

# Event states

The event states are listed in the table below.

If such event is received on client side, it may be processed in a corresponding object (`Stream`, `Session`, `Call`, etc).

Examples:

1. Handle success connection to WCS server

```
session.on(SESSION_STATUS.ESTABLISHED, function(){
    trace("connection established");
});
```

2. Handle success stream publishing to WCS server

```
stream.on(STREAM_STATUS.PUBLISHING, function(){
    trace("stream is publishing");
});
```

Event	Status field values	Description
ConnectionStatusEvent	<code>ESTABLISHED</code> - Connection is established <code>FAILED</code> - Error while establishing connection <code>DISCONNECTED</code> - Connection is lost	The status of establishing connection to the WCS server via Websocket
RegistrationStatusEvent	<code>REGISTERED</code> - SIP account is registered <code>UNREGISTERED</code> - SIP account is unregistered <code>AUTHENTICATION_FAIL</code> - Authentication error	SIP registration status

Event	Status field values	Description
CallStatusEvent	<p><b>TRYING</b> - SIP returned TRYING</p> <p><b>RING</b> - SIP returned 183</p> <p><b>RINGING</b></p> <p><b>SESSION_PROGRESS</b> - SIP returned 183</p> <p><b>SESSION_PROGRESS</b></p> <p><b>BUSY</b> - SIP returned 486 or 600</p> <p><b>BUSY</b></p> <p><b>ESTABLISHED</b> - SIP returned 200 OK to INVITE, call is established</p> <p><b>HOLD</b> - SIP call is put on hold</p> <p><b>FINISH</b> - SIP call is finished</p> <p><b>FAILED</b> - SIP call has failed or ended with an error</p>	SIP call status
TransferStatusEvent	<p><b>ACCEPTED</b> - SIP call transfer request is accepted</p> <p><b>TRYING</b> - An attempt to transfer the call</p> <p><b>COMPLETED</b> - Transfer is completed</p> <p><b>FAILED</b> - Transfer has failed</p>	Call transfer status
StreamStatusEvent	<p><b>PUBLISHING</b> - Stream is publishing</p> <p><b>UNPUBLISHED</b> - Publishing finished</p> <p><b>PLAYING</b> - Stream is playing</p> <p><b>STOPPED</b> - Playback is stopped</p> <p><b>FAILED</b> - Error while publishing or playing the stream</p>	Stream status
StreamKeepAliveEvent	<p><b>PUBLISHING</b> - Stream is publishing</p> <p><b>PLAYING</b> - Stream is playing</p>	Keep-alive of the stream
SessionDebugStatusEvent	<p><b>STARTED</b> - Debug logs are enabled</p> <p><b>STOPPED</b> - Debug logs are disabled</p>	Debug session status
DataStatusEvent	<p><b>ACCEPTED</b> - Data are accepted for processing</p> <p><b>FAILED</b> - An error occurred</p>	Data sending status