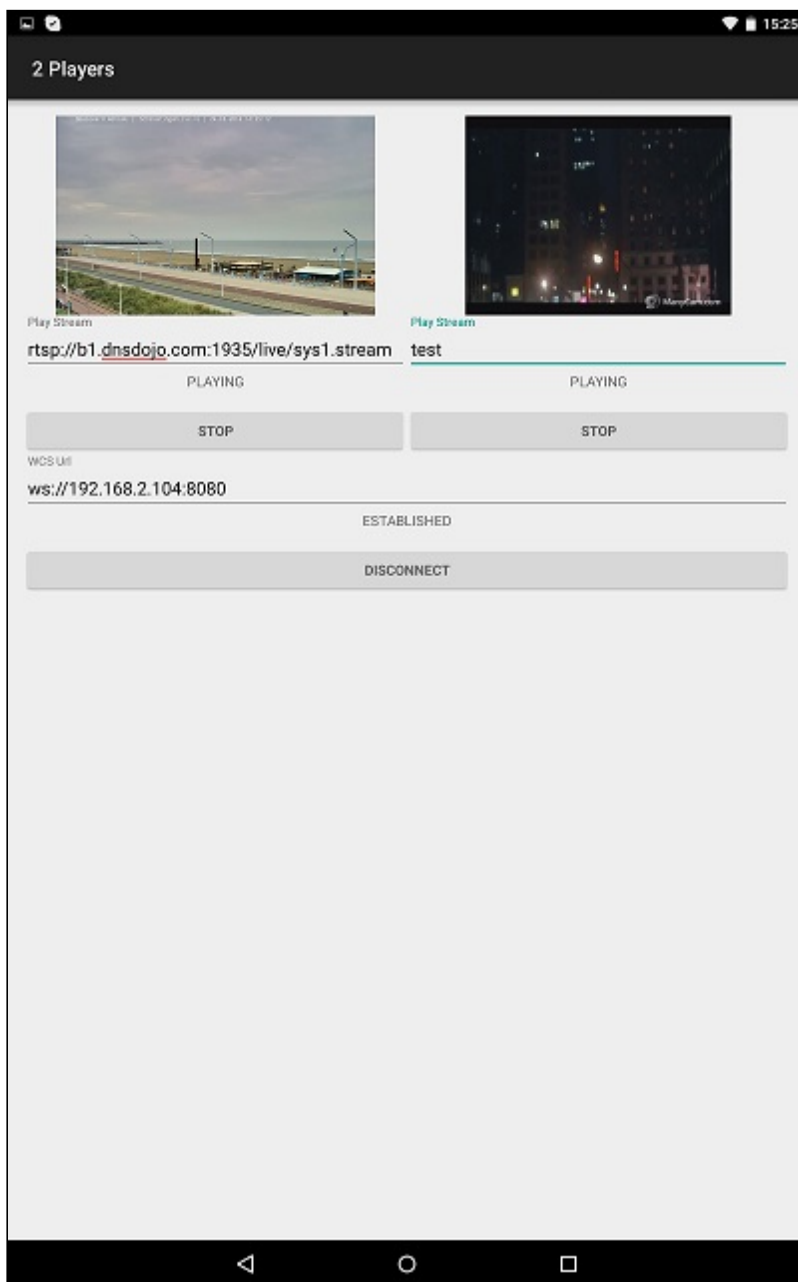


# Android 2 Players

## Example of Android application with two players

This example demonstrates how two or more players can be displayed in one application. Each of the players can be used to play a different stream.



## Analyzing the example code

To analyze the code, let's take class `TwoPlayersActivity.java` of the `2players` example, which can be downloaded with corresponding build `1.0.1.38`.

## 1. API initialization

`Flashphoner.init()` [code](#)

For initialization, `Context` object is passed to the `init()` method.

```
Flashphoner.init(this);
```

## 2. Session creation

`Flashphoner.createSession()` [code](#)

`SessionOptions` object with the URL of WCS server is passed to `createSession()` method:

```
SessionOptions sessionOptions = new
SessionOptions(mWcsUrlView.getText().toString());
sessionOptions.setRemoteRenderer(remote2Render);

/**
 * Session for connection to WCS server is created with method
 * createSession().
 */
session = Flashphoner.createSession(sessionOptions);
```

## 3. Connection to the server

`Session.connect()` [code](#)

```
session.connect(new Connection());
```

## 4. Receiving the event confirming successful connection

`session.onConnected()` [code](#)

```
@Override
public void onConnected(final Connection connection) {
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            mConnectButton.setText(R.string.action_disconnect);
            mConnectButton.setTag(R.string.action_disconnect);
            mConnectButton.setEnabled(true);
            mConnectStatus.setText(connection.getStatus());
            mPlay1Button.setEnabled(true);
            mPlay2Button.setEnabled(true);
        }
    });
}
```

```

    }
    });
}

```

## 5. Playback of video stream 1

`Session.createStream()`, `Stream.play()` [code](#)

`StreamOptions` object with the following parameters is passed to the `createStream()` method:

- name of the stream to playback
- `SurfaceViewRenderer remote1Renderer` to display the stream 1

```

StreamOptions streamOptions = new
StreamOptions(mPlay1StreamView.getText().toString());
streamOptions.setRenderer(remote1Render);

/**
 * Stream is created with method Session.createStream().
 */
play1Stream = session.createStream(streamOptions);

/**
 * Callback function for stream status change is added to make appropriate
changes in controls of the interface when stream is being played.
 */
play1Stream.on(new StreamStatusEvent() {
    ...
});

/**
 * Method Stream.play() is called to start playback of the stream.
 */
play1Stream.play();

```

## 6. Playback of video stream 2

`Session.createStream()`, `Stream.play()` [code](#)

`StreamOptions` object with the following parameters is passed to the `createStream()` method:

- name of the stream to playback
- `SurfaceViewRenderer remote2Renderer` to display the stream 1

```

StreamOptions streamOptions = new
StreamOptions(mPlay2StreamView.getText().toString());
streamOptions.setRenderer(remote2Render);

```

```

/**
 * Stream is created with method Session.createStream().
 */
play2Stream = session.createStream(streamOptions);

/**
 * Callback function for stream status change is added to make appropriate
 * changes in controls of the interface when stream is being played.
 */
play2Stream.on(new StreamStatusEvent() {
    ...
});

/**
 * Method Stream.play() is called to start playback of the stream.
 */
play2Stream.play();

```

## 7. Stream 1 playback stop

`Stream.stop()` [code](#)

```

play1Stream.stop();
play1Stream = null;

```

## 8. Stream 2 playback stop

`Stream.stop()` [code](#)

```

play2Stream.stop();
play2Stream = null;

```

## 9. Session disconnection

`Session.disconnect()` [code](#)

```

session.disconnect();

```

## 10. Receiving the event confirming successful disconnection

`session.onDisconnection()` [code](#)

```

@Override
public void onDisconnection(final Connection connection) {
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            mConnectButton.setText(R.string.action_connect);
        }
    });
}

```

```
mConnectButton.setTag(R.string.action_connect);
mConnectButton.setEnabled(true);
mPlay1Button.setText(R.string.action_play);
mPlay1Button.setTag(R.string.action_play);
mPlay1Button.setEnabled(false);
mPlay2Button.setText(R.string.action_play);
mPlay2Button.setTag(R.string.action_play);
mPlay2Button.setEnabled(false);
mConnectStatus.setText(connection.getStatus());
mPlay1Status.setText("");
mPlay2Status.setText("");
    }
});
}
```