

# SurfaceViewRenderer management in Android application

- [Color and transparency modification](#)
- [Last frame clearing](#)

## Color and transparency modification

By default, SurfaceViewRenderer for displaying video stream has black color, but it can be made transparent if necessary. To do this, the following modules should be imported:

```
import android.graphics.PixelFormat;
```

and setZOrderOnTop() and setFormat() methods should be used:

```
localRender = (SurfaceViewRenderer) findViewById(R.id.local_video_view);
localRender.setZOrderOnTop(true);
localRender.getHolder().setFormat(PixelFormat.TRANSPARENT);
```

To set SurfaceViewRenderer color, setBackgroundColor() method should be used:

```
remoteRender = (SurfaceViewRenderer) findViewById(R.id.remote_video_view);
remoteRender.setZOrderOnTop(true);
remoteRender.getHolder().setFormat(PixelFormat.TRANSLUCENT);
remoteRender.setBackgroundColor(0xFFFF8800); //orange in RGB
```

## Last frame clearing

When stream playback is finished, last frame still displayed on SurfaceViewRenderer. This frame can be cleared if necessary, then surface will be filled black. To do this, the following modules should be imported:

```
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.PixelFormat;
import android.graphics.PorterDuff;
```

and release() method should be invoked:

```
playStream.stop();
playStream = null;
remoteRender.release();
```

After this method invocation no further operation with surface is possible.

If this does not work, or surface is supposed to use repeatedly, SurfaceViewRenderer can be simply hidden. For example, when "Play" button is pressed, make remoteRender visible

```
mPlayButton.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View view) {
        mPlayButton.setEnabled(false);
        if (mPlayButton.getTag() == null || Integer.valueOf(R.string.action_play).equals(mPlayButton.
getTag())) {
            remoteRender.setVisibility(View.VISIBLE);
            ...
            playStream = session.createStream(streamOptions);
            ...
        }
    }
});
```

And when Stop button is pressed, stop stream playback and make remoteRender invisible

```
        ...
    } else {
        playStream.stop();
        playStream = null;
        remoteRender.setVisibility(View.GONE);
    }
    ...
}
```