

# Блокировка и разблокировка кнопок в Android приложении

Многие операции при публикации видео, например, установка Websocket соединения, а затем собственно публикация, занимают определенное время. На это время, до перехода устройства в очередное стабильное состояние, желательно блокировать такие действия пользователя, как повторные нажатия на кнопки в приложении.

Для этого создайте в приложении методы `muteButton` для запрета нажатия на все используемые кнопки, например

code

```
private void muteButton() {
    mStartButton.setEnabled(false);
    mTestButton.setEnabled(false);
    mSwitchCameraButton.setEnabled(false);
    mSwitchFlashlightButton.setEnabled(false);
    mSwitchRenderButton.setEnabled(false);
}
```

и `unmuteButton`, для разрешения нажатия на кнопки

code

```
private void unmuteButton() {
    mStartButton.setEnabled(true);
    mSwitchCameraButton.setEnabled(true);
    mSwitchRenderButton.setEnabled(true);
    if (mSendVideo.isChecked()) {
        mSwitchCameraButton.setEnabled(true);
    }
    if (Flashphoner.isFlashlightSupport()) {
        mSwitchFlashlightButton.setEnabled(true);
    }
}
```

Затем эти методы следует вызывать:

- `muteButton()` - в начале операции
- `unmuteButton()` - по окончании операции

Например, при переключении камеры `muteButton` должен быть вызван после нажатия пользователем кнопки `Switch Camera`, а `unmuteButton` после того, как переключение камеры завершилось успехом или ошибкой

code

```
mSwitchCameraButton.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View view) {
        if (publishStream != null) {
            turnOffFlashlight();
            muteButton();
            publishStream.switchCamera(new CameraSwitchHandler() {
                @Override
                public void onCameraSwitchDone(boolean var1) {
                    runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            if (mStartButton.getTag() == null ||
                                Integer.valueOf(R.string.action_stop).equals(mStartButton.getTag())) {
                                unmuteButton();
                            }
                        }
                    });
                }
            });
        }

        @Override
        public void onCameraSwitchError(String var1) {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    unmuteButton();
                }
            });
        }
    }
});
```