

Обработка ошибок Android SDK

- [Список ошибок](#)
- [Пример кода обработки ошибок](#)

При получении статуса потока или звонка 'FAILED', метод `getInfo()` возвращает текстовое описание возникшей ошибки.

Список ошибок

Ошибка	Текстовое описание
Класс StreamStatusInfo	
FAILED_BY_ICE_ERROR	Failed by ICE error
FAILED_BY_ICE_TIMEOUT	Failed by ICE timeout
FAILED_BY_ICE_KEEP_ALIVE	Failed by ICE keep alive
FAILED_BY_DTLS_FINGERPRINT_ERROR	Failed by DTLS fingerprint error
FAILED_BY_DTLS_ERROR	Failed by DTLS error
FAILED_BY_HLS_WRITER_ERROR	Failed by HLS writer error
FAILED_BY_RTMP_WRITER_ERROR	Failed by RTMP writer error
FAILED_BY_RTP_ACTIVITY	Failed by RTP activity
STOPPED_BY_SESSION_DISCONNECT	Stopped by session disconnect
STOPPED_BY_REST_TERMINATE	Stopped by rest /terminate
STOPPED_BY_PUBLISHER_STOP	Stopped by publisher stop
STOPPED_BY_USER	Stopped by user
FAILED_BY_ERROR	Failed by error
FAILED_TO_ADD_STREAM_TO_PROXY	Failed to add stream to proxy
DISTRIBUTOR_STOPPED	Distributor stopped
PUBLISH_STREAM_IS_NOT_READY	Publish stream is not ready
STREAM_NOT_FOUND	Stream not found
STREAM_NAME_ALREADY_IN_USE	Stream name is already in use
MEDIASESSION_ID_NULL	MediaSessionId is null
MEDIASESSION_ID_ALREADY_IN_USE	MediaSessionId is already in use
SESSION_NOT_READY	Session not ready
SESSION_DOES_NOT_EXIST	Session does not exist
RTSP_HAS_WRONG_FORMAT	Rtsp has wrong format
FILE_HAS_WRONG_FORMAT	File has wrong format
FAILED_TO_CONNECT_TO_RTSP_STREAM	Failed to connect to rtsp stream
RTSP_STREAM_NOT_FOUND	Rtsp stream not found
RTSPAGENT_SHUTDOWN	RtspAgent shutdown
STREAM_FAILED	Stream failed
NO_COMMON_CODECS	No common codecs
BAD_URI	Bad URI
GOT_EXCEPTION_WHILE_STREAMING_FILE	Got exception while streaming file
REQUESTED_STREAM_SHUTDOWN	Requested stream shutdown

FAILED_TO_READ_FILE	Failed to read file
FILE_NOT_FOUND	File not found
FAILED_TO_CONNECT_TO_ORIGIN_STREAM	Failed to connect to origin stream
CDN_STREAM_NOT_FOUND	CDN stream not found
FAILED_TO_GET_AGENT_STORAGE	Failed to get agent storage
AGENT_SERVICING_ORIGIN_STREAM_IS_SHUTTING_DOWN	Agent servicing origin stream is shutting down
TERMINATED_BY_KEEP_ALIVE	Terminated by keep-alive
TRANSCODING_REQUIRED_BUT_DISABLED	Transcoding is required, but disabled
NO_AVAILABLE_TRANSCODERS	No available transcoder nodes in CDN
Kласс CallStatusInfo	
FAILED_BY_SESSION_CREATION	Failed by session creation
FAILED_BY_ICE_ERROR	Failed by ICE error
FAILED_BY_RTP_ACTIVITY	Failed by RTP activity
FAILED_BY_RTMP_WRITER_ERROR	Failed by RTMP writer error
FAILED_BY_DTLS_FINGERPRINT_ERROR	Failed by DTLS fingerprint error
FAILED_BY_DTLS_ERROR	Failed by DTLS error
FAILED_BY_ERROR	Failed by error
FAILED_BY_REQUEST_TIMEOUT	Failed by request timeout
TRANSCODING_REQUIRED_BUT_DISABLED	Transcoding is required, but disabled

Пример кода обработки ошибок

В качестве примера рассмотрим код класса [StreamingMinActivity.java](#) версии с хэшем 17fa60baa5a21bc7398338c530bd7314e0cbfca3, которая доступна для скачивания в сборке [1.0.1.49](#)

1. Обработка ошибок при воспроизведении потока

```
stream.getInfo()code
```

```

        playStream.on(new StreamStatusEvent() {
            @Override
            public void onStreamStatus(final Stream stream, final StreamStatus streamStatus) {
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        ...
                        if (StreamStatus.FAILED.equals(streamStatus)){
                            switch (stream.getInfo()){
                                case StreamStatusInfo.SESSION_DOES_NOT_EXIST:
                                    mPlayStatus.setText(streamStatus+": Actual session does not
exist");

                                    break;
                                case StreamStatusInfo.STOPPED_BY_PUBLISHER_STOP:
                                    mPlayStatus.setText(streamStatus+": Related publisher stopped
its stream or lost connection");

                                    break;
                                case StreamStatusInfo.SESSION_NOT_READY:
                                    mPlayStatus.setText(streamStatus+": Session is not initialized
or terminated on play ordinary stream");

                                    break;
                                case StreamStatusInfo.RTSP_STREAM_NOT_FOUND:
                                    mPlayStatus.setText(streamStatus+": Rtp stream not found where
agent received '404-Not Found'");

                                    break;
                                case StreamStatusInfo.FAILED_TO_CONNECT_TO_RTSP_STREAM:
                                    mPlayStatus.setText(streamStatus+": Failed to connect to rtp
stream");

                                    break;
                                case StreamStatusInfo.FILE_NOT_FOUND:
                                    mPlayStatus.setText(streamStatus+": File does not exist, check
filename");

                                    break;
                                case StreamStatusInfo.FILE_HAS_WRONG_FORMAT:
                                    mPlayStatus.setText(streamStatus+": File has wrong format on
play vod, this format is not supported");

                                    break;
                                default:{
                                    mPlayStatus.setText(stream.getInfo());
                                }
                            }
                        } else {
                            mPlayStatus.setText(streamStatus.toString());
                        }
                    }
                });
            }
        });

```

2. Обработка ошибок при публикации потока

`stream.getInfo()`[code](#)

```

        if (StreamStatus.FAILED.equals(streamStatus)){
            switch (stream.getInfo()){
                case StreamStatusInfo.STREAM_NAME_ALREADY_IN_USE:
                    mPublishStatus.setText(streamStatus+": Server already has a
publish stream with the same name, try using different one");

                    break;
                default:{
                    mPlayStatus.setText(stream.getInfo());
                }
            }
        } else {
            mPublishStatus.setText(streamStatus.toString());
        }

```