2016

AccessibilityNodeInfo.AccessibilityAction

中国信息无障碍产品联盟&信息无障碍研究会　译制

20161208

目录

翻译声明

翻译机构：信息无障碍研究会（ARA） 中国信息无障碍产品联盟（CAPA）

译者：刘辉

审阅：李鸿利、刘彪、沈广荣

本文档翻译自谷歌官方文档《AccessibilityNodeInfo.AccessibilityAction》。如您对翻译文档内容有异议，请将原文文档做为主要参考，原文版权由 Google 持有并保留。

AccessibilityNodeInfo.AccessibilityAction

添加于 API 级别 21。

public static final class AccessibilityNodeInfo.AccessibilityAction

继承自 Object

java.lang.Object

↳ android.view.accessibility.AccessibilityNodeInfo.AccessibilityAction

该类定义了一个可以在 AccessibilityNodeInfo 上执行的操作。每一个操作都有一个唯一的 id，且该 id 是强制的、可选的数据。

有三种操作类别：

- **标准操作**：这些操作被平台标准 UI 部件报告和处理。在该类中每一个标准操作定义了一个静态常量，例如，ACTION_FOCUS。
- **自定义操作**：这些操作被自定义部件报告和处理，即，这些操作不被包含在 UI 工具包中。例如，一个应用可能定义一个自定义操作来清除用户历史记录。
- **重写标准操作**：这些操作重写标准操作来自定义他们。例如，一个应用添加一个标签到 ACTION_CLICK 操作来告知该操作清除浏览历史。

一般使用 onInitializeAccessibilityNodeInfo(AccessibilityNodeInfo) 中的 addAction(AccessibilityAction) 将操作添加到 AccessibilityNodeInfo，并且在 performAccessibilityAction(int, Bundle)中执行。

注意：支持这些操作的视图，应该激活带有 IMPORTANT_FOR_ACCESSIBILITY_YE 的 setImportantForAccessibility(int)，来保证 AccessibilityService 可以获取被支持操作的设置。

1.概览

1.1 字段

public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_ACCESSIBILITY_FOCUS 赋予节点无障碍焦点的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_CLEAR_ACCESSIBILITY_FOCUS 清除节点无障碍焦点的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_CLEAR_FOCUS 清除节点输入焦点的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_CLEAR_SELECTION 取消选择节点的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_CLICK 点击节点信息的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_COLLAPSE 折叠可展开节点的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_CONTEXT_CLICK 上下文点击节点的操作。
public static final	ACTION_COPY

AccessibilityNodeInfo.AccessibilityAction	复制当前选择到剪贴板的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_CUT 剪切当前选择并放置到剪贴板的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_DISMISS 关闭一个可关闭节点的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_EXPAND 展开一个可展开节点的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_FOCUS 赋予节点输入焦点的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_LONG_CLICK 点击长按节点的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_NEXT_AT_MOVEMENT_GRANULARITY 以给定移动粒度，请求去到节点文本中的下一个实体的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_NEXT_HTML_ELEMENT 移动到给定类型的下一个 HTML 元素的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_PASTE 粘贴当前剪贴板内容的操作。

public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_PREVIOUS_AT_MOVEMENT_GRANULARITY 以给定移动粒度，请求去到节点文本中的上一个实体的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_PREVIOUS_HTML_ELEMENT 移动到给定类型的上一个 HTML 元素的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_SCROLL_BACKWARD 向后滚动节点内容的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_SCROLL_DOWN 向下滚动节点内容的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_SCROLL_FORWARD 向前滚动节点内容的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_SCROLL_LEFT 向左滚动节点内容的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_SCROLL_RIGHT 向右滚动节点内容的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_SCROLL_TO_POSITION 滚动节点内容让指定集合位置在屏幕上可见的操作。
public static final	ACTION_SCROLL_UP

AccessibilityNodeInfo.AccessibilityAction	向上滚动节点内容的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_SELECT 选择节点的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_SET_PROGRESS 设置进度的操作，进度值在RangeInfo.getMin()和RangeInfo.getMax()之间。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_SET_SELECTION 设置选择的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_SET_TEXT 设置节点文本的操作。
public static final AccessibilityNodeInfo.AccessibilityAction	ACTION_SHOW_ON_SCREEN 请求让节点边框矩形在屏幕上可见的操作，如果有必要就滚动。

1.2 公有构造函数

AccessibilityNodeInfo.AccessibilityAction(int actionId, CharSequence label)

创建一个新的 AccessibilityAction。

1.3 公有方法

boolean	equals(Object other)

	标识某些其他对象是否“等同于”该对象。
int	getId() 获取该操作的 id。
CharSequence	getLabel() 获取该操作的 label。
int	hashCode() 返回该对象的哈希编码值。
String	toString() 返回该对象的字符串表示。

1.4 继承方法

继承自 java.lang.Object 类。

Object	clone() 创建并返回该对象的复本。
boolean	equals(Object obj) 标识某些其他对象是否"等同于"该对象。
void	finalize() 当垃圾收集器确认再也没有该对象的引用时，垃圾收集器调用该方法。
final Class<?>	getClass()

	返回该对象的运行类。
int	hashCode() 为该对象返回哈希编码值。
final void	Notify() 唤醒在对象监视器上等待的单线程。
final void	notifyAll() 唤醒在对象监视器上等待的所有线程。
String	toString() 返回对象的字符串表示。
final void	wait(long millis, int nanos) 导致当前线程等待，直到另一线程为该对象调用 Notify()方法或 notifyAll()方法，或某些其他线程中断当前线程，或一定数量的实时运行已经停止。
final void	wait(long millis) 导致当前线程等待，直到另一线程为该对象调用 Notify()方法或 notifyAll()方法，或一定数量的实时运行已经停止。
final void	wait() 导致当前线程等待，直到另一线程为该对象调用 Notify()方法或 notifyAll()方法。

2.字段

2.1ACTION_ACCESSIBILITY_FOCUS

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_ACCESSIBILITY_FOCUS

赋予节点无障碍焦点的操作。

2.2ACTION_CLEAR_ACCESSIBILITY_FOCUS

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_CLEAR_ACCESSIBILITY_FOCUS

清除节点无障碍焦点的操作。

2.3ACTION_CLEAR_FOCUS

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_CLEAR_FOCUS

清除节点输入焦点的操作。

2.4ACTION_CLEAR_SELECTION

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_CLEAR_SELECTION

取消选择节点的操作。

2.5ACTION_CLICK

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_CLICK

点击节点信息的操作。

2.6ACTION_COLLAPSE

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_COLLAPSE

折叠可展开节点的操作。

2.7ACTION_CONTEXT_CLICK

添加于 API 级别 23。

AccessibilityNodeInfo.AccessibilityAction ACTION_CONTEXT_CLICK

上下文点击节点的操作。

2.8ACTION_COPY

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_COPY

复制当前选择到剪贴板的操作。

2.9ACTION_CUT

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_CUT

剪贴当前选择并将其放置到剪贴板的操作。

2.10ACTION_DISMISS

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_DISMISS

关闭一个可关闭节点的操作。

2.11ACTION_EXPAND

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_EXPAND

展开一个可展开节点的操作。

2.12ACTION_FOCUS

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_FOCUS

赋予节点输入焦点的操作。

2.13ACTION_LONG_CLICK

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_LONG_CLICK

点击长按节点的操作。

2.14ACTION_NEXT_AT_MOVEMENT_GRANULARITY

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction
ACTION_NEXT_AT_MOVEMENT_GRANULARITY

以给定移动粒度，请求去到节点文本中的下一个实体的操作。例如，移动到下一个字符、单词等。

参数：
AccessibilityNodeInfo.ACTION_ARGUMENT_MOVEMENT_GRANULARITY_INT，
AccessibilityNodeInfo.ACTION_ARGUMENT_EXTEND_SELECTION_BOOLEAN

样例：移动到上一个字符，并且不会扩大选择。

```
Bundle arguments = new Bundle();
arguments.putInt(AccessibilityNodeInfo.ACTION_ARGUMENT_MOVEMENT_GRANULARITY_INT,AccessibilityNodeInfo.MOVEMENT_GRANULARITY_CHARACTER);

arguments.putBoolean(AccessibilityNodeInfo.ACTION_ARGUMENT_EXTEND_SELECTION_BOOLEAN,false);

info.performAction(AccessibilityAction.ACTION_NEXT_AT_MOVEMENT_GR
```

```
ANULARITY.getId(),arguments);
```

参见：

AccessibilityNodeInfo.ACTION_ARGUMENT_MOVEMENT_GRANULARITY_INT

AccessibilityNodeInfo.ACTION_ARGUMENT_EXTEND_SELECTION_BOOLEAN

AccessibilityNodeInfo.ACTION_ARGUMENT_EXTEND_SELECTION_BOOLEAN

AccessibilityNodeInfo.getMovementGranularities()

AccessibilityNodeInfo.MOVEMENT_GRANULARITY_CHARACTER

AccessibilityNodeInfo.MOVEMENT_GRANULARITY_WORD

AccessibilityNodeInfo.MOVEMENT_GRANULARITY_LINE

AccessibilityNodeInfo.MOVEMENT_GRANULARITY_PARAGRAPH

AccessibilityNodeInfo.MOVEMENT_GRANULARITY_PAGE

2.15ACTION_NEXT_HTML_ELEMENT

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_NEXT_HTML_ELEMENT

移动到给定类型的下一个 HTML 元素的操作。例如，移动到 BUTTON、INPUT、TABLE 等。

参数：
AccessibilityNodeInfo.ACTION_ARGUMENT_HTML_ELEMENT_STRING

样例：

```
Bundle arguments = new Bundle();
arguments.putString(AccessibilityNodeInfo.ACTION_ARGUMENT_HTML_ELEMENT_STRING, "BUTTON");
info.performAction(AccessibilityAction.ACTION_NEXT_HTML_ELEMENT.getId(), arguments);
```

2.16ACTION_PASTE

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_PASTE

粘贴当前剪贴板内容的操作。

2.17ACTION_PREVIOUS_AT_MOVEMENT_GRANULARITY

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction
ACTION_PREVIOUS_AT_MOVEMENT_GRANULARITY

以给定移动粒度，请求去到节点文本中的上一个实体的操作。例如，移动到下一个字符、单词等。

参数：

AccessibilityNodeInfo.ACTION_ARGUMENT_MOVEMENT_GRANULARITY_INT，
AccessibilityNodeInfo.ACTION_ARGUMENT_EXTEND_SELECTION_BOOLEAN

样例：移动到下一个字符，并且不会扩大选择。

```
Bundle arguments = new Bundle();
arguments.putInt(AccessibilityNodeInfo.ACTION_ARGUMENT_MOVEMENT_
GRANULARITY_INT,
AccessibilityNodeInfo.MOVEMENT_GRANULARITY_CHARACTER);

arguments.putBoolean(AccessibilityNodeInfo.ACTION_ARGUMENT_EXTEND_
SELECTION_BOOLEAN,false);

info.performAction(AccessibilityAction.ACTION_PREVIOUS_AT_MOVEMENT
_GRANULARITY.getId(),arguments);
```

参见：

AccessibilityNodeInfo.ACTION_ARGUMENT_MOVEMENT_GRANULARITY_INT

AccessibilityNodeInfo.ACTION_ARGUMENT_EXTEND_SELECTION_BOOLEAN

AccessibilityNodeInfo.setMovementGranularities(int)

AccessibilityNodeInfo.getMovementGranularities()

AccessibilityNodeInfo.MOVEMENT_GRANULARITY_CHARACTER

AccessibilityNodeInfo.MOVEMENT_GRANULARITY_WORD

AccessibilityNodeInfo.MOVEMENT_GRANULARITY_LINE

AccessibilityNodeInfo.MOVEMENT_GRANULARITY_PARAGRAPH

AccessibilityNodeInfo.MOVEMENT_GRANULARITY_PAGE

2.18ACTION_PREVIOUS_HTML_ELEMENT

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_PREVIOUS_HTML_ELEMENT

移动到给定类型的上一个 HTML 元素的操作。例如，移动到 BUTTON、INPUT、TABLE 等。

参数：

AccessibilityNodeInfo.ACTION_ARGUMENT_HTML_ELEMENT_STRING

样例：

```
Bundle arguments = new Bundle();
arguments.putString(AccessibilityNodeInfo.ACTION_ARGUMENT_HTML_ELEMENT_STRING, "BUTTON");
info.performAction(AccessibilityAction.ACTION_PREVIOUS_HTML_ELEMENT.getId(), arguments);
```

2.19ACTION_SCROLL_BACKWARD

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_SCROLL_BACKWARD

向后滚动节点内容的操作。

2.20ACTION_SCROLL_DOWN

添加于 API 级别 23。

AccessibilityNodeInfo.AccessibilityAction ACTION_SCROLL_DOWN

向下滚动节点内容的操作。

2.21ACTION_SCROLL_FORWARD

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_SCROLL_FORWARD

向前滚动节点内容的操作。

2.22ACTION_SCROLL_LEFT

添加于 API 级别 23。

AccessibilityNodeInfo.AccessibilityAction ACTION_SCROLL_LEFT

向左滚动节点内容的操作。

2.23ACTION_SCROLL_RIGHT

添加于 API 级别 23。

AccessibilityNodeInfo.AccessibilityAction ACTION_SCROLL_RIGHT

向右滚动节点内容的操作。

2.24ACTION_SCROLL_TO_POSITION

添加于 API 级别 23。

AccessibilityNodeInfo.AccessibilityAction ACTION_SCROLL_TO_POSITION

滚动节点内容让指定集合位置在屏幕上可见的操作。

参数：

ACTION_ARGUMENT_ROW_INT

ACTION_ARGUMENT_COLUMN_INT

参见：getCollectionInfo()

2.25ACTION_SCROLL_UP

添加于 API 级别 23。

AccessibilityNodeInfo.AccessibilityAction ACTION_SCROLL_UP

向上滚动节点内容的操作。

2.26ACTION_SELECT

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_SELECT

选择节点的操作。

2.27ACTION_SET_PROGRESS

添加于 API 级别 24。

AccessibilityNodeInfo.AccessibilityAction ACTION_SET_PROGRESS

设置进度的操作，进度在 RangeInfo.getMin()和 RangeInfo.getMax()之间。它应该使用与 RangeInfo.getType()相同值类型。

参数：ACTION_ARGUMENT_PROGRESS_VALUE

参见：AccessibilityNodeInfo.RangeInfo

2.28ACTION_SET_SELECTION

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_SET_SELECTION

设置选择的操作。没有参数的时候，执行该操作清除选择。

参数：

AccessibilityNodeInfo.ACTION_ARGUMENT_SELECTION_START_INT，
AccessibilityNodeInfo.ACTION_ARGUMENT_SELECTION_END_INT

样例：

```
Bundle arguments = new Bundle();
arguments.putInt(AccessibilityNodeInfo.ACTION_ARGUMENT_SELECTION_START_INT, 1);
arguments.putInt(AccessibilityNodeInfo.ACTION_ARGUMENT_SELECTION_END_INT, 2);
info.performAction(AccessibilityAction.ACTION_SET_SELECTION.getId(), arguments);
```

参见：

AccessibilityNodeInfo.ACTION_ARGUMENT_SELECTION_START_INT

AccessibilityNodeInfo.ACTION_ARGUMENT_SELECTION_END_INT

2.29ACTION_SET_TEXT

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction ACTION_SET_TEXT

设置节点文本的操作。在没有参数的情况下执行该操作，使用 null 或者空 CharSequence 将会清除文本。该操作将会把光标放置在文本末尾。

参数：

AccessibilityNodeInfo.ACTION_ARGUMENT_SET_TEXT_CHARSEQUENCE

样例：

```
Bundle arguments = new Bundle();
arguments.putCharSequence(AccessibilityNodeInfo.ACTION_ARGUMENT_SET_TEXT_CHARSEQUENCE,"android");
info.performAction(AccessibilityAction.ACTION_SET_TEXT.getId(), arguments);
```

2.30ACTION_SHOW_ON_SCREEN

添加于 API 级别 23。

AccessibilityNodeInfo.AccessibilityAction ACTION_SHOW_ON_SCREEN

请求节点让其边界矩形在屏幕上可见的操作，如果有必要可以滚动。

参见：

requestRectangleOnScreen(Rect)

3.公有构造函数

3.1AccessibilityNodeInfo.AccessibilityAction

添加于 API 级别 21。

AccessibilityNodeInfo.AccessibilityAction (int actionId, CharSequence label)

创建一个新的无障碍操作。为了添加一个不具有指定标签的标准操作，使用静态常量。也可以重写一个标准操作的描述。下面是一个样例，讲述怎样通过添加一个自定义标签添加标准点击操作。

```
AccessibilityAction action = new AccessibilityAction(
AccessibilityAction.ACTION_ACTION_CLICK, getLocalizedLabel());

node.addAction(action);
```

参数：

actionId	int：该操作的 id。该操作要么是一个标准操作，或者是应用中的特定操作。在该情况下，需要使用一个资源标识器。
label	CharSequence：新无障碍操作的标签。

4.公有方法

4.1equals

添加于 API 级别 21。

boolean equals (Object object)

标识某些其他对象是否“等同于”该对象。

该 equals 方法在非空对象索引上实现等价关系：

- 自反：对于任何非空索引值 x，x.equals(x)应该返回 true。
- 对称：对于任何非空索引值 x 和 y，当且仅当 y.equals(x)返回 true 时，x.equals(y)应该返回 true。
- 传递：对于任何非空索引值 x、y 和 z，如果 x.equals(y)返回 true 且 y.equals(z)返回 true，则 x.equals(z)应该返回 true。
- 一致：对于任何非空索引值 x 和 y，x.equals(y)的多次调用一致返回 true，或一致返回 false，在已修改对象的 equals 比较中，未提供任何信息。
- 对于任何非空索引值 x，x.equals(null)应该返回 false。

Object 类的 equals 方法在对象上实现了最可能区分的等价关系；因此，对于任何非空索引值 x 和 y，当且只有当 x 和 y 指向同一对象的时候，该方法返回 true（x==y 具有相同的值），返回 true。

注意，一般来说，无论该方法何时被重写，必须重写 hashCode 方法，保持 hashCode 方法的总合约，这说明等价对象必须有等价哈希编码。

参数：

Object	Object: 要比较的索引对象。

返回值：

boolean	如果该对象于 obj 参数相同，返回 true，否则，返回 false。

4.2getId

添加于 API 级别 21。

int getId ()

获取该操作的 id。

返回值：

int	操作 id。

4.3getLabel

添加于 API 级别 21。

CharSequence getLabel ()

获取该操作的标签。该方法的目的是给用户描述该操作。

返回值：

CharSequence	标签。

4.4hashCode

添加于 API 级别 21。

int hashCode ()

返回对象的哈希编码值。该方法支持哈希表格的优点，例如 HashMap 提供的用途。

hashCode 的一般效用：

- 在一个 JAVA 应用执行过程中，无论何时在同一对象上被激活超过一次，hashCode 方法必须一致性地返回相同整数，该整数在 equals 比较中不提供任何关于被修改对象的信息。从一个应用的一次执行到相同应用的另一次执行，该整数不需要保持一致。
- 如果根据 equals(Object)方法，两个对象相同，然后在两个对象上分别调用 hashCode 方法，一定会生成相同的整数结果。
- 如果依据 equals(java.lang.Object)方法，两个对象不相同，不需要在两个对象上分别调用 hashCode 方法生成不同的整数结果。但是，开发者应该清楚知道，为不同对象生成不同整数结果会提升哈希表格的表现。

作为一个合理的实践，被 Object 对象定义的 hashCode 方法，为不同对象返回不同的整数。（这通常通过将转换对象外部地址为一个整体实现，但是该实现技术不是 Java™ 编程语言必须的。）

返回值：

int	该对象的哈希代码值。

4.5toString

添加于 API 级别 21。

String toString ()

返回一个代表对象的字符串。一般来说，toString 方法返回一个字符串，以文本表示该对象。结果应该是个精简翔实的表示，且容易被用户阅读。推荐所有子类重写该方法。

Object 类的 toString 方法返回一个包含实例对象类名的字符串，包含字符“@”和对象哈希码的十六进制表示。换句话说，该方法返回的字符串等于下值：

getClass().getName() + '@' + Integer.toHexString(hashCode())

返回值：

String	对象的一个字符串表示。