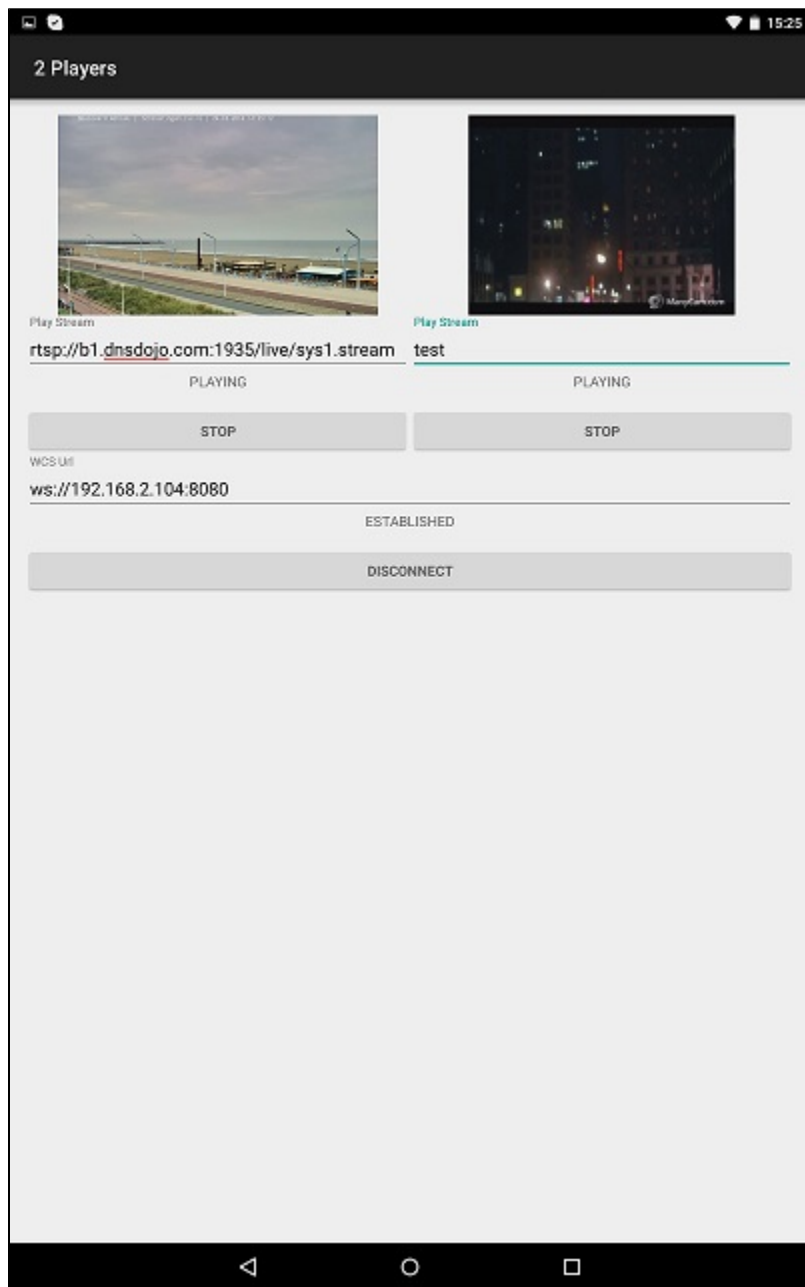


Android 2 Players

Пример Android-приложения с двумя плеерами

Данный пример показывает, как отобразить в одном приложении более одного плеера, каждый из которых может играть свой видеопоток.



Работа с кодом примера

Для разбора кода возьмем класс [TwoPlayersActivity.java](#) примера 2players, который доступен для скачивания в соответствующей сборке [1.0.1.38](#).

1. Инициализация API.

Flashphoner.init() [код](#)

При инициализации методу init() передается объект Context.

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

2. Создание сессии

Flashphoner.createSession() [код](#)

Методу передается объект SessionOptions со следующими параметрами

- URL WCS-сервера

```
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. Подключение к серверу.

Session.connect() [код](#)

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

4. Получение от сервера события, подтверждающего успешное соединение.

session.onConnected() [код](#)

```
@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. Воспроизведение первого видеопотока.

Session.createStream(), Stream.play() [код](#)

При создании потока методу Session.createStream() передается объект StreamOptions с параметрами:

- имя видеопотока для воспроизведения;
- SurfaceViewRenderer remote1Renderer, который будет использоваться для воспроизведения видеопотока

```

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. Воспроизведение второго видеопотока.

Session.createStream(), Stream.play() [код](#)

При создании потока методу Session.createStream() передается объект StreamOptions с параметрами:

- имя видеопотока для воспроизведения;
- SurfaceViewRenderer remote2Renderer, который будет использоваться для воспроизведения видеопотока

```

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. Остановка первого видеопотока.

Stream.stop() [код](#)

```

play1Stream.stop();
play1Stream = null;

```

8. Остановка второго видеопотока.

Stream.stop() [код](#)

```
play2Stream.stop();  
play2Stream = null;
```

9. Закрытие соединения.

Session.disconnect() [код](#)

```
session.disconnect();
```

10. Получение события, подтверждающего разъединение.

session.onDisconnection() [код](#)

```
@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("");  
        }  
    });  
}
```