公共事件

1 何为公共事件

公共事件是由鸿蒙操作系统的公共事件服务(Common Event Service, CES)提供的。CES可以为应用程序提供订阅、退订和发布公共事件的能力。

系统：电量不足 -> CES -> 应用 -> 省电模式
系统：WIFI断开 -> CES -> 应用 -> 流量提示

另外，应用程序之间也可以相互发送公共事件。

2 公共事件的类型

根据发布者的不同，公共事件分为由操作系统发布的系统公共事件，和由应用程序发布的自定义公共事件。

根据使用方式的不同，公共事件分为：

- 普通公共事件

```
        接受者
       /
发布者 - 接受者
       \
        接受者
```

- 有序公共事件

```
发布者 - 接受者 - 接受者 x 接受者
```

- 粘性公共事件
  公共事件在发布后会驻留一段时间，即使在其被发布后订阅该事件的应用也能接收到该事件。

```
                 接受者
                /
发布者 - 驻留区 - 接受者
                \
                 接受者
```

3 订阅系统公共事件

CommonEventSupport类中包含了系统公共事件字符串：

```
COMMON_EVENT_AIRPLANE_MODE_CHANGED - 飞行模式改变
COMMON_EVENT_BATTERY_CHANGED       - 电池状态改变
COMMON_EVENT_BATTERY_LOW           - 低电量警报
COMMON_EVENT_BATTERY_OKAY          - 低电量警报解除
COMMON_EVENT_CHARGING              - 开始充电
COMMON_EVENT_DISCHARGING           - 停止充电
COMMON_EVENT_BOOT_COMPLETED        - 系统启动完毕
COMMON_EVENT_DATE_CHANGED          - 日期改变
COMMON_EVENT_CONFIGURATION_CHANGED - 系统配置改变
COMMON_EVENT_DRIVE_MODE            - 进入驾驶模式
COMMON_EVENT_HOME_MODE             - 进入家庭模式
COMMON_EVENT_OFFICE_MODE           - 进入办公室模式
COMMON_EVENT_HWID_LOGIN            - 登录华为账号
COMMON_EVENT_HWID_LOGOUT           - 退出华为账号
COMMON_EVENT_LOCALE_CHANGED        - 语言区域改变
COMMON_EVENT_SCREEN_OFF            - 熄屏
COMMON_EVENT_SCREEN_ON             - 亮屏
COMMON_EVENT_WIFI_CONN_STATE       - WiFi状态改变
...
```

```
MatchingSkills(需要订阅的公共事件)
               |
               v
CommonEventSubscribeInfo(订阅参数)
               |
               v
CommonEventSubscriber(公共事件订阅者)
               |
     subcribeCommonEvent(订阅)
   unsubcribeCommonEvent(退订)
               v
CommonEventManager(公共事件管理器)
```

公共事件：

```
MatchingSkills events = new MatchingSkills();
events.addEvent(CommonEventSupport.COMMON_EVENT_...);
```

订阅参数：

```
CommonEventSubscribeInfo info = new CommonEventSubscribeInfo(events);
```

订阅者：

```
CommonEventSubscriber subscriber = new CommonEventSubscriber(info) {
    @Override
    public void onReceiveEvent(CommonEventData data) {
        // 处理事件
    }
};
```

订阅:

```
CommonEventManager.subscribeCommonEvent(subscriber);
```

退订:

```
CommonEventManager.unsubscribeCommonEvent(subscriber);
```

> 例程：SystemEvent

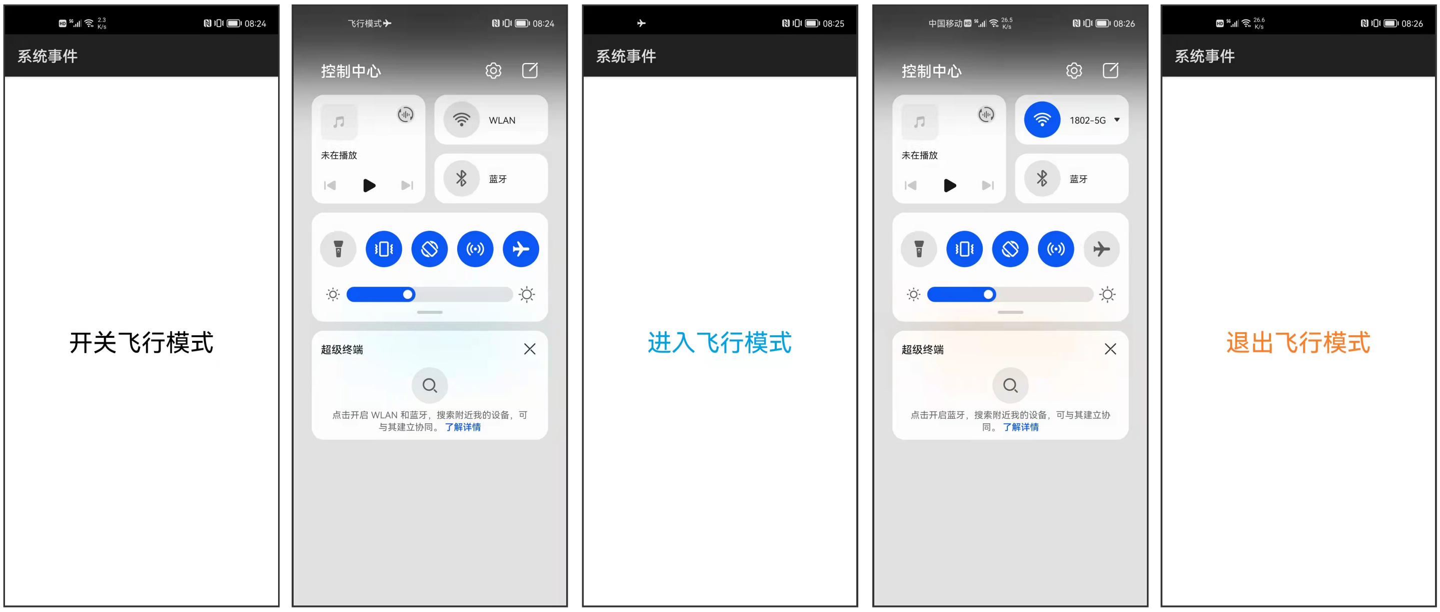
...\SystemEvent\entry\src\main\resources\base\layout\ability_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<DirectionalLayout
    xmlns:ohos="http://schemas.huawei.com/res/ohos"
    ohos:height="match_parent"
    ohos:width="match_parent"
    ohos:alignment="center"
    ohos:orientation="vertical">

    <Text
        ohos:id="$+id:txt"
        ohos:height="match_content"
        ohos:width="match_content"
        ohos:text="开关飞行模式"
        ohos:text_size="32fp"
        />

</DirectionalLayout>
```

...\SystemEvent\entry\src\main\java\com\minwei\systemevent\slice\MainAbilitySlice.java

```
public class MainAbilitySlice extends AbilitySlice {
    private static final HiLogLabel label = new HiLogLabel(
        HiLog.LOG_APP, 0x00101,
        MainAbilitySlice.class.getCanonicalName());

    private CommonEventSubscriber subscriber;

    @Override
    public void onStart(Intent intent) {
        ...
        MatchingSkills events = new MatchingSkills();
        events.addEvent(
            CommonEventSupport.COMMON_EVENT_AIRPLANE_MODE_CHANGED);

        CommonEventSubscribeInfo info =
            new CommonEventSubscribeInfo(events);

        subscriber = new CommonEventSubscriber(info) {
            @Override
            public void onReceiveEvent(CommonEventData data) {
                Text text = (Text)findComponentById(
                    ResourceTable.Id_txt);

                if (data.getIntent().getBooleanParam("state", true)) {
                    text.setText("进入飞行模式");
                    text.setTextColor(new Color(0xff00a2e8));
                }
                else {
                    text.setText("退出飞行模式");
                    text.setTextColor(new Color(0xffff7f27));
                }
            }
        };

        try {
            CommonEventManager.subscribeCommonEvent(subscriber);
        }
        catch (RemoteException exception) {
            HiLog.error(label, exception.getLocalizedMessage());
        }
    }
    ...
    @Override
    protected void onStop() {
        super.onStop();

        try {
            CommonEventManager.unsubscribeCommonEvent(subscriber);
        }
        catch (RemoteException exception) {
            HiLog.error(label, exception.getLocalizedMessage());
        }
    }
}
```

运行效果如下图所示：



4 自定义公共事件

4.1 发布公共事件

```
CommonEventManager.publishCommonEvent(
    CommonEventData         data,      // 公共事件数据
                                       // 通过Intent携带公共事件字符串
    CommonEventPublishInfo info,       // 公共事件发布信息
                                       // 公共事件的类型和订阅者所需权限
    CommonEventSubscriber  subscriber // 公共事件订阅者处理情况
);
```

4.1.1 发布普通公共事件

```
Intent intent = new Intent();
Operation operation = new OperationBuilder()
    .withAction("NORMAL_COMMON_EVENT")
    .build();
intent.setOperation(operation);

CommonEventData data = new CommonEventData(intent);

try {
    CommonEventManager.publishCommonEvent(data);
}
catch (RemoteException exception) {
    HiLog.error(label, exception.getLocalizedMessage());
}
```

4.1.2 发布有序公共事件

```
Intent intent = new Intent();
Operation operation = new OperationBuilder()
    .withAction("ORDERED_COMMON_EVENT")
    .build();
intent.setOperation(operation);

CommonEventData data = new CommonEventData(intent);
data.setCode(1);
data.setData("发布者->");

CommonEventPublishInfo info = new CommonEventPublishInfo();
info.setOrdered(true);

CommonEventSubscriber subscriber = createSubscriber();

try {
    CommonEventManager.publishCommonEvent(data, info, subscriber);
}
catch (RemoteException exception) {
    HiLog.error(label, exception.getLocalizedMessage());
}

private CommonEventSubscriber createSubscriber() {
    MatchingSkills events = new MatchingSkills();

    CommonEventSubscribeInfo info =
        new CommonEventSubscribeInfo(events);

    return new CommonEventSubscriber(info) {
        @Override
        public void onReceiveEvent(CommonEventData data) {
            new ToastDialog(getContext())
                .setText("[" + getCode() + "] " +
                getData() + "我")
                .setDuration(5000)
                .setAlignment(LayoutAlignment.CENTER)
                .setOffset(0, 256)
                .show();
        }
    };
}
```

4.1.3 发布粘性公共事件

```
Intent intent = new Intent();
Operation operation = new OperationBuilder()
    .withAction("STICKY_COMMON_EVENT")
    .build();
intent.setOperation(operation);

CommonEventData data = new CommonEventData(intent);

CommonEventPublishInfo info = new CommonEventPublishInfo();
info.setSticky(true);

try {
    CommonEventManager.publishCommonEvent(data, info);
}
catch (RemoteException exception) {
    HiLog.error(label, exception.getLocalizedMessage());
}
```

config.json

```
"reqPermissions": [
  {
    "name": "ohos.permission.COMMONEVENT_STICKY"
  }
]
```

4.2 订阅公共事件

```
MatchingSkills events = new MatchingSkills();
events.addEvent("NORMAL_COMMON_EVENT");
events.addEvent("ORDERED_COMMON_EVENT");
events.addEvent("STICKY_COMMON_EVENT");

CommonEventSubscribeInfo info =
    new CommonEventSubscribeInfo(events);
info.setPriority(1);

subscriber = new CommonEventSubscriber(info) {
    @Override
    public void onReceiveEvent(CommonEventData data) {
        Text text =
            (Text)findComponentById(ResourceTable.Id_txt);
        text.append(data.getIntent().getAction() + "\n");

        if (data.getIntent().getAction() ==
            "ORDERED_COMMON_EVENT") {
            text.append("[" + data.getCode() + "] " +
                data.getData() + "我\n");

            setCode(2);
            setData(data.getData() + "订阅者->");
        }
    }
};

try {
    CommonEventManager.subscribeCommonEvent(subscriber);
}
catch (RemoteException exception) {
    HiLog.error(label, exception.getLocalizedMessage());
}
```

4.3 订阅并阻断公共事件

```
MatchingSkills events = new MatchingSkills();
events.addEvent("NORMAL_COMMON_EVENT");
events.addEvent("ORDERED_COMMON_EVENT");
events.addEvent("STICKY_COMMON_EVENT");

CommonEventSubscribeInfo info =
    new CommonEventSubscribeInfo(events);

subscriber = new CommonEventSubscriber(info) {
    @Override
    public void onReceiveEvent(CommonEventData data) {
        Text text =
            (Text)findComponentById(ResourceTable.Id_txt);
        text.append(data.getIntent().getAction() + "\n");

        if (data.getIntent().getAction() ==
            "ORDERED_COMMON_EVENT") {
            text.append("[" + data.getCode() + "] " +
                data.getData() + "我\n");

            setCode(3);
            setData(data.getData() + "终结者->");

            abortCommonEvent();
        }
    }
};

try {
    CommonEventManager.subscribeCommonEvent(subscriber);
}
catch (RemoteException exception) {
    HiLog.error(label, exception.getLocalizedMessage());
}
```

4.4 退订公共事件

```
try {
    CommonEventManager.unsubscribeCommonEvent(subscriber);
}
catch (RemoteException exception) {
    HiLog.error(label, exception.getLocalizedMessage());
}
```

> 例程：Publisher

...\Publisher\entry\src\main\config.json

```
{
  ...
  "module": {
    ...
    "reqPermissions": [
      {
        "name": "ohos.permission.COMMONEVENT_STICKY"
      }
    ],
    ...
  }
}
```

...\Publisher\entry\src\main\resources\base\graphic\background_button_normal.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:ohos="http://schemas.huawei.com/res/ohos"
    ohos:shape="rectangle">
    <corners ohos:radius="100"/>
    <solid ohos:color="#00a2e8"/>
</shape>
```

...\Publisher\entry\src\main\resources\base\graphic\background_button_ordered.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:ohos="http://schemas.huawei.com/res/ohos"
    ohos:shape="rectangle">
    <corners ohos:radius="100"/>
    <solid ohos:color="#22b14c"/>
</shape>
```

...\Publisher\entry\src\main\resources\base\graphic\background_button_sticky.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:ohos="http://schemas.huawei.com/res/ohos"
    ohos:shape="rectangle">
    <corners ohos:radius="100"/>
    <solid ohos:color="#ff7f27"/>
</shape>
```

...\Publisher\entry\src\main\resources\base\layout\ability_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<DirectionalLayout
    xmlns:ohos="http://schemas.huawei.com/res/ohos"
    ohos:height="match_parent"
    ohos:width="match_parent"
    ohos:alignment="center"
    ohos:orientation="vertical">

    <Button
        ohos:id="$+id:btnNormal"
        ohos:height="match_content"
        ohos:width="match_content"
        ohos:left_padding="30vp"
        ohos:top_padding="10vp"
        ohos:right_padding="30vp"
        ohos:bottom_padding="10vp"
        ohos:background_element="$graphic:background_button_normal"
        ohos:text="普通公共事件"
        ohos:text_size="28fp"
        ohos:text_color="#ffffff"
        />

    <Button
        ohos:id="$+id:btnOrdered"
        ohos:height="match_content"
        ohos:width="match_content"
        ohos:left_padding="30vp"
        ohos:top_padding="10vp"
        ohos:right_padding="30vp"
        ohos:bottom_padding="10vp"
        ohos:top_margin="60vp"
        ohos:background_element="$graphic:background_button_ordered"
        ohos:text="有序公共事件"
        ohos:text_size="28fp"
        ohos:text_color="#ffffff"
        />

    <Button
        ohos:id="$+id:btnSticky"
        ohos:height="match_content"
        ohos:width="match_content"
        ohos:left_padding="30vp"
        ohos:top_padding="10vp"
        ohos:right_padding="30vp"
        ohos:bottom_padding="10vp"
        ohos:top_margin="60vp"
        ohos:background_element="$graphic:background_button_sticky"
        ohos:text="粘性公共事件"
        ohos:text_size="28fp"
        ohos:text_color="#ffffff"
        />

</DirectionalLayout>
```

```
public class MainAbilitySlice extends AbilitySlice {
    private static final HiLogLabel label = new HiLogLabel(
        HiLog.LOG_APP, 0x00101,
        MainAbilitySlice.class.getCanonicalName());

    @Override
    public void onStart(Intent intent) {
        ...
        findComponentById(ResourceTable.Id_btnNormal)
            .setClickedListener(component -> publishNormalCommonEvent());
        findComponentById(ResourceTable.Id_btnOrdered)
            .setClickedListener(component -> publishOrderedCommonEvent());
        findComponentById(ResourceTable.Id_btnSticky)
            .setClickedListener(component -> publishStickyCommonEvent());
    }
    ...
    private void publishNormalCommonEvent() {
        Intent intent = new Intent();
        Operation operation = new OperationBuilder()
            .withAction("NORMAL_COMMON_EVENT")
            .build();
        intent.setOperation(operation);

        CommonEventData data = new CommonEventData(intent);

        try {
            CommonEventManager.publishCommonEvent(data);
        }
        catch (RemoteException exception) {
            HiLog.error(label, exception.getLocalizedMessage());
        }
    }

    private void publishOrderedCommonEvent() {
        Intent intent = new Intent();
        Operation operation = new OperationBuilder()
            .withAction("ORDERED_COMMON_EVENT")
            .build();
        intent.setOperation(operation);

        CommonEventData data = new CommonEventData(intent);
        data.setCode(1);
        data.setData("发布者->");

        CommonEventPublishInfo info = new CommonEventPublishInfo();
        info.setOrdered(true);

        CommonEventSubscriber subscriber = createSubscriber();

        try {
            CommonEventManager.publishCommonEvent(data, info, subscriber);
        }
        catch (RemoteException exception) {
            HiLog.error(label, exception.getLocalizedMessage());
        }
    }

    private void publishStickyCommonEvent() {
        Intent intent = new Intent();
        Operation operation = new OperationBuilder()
            .withAction("STICKY_COMMON_EVENT")
            .build();
        intent.setOperation(operation);

        CommonEventData data = new CommonEventData(intent);

        CommonEventPublishInfo info = new CommonEventPublishInfo();
        info.setSticky(true);

        try {
            CommonEventManager.publishCommonEvent(data, info);
        }
        catch (RemoteException exception) {
            HiLog.error(label, exception.getLocalizedMessage());
        }
    }

    private CommonEventSubscriber createSubscriber() {
        MatchingSkills events = new MatchingSkills();

        CommonEventSubscribeInfo info =
            new CommonEventSubscribeInfo(events);

        return new CommonEventSubscriber(info) {
            @Override
            public void onReceiveEvent(CommonEventData data) {
                new ToastDialog(getContext())
                    .setText("[" + getCode() + "] " +
                    getData() + "我")
                    .setDuration(5000)
                    .setAlignment(LayoutAlignment.CENTER)
                    .setOffset(0, 256)
                    .show();
            }
        };
    }
}
```

> 例程：Subscriber

...\Subscriber\entry\src\main\resources\base\layout\ability_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<DirectionalLayout
    xmlns:ohos="http://schemas.huawei.com/res/ohos"
    ohos:height="match_parent"
    ohos:width="match_parent">

    <ScrollView
        ohos:height="match_parent"
        ohos:width="match_parent"
        ohos:rebound_effect="true">

        <Text
            ohos:id="$+id:txt"
            ohos:height="match_content"
            ohos:width="match_parent"
            ohos:multiple_lines="true"
            ohos:text_size="20fp"
            ohos:text_alignment="horizontal_center"
            />

    </ScrollView>

</DirectionalLayout>
```

...\Subscriber\entry\src\main\java\com\minwei\subscriber\slice\MainAbilitySlice.java

```
public class MainAbilitySlice extends AbilitySlice {
    private static final HiLogLabel label = new HiLogLabel(
        HiLog.LOG_APP, 0x00101,
        MainAbilitySlice.class.getCanonicalName());

    private CommonEventSubscriber subscriber;

    @Override
    public void onStart(Intent intent) {
        ...
        MatchingSkills events = new MatchingSkills();
        events.addEvent("NORMAL_COMMON_EVENT");
        events.addEvent("ORDERED_COMMON_EVENT");
        events.addEvent("STICKY_COMMON_EVENT");

        CommonEventSubscribeInfo info =
            new CommonEventSubscribeInfo(events);
        info.setPriority(1);

        subscriber = new CommonEventSubscriber(info) {
            @Override
            public void onReceiveEvent(CommonEventData data) {
                Text text =
                    (Text)findComponentById(ResourceTable.Id_txt);
                text.append(data.getIntent().getAction() + "\n");

                if (data.getIntent().getAction() ==
                    "ORDERED_COMMON_EVENT") {
                    text.append("[" + data.getCode() + "] " +
                        data.getData() + "我\n");

                    setCode(2);
                    setData(data.getData() + "订阅者->");
                }
            }
        };

        try {
            CommonEventManager.subscribeCommonEvent(subscriber);
        }
        catch (RemoteException exception) {
            HiLog.error(label, exception.getLocalizedMessage());
        }
    }
    ...
    @Override
    protected void onStop() {
        super.onStop();

        try {
            CommonEventManager.unsubscribeCommonEvent(subscriber);
        }
        catch (RemoteException exception) {
            HiLog.error(label, exception.getLocalizedMessage());
        }
    }
}
```

> 例程：Terminator

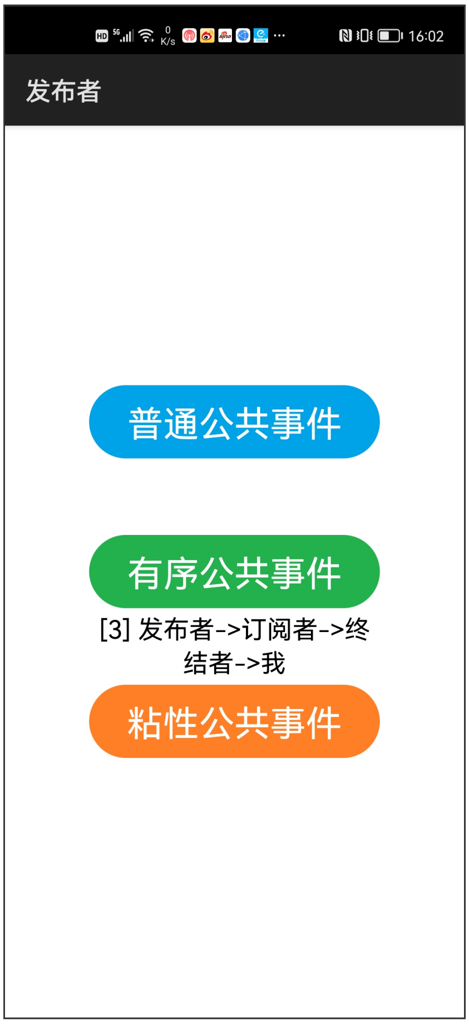
...\Terminator\entry\src\main\resources\base\layout\ability_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<DirectionalLayout
    xmlns:ohos="http://schemas.huawei.com/res/ohos"
    ohos:height="match_parent"
    ohos:width="match_parent">

    <ScrollView
        ohos:height="match_parent"
        ohos:width="match_parent"
        ohos:rebound_effect="true">

        <Text
            ohos:id="$+id:txt"
            ohos:height="match_content"
            ohos:width="match_parent"
            ohos:multiple_lines="true"
            ohos:text_size="20fp"
            ohos:text_alignment="horizontal_center"
            />

    </ScrollView>

</DirectionalLayout>
```

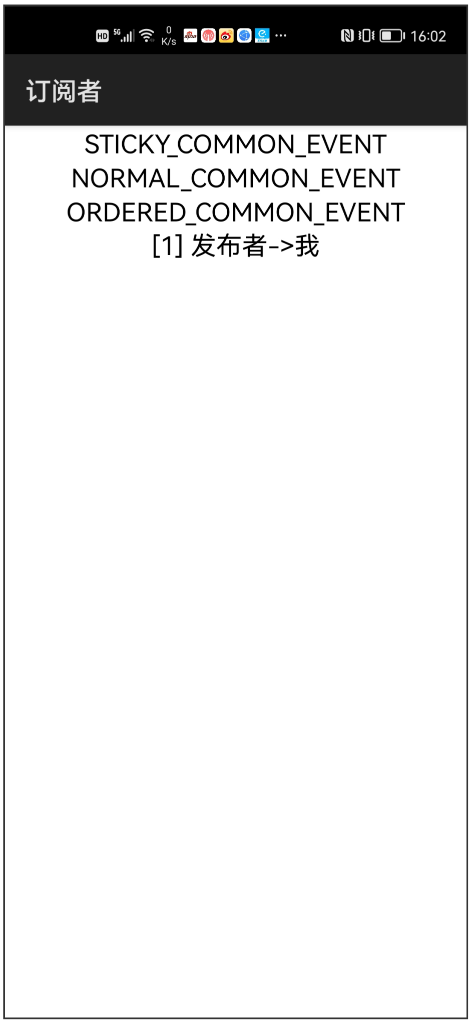
...\Terminator\entry\src\main\java\com\minwei\terminator\slice\MainAbilitySlice.java

```
public class MainAbilitySlice extends AbilitySlice {
    private static final HiLogLabel label = new HiLogLabel(
        HiLog.LOG_APP, 0x00101,
        MainAbilitySlice.class.getCanonicalName());

    private CommonEventSubscriber subscriber;

    @Override
    public void onStart(Intent intent) {
        ...
        MatchingSkills events = new MatchingSkills();
        events.addEvent("NORMAL_COMMON_EVENT");
        events.addEvent("ORDERED_COMMON_EVENT");
        events.addEvent("STICKY_COMMON_EVENT");

        CommonEventSubscribeInfo info =
            new CommonEventSubscribeInfo(events);

        subscriber = new CommonEventSubscriber(info) {
            @Override
            public void onReceiveEvent(CommonEventData data) {
                Text text =
                    (Text)findComponentById(ResourceTable.Id_txt);
                text.append(data.getIntent().getAction() + "\n");

                if (data.getIntent().getAction() ==
                    "ORDERED_COMMON_EVENT") {
                    text.append("[" + data.getCode() + "] " +
                        data.getData() + "我\n");

                    setCode(3);
                    setData(data.getData() + "终结者->");

                    abortCommonEvent();
                }
            }
        };

        try {
            CommonEventManager.subscribeCommonEvent(subscriber);
        }
        catch (RemoteException exception) {
            HiLog.error(label, exception.getLocalizedMessage());
        }
    }
    ...
    @Override
    protected void onStop() {
        super.onStop();

        try {
            CommonEventManager.unsubscribeCommonEvent(subscriber);
        }
        catch (RemoteException exception) {
                HiLog.error(label, exception.getLocalizedMessage());
            }
        }
    }
```

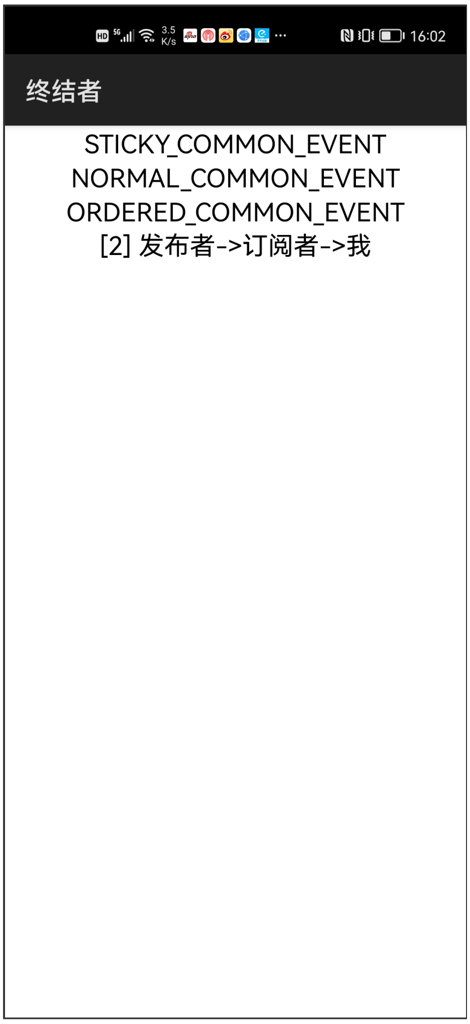
运行效果如下图所示：



[3] 发布者->订阅者->终
结者->我



STICKY_COMMON_EVENT
NORMAL_COMMON_EVENT
ORDERED_COMMON_EVENT
[1] 发布者->我



STICKY_COMMON_EVENT
NORMAL_COMMON_EVENT
ORDERED_COMMON_EVENT
[2] 发布者->订阅者->我

更多精彩，敬请期待……

达内集团C++教学部
2021年9月26日