過API可以與IMF及其IME交互實現更細緻的控制，如顯示或輸入區域，用戶選取某輸入法等。

當你編寫自己的文本編輯器的時候，你要實現onCreateInputConnection(EditorInfo)以返回你的InputConnection的接口實例，用來允許IME和你的文本編輯域來交互。

輸入法(Input Methods)

一個輸入法（IME）是Service（服務）的實現，通常繼承自 InputMethodService。IME提供核心的 InputMethod接口，儘管提供InputMethod通常是由InputMethodService來處理，而IME的實現只需要處理更高層的API。

更多信息參閱InputMethodService。

安全(Security)



輸入法關係到許多安全問題，因其基本不受約束地驅動UI並監視用戶輸入。Android輸入法框架還允許調度第三方IME，所以必須小心以限制他們的選擇和相互作用。

以下是IMF背後安全架構的要點：

- ☒ 只允許系統訪問經BIND_INPUT_METHOD權限許可訪問IME的InputMethod接口。通過綁定到要求這個權限的服務來強制實現這一點。所以系統可以保證沒有不被信任的客戶端在它的控制之外訪問到當前的輸入法。
- ☒ IMF中可能有許多客戶進程，但在同一時間只有一個是激活的。未激活客戶端不能與IMF核心交互通過下述機制實現。
- ☒ 輸入法客戶端只可訪問InputMethodSession接口。每個客戶端創建一個接口實例，只有與激活客戶相關聯的會話的調用纔會被IME處理。這點通過普通IME的 AbstractInputMethodService執行，必須由IME顯式的處理，而IME正是InputMethodSession的自定義實現。
- ☒ 只有激活的InputConnection接受操作。IMF通知每個客戶進程是否激活，IME忽略非激活進程對當前輸入連接的調用。這確保了當前IME只將事件和文本編輯交付用戶可見焦點的UI。
- ☒ 一個IME永遠不能在屏幕關閉時與 InputConnection交互。這是通過當屏幕關閉時所有客戶端無效，並防止不良IME在用戶無法關注其行為時驅動UI。
- ☒ 客戶應用程序可以要求系統選取一個新的IME，但不能編程式選擇之一。這是避免惡意程序在用戶導航到其它應用程序時，選擇自己的IME並保持運行。也就是說，IME可以編程式的要求系統選擇其它IME，因它完全控制了用戶輸入。

用戶在可切換至一個新的IME前必須顯式的啓用它，以確認系統瞭解它，使其處於可用狀態。

三、常量

```
public static final int HIDE_IMPLICIT_ONLY
```

hideSoftInputFromWindow(IBinder, int)中的標誌，表示如果用戶未顯式地顯示軟鍵盤窗口，則隱藏窗口。

常量值: 1 (0x00000001)

```
public static final int HIDE_NOT_ALWAYS
```



`hideSoftInputFromWindow(IBinder, int)`中的標誌，表示軟鍵盤窗口總是隱藏，除非開始時以`SHOW_FORCE_HIDDEN`顯示。

常量值: 2 (0x00000002)

public static final int **RESULT_HIDDEN**

`showSoftInput(View, int, ResultReceiver)`和`hideSoftInputFromWindow(IBinder, int, ResultReceiver)`中`ResultReceiver`結果代碼標誌：軟鍵盤窗口從顯示切換到隱藏時的狀態。

常量值: 3 (0x00000003)

public static final int **RESULT_SHOWN**

`showSoftInput(View, int, ResultReceiver)`和`hideSoftInputFromWindow(IBinder, int, ResultReceiver)`中`ResultReceiver`結果代碼標誌：軟鍵盤窗口從隱藏切換到顯示時的狀態。

常量值: 2 (0x00000002)

public static final int **RESULT_UNCHANGED_HIDDEN**

`showSoftInput(View, int, ResultReceiver)`和`hideSoftInputFromWindow(IBinder, int, ResultReceiver)`中`ResultReceiver`結果代碼標誌：軟鍵盤窗口不變保持隱藏時的狀態。

常量值: 1 (0x00000001)

public static final int **RESULT_UNCHANGED_SHOWN**

`showSoftInput(View, int, ResultReceiver)`和`hideSoftInputFromWindow(IBinder, int, ResultReceiver)`中`ResultReceiver`結果代碼標誌：軟鍵盤窗口不變保持顯示時的狀態。

常量值: 0 (0x00000000)

public static final int **SHOW_FORCED**



`showSoftInput(View, int)`標誌，表示用戶強制打開輸入法（如長按菜單鍵），一直保持打開直至只有顯式關閉。

常量值: 2 (0x00000002)

public static final int **SHOW_IMPLICIT**

`showSoftInput(View, int)`標誌，表示隱式顯示輸入窗口，非用戶直接要求。窗口可能不顯示。

常量值: 1 (0x00000001)

四、公共方法

public void **displayCompletions** (View view, CompletionInfo[] completions)

（譯者注：輸入法自動完成）

public InputMethodSubtype **getCurrentInputMethodSubtype** ()

（譯者注：獲取當前輸入法類型？）



public List<InputMethodInfo> **getEnabledInputMethodList** ()

（譯者注：獲取已啓用輸入法列表？）

public List<InputMethodSubtype> **getEnabledInputMethodSubtypeList** (InputMethodInfo imi, boolean allowsImplicitlySelectedSubtypes)

public List<InputMethodInfo> **getInputMethodList** ()

（譯者注：獲取輸入法列表）

public Map<InputMethodInfo, List<InputMethodSubtype>> **getShortcutInputMethods/**  **types** () 

public void **hideSoftInputFromInputMethod** (IBinder token, int flags)

關閉/隱藏輸入法軟鍵盤區域，用戶不再看到或與其交互。只能由當前激活輸入法調用，因需令牌(token)驗證。

參數

token 在輸入法啟動時提供令牌驗證，驗證後可對其進行操作。

flags 提供額外的操作標誌。當前可以為0或 [HIDE_IMPLICIT_ONLY](#), [HIDE_NOT_ALWAYS](#) 等位設置。

public boolean **hideSoftInputFromWindow** (IBinder windowToken, int flags)

[hideSoftInputFromWindow\(IBinder, int, ResultReceiver\)](#)的無返回值版：從窗口上下文中確定當前接收輸入的窗口，隱藏其輸入法窗口

參數

windowToken 由窗口請求[View.getWindowToken\(\)](#)返回得到的令牌(token)。

flags 提供額外的操作標誌。當前可以為0或 [HIDE_IMPLICIT_ONLY](#)位設置。

public boolean **hideSoftInputFromWindow** (IBinder windowToken, int flags, ResultReceiver resultReceiver)

從窗口上下文中確定當前接收輸入的窗口，要求隱藏其軟鍵盤窗口。它可由用戶調用並得到結果而不僅僅是顯式要求輸入法窗口隱藏。

參數

windowToken 由窗口請求[View.getWindowToken\(\)](#)返回得到的令牌(token)。

flags 提供額外的操作標誌。當前可以為0或 [HIDE_IMPLICIT_ONLY](#)位設置。

resultReceiver 如不為空，當IME處理請求告訴你完成時調用。你收到的結果碼可以是[RESULT_UNCHANGED_SHOWN](#), [RESULT_UNCHANGED_HIDDEN](#), [RESULT_SHOWN](#), 或[RESULT_HIDDEN](#)。 ^

public void **hideStatusIcon** (IBinder imeToken)

(譯者注：隱藏狀態欄圖標？)

public boolean **isAcceptingText** ()

當前服務視圖接受全文編輯返回真。沒有輸入法聯接為false，這時其只能處理原始按鍵事件。

public boolean **isActive** (View view)

視圖為當前輸入的激活視圖時返回真。

public boolean **isActive** ()

輸入法中的任意視圖激活時返回真。

public boolean **isFullscreenMode** ()

判斷相關輸入法是否以全屏模式運行。全屏時，完全覆蓋你的UI時，返回真，否則返回假。

public boolean **isWatchingCursor** (View view)

如當前輸入法要看到輸入編輯者的光標位置時返回真。

public void **restartInput** (View view)

如有輸入法聯接至視圖，重啓輸入以顯示新的內容。可在以下情況時調用此方法：視圖的文字導致輸入法外觀變化或有按鍵輸入流，如應用程序調用TextView.setText()時。

參數

view 文字發生變化的視圖。



public void **sendAppPrivateCommand** (View view, String action, Bundle data)

對當前輸入法調用 `InputMethodSession.appPrivateCommand()`。

參數

- view** 可選的發送命令的視圖，如你要發送命令而不考慮視圖附加到輸入法，此項可以為空。
- action** 執行的命令名稱。必須是作用域的名稱，如前綴包名稱，這樣不同的開發者就不會創建衝突的命令。
- data** 命令中包含的任何數據。

public boolean **setCurrentInputMethodSubtype** (InputMethodSubtype subtype)

（譯者注：此方法為3.0中新增的方法）

public void **setInputMethod** (IBinder token, String id)

強制切換到新輸入法部件。只能由持有token的應用程序(application)或服務(service) 調用當前激活輸入法。

參數

- token** 在輸入法啟動時提供令牌驗證，驗證後可對其進行操作。
- id** 切換到新輸入法的唯一標識。

public void **setInputMethodAndSubtype** (IBinder token, String id, InputMethodSubtype subtype)

強制切換到一個新的輸入法和指定的類型。只能由持有token的應用程序(application)或服務(service) 調用當前激活輸入法。（譯者注：此方法為3.0中新增的方法）

參數

- token** 在輸入法啟動時提供令牌驗證，驗證後可對其進行操作。
- id** 切換到新輸入法的唯一標識。



subtype 切換到新輸入法的新類型。

```
public void showInputMethodAndSubtypeEnabler (String topId)
```

(譯者注：此方法為3.0中新增的方法)

```
public void showInputMethodPicker ()
```

(譯者注：顯示輸入法菜單列表)

```
public boolean showSoftInput (View view, int flags, ResultReceiver resultReceiver)
```

如需要，顯式要求當前輸入法的軟鍵盤區域向用戶顯示。當用戶與視圖交互，用戶表示要開始執行輸入操作時，可以調用此方法。

參數

view 當前焦點視圖，可接受軟鍵盤輸入。

flags 提供額外的操作標誌。當前可以是0或[SHOW_IMPLICIT](#) 位設置。

resultReceiver 如不為空，當IME處理請求告訴你完成時調用。你收到的結果碼可以是[RESULT_UNCHANGED_SHOWN](#), [RESULT_UNCHANGED_HIDDEN](#), [RESULT_SHOWN](#), 或 [RESULT_HIDDEN](#)。

```
public boolean showSoftInput (View view, int flags)
```

[showSoftInput\(View, int, ResultReceiver\)](#)的無返回值版：如需要，顯式要求當前輸入法的軟鍵盤區域向用戶顯示。

參數

view 當前焦點視圖，可接受軟鍵盤輸入。

flags 提供額外的操作標誌。當前可以是0或[SHOW_IMPLICIT](#) 位設置。

```
public void showSoftInputFromInputMethod (IBinder token, int flags)
```



顯示輸入法的軟鍵盤區域，這樣用戶可以到看到輸入法窗口並能與其交互。只能由當前激活輸入法調用，因需令牌(token)驗證。

參數

token 在輸入法啟動時提供令牌驗證，驗證後可對其進行操作。

flags 提供額外的操作標誌。可以是0或 [SHOW_IMPLICIT](#), [SHOW_FORCED](#)位設置。

```
public void showStatusIcon (IBinder imeToken, String packageName, int iconId)
```

(譯者注：顯示狀態欄圖標？)

```
public boolean switchToLastInputMethod (IBinder imeToken)
```

```
public void toggleSoftInput (int showFlags, int hideFlags)
```

(譯者注：切換軟鍵盤)

```
public void toggleSoftInputFromWindow (IBinder windowToken, int showFlags, int hideFlags)
```

本方法切換輸入法的窗口顯示。如輸入窗口已顯示，它隱藏。如無輸入窗口則顯示。

參數

windowToken 由窗口請求[View.getWindowToken\(\)](#)返回得到的令牌(token)。

showFlags 提供額外的操作標誌。當前可以為0或 [HIDE_IMPLICIT_ONLY](#)位設置。

hideFlags 提供額外的操作標誌。可以是0或 [HIDE_IMPLICIT_ONLY](#), [HIDE_NOT_ALWAYS](#)位設置。

```
public void updateCursor (View view, int left, int top, int right, int bottom)
```

返回窗口的當前光標位置。



public void **updateExtractedText** (View view, int token, ExtractedText text)

(譯者注：當內容變化時文本編輯器調用此方法，通知其新提取文本。)

public void **updateSelection** (View view, int selStart, int selEnd, int candidatesStart, int candidates
End)

返回當前選擇區域。

發表評論

登錄以後才評論...

登录

所有評論

還沒有人評論，想成為第一個評論的人麼？請在上方評論欄輸入並且點擊發布。

相關文章

eclipse 閱讀代碼 快捷鍵 快速方便

1. Ctrl+左鍵這個是大多數人經常用到的，用來查看變量、方法、類的定義2. Ctrl+O查看一個類的綱要，列出其方法和成員變量。提示：再多按一次Ctrl+O，可以列出該類繼承的方法和變量。助記："O"--->"Outline"--->"綱要"3. Ctrl+T查看一個類的繼承關係樹，是自頂向下的，再多按一次Ctrl+T，會換成自底向上的顯示結構。提示：選中一個方法名，按Ctrl+T，可以查看到

🕒 锁晴 🕒 2018-08-23 14:08



退出 android 應用程序

1. finish()方法該方法可以結束當前 Activity，但是如果你的 App 有很多 Activity 的話，使用該方法顯得有點捉襟見肘了。另外，還有一個方法 finishActivity (int requestCode) ,關於這個方法，先看看 sdk 的 api 說明吧！
public void finishActivity (int requestCode)Since: API Le

🕒 锁晴 🕒 2018-08-23 14:08

[Android] ImageView.ScaleType 設置圖解

ImageView的ScaleType決定了圖片在View上顯示時的樣子，如進行何種比例的縮放，及顯示圖片的整體還是部分，等等。設置的方式包括：1. 在layout xml中定義android:scaleType="CENTER"2. 或在代碼中調用imageView.setScaleType(ImageView.ScaleType.CENTER);接下來，將對ScaleType的值和對應的顯示效

🕒 锁晴 🕒 2018-08-23 14:08

Android Fragment和FragmentActivity區別和用法

Android Fragment和FragmentActivity區別和用法Android Fragment是Android4.0以上纔有的；而FragmentActivity是爲了兼容4.0以下版本的Fragment使用的。所以如果你想兼容4.0以下Android版本使用Fragment的話，框架Activity需要繼承FragmentActivity，FragmentActivity這個類是在

🕒 锁晴 🕒 2018-08-23 14:07
