

**朗国批注手笔分离api使用说明**

**版权所有**

**广州朗国电子科技有限公司 2017。保留一切权利。**

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

**注意**

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

**Rights Reserved**

Guangzhou LANGO Electronic Technology Co., Ltd. (LANGO) keeps all rights **RESERVED**.

Without a written permission, no unit and individual can copy the contents of this document in part or whole, or transmit it in any form.

**NOTE**

This document may be updated irregularly for upgrade version or other reasons. Unless some additional agreements, there is no guarantee for any explicitly and implicitly made by statements, information or suggestions in this document.

**文档对象**

本文档主要适用于：

FAE，RD，APP开发小组成员

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修订日期 | 修订人 | 邮箱 | 文档版本 | 修订说明 |
| 20240810 | 吴志豪 | [Zhihao.wu@lango-tech.cn](mailto:Zhihao.wu@lango-tech.cn) | 1.0.0 | API 首次修订 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[一、 方案描述 6](#_Toc26309)

[二、 接口说明： 8](#_Toc11951)

[三、 使用注意 10](#_Toc10185)

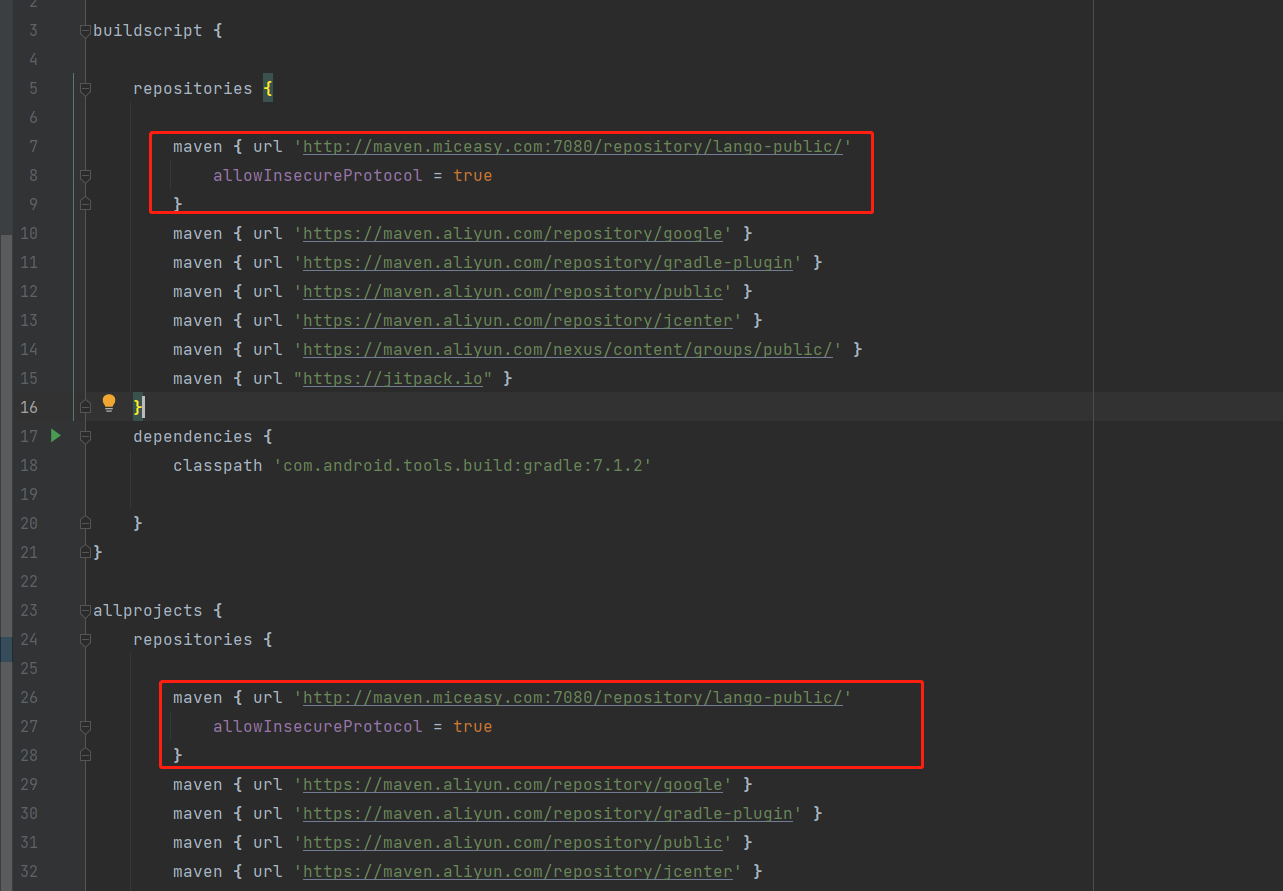
# **方案描述**

在批注中，如要实现手笔分离的效果，如使用笔书写，手触摸UI。这样用户就可以不用退出批注，而不影响正常的UI交互。在常规的app应用是无法实现这种效果，需要与系统相配合，本方案提供了一些api。供批注接入使用，可以控制触摸事件 如何的分发。

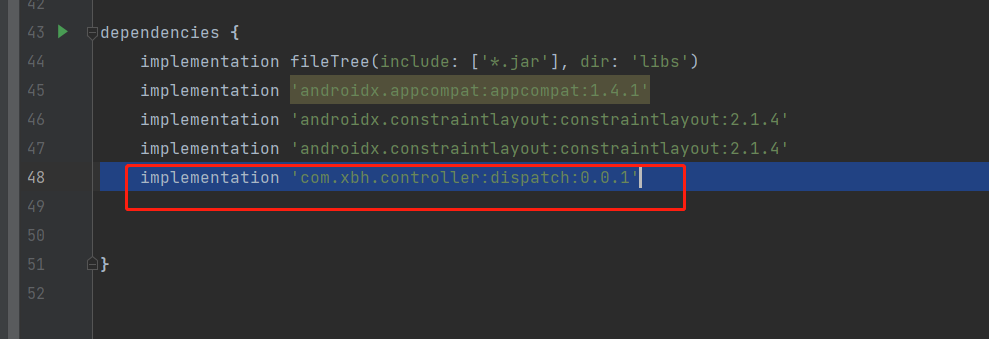
## 

本方案适用于批注如示例所示，包含全屏透明可书写区域，由surfaceView或其它view实现。包含 批注的工具栏区域。

在项目根目录build.gradle ，添加朗国公开maven仓库



在具体的module中build.gradle 导入dispatch 工具库。



Sync 之后即可。

最新api版本可通过：<http://maven.miceasy.com:7080/#browse/browse:lango-public:com%2Fxbh%2Fcontroller%2Fdispatch> 访问查询。

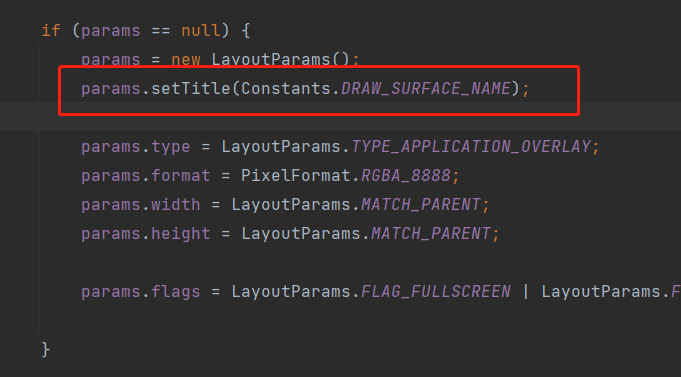
# **接口说明：**

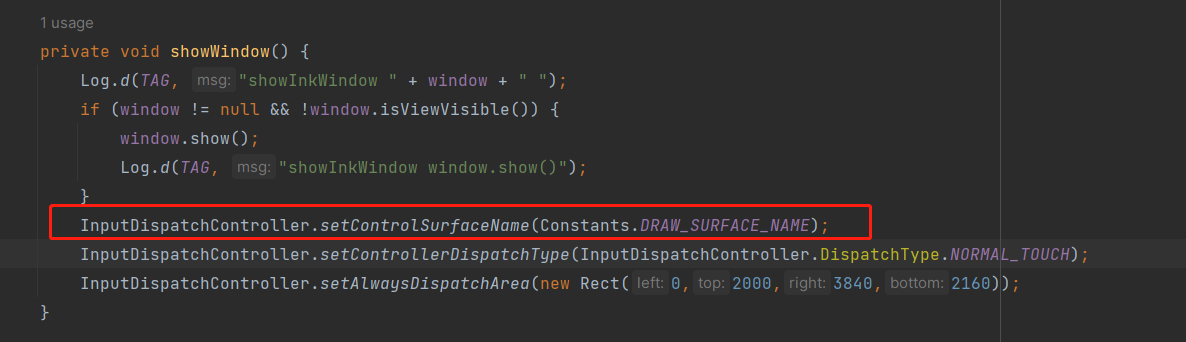
接口包路径为：com.xbh.dispatch.InputDispatchController

|  |  |
| --- | --- |
| 函数名 | 备注 |
| setControlSurfaceName(String name) | Set 要控制的surface name |
| setControllerDispatchType(int dispatchType) | Set 触摸事件 分发类型 |
| setAlwaysDispatchArea(Rect rect) | Set 工具栏区域，工具栏区域的点击事件，不受dispatch type控制。 |
| Clear() | 清除属改值，回到默认状态. |

1. setControlSurfaceName(String name)

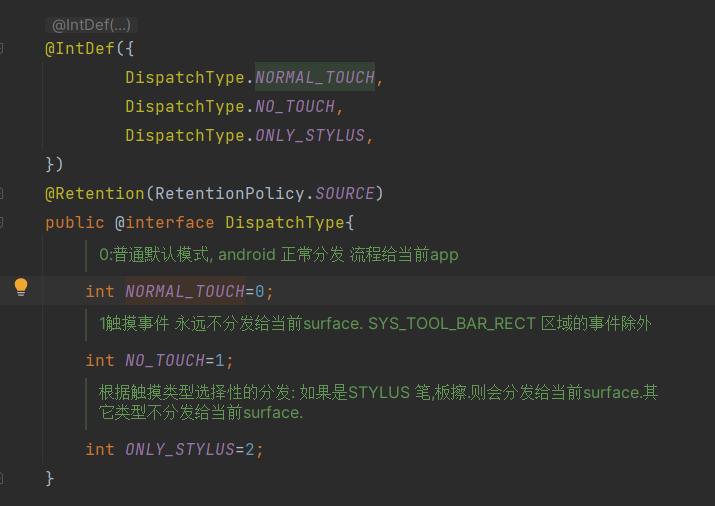
Set 要控制的surface name，因为在framework中，应用对应于一个surface。包含surface name，宽高等属性。所以这个方法就是告诉系统，要对哪个surface name进行特殊的事件分发处理。一般默认的surface name是应用的包名，或activity 名称。可以通过set title修改surfaceName。如通过windowManager 添加view 时，可以setTitle。也就是对应surface name。请保持这两个值一致。





setControllerDispatchType（int dispatchType)

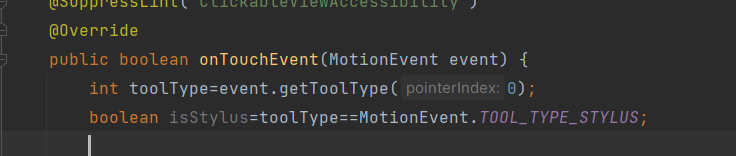
Set 分发的策略类型，有三个值：



NORMAL\_TOUCH：触摸事件正常分发给当前应用，可在当前模式正常书写等.

NO\_TOUCH : 触摸事件不分发给当前应用。在默认情况下，android会把触摸事件分发给最顶层的surface window。当set NO\_TOUCH 后。即使当前 应用处理最顶层，也不会接收到触摸事件，事件会穿透下去，给下一层的应用处理。

ONLY\_STYLUS：手笔分离，当set ONLY\_STYLUS后，只有类型为 STYLUS 的触摸，（stylus 包含笔与板擦）才会分发给当前应用。其它类型，则分发给下一层应用。手笔分离需要触摸框支持区分 书写笔与手指。可通过Motionevnt.getToolType(0) 获取触摸类型验证.



setAlwaysDispatchArea(Rect r);

如上所述，当setControllerDispatchType 为NO\_TOUCH 与 ONLY\_STYLUS，部分触摸会穿透分发到下一层应用上，而批注一般有工具栏区域，对工具栏区域的触摸，是不需要区分触摸类型的，应该按照默认的android 分发机制。所以在 setDispatchType 为NO\_TOUCH 或ONLY\_STYLUS，设置一个白名单区域。这个区域的点击的事件，会分发给当前应用，可保证 工具栏按钮等正常的响应。

# **使用注意**

1、由于setXxxx方法是写入到propert属性值。当多个应用切换时，propert 值可能会改变。所以当批注应用可见 show时，应重新调用一次所有的setXxxx方法。以保证set为当前应用的值。

2、当在 NO\_TOUCH 、ONLY\_STYLUS 类型时，如果批注的工具栏区域，如大小，位置发生改变，需及时更新setAlwaysDispatchArea，不然此区域的事件不会分发给当前应用，导致工具栏不能响应点击等。