

# The match between client invocations and REST methods

The table below lists methods, types and corresponding REST hooks and events related to these REST hooks. The **Web SDK** field illustrates how to invoke methods and handle events on the client's side.

Description	Web SDK	REST hook	Events
<b>connect</b>			
Establish connection to WCS	<code>Flashphoner.createSession()</code>	connect	ConnectionStatusEvent, RegistrationStatusEvent
<b>Direct invocations - Streaming</b>			
Play the stream	<code>stream.play()</code>	playStream	StreamStatusEvent
Pause playback of the stream	<code>stream.pause()</code>	pauseStream	StreamStatusEvent
Stop playback of the stream	<code>stream.stop()</code>	stopStream	StreamStatusEvent
Publish the stream	<code>stream.publish()</code>	publishStream	StreamStatusEvent
Make a snapshot of the stream	<code>stream.snapshot()</code>	snapshot	StreamStatusEvent
Stop publishing the stream	<code>stream.stop()</code>	unPublishStream	StreamStatusEvent
<b>Direct invocations - Calls</b>			
Make an outbound call	<code>call.call()</code>	call	CallStatusEvent
Send tone signal (DTMF)	<code>call.sendDTMF()</code>	sendDtmf	\-
Answer an incoming call	<code>call.answer()</code>	answer	CallStatusEvent
Put the call on hold	<code>call.hold()</code>	hold	CallStatusEvent

Description	Web SDK	REST hook	Events
Release the call from hold	<code>call.unhold()</code>	unhold	CallStatusEvent
Transfer the call to another number	<code>call.transfer()</code>	transfer	TransferStatusEvent
Hang up the call	<code>call.hangup()</code>	hangup	CallStatusEvent
<b>Incoming invocations – Calls</b>			
Receive the incoming call	<code>session.on(SESSION_STATUS.INCOMING_CALL, handler)</code>	OnCallEvent, CallStatusEvent	
<b>Direct and incoming invocations – Data</b>			
Send data	<code>session.sendData()</code>	OnDataEvent, DataStatusEvent	
Receive data	<code>session.on(SESSION_STATUS.APPLICATION_DATA, handler)</code>	OnDataEvent, DataStatusEvent	
<b>Direct invocations – Diagnostics</b>			
Send a bug report to save on the server	<code>session.submitBugReport()</code>	submitBugReport	BugReportStatusEvent
Save client logs on the server	<code>session.pushLogs()</code>	pushLogs	\-
Enable debug logs for streaming	<code>session.startDebug()</code>	sessionDebug	SessionDebugStatusEvent
Disable debug logs for streaming	<code>session.stopDebug()</code>	sessionDebug	SessionDebugStatusEvent
<b>Session events</b>			
Connection status	<code>session.on(SESSION_STATUS..., handler)</code>	ConnectionStatusEvent	ConnectionStatusEvent
Registration status	<code>session.on(SESSION_STATUS..., handler)</code>	RegistrationStatusEvent	RegistrationStatusEvent
Session debug status	<code>session.on(SESSION_STATUS.DEBUG, handler)</code>	SessionDebugStatusEvent	SessionDebugStatusEvent
<b>Stream events</b>			

Description	Web SDK	REST hook	Events
Stream status	<code>stream.on(STREAM_STATUS..., handler)</code>	StreamStatusEvent	StreamStatusEvent
<b>Call events</b>			
Call status	<code>call.on(CALL_STATUS..., handler)</code>	CallStatusEvent	CallStatusEvent
<b>Other events</b>			
The event sent by the server to backend while publishing / playing the stream	Not used by the client	StreamKeepAliveEvent	StreamKeepAliveEvent