



ANDROID - DRAG AND DROP

T. Shaden AlZeer

OUTLINE

The Drag/Drop Process ➤

The DragEvent Class ➤

Listening for Drag Event ➤

Starting a Drag Event ➤

عملية السحب / الإفلات

فئة DragEvent ➤

الاستماع لحدث السحب ➤

بدء حدث السحب ➤

INTRODUCTION

Android drag/drop framework allows your users to move data from one View to another View in the current layout using a graphical .drag and drop gesture

يسمح إطار عمل السحب / الإفلات في Android للمستخدمين بنقل البيانات من طريقة عرض إلى طريقة عرض أخرى في التخطيط الحالي باستخدام إيماءة السحب والإفلات الرسومية.

.As of **API 11** drag and drop of view onto other views or view groups is supported

: The framework includes following three important components to support drag & drop functionality

- .1 **Drag event class**
- .2 **Drag listeners**
- .3 **Helper methods and classes**
- .4 اعتبارًا من API 11 ، يتم دعم السحب وإسقاط العرض على طرق العرض أو مجموعات العرض الأخرى.
- .5 يتضمن إطار العمل التالي ثلاثة مكونات مهمة لدعم وظيفة السحب والإفلات:
- .6 اسحب فئة الحدث.
- .7 اسحب المستمعين.
- .8 طرق وفصول المساعد.

THE DRAG/DROP PROCESS

– There are basically four steps or states in the drag and drop process

هناك أربع خطوات أو حالات في الأساس في عملية السحب والإفلات





Started

**Continuin
g**

Dropped

Ended

Started

- This event occurs when you start dragging an item in a layout, your application .calls *startDrag()* method to tell the system to start a drag
- The arguments inside *startDrag()* method provide the data to be dragged, metadata for this .data, and a callback for drawing the drag shadow
- The system first responds by calling back to your application to get a drag shadow. It then .displays the drag shadow on the device
- Next, the system sends a drag event with action type *ACTION_DRAG_STARTED* to the .registered drag event listeners for all the View objects in the current layout
- To continue to receive drag events, including a possible drop event, a drag event listener must return **true**, If the drag event listener returns false, then it will not receive drag events for the current operation until the system sends a drag event with action type .ACTION_DRAG_ENDED

بدأت

يحدث هذا الحدث عندما تبدأ في سحب عنصر في تخطيط ، يستدعي التطبيق الخاص بك طريقة `startDrag ()` لإخبار النظام ببدء السحب.

توفر الوسائط داخل طريقة `startDrag ()` البيانات المراد سحبها والبيانات الوصفية لهذه البيانات واستدعاء لرسم ظل السحب.

يستجيب النظام أولاً عن طريق معاودة الاتصال بتطبيقك للحصول على ظل السحب. ثم يعرض ظل السحب على الجهاز.

بعد ذلك ، يرسل النظام حدث سحب بنوع الإجراء `ACTION_DRAG_STARTED` إلى مستمعي حدث السحب المسجلين لجميع كائنات العرض في التخطيط الحالي.

للاستمرار في تلقي أحداث السحب ، بما في ذلك حدث الإسقاط المحتمل ، يجب أن يعود مستمع حدث السحب صحيحاً ، إذا قام مستمع حدث السحب بإرجاع خطأ ، فلن يتلقى أحداث السحب للعملية الحالية حتى يرسل النظام حدث سحب بنوع الإجراء `ACTION_DRAG_ENDED`.



Continuing

.The user continues the drag

System sends ACTION_DRAG_ENTERED action followed by ACTION_DRAG_LOCATION action to the registered drag event listener for the View where dragging point enters

The listener may choose to alter its View object's appearance in response to the event or can react by highlighting its View

The drag event listener receives a ACTION_DRAG_EXITED action after the users has moved the drag shadow outside the bounding box of the View

استمرار

يستمر المستخدم في السحب.

يرسل النظام إجراء ACTION_DRAG_ENTERED متبوعاً بإجراء ACTION_DRAG_LOCATION إلى مستمع حدث السحب المسجل للعرض حيث تدخل نقطة السحب.

قد يختار المستمع تغيير مظهر كائن العرض الخاص به استجابةً للحدث أو يمكنه الرد من خلال إبراز طريقة العرض الخاصة به.

يتلقى مستمع حدث السحب إجراء ACTION_DRAG_EXITED بعد أن يقوم المستخدم بتحريك ظل السحب خارج المربع المحيط للعرض.



Dropped

The user releases the dragged item within the bounding box of a `.View`

The system sends the View object's listener a drag event with action `.type ACTION_DROP`

إسقاط

يقوم المستخدم بتحرير العنصر المسحوب داخل المربع المحيط للعرض.

يرسل النظام إلى مستمع الكائن "عرض" حدث سحب بنوع الإجراء `ACTION_DROP`.



Ended

Just after the action type ACTION_DROP, the system sends out a drag event with action type ACTION_DRAG_ENDED to indicate that the drag operation is over

انتهى

بعد نوع الإجراء ACTION_DROP مباشرة ، يرسل النظام حدث سحب بنوع الإجراء ACTION_DRAG_ENDED للإشارة إلى انتهاء عملية السحب..

THE DRAGEVENT CLASS

The **DragEvent** represents an event that is sent out by the system at various times during a drag and drop operation

This class provides few Constants and important methods which we use during Drag/Drop process

يمثل DragEvent حدثاً يتم إرساله بواسطة النظام في أوقات مختلفة أثناء عملية السحب والإفلات.
توفر هذه الفئة عدداً قليلاً من الثوابت والأساليب المهمة التي نستخدمها أثناء عملية السحب / الإفلات.

CONSTANTS

Sr.No.	Constants & Description
1	ACTION_DRAG_STARTED Signals the start of a drag and drop operation.
2	ACTION_DRAG_ENTERED Signals to a View that the drag point has entered the bounding box of the View.
3	ACTION_DRAG_LOCATION Sent to a View after ACTION_DRAG_ENTERED if the drag shadow is still within the View object's bounding box.
4	ACTION_DRAG_EXITED Signals that the user has moved the drag shadow outside the bounding box of the View.
5	ACTION_DROP Signals to a View that the user has released the drag shadow, and the drag point is within the bounding box of the View.
6	ACTION_DRAG_ENDED Signals to a View that the drag and drop operation has concluded.

METHODS

Sr.No.	Constants & Description
1	int getAction() Inspect the action value of this event..
2	ClipData getClipData() Returns the ClipData object sent to the system as part of the call to startDrag().
3	ClipDescription getClipDescription() Returns the ClipDescription object contained in the ClipData.
4	boolean getResult() Returns an indication of the result of the drag and drop operation.
5	float getX() Gets the X coordinate of the drag point.
6	float getY() Gets the Y coordinate of the drag point.
7	String toString() Returns a string representation of this DragEvent object.

LISTENING FOR DRAG EVENT

If you want any of your views within a Layout should respond Drag event then your view either implements **View.OnDragListener** or .setup **onDragEvent(DragEvent)** callback method

When the system calls the method or listener, it passes to them a DragEvent object .explained above. You can have both a listener and a callback method for View object

If this occurs, the system first calls the listener and then defined callback as long as .listener returns true

The combination of the *onDragEvent(DragEvent)* method and *View.OnDragListener* is analogous to the combination of the **onTouchEvent()** and **View.OnTouchListener** used with touch events in old .versions of Android

إذا كنت تريد أي من طرق العرض الخاصة بك ضمن Layout يجب أن تستجيب لحدث السحب ، فإن طريقة العرض الخاصة بك إما تنفذ طريقة العرض `View.OnDragListener` أو الإعداد `onDragEvent` (`DragEvent`).

عندما يستدعي النظام الطريقة أو المستمع ، فإنه يمرر إليهم كائن `DragEvent` الموضح أعلاه. يمكن أن يكون لديك كل من المستمع وطريقة رد الاتصال لكائن العرض. في حالة حدوث ذلك ، يقوم النظام أولاً باستدعاء المستمع ثم تحديد رد الاتصال طالما أن المستمع يعود صحيحاً.

الجمع بين طريقة (`DragEvent`) `onDragEvent` و `View.OnDragListener` يماثل الجمع بين `onTouchEvent ()` و `View.OnTouchListener` المستخدمة مع أحداث اللمس في الإصدارات القديمة من Android.

STARTING A DRAG EVENT

You start with creating a **ClipData** and **ClipData.Item** for the data being moved. As part of the *ClipData* object, supply metadata that is stored in a **ClipDescription** object within the *ClipData*

For a drag and drop operation that does not represent data movement, you may want to use **null** instead of an actual object

Next either you can extend **View.DragShadowBuilder** to create a drag shadow for dragging the view or simply you can use *View.DragShadowBuilder(View)* to create a default drag shadow that's the same size as the *View* argument passed to it, with the touch point centered in the drag shadow

تبدأ بإنشاء ClipData و ClipData.Sem للبيانات التي يتم نقلها. كجزء من كائن ClipData قم بتوفير بيانات التعريف المخزنة في كائن ClipDescription داخل ClipData.

بالنسبة لعملية السحب والإفلات التي لا تمثل حركة البيانات ، قد ترغب في استخدام القيمة الخالية بدلاً من الكائن الفعلي.

بعد ذلك ، يمكنك إما توسيع العرض. اسحب ShadowBuilder لإنشاء ظل سحب لسحب العرض أو يمكنك ببساطة استخدام العرض. تتمحور النقطة في ظل السحب

EXAMPLE

```
private final class ChoiceTouchListener implements View.OnTouchListener {
```

```
    @Override
```

```
    public boolean onTouch(View v, MotionEvent event) {
```

```
        if ((event.getAction() == ACTION_DOWN) && (((ImageView) v).getDrawable() != null))
```

```
        {
```

```
            ClipData data = ClipData.newPlainText("", "");
```

```
            View.DragShadowBuilder shadowBuilder = new View.DragShadowBuilder(v);
```

```
            v.startDrag(data, shadowBuilder, v, 0);
```

```
            return true;
```

```
        } else {
```

```
            return false;
```

```
        }
```

```
    }
```

```
}
```

```
private class ChoiceDragListener implements View.OnDragListener {  
    @Override  
    public boolean onDrag(View v, DragEvent event) {  
        switch (event.getAction()) {  
            case DragEvent.ACTION_DRAG_STARTED:  
                break;  
            case DragEvent.ACTION_DRAG_ENTERED:  
                break;  
            case DragEvent.ACTION_DRAG_EXITED:  
                break;  
            case DragEvent.ACTION_DROP:  
                ImageView view = (ImageView) event.getLocalState();  
                ((ImageView) v).setImageDrawable(getResources().getDrawable(R.drawable.android));  
                ((ImageView) view).setImageDrawable(null);  
                break;  
            case DragEvent.ACTION_DRAG_ENDED:  
                break; }  
        return true; }}  

```



ANY QUESTIONS?